Documentation 8.0

ZABBIX

02.04.2026

Contents

Manuale utente **6**

Benvenuti nella documentazione di Zabbix **6**

Inizia a usare Zabbix . . . 6
Zabbix Cloud . . . 6
Centro sviluppatori . . . 6
Community e altre risorse . . . 6
1 Installazione e primi passi . . . 6
1 Installazione dell'interfaccia web . . . 7
2 Guide rapide introduttive . . . 13
3 Installazione: guide aggiuntive . . . 33
4 Aggiornamento . . . 72
5 Note di aggiornamento per Zabbix 8.0 . . . 87
Problemi noti . . . 89
Requisiti . . . 97
2 Novità di Zabbix 8.0 . . . 106
Template . . . 107
Item . . . 108
Plugin . . . 108
Low-level discovery . . . 109
Processi . . . 109
Autenticazione . . . 110
Widget . . . 110
Frontend . . . 110
Documentazione . . . 112
Novità in Zabbix 8.0.x . . . 112
3 Appliance Zabbix . . . 112
4 Processi di Zabbix . . . 116
1 Server . . . 116
2 Agent . . . 125
3 Agent 2 . . . 128
4 Proxy . . . 137
5 Java gateway . . . 142
6 Sender . . . 145
7 Get . . . 146
8 JS . . . 147
9 Web service . . . 148
5 Configurazione . . . 149
1 Host e gruppi di host . . . 159
2 Item . . . 177
3 Rilevamento dei problemi con i trigger . . . 392
4 Eventi . . . 442
5 Correlazione degli eventi . . . 446
6 Etichettatura . . . 453
7 Visualizzazione . . . 456
8 Modelli e gruppi di modelli . . . 484
9 Modelli pronti all'uso . . . 484
10 Notifiche sugli eventi . . . 498
11 Macro . . . 548
12 Utenti e gruppi di utenti . . . 562
13 Conservazione dei segreti . . . 571

14 Rapporti pianificati . . . 579
15 Esportazione dati . . . 582
6 Monitoraggio dei servizi . . . 588
1 Albero dei servizi . . . 589
2 SLA . . . 593
3 Esempio di configurazione . . . 594
7 Monitoraggio web . . . 598
1 Elementi di monitoraggio Web . . . 608
2 Scenario reale . . . 609
8 Monitoraggio delle macchine virtuali . . . 616
1 Elementi chiave di monitoraggio VMware . . . 618
2 Campi chiave del rilevamento delle macchine virtuali . . . 636
3 Esempi JSON per elementi VMware . . . 640
4 Esempio di configurazione del monitoraggio VMware . . . 645
9 Manutenzione . . . 650
10 Espressioni regolari . . . 655
11 Riconoscimento del problema . . . 660
1 Soppressione del problema . . . 663
12 Esportazione/importazione della configurazione . . . 664
1 Gruppi di template . . . 666
2 Gruppi di host . . . 666
3 Template . . . 666
4 Host . . . 683
5 Mappe di rete . . . 699
6 Tipi di supporto . . . 707
13 Individuazione . . . 714
1 Individuazione all’interno della rete . . . 714
2 Registrazione automatica dell’agente attivo . . . 722
3 Individuazione di basso livello . . . 726
14 Monitoraggio distribuito . . . 780
1 Proxy . . . 781
15 Crittografia . . . 789
1 Utilizzo dei certificati . . . 796
2 Utilizzo delle chiavi precondivise . . . 803
3 Risoluzione dei problemi . . . 806
16 Interfaccia web . . . 808
1 Menù . . . 809
2 Sezioni di frontend . . . 818
3 Impostazioni utente . . . 1075
4 Ricerca globale . . . 1080
5 Modalità di manutenzione del frontend . . . 1082
6 Parametri di pagina . . . 1083
7 Definizioni . . . 1086
8 Creare il tuo tema . . . 1086
9 Modalità di debug . . . 1087
10 Cookie utilizzati da Zabbix . . . 1088
11 Fusi orari . . . 1089
12 Reimpostazione della password . . . 1090
13 Selettori del periodo di tempo e dell’host . . . 1090
17 Best practice . . . 1092
1 Best practices per la sicurezza . . . 1092
2 Best practices per la configurazione . . . 1100
18 API . . . 1101
Riferimento ai metodi . . . 1106
Appendice 1. Commento di riferimento . . . 1926
Appendice 2. Modifiche dalla 7.4 alla 8.0 . . . 1932
Appendice 3. Modifiche nella versione 8.0 . . . 1933
19 Estensioni . . . 1933
1 Moduli caricabili . . . 1936
3 moduli frontend . . . 1945
20 Appendici . . . 1945
1 Installazione e configurazione . . . 1946
2 Configurazione del processo . . . 2005

3 Protocolli . . . 2092
4 Item . . . 2117
5 Funzioni supportate . . . 2154
6 Macro . . . 2154
7 Simboli delle unità . . . 2191
8 Sintassi del periodo di tempo . . . 2192
9 Esecuzione dei comandi . . . 2193
10 Compatibilità della versione . . . 2194
12 Libreria a collegamento dinamico di Zabbix sender per Windows . . . 2195
13 Aggiornamento del monitoraggio dei servizi . . . 2196
14 Altri problemi . . . 2196
16 Esempi di escaping . . . 2197
21 Guide rapide di riferimento . . . 2199
Panoramica . . . 2199
1 Monitora Linux con l'agente Zabbix . . . 2199
2 Monitorare Windows con Zabbix agent . . . 2203
3 Monitora Apache tramite HTTP . . . 2207
4 Monitorare MySQL con Zabbix agent 2 . . . 2211
5 Monitorare VMware con Zabbix . . . 2219
6 Monitorare il traffico di rete con Zabbix . . . 2222
7 Monitorare il traffico di rete utilizzando controlli attivi . . . 2225
8 Monitorare i siti web con gli item Browser . . . 2228
9 Monitorare i certificati dei siti web con Zabbix agent 2 (passivo) . . . 2232
10 Monitorare uno switch o un router di rete con Zabbix . . . 2237
11 Monitorare il log eventi di Windows utilizzando controlli attivi . . . 2244
Cos'è Zabbix . . . 2248

Zabbix Cloud **2252**

Zabbix Cloud **2252**
Configurare e gestire Zabbix Cloud . . . 2252
Scopri Zabbix Cloud . . . 2253
Distribuire Zabbix nel cloud . . . 2253
Configurazione del nodo . . . 2257
Panoramica . . . 2257
Scheda Panoramica . . . 2258
Scheda Filtri di accesso . . . 2259
Scheda Crittografia . . . 2260
Scheda Backup . . . 2261
Scheda Cronologia . . . 2261
Scheda Manutenzione . . . 2262
Scheda Upgrade . . . 2263
Aggiunta di utenti . . . 2265
Introduzione . . . 2265
Aggiunta di utenti alle organizzazioni . . . 2265
Aggiunta di utenti ai nodi . . . 2266
Differenze principali di Zabbix Cloud . . . 2267

Centro sviluppatori **2269**

Centro sviluppatori **2269**
Sviluppare su Zabbix . . . 2269
Plugin . . . 2269
Interfacce plugin . . . 2275
Libreria Python per Zabbix . . . 2278
Installazione . . . 2278
Guida rapida . . . 2278
Utilizzo della API di Zabbix . . . 2282
Raccogliere dati da Zabbix agent . . . 2285
Inviare dati al server o al proxy Zabbix . . . 2286
Logging di debug . . . 2290
Moduli . . . 2291
Struttura del file del modulo . . . 2291
Widget . . . 2299

Tutorial . . . 2307
Esempi . . . 2333
Modifiche allo sviluppo delle estensioni . . . 2333

Pagine man di Zabbix **2334**
zabbix_agent2 . . . 2334
NOME . . . 2334
SINOSSI . . . 2334
DESCRIZIONE . . . 2334
OPZIONI . . . 2334
FILE . . . 2335
VEDI ANCHE . . . 2335
Indice . . . 2335
zabbix_agentd . . . 2336
NOME . . . 2336
SINOSSI . . . 2336
DESCRIZIONE . . . 2336
OPZIONI . . . 2336
FILE . . . 2337
VEDI ANCHE . . . 2337
Indice . . . 2337
zabbix_get . . . 2338
NOME . . . 2338
SINOSSI . . . 2338
DESCRIZIONE . . . 2338
OPZIONI . . . 2338
ESEMPI . . . 2339
VEDI ANCHE . . . 2339
Indice . . . 2339
zabbix_js . . . 2340
NOME . . . 2340
SINOSSI . . . 2340
DESCRIZIONE . . . 2340
OPZIONI . . . 2340
ESEMPI . . . 2340
VEDI ANCHE . . . 2340
Indice . . . 2341
zabbix_proxy . . . 2341
NOME . . . 2341
SINOSSI . . . 2341
DESCRIZIONE . . . 2341
OPZIONI . . . 2341
FILE . . . 2342
VEDI ANCHE . . . 2342
Indice . . . 2342
zabbix_sender . . . 2343
NOME . . . 2343
SINOSSI . . . 2343
DESCRIZIONE . . . 2343
OPZIONI . . . 2343
STATO DI USCITA . . . 2346
ESEMPI . . . 2346
VEDI ANCHE . . . 2346
Indice . . . 2346
zabbix_server . . . 2347
NOME . . . 2347
SINOSSI . . . 2347
DESCRIZIONE . . . 2347
OPZIONI . . . 2347
FILE . . . 2349
VEDI ANCHE . . . 2349
Indice . . . 2349
zabbix_web_service . . . 2349

NOME 2349
SINOSSI 2350
DESCRIZIONE 2350
OPZIONI 2350
FILE 2350
VEDI ANCHE 2350
Indice 2350

Avviso sul copyright **2350**

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 rapide introduttive Guide concise e orientate alle attività che ti accompagnano nelle nozioni di base, 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.

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.

1 Installazione e primi passi

Per configurare Zabbix da zero, seguire questo processo:

1. Assicurarsi 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 maggiori dettagli su ciascun passaggio.

Software richiesto Zabbix funziona bene con il seguente software (spesso chiamato stack LAMP):

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

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

Per informazioni più dettagliate, come le versioni software supportate, vedere Software richiesto.

Installare Zabbix Il modo **più semplice** per installare Zabbix è tramite i **pacchetti della distribuzione**. I pacchetti installano le dipendenze software minori.

Istruzioni di installazione precise, adattate al proprio sistema operativo, sono disponibili sul **sito web di Zabbix**.

Selezionare la propria piattaforma e seguire le istruzioni fornite. Per iniziare, si consiglia di installare uno Zabbix server, uno Zabbix agent 2 e il frontend (interfaccia web). In seguito sarà possibile aggiungere Zabbix proxy e altri agent.

Una parte del processo di installazione consiste nella creazione di un database (per conservare i dati di configurazione e cronologia) 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), in modo che 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.

Si può iniziare dalla guida rapida per monitorare un server Linux.

1 Installazione dell'interfaccia web

Questa sezione fornisce istruzioni passo passo per installare l'interfaccia web di Zabbix. Segui questi passaggi dopo aver installato il backend di Zabbix: Zabbix server, agent e database.

Il frontend di Zabbix è scritto in PHP, quindi per eseguirlo è necessario un server web supportato da PHP.

> **Note:**
> Per saperne di più sulla configurazione del protocollo SSL per il frontend di Zabbix, consulta queste best practices.

Schermata di benvenuto

Apri l'URL del frontend di Zabbix nel browser. Se hai installato Zabbix dai pacchetti, l'URL è:

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

Dovresti vedere la prima schermata della procedura guidata di installazione del frontend.

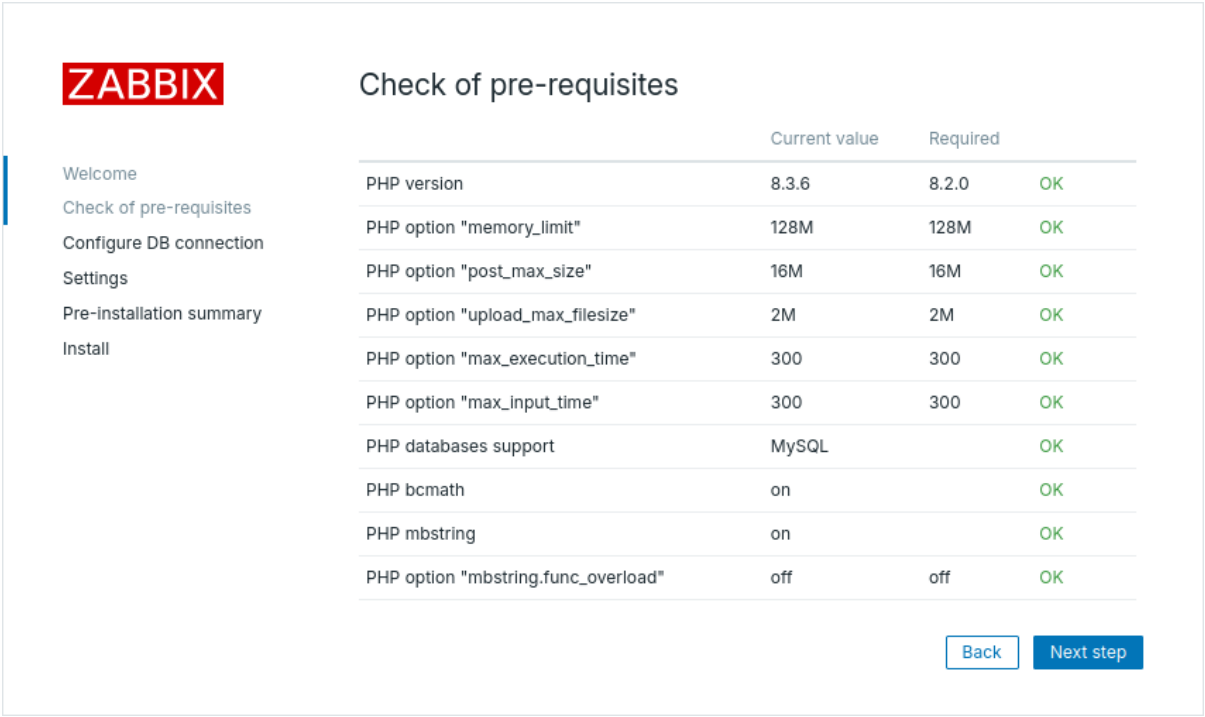
Usa il menu a discesa *Lingua predefinita* per cambiare la lingua predefinita del sistema e continuare il processo di installazione nella lingua selezionata (facoltativo). Per maggiori informazioni, vedi Installazione di lingue aggiuntive del frontend.

Tieni presente che impostando la lingua su *English (en_US)* verrà abilitato anche il formato statunitense di ora/data nel frontend.



Verifica dei prerequisiti

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



Prerequisito	Valore minimo	Descrizione
Versione PHP	8.2.0	
Opzione PHP "memory_limit"	128MB	In php.ini: memory_limit = 128M
Opzione PHP "post_max_size"	16MB	In php.ini: post_max_size = 16M
Opzione PHP "upload_max_filesize"	2MB	In php.ini: upload_max_filesize = 2M
Opzione PHP "max_execution_time"	300 secondi	In php.ini: max_execution_time = 300 (sono consentiti anche i valori 0 e -1)
Opzione PHP "max_input_time"	300 secondi	In php.ini: max_input_time = 300 (sono consentiti anche i valori 0 e -1)

Prerequisito	Valore minimo	Descrizione
Supporto PHP per database	Uno tra: MySQL, PostgreSQL	Vedere Requirements per l'elenco di tutte le estensioni PHP obbligatorie e facoltative. Si noti che i prerequisiti facoltativi non soddisfatti vengono visualizzati con lo stato rosso *Warning* e il processo di configurazione può continuare anche se non sono soddisfatti.
PHP bcmath	deve essere abilitato	
PHP mbstring	deve essere abilitato	
Opzione PHP "mbstring.func_overload"	deve essere disabilitata	In php.ini: mbstring.func_overload = 0
Opzione PHP "session.auto_start"	deve essere disabilitata	In php.ini: session.auto_start = 0
Opzione PHP "arg_separator.output"	&	In php.ini: arg_separator.output = "&" (è consentito anche il valore "&")

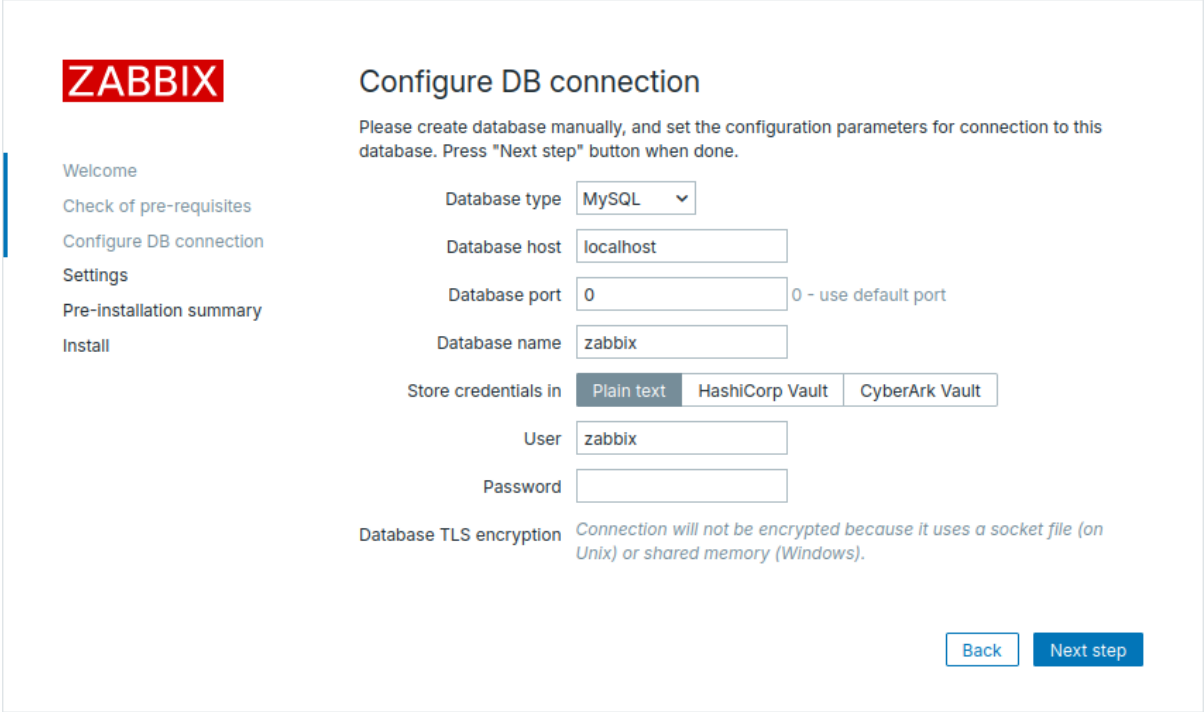
Attention:
Se è necessario modificare l'utente o il gruppo di utenti di Apache, verificare i permessi della cartella di sessione; in caso contrario, la configurazione di Zabbix potrebbe non essere in grado di continuare.

Configurare la connessione al DB

Inserire i dettagli per la connessione al database. Il database Zabbix deve essere già stato creato.

Per MySQL, inserendo `localhost` o lasciando vuoto il campo *Database host* si ottiene una connessione tramite il socket Unix predefinito. Il modulo di configurazione non fornisce un campo separato *Database socket*, quindi per utilizzare un socket personalizzato è necessario configurarlo nelle impostazioni del server Zabbix (ad esempio, con `DBSocket=` in zabbix_server.conf). In questo modo il frontend rimane allineato alle impostazioni di connessione tra server e database.

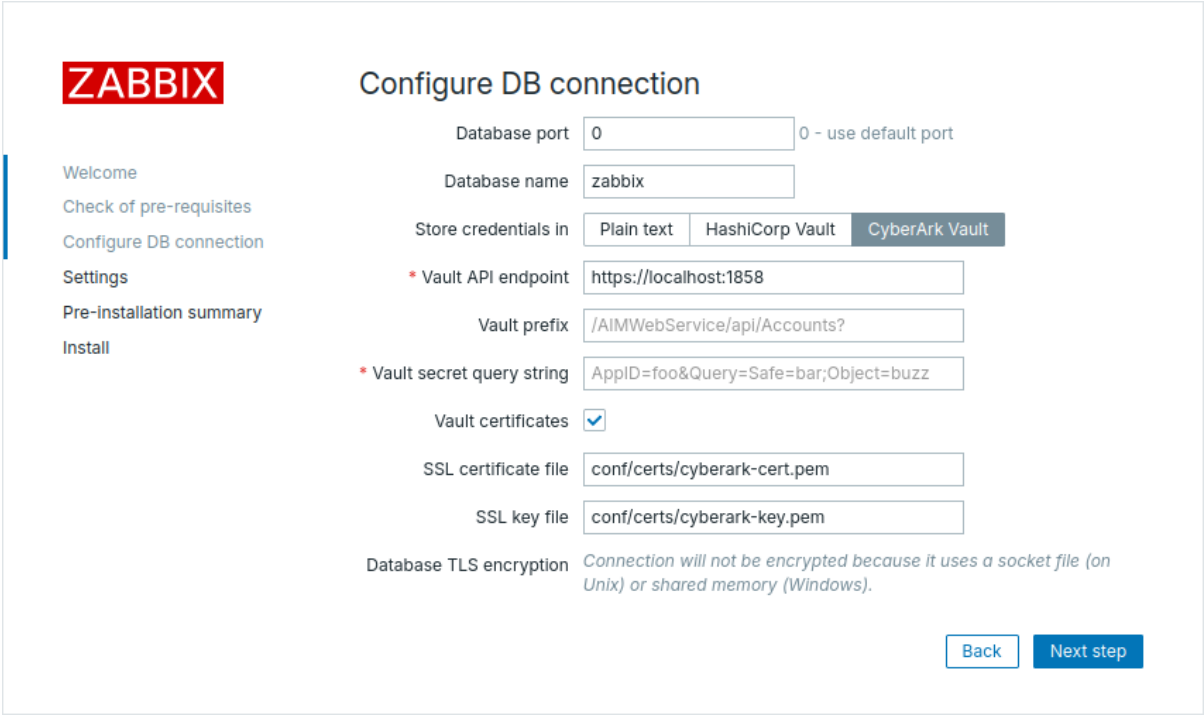
Per PostgreSQL, se il campo *Database host* viene lasciato vuoto, viene utilizzato il socket Unix-domain predefinito. Se invece viene inserito un percorso del socket (ad esempio, `/var/run/pgbouncer`), verrà utilizzato quel socket Unix-domain.



Se l'opzione *Database TLS encryption* è selezionata, nel modulo compariranno campi aggiuntivi per configurare la connessione TLS al database (solo MySQL o PostgreSQL).

Se *Store credentials in* è impostato su HashiCorp Vault o CyberArk Vault, saranno disponibili parametri aggiuntivi:

- per HashiCorp Vault: endpoint API del Vault, prefisso del vault, percorso del segreto e token di autenticazione;
- per CyberArk Vault: endpoint API del Vault, prefisso del vault, stringa di query del segreto e certificati. Se si seleziona la casella *Vault certificates*, compariranno due nuovi campi per specificare i percorsi del file del certificato SSL e del file della chiave SSL.

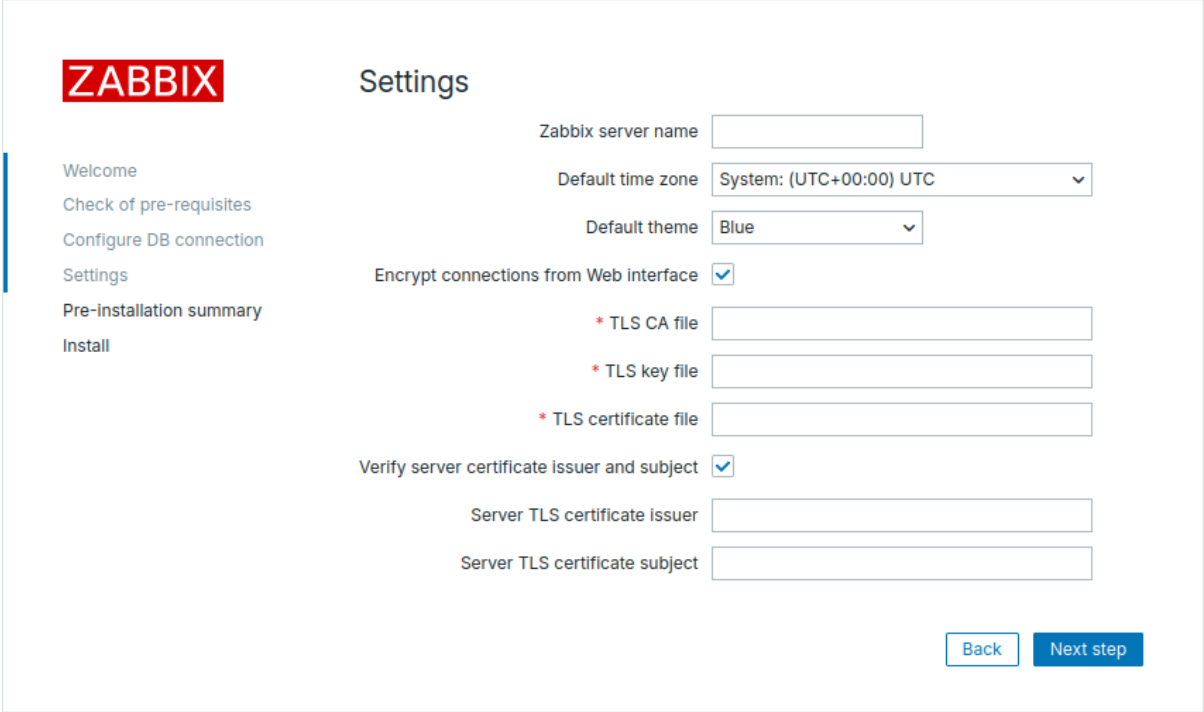


Impostazioni

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

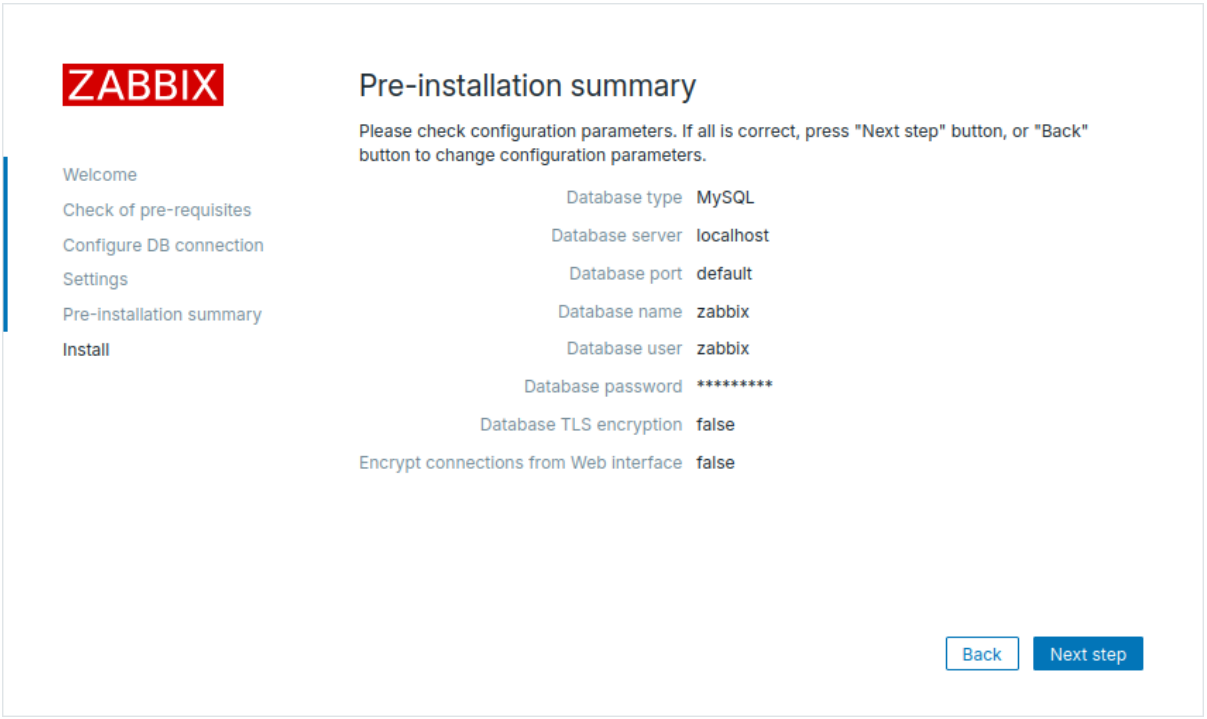
Imposta il fuso orario e il tema predefiniti per il frontend.

Se l'opzione *Crittografa le connessioni dall'interfaccia Web* è selezionata, nel modulo verranno visualizzati campi aggiuntivi per la configurazione della connessione TLS tra il server Zabbix e il frontend.

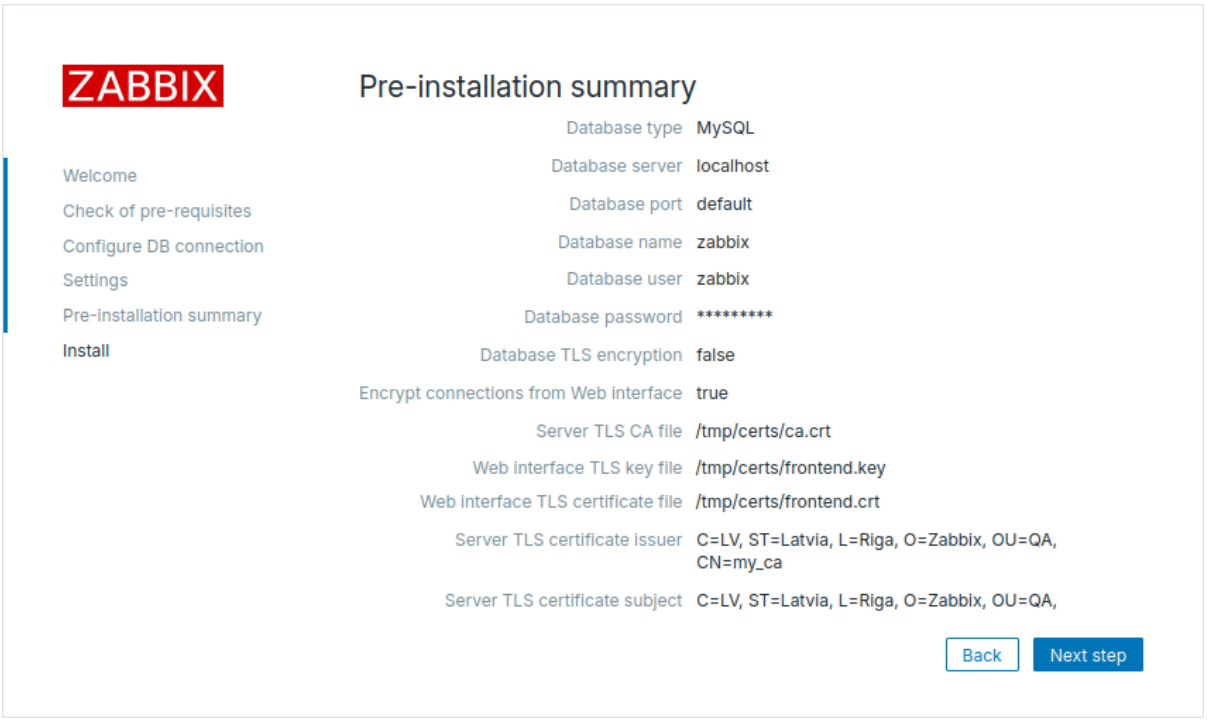


Riepilogo preinstallazione

Rivedi un riepilogo delle impostazioni.

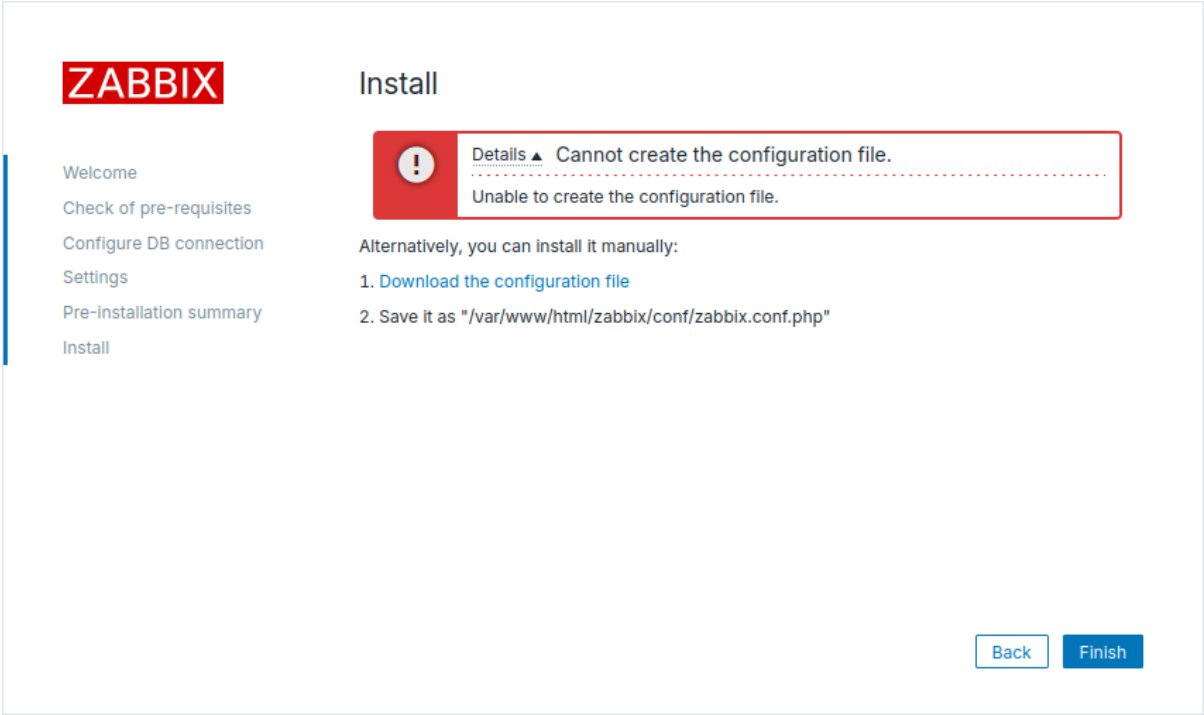


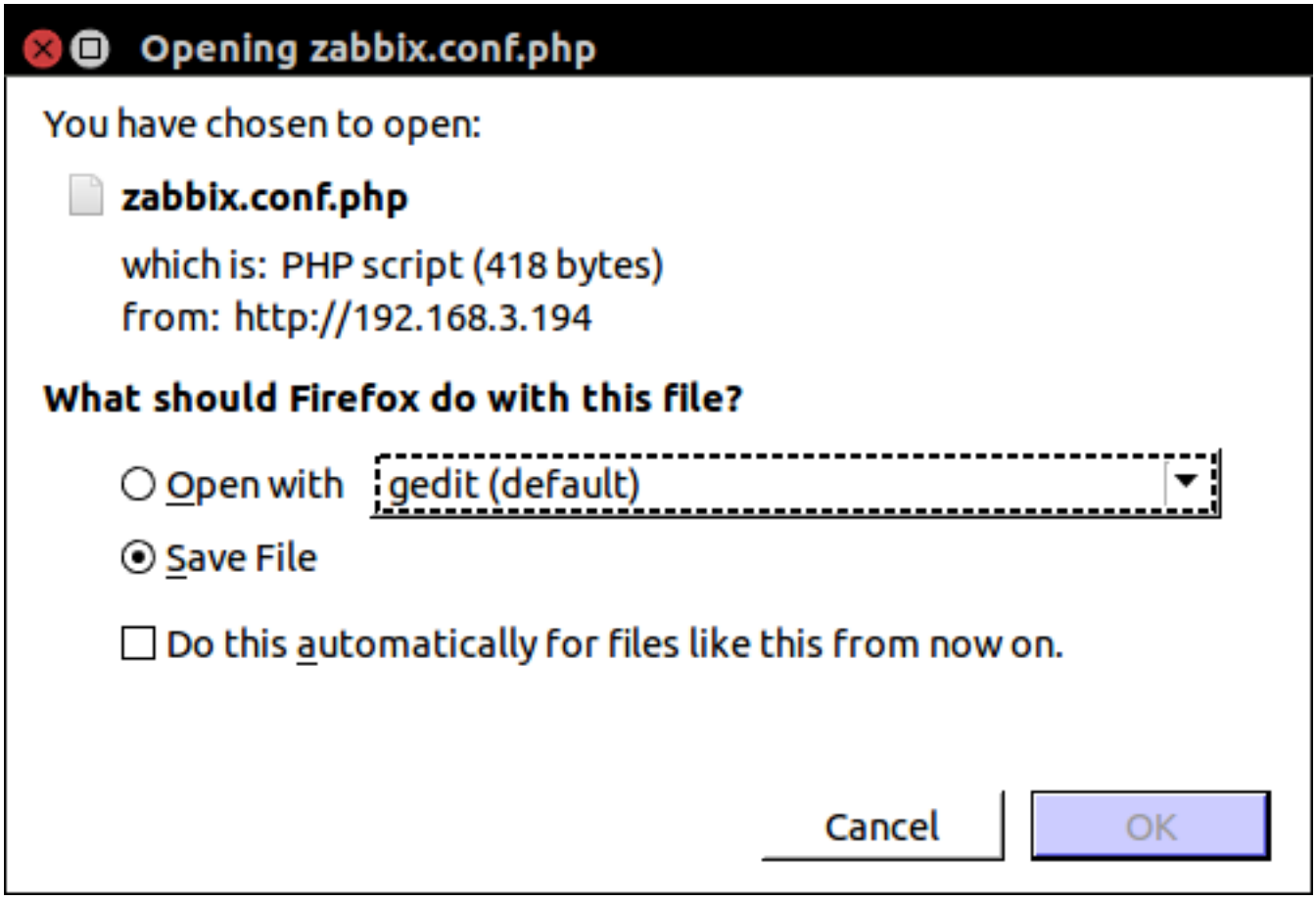
La sottopagina mostrerà i dati se è stata aggiunta la configurazione TLS.



Installazione

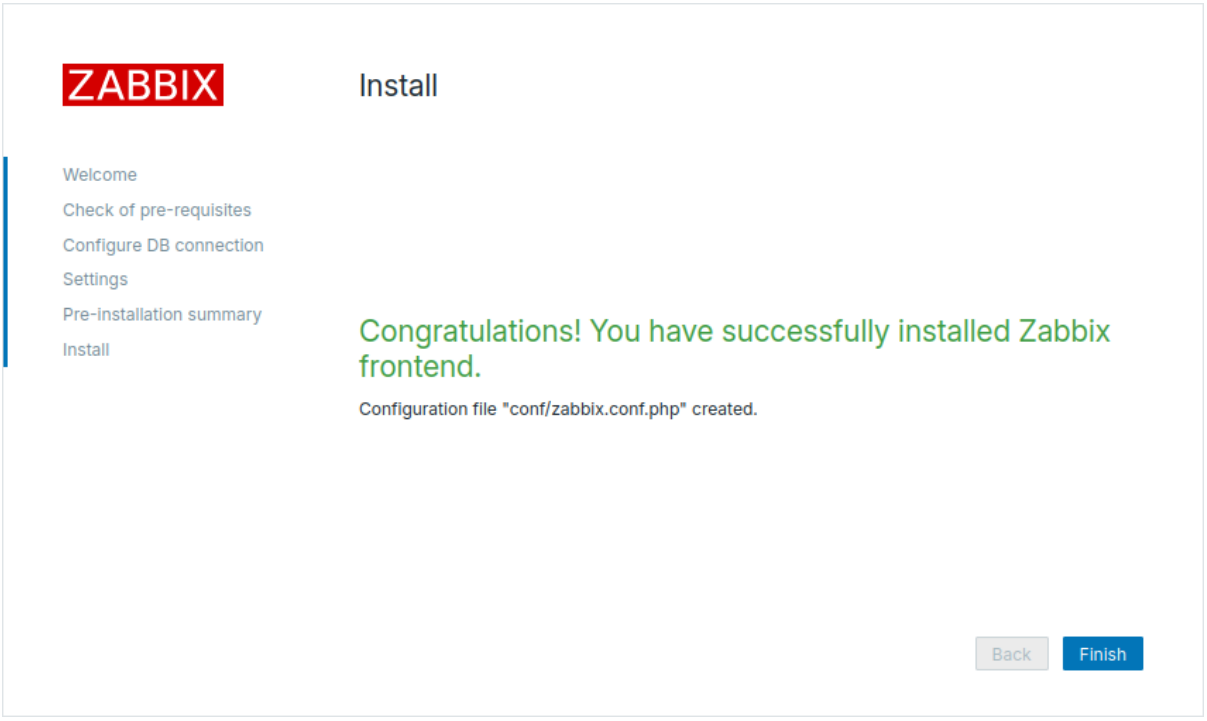
Se si installa Zabbix dai sorgenti, scaricare il file di configurazione e inserirlo in conf/ nella sottodirectory dei documenti HTML del webserver in cui sono stati copiati i file PHP di Zabbix.





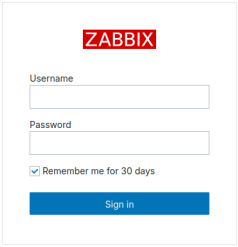
Note:
Se l'utente del webserver dispone dei permessi di scrittura sulla directory conf/, il file di configurazione verrà salvato automaticamente e sarà possibile passare subito al passaggio successivo.

Completare l'installazione.



Accesso

Il frontend di Zabbix è pronto! Il nome utente predefinito è **Admin**, la password **zabbix**.



Procedi alla guida al monitoraggio di Linux.

2 Guide rapide introduttive

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

Usa queste guide rapide per iniziare a monitorare risorse comuni:

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

Le guide rapide introduttive si concentrano sull'uso 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 obiettivo di monitoraggio (dispositivo, applicazione, ecc.), modificarlo e poi applicarlo a più obiettivi di monitoraggio simili.

Sebbene i template siano adatti per un monitoraggio immediato, potresti anche essere interessato 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

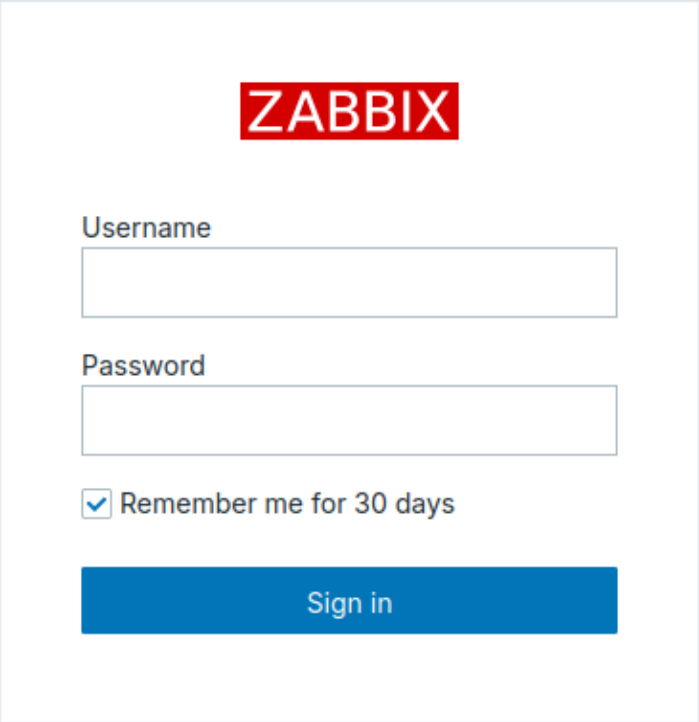
- Ricezione della notifica di problema
- Nuovo template

1 Accesso e configurazione dell'utente

Panoramica

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

Accesso



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

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

Accesso persistente

Per rimanere connesso fino a 30 giorni, seleziona *Ricorda per 30 giorni* prima di fare clic su *Accedi*.

Ricorda per 30 giorni abilitato:

- La tua sessione rimane attiva per 30 giorni.
- *Disconnessione automatica* viene ignorata, mantenendoti connesso fino al termine del periodo.
- Verrai connesso automaticamente nelle visite future entro 30 giorni senza reinserire le credenziali.

Ricorda per 30 giorni disabilitato:

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

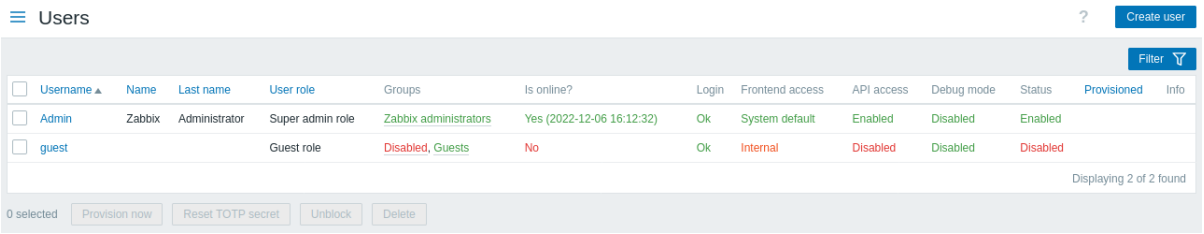
Protezione contro gli attacchi brute force

In caso di cinque tentativi consecutivi di accesso non riusciti, l'interfaccia di Zabbix verrà messa in pausa per 30 secondi per prevenire attacchi brute force e attacchi a dizionario.

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

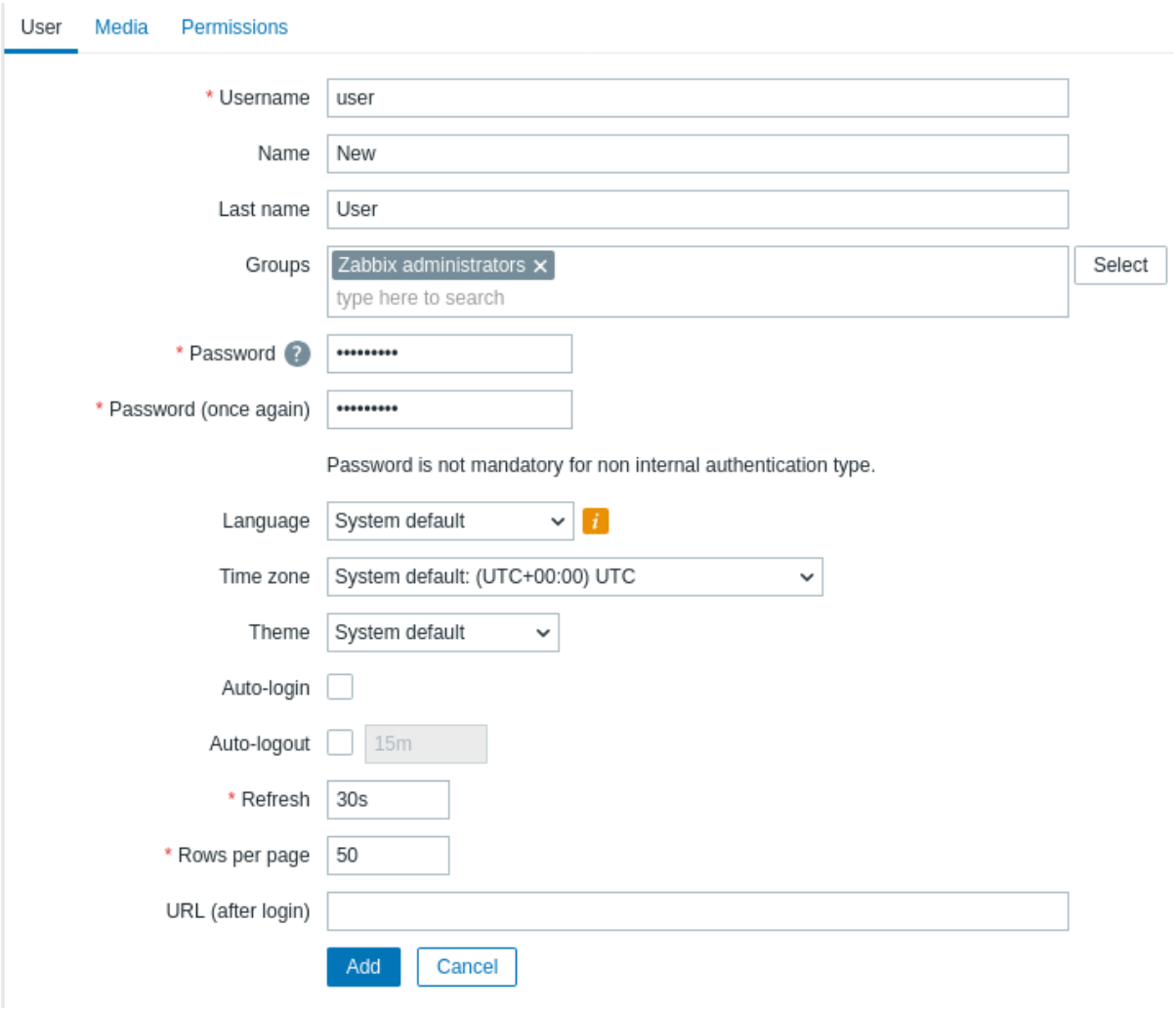
Aggiunta di un utente

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



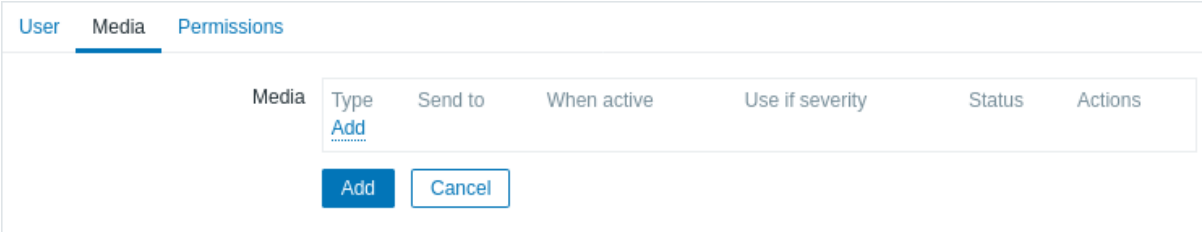
Per aggiungere un nuovo utente, seleziona *Create user* nell'angolo in alto a destra.

Nel modulo del nuovo utente, assicurati di aggiungere l'utente a uno dei user group esistenti, ad esempio 'Zabbix administrators'.



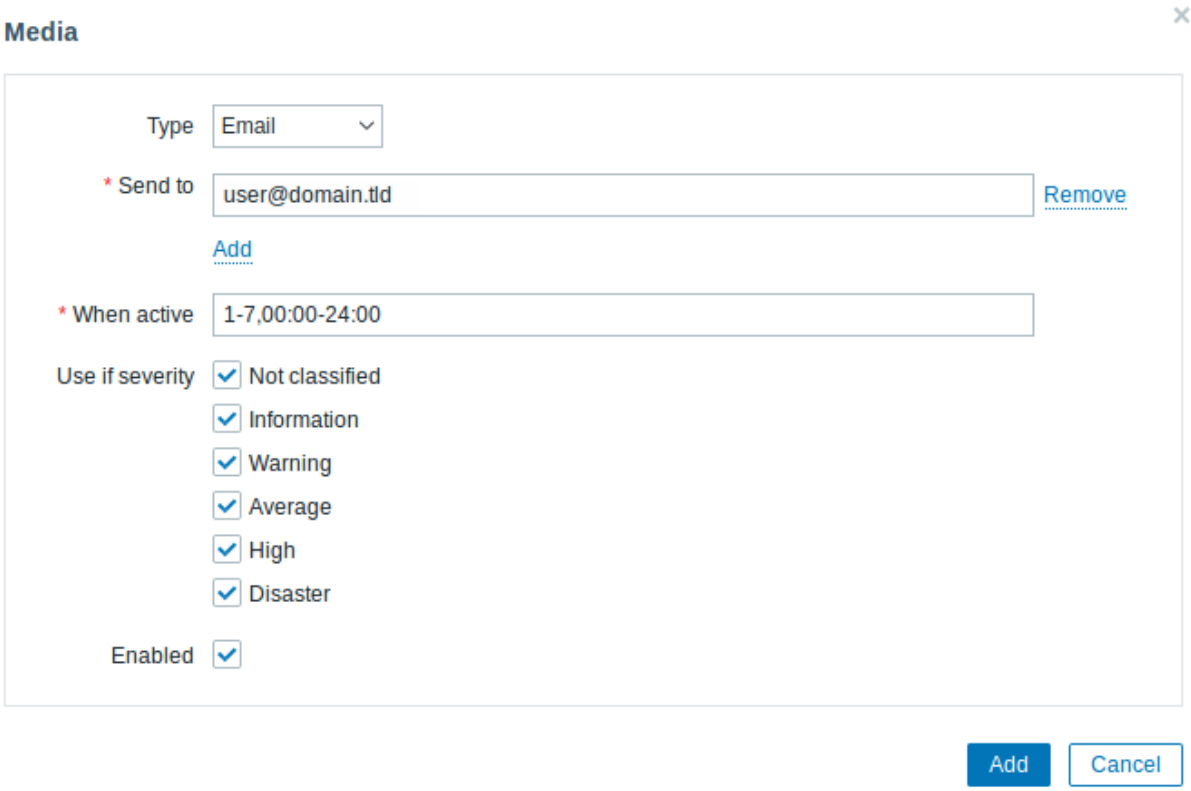
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso. Per i dettagli sui campi di input di questo modulo di configurazione, consulta la pagina User settings.

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



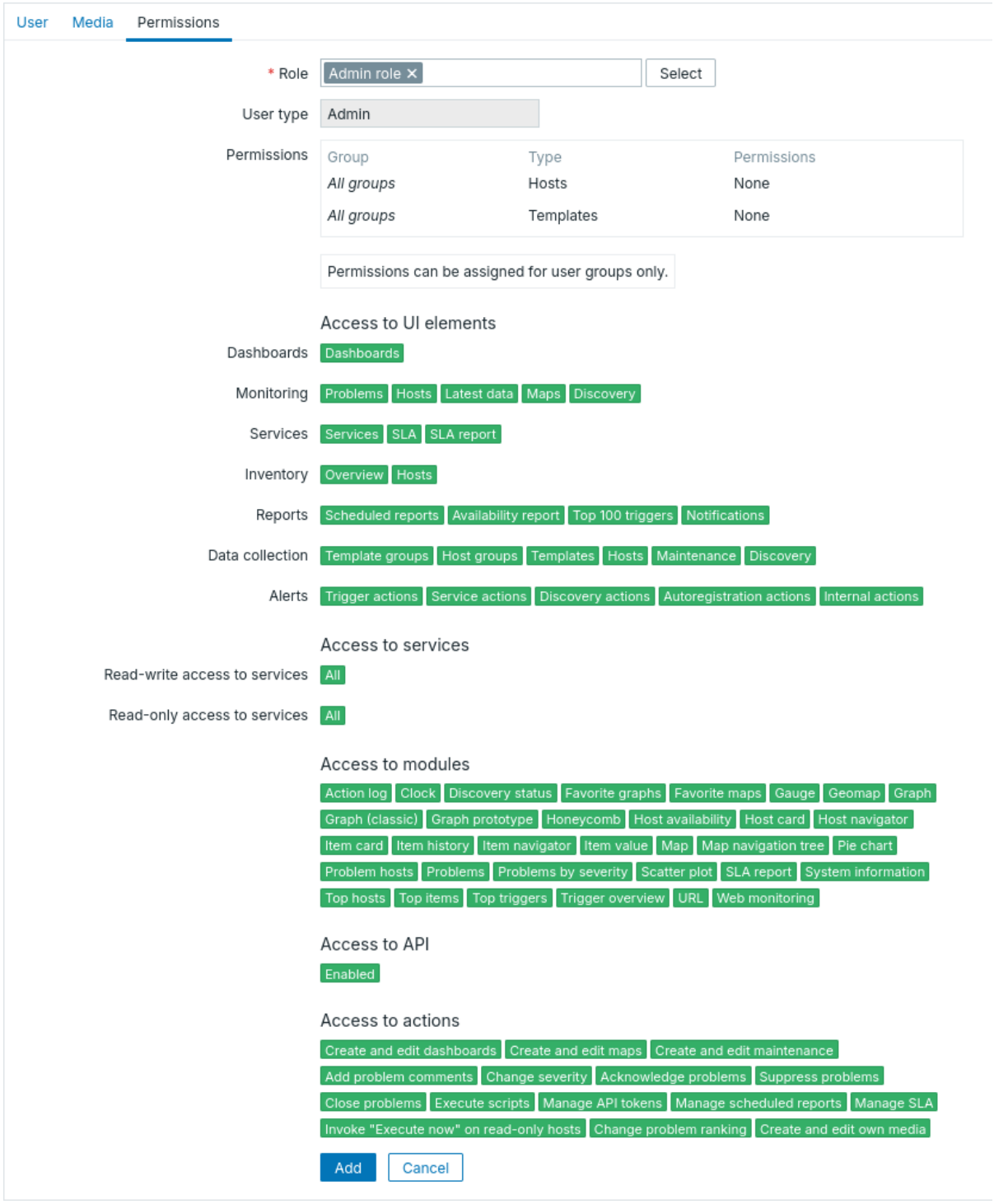
Nella finestra popup, inserisci l'indirizzo email dell'utente.

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



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

La scheda Permissions contiene un campo obbligatorio, *Role*. Il ruolo determina quali elementi del frontend l'utente può visualizzare e quali azioni gli è consentito eseguire. Premi 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, tranne Administration. In seguito, potrai modificare i permessi oppure creare altri ruoli utente. Dopo aver selezionato un ruolo, i permessi appariranno nella stessa scheda:

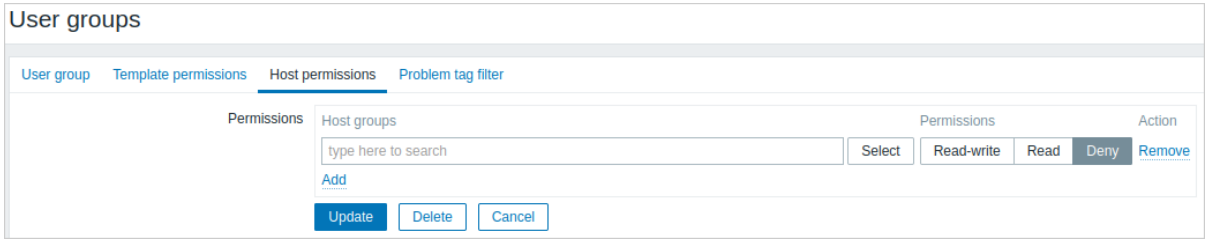


Fai clic su *Add* nel modulo delle proprietà dell'utente per salvare l'utente. Il nuovo utente compare nell'elenco utenti.



Aggiunta dei permessi

Per impostazione predefinita, un nuovo utente non dispone di alcun permesso 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 di *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 visualizzare l'elenco dei gruppi di host. Questo utente deve avere accesso in sola lettura al gruppo *Linux servers*, quindi seleziona la casella appropriata nell'elenco e fai clic su *Select* per confermare la scelta.



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

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

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

Fatto! Ora puoi provare ad accedere utilizzando le credenziali del nuovo utente.

2 Nuovo host

Panoramica

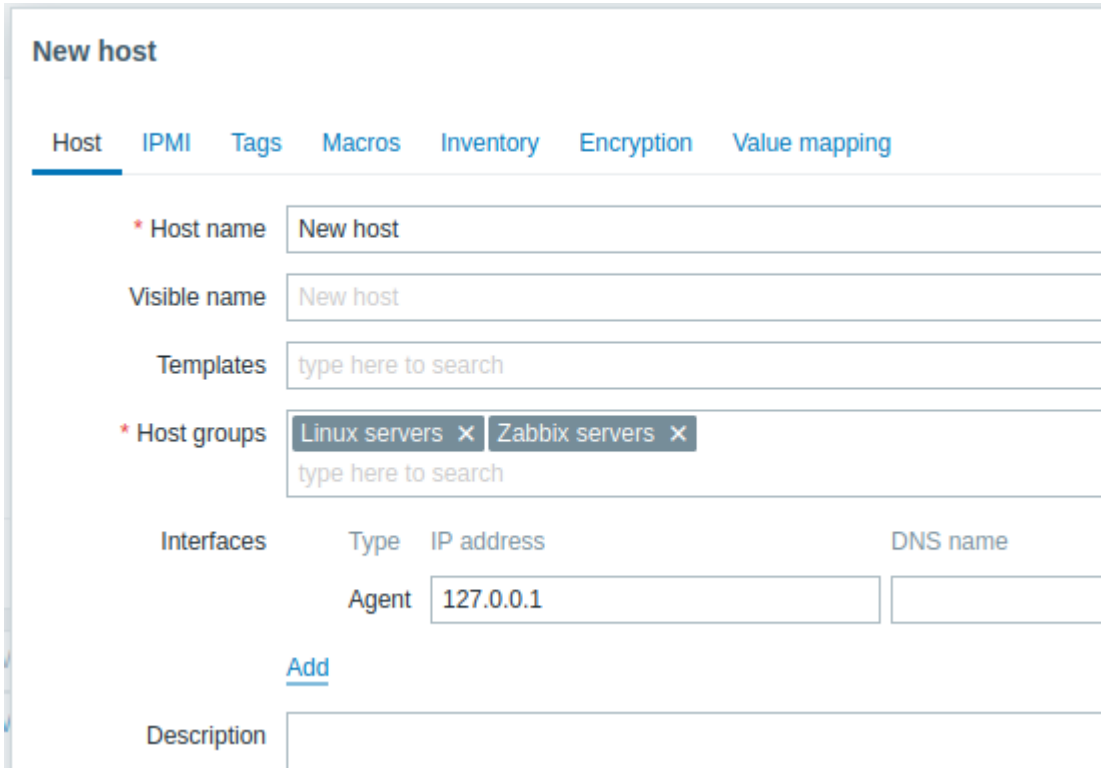
In questa sezione imparerai come configurare un nuovo host.

Un host in Zabbix è un'entità di rete (fisica, virtuale) che desideri 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 nelle sezioni di menu *Raccolta dati* > *Host* e *Monitoraggio* > *Host*. Esiste già un host predefinito, chiamato "Zabbix server", ma vogliamo imparare ad aggiungerne un altro.

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



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

Il minimo indispensabile da inserire qui è:

Nome host

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

Gruppi host

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

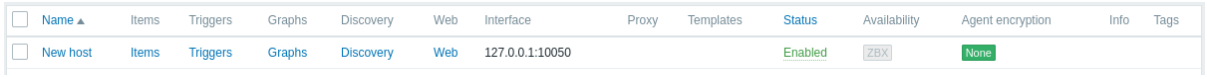
Note:
Tutti i permessi di accesso vengono assegnati ai gruppi host, non ai singoli host. Per questo motivo un host deve appartenere ad almeno un gruppo.

Interfacce: indirizzo IP

- Sebbene tecnicamente non sia un campo obbligatorio, un'interfaccia host è necessaria per raccogliere determinate metriche. Per utilizzare i controlli passivi di Zabbix agent, specificare in questo campo l'IP o il DNS dell'agent. Si noti che è necessario specificare anche l'IP o il DNS di Zabbix server nella direttiva 'Server' del file di configurazione di Zabbix agent. 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 indicatori della disponibilità dell'host per ciascuna interfaccia. Abbiamo definito un'interfaccia Zabbix agent, quindi possiamo usare l'icona di disponibilità dell'agent (con la scritta 'ZBX') per comprendere la disponibilità dell'host:

- ZBX - lo stato dell'host non è ancora stato determinato; non è stato ancora eseguito alcun controllo delle metriche
- ZBX - l'host è disponibile, un controllo delle metriche è andato a buon fine
- ZBX - l'host non è disponibile, un controllo delle metriche non è riuscito (spostare il cursore del mouse sull'icona per visualizzare il messaggio di errore). Potrebbe esserci un errore di comunicazione, probabilmente causato da credenziali dell'interfaccia errate. Verificare che Zabbix server sia in esecuzione e provare anche ad aggiornare la pagina più tardi.

3 Nuovo item

Panoramica

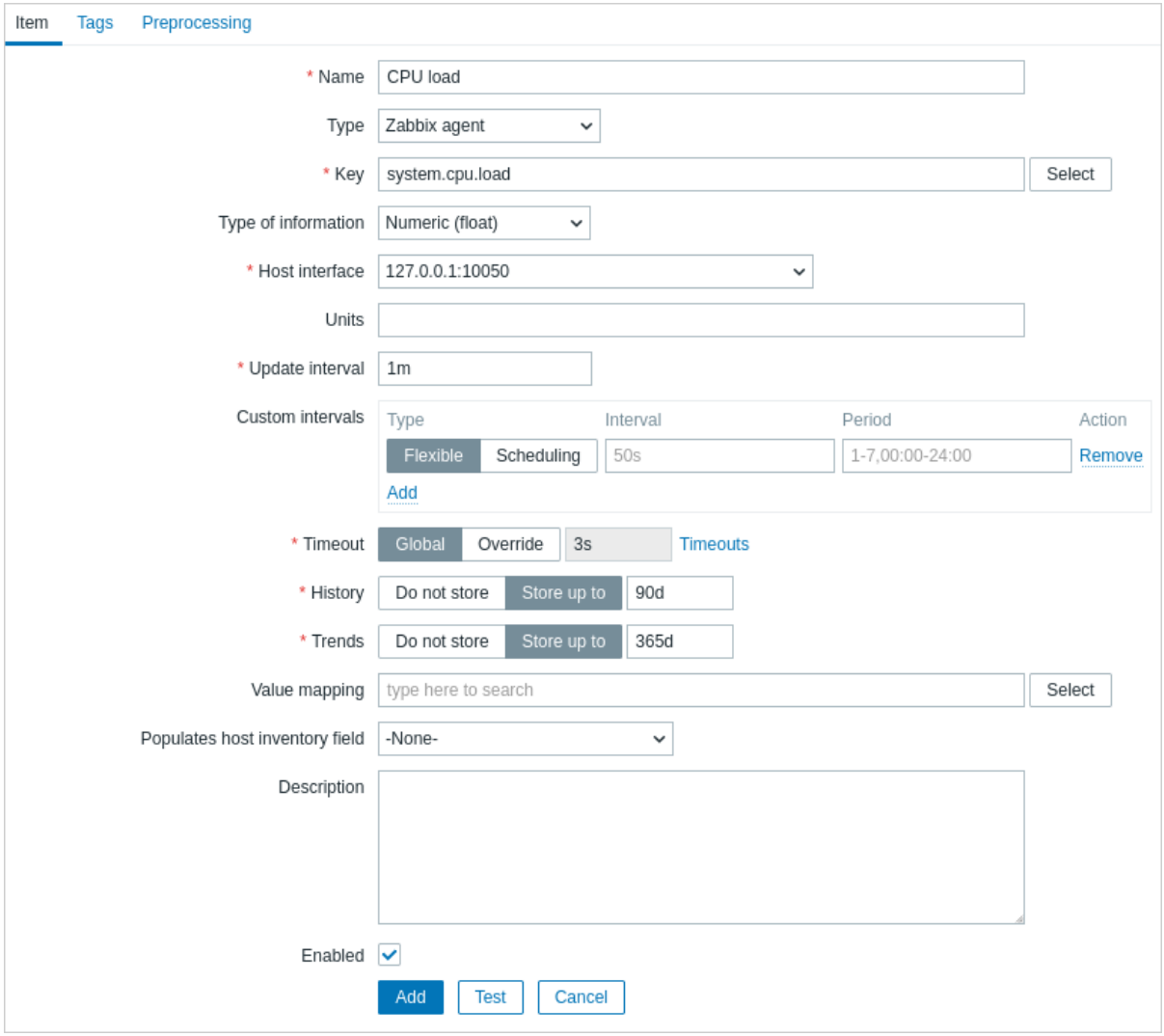
In questa sezione imparerai come configurare un item.

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

Aggiunta di un item

Tutti gli item sono raggruppati attorno agli host. Per questo motivo, per configurare un item di esempio andiamo in *Raccolta dati* > *Host* e troviamo il "Nuovo host" che abbiamo creato.

Fare clic sul collegamento *Item* nella riga di "Nuovo host", quindi fare clic su *Crea item*. Si aprirà un modulo di configurazione dell'item.



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

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

Nome

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

Chiave

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

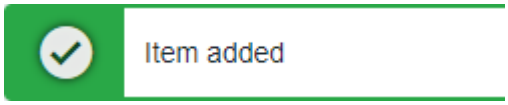
Tipo di informazione

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

Note:
Si consiglia inoltre di ridurre il numero di giorni per cui verrà conservato lo storico dell'item, a 7 o 14. Questa è una buona pratica per alleggerire il database dall'archiviazione di molti valori storici.

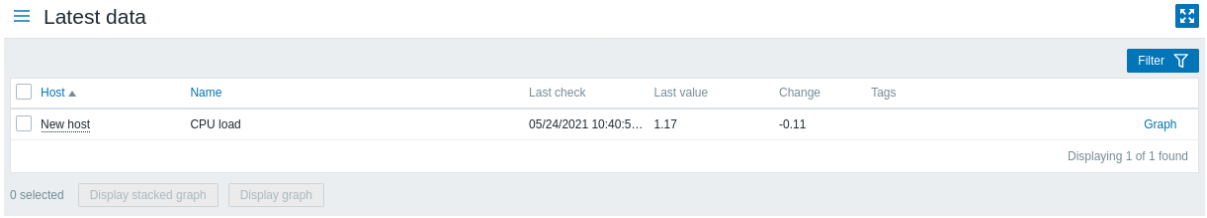
Le altre opzioni possono essere lasciate invariate, utilizzando i valori predefiniti.

Al termine, fare clic su *Aggiungi*. Il nuovo item dovrebbe comparire nell'elenco degli item e dovrebbe essere visualizzato un messaggio di successo.



Visualizzazione dei dati

Una volta definito un item, potresti voler verificare se sta effettivamente raccogliendo dati. Per farlo, vai in *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, questa è la frequenza con cui il server legge le modifiche alla configurazione e rileva i nuovi item da eseguire.

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

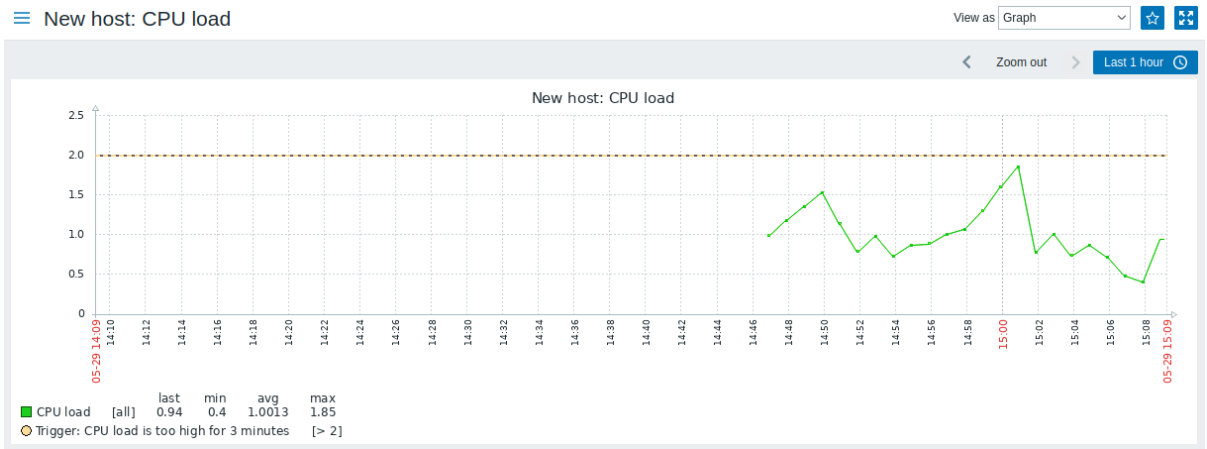
Se non visualizzi informazioni sull'item come nello screenshot, assicurati che:

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

Grafici

Dopo che l'item ha funzionato per un po', potrebbe essere il momento di vedere qualcosa di visivo. I grafici semplici sono disponibili per qualsiasi item numerico monitorato senza alcuna configurazione aggiuntiva. Questi grafici vengono generati in fase di esecuzione.

Per visualizzare il grafico, andare su *Monitoring* > *Latest data* e fare clic sul collegamento 'Graph' accanto all'item.



4 Nuovo trigger

Panoramica

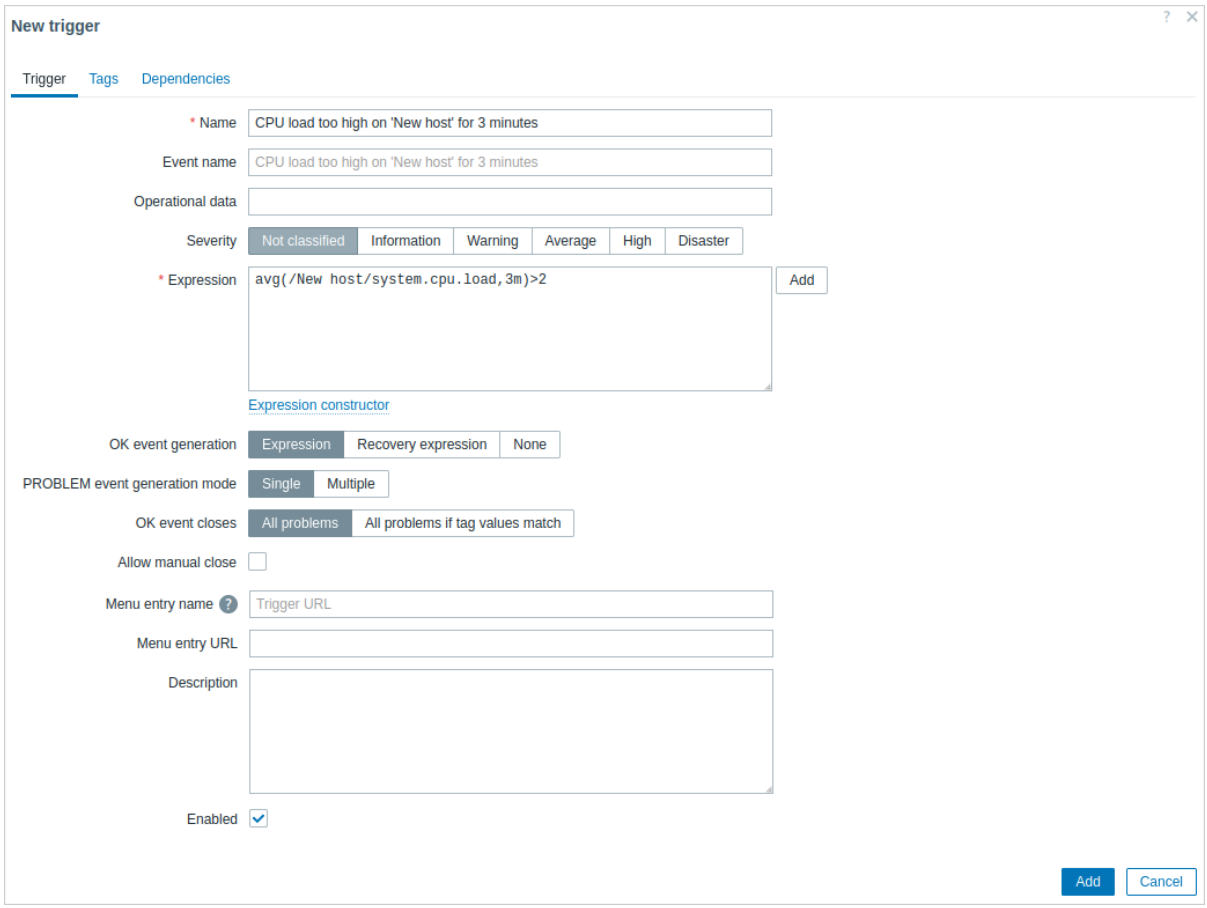
In questa sezione imparerai come configurare un trigger.

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 costituisce un livello accettabile per i dati.

Se tale livello viene superato dai dati in arrivo, un trigger "si attiverà" oppure passerà allo stato 'Problema', indicando che è successo qualcosa che potrebbe richiedere attenzione. Se il livello torna ad essere accettabile, il trigger ritorna allo stato 'Ok'.

Aggiunta di un trigger

Per configurare un trigger per il nostro item, vai su *Raccolta dati* > *Host*. Fai clic sul collegamento *Trigger* nella riga di 'New host', quindi fai clic su *Crea trigger*. Si aprirà un modulo di configurazione del trigger.



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

Nome

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

Espressione

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

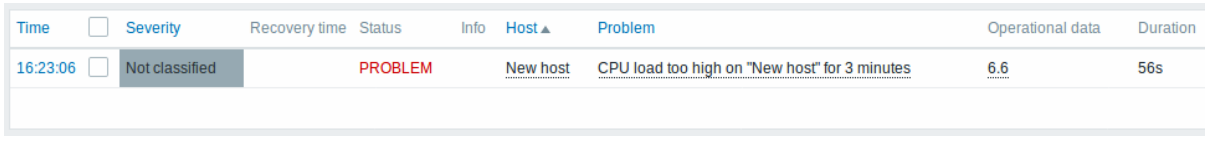
Questa è l'espressione del trigger. Assicurati che l'espressione sia inserita correttamente, fino all'ultimo simbolo. La chiave item qui (system.cpu.load) viene utilizzata per fare riferimento all'item. Questa particolare espressione dice fondamentalmente che la soglia del problema viene superata quando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

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

Visualizzazione dello stato del trigger

Con un trigger definito, potresti essere interessato a vederne lo 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 recente modifica dello stato del trigger, avvenuta negli ultimi 30 minuti.

5 Ricezione della notifica di problema

Panoramica

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

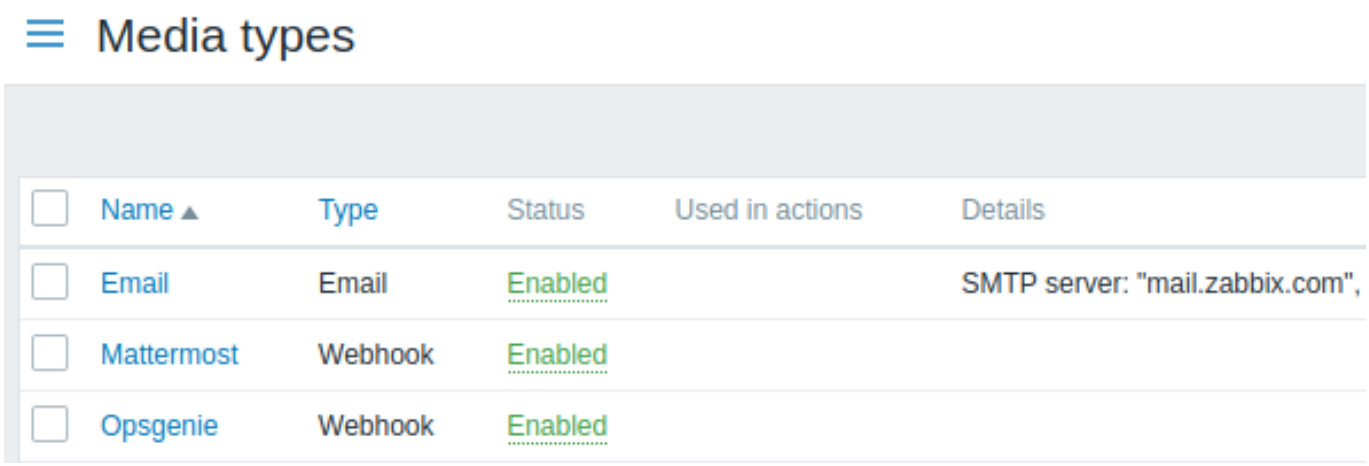
Con gli item che raccolgono dati e i trigger progettati per ”attivarsi” in caso di situazioni problematiche, è utile anche disporre di un meccanismo di avviso per notificare eventi importanti anche quando il frontend di Zabbix non viene controllato attivamente.

Questo è ciò che fanno le notifiche. Poiché l’email è il metodo di consegna più diffuso per le notifiche di problema, vedremo come configurare una notifica email.

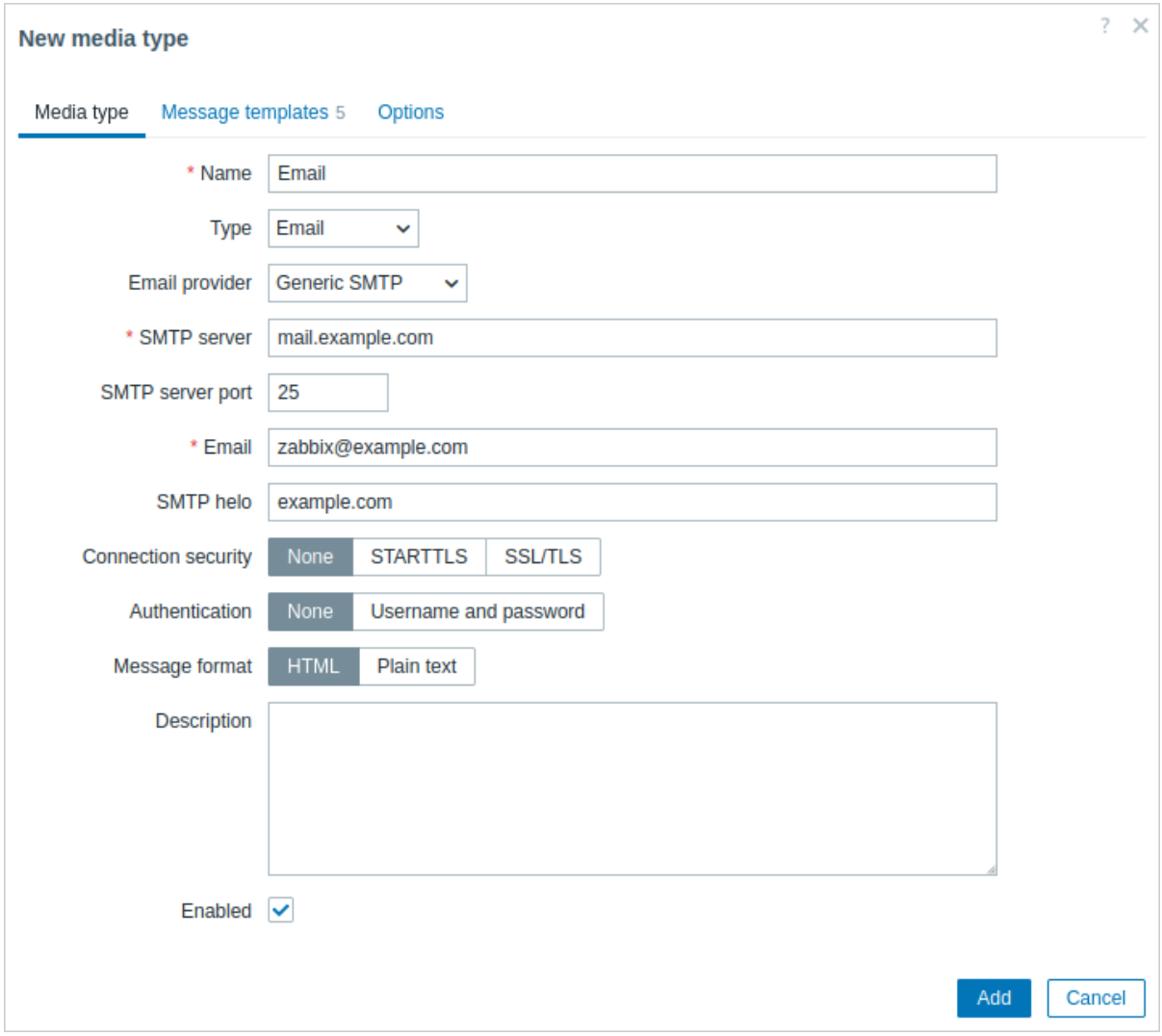
Impostazioni email

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

Per configurare le impostazioni email, vai su *Avvisi* > *Tipi di supporto* e fai clic su *Email* nell’elenco dei tipi di supporto predefiniti.



Si aprirà un modulo di configurazione delle impostazioni email.



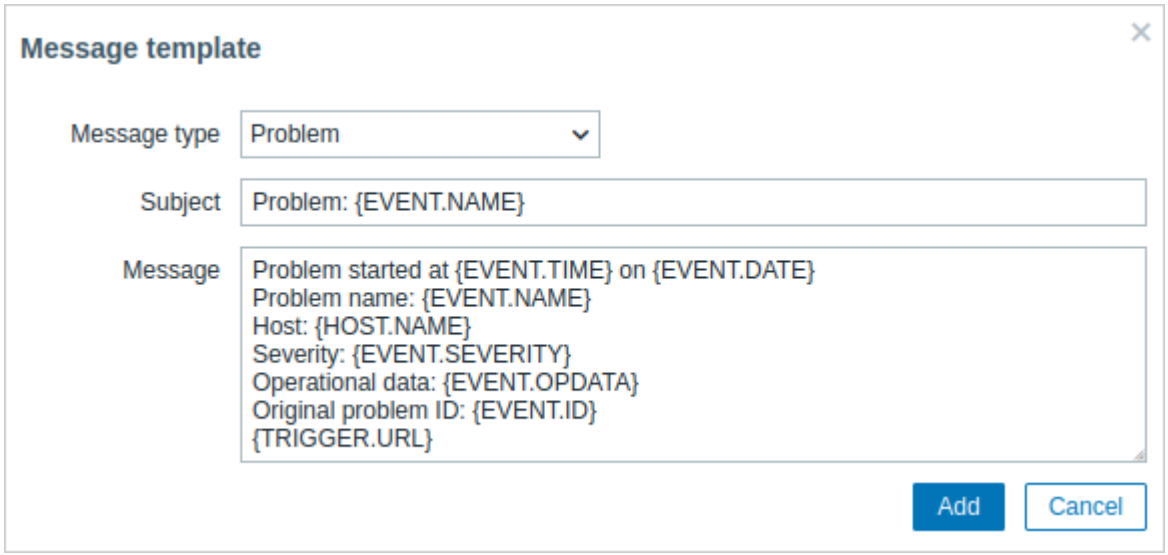
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Nella scheda *Tipo di supporto*, imposta i valori di *Server SMTP*, *SMTP helo* e *Email* in modo appropriato per il tuo ambiente.

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

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

Fai clic su *Aggiungi* per creare un template di messaggio e seleziona *Problema* come tipo di messaggio.

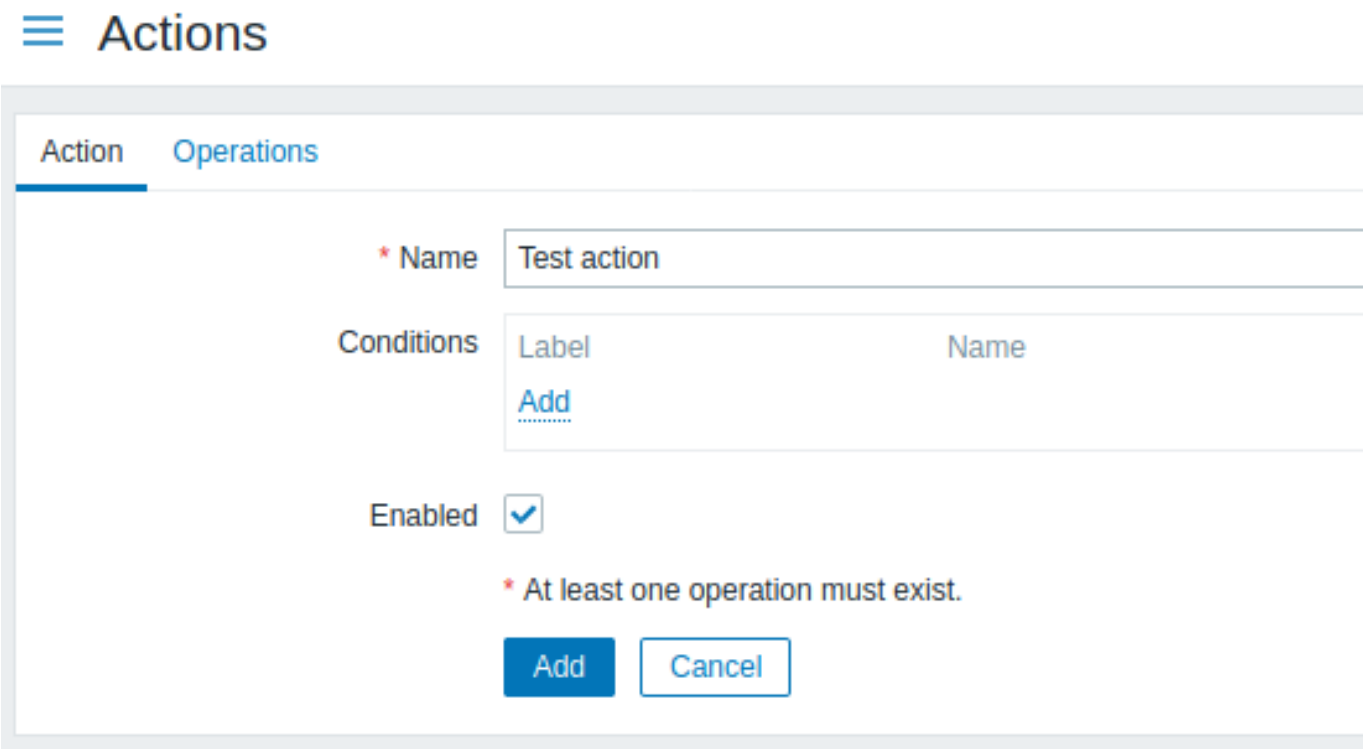


Fai clic su *Aggiungi* quando hai finito e salva il modulo.

Ora hai configurato *Email* come tipo di supporto funzionante. Il tipo di supporto deve anche essere collegato agli utenti definendo indirizzi di consegna specifici (come abbiamo fatto durante la configurazione di un nuovo utente), altrimenti non verrà utilizzato.

Nuova azione

L'invio delle notifiche è una delle funzioni svolte dalle azioni in Zabbix. Pertanto, per configurare una notifica, vai su *Avvisi* > *Azioni* > *Azioni trigger* e fai clic su *Crea azione*.

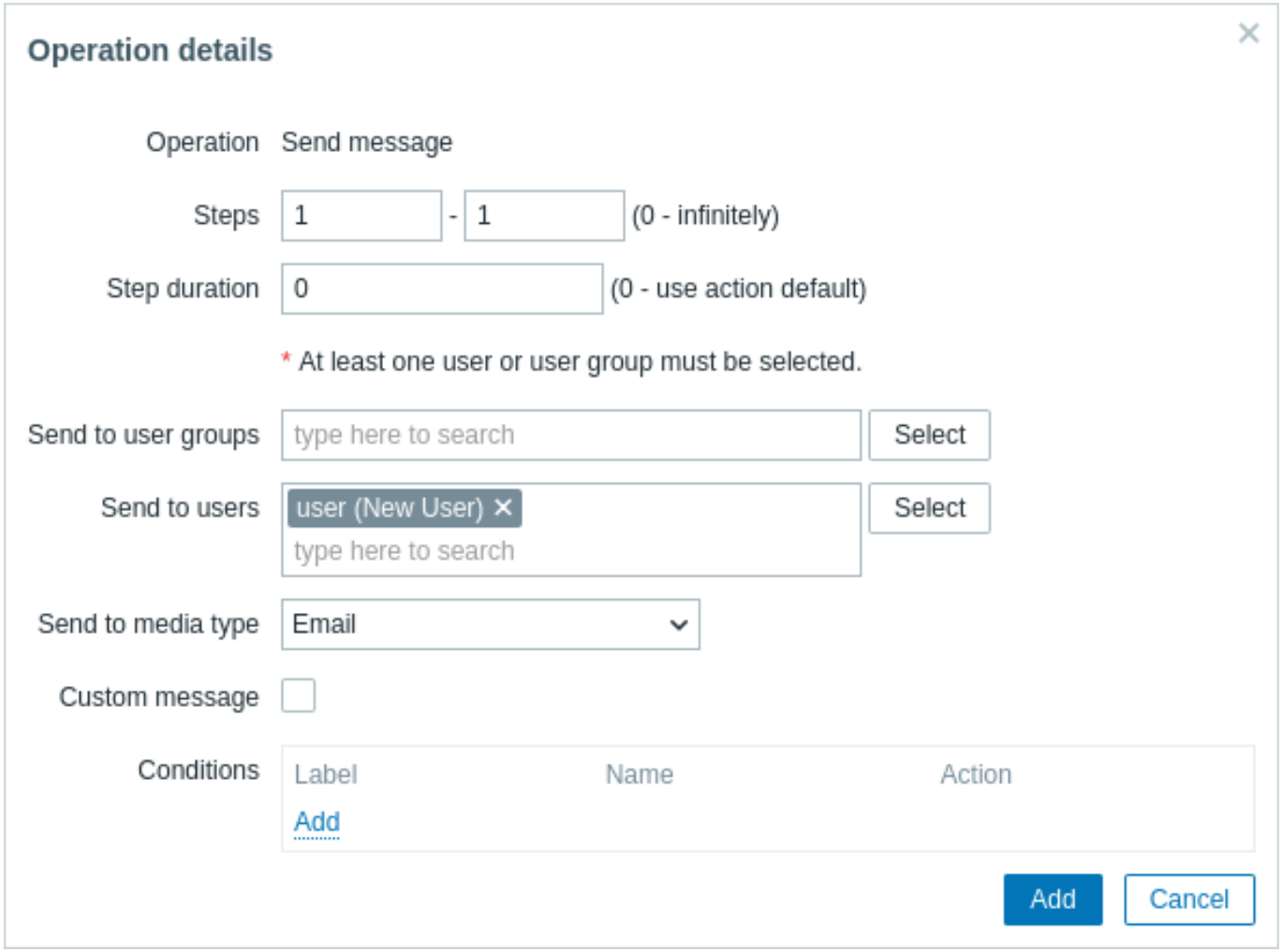


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

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

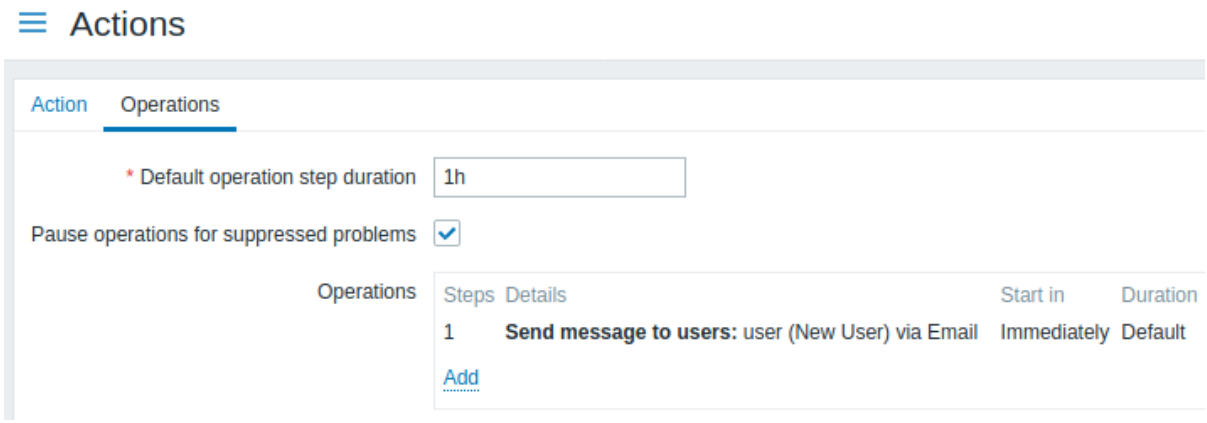
Nel caso più semplice, se non aggiungiamo condizioni più specifiche, l'azione verrà eseguita a ogni cambiamento del trigger da 'Ok' a 'Problema'.

Dobbiamo ancora definire cosa deve fare l'azione: questo viene fatto nella scheda *Operazioni*. Fai clic su *Aggiungi* nel blocco *Operazioni*, che aprirà un nuovo modulo di operazione.



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

Qui, fai clic su *Seleziona* nel blocco *Invia agli utenti* e seleziona l'utente ('user') che abbiamo definito. Seleziona "Email" come valore di *Invia al tipo di supporto*. Una volta fatto, fai clic su *Aggiungi* e l'operazione dovrebbe essere aggiunta:



Questo è tutto per una semplice configurazione dell'azione, quindi fai clic su *Aggiungi* nel modulo dell'azione.

Ricezione di una notifica

Ora che la consegna delle notifiche è configurata, sarebbe bello riceverne effettivamente una. Per facilitare ciò, possiamo aumentare di proposito il carico sul nostro host, in modo che il nostro trigger "si attivi" 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 "si attivi", il valore di 'CPU Load' deve superare '2' per 3 minuti consecutivi. Quando ciò accade:

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

Attention:
Se le notifiche non funzionano:
- verifica ancora una volta che sia le impostazioni email sia l'azione 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 *Adding user*. L'utente, facendo parte del gruppo utenti 'Zabbix administrators', deve avere almeno accesso in lettura al gruppo host 'Linux servers' a cui appartiene il nostro host.
- Inoltre, puoi controllare il log delle azioni andando in *Reports* > *Action log*.

6 Nuovo template

Panoramica

In questa sezione imparerai come configurare un template.

In precedenza abbiamo imparato come configurare un item, un trigger e come ricevere una notifica di problema per l'host.

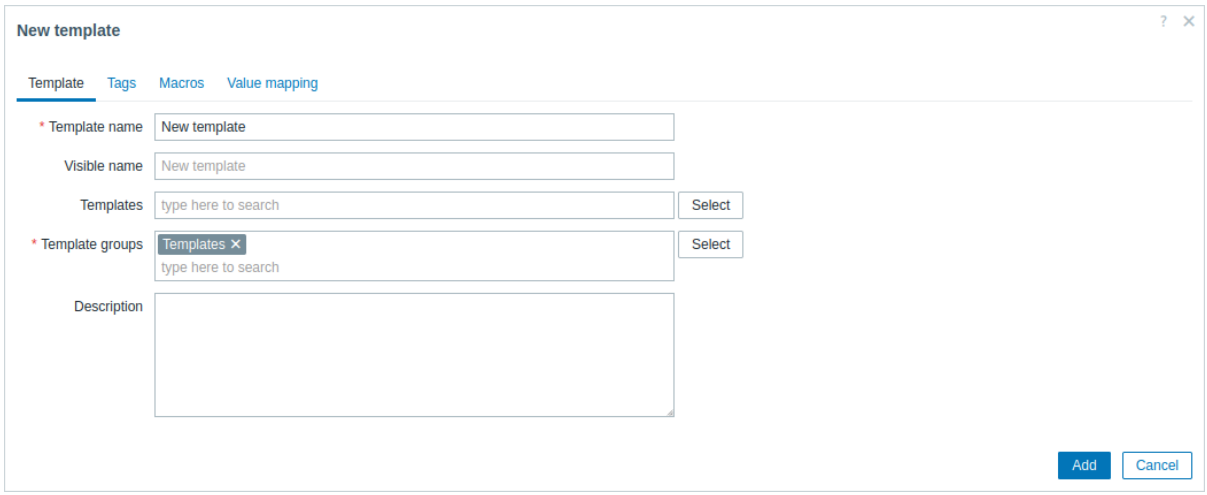
Sebbene tutti questi passaggi offrano di per sé una grande flessibilità, potrebbero sembrare molti se necessari, ad esempio, per mille host. Un po' di automazione sarebbe utile.

È qui che i template vengono in aiuto. I template consentono di raggruppare item, trigger e altre entità utili, in modo che possano essere riutilizzati più e più volte applicandoli agli host in un unico passaggio.

Quando un template è collegato a un host, l'host eredita tutte le entità del template. In pratica, un insieme di controlli già predisposto può essere applicato molto rapidamente.

Aggiunta di un template

Per iniziare a lavorare con i template, dobbiamo prima crearne uno. Per farlo, in *Raccolta dati* > *Template* fare clic su *Crea template*. Si aprirà un modulo di configurazione del template.



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

I parametri obbligatori da inserire qui sono:

Nome del template

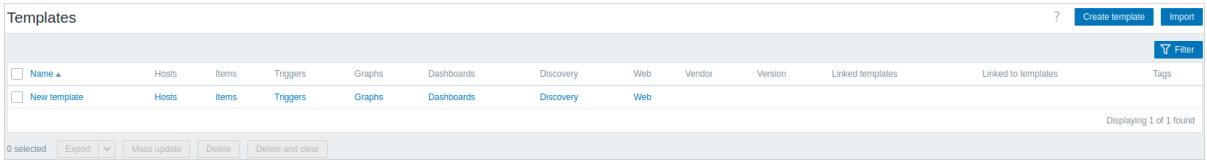
- Inserire un nome per il template. Sono consentiti caratteri alfanumerici, spazi e trattini bassi.

Gruppi di template

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

Note:
I permessi di accesso ai gruppi di template vengono assegnati nella configurazione del gruppo utenti, nella scheda **Permessi template**, nello stesso modo dei permessi dei 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 *Aggiungi*. Il nuovo template dovrebbe essere visibile nell'elenco dei template. È anche possibile usare il filtro per trovare il proprio template.



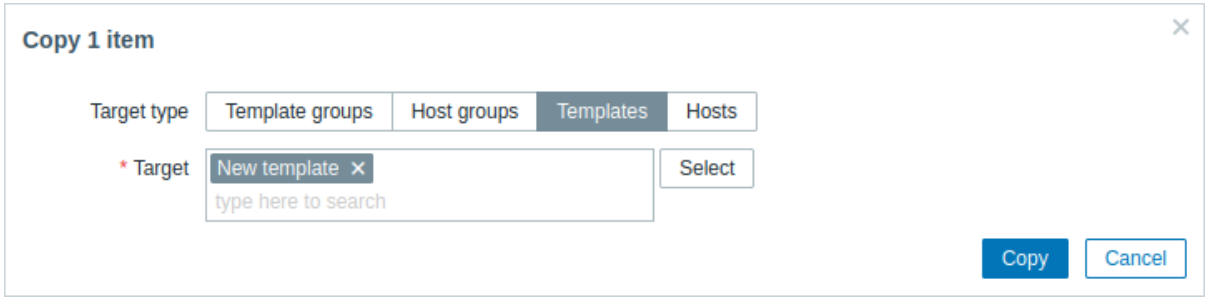
Come si può vedere, il template è presente, ma non contiene ancora nulla: nessun item, trigger o altre entità.

Aggiunta di un item al template

Per aggiungere un item al template, aprire l'elenco degli item di 'New host' andando su *Data collection* → *Hosts* e facendo clic su *Items* accanto a 'New host'.

Quindi:

- Selezionare la casella di controllo dell'item 'CPU Load' nell'elenco.
- Fare clic su *Copy* sotto l'elenco.
- Selezionare la scheda *Templates*.
- Selezionare il template in cui copiare l'item.



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

- Fare clic su *Copy*.

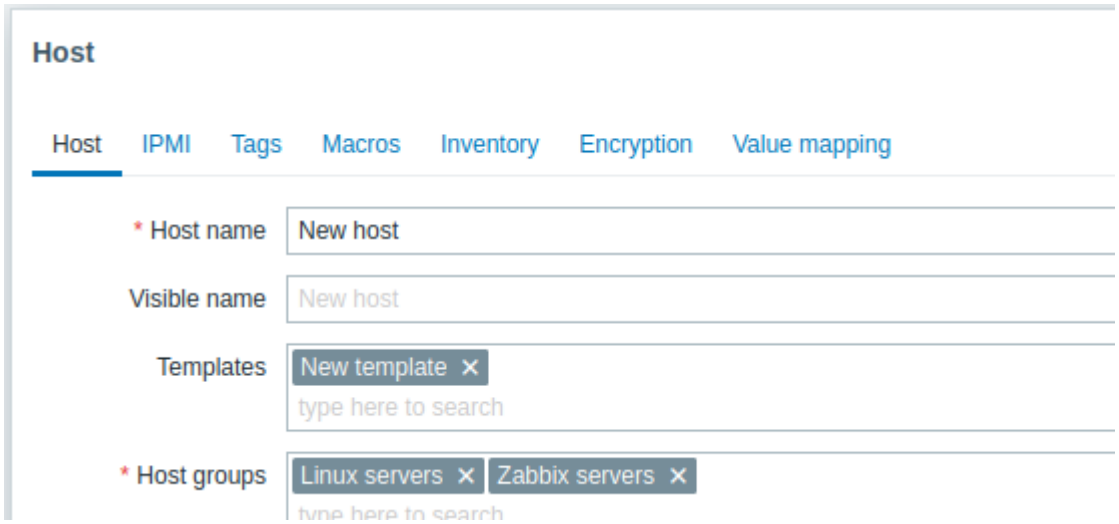
Se ora si va su *Data collection* > *Templates*, 'New template' dovrebbe contenere un nuovo item.

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

Collegamento di un template a un host

Con il template pronto, non resta che aggiungerlo a un host. Per farlo, vai su *Data collection* > *Hosts*, fai clic su 'New host' per aprire il relativo modulo di configurazione e individua il campo **Templates**.

Inizia a digitare *New template* nel campo *Templates*. Il nome del template che abbiamo creato dovrebbe comparire nell'elenco a discesa. Scorri verso il basso per selezionarlo. Verifica che compaia nel campo *Templates*.



Fai clic su *Update* 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. Qualsiasi modifica agli item, trigger e ad altre entità a livello di template verrà propagata 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 il monitoraggio molto rapidamente, puoi collegare quello più appropriato a un host, ma tieni presente che questi template devono essere ottimizzati in base al 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 server web Apache

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

1. Assicurati di avere accesso alle statistiche di Apache.
2. Applica il template di monitoraggio di Apache.
3. Controlla gli ultimi dati.

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 server web 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 server web Apache, eseguire:

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

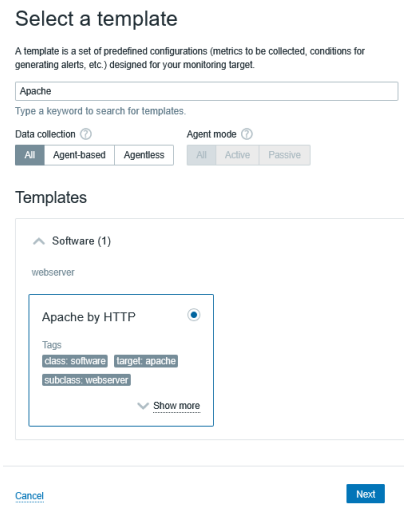
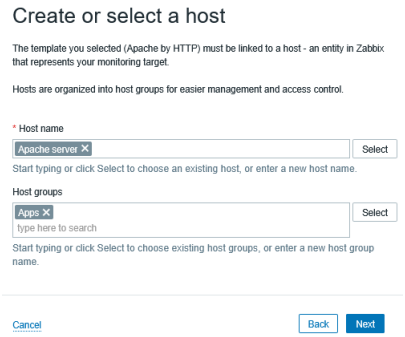
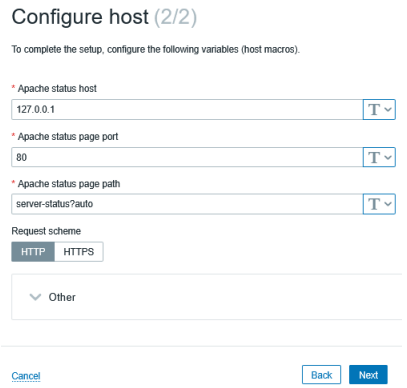
Se nell'output viene visualizzato "`status_module (shared)`", il modulo status è già abilitato. In caso contrario, è necessario abilitare il modulo status.

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, avviare la Procedura guidata host (in *Raccolta dati* > *Host*):

- **Selezionare il template** ("Apache by HTTP")
- Creare un host Apache e aggiungervi un gruppo. Questo host virtuale conterrà i dati relativi ad Apache.
- Configurare l'host. In particolare, inserire l'indirizzo del server web Apache (127.0.0.1 su una macchina locale, indirizzo IP su una macchina remota).

Passo 1	Passo 2	Passo 3
		
Selezione del template.	Creazione di host/gruppo host.	Configurazione dell'host.

È possibile lasciare le altre impostazioni ai valori predefiniti.

Per completare la Procedura guidata host, fare clic su *Create* e *Finish*.

Visualizza gli ultimi dati



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

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

Monitoraggio di 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. Controlla gli ultimi dati.

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

Sul server Linux verrà utilizzato un agent Zabbix per monitorarlo. Le istruzioni per installare l'agent verranno fornite nel corso della 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*.

Applicare il template di monitoraggio Linux

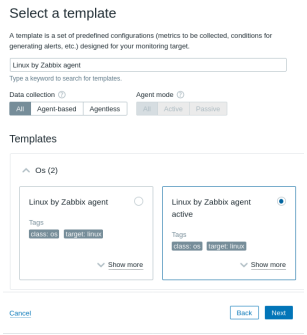
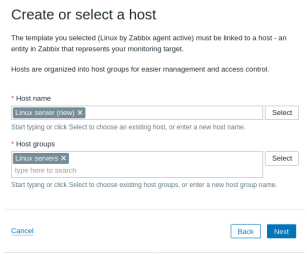
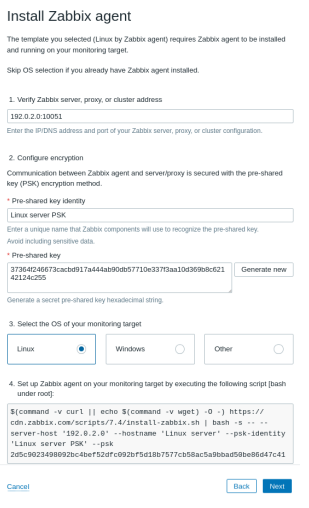
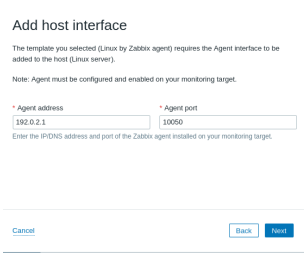
In Zabbix questo template è pronto all'uso. Contiene elementi preconfigurati pronti per la raccolta dei dati, la visualizzazione e l'analisi.

Per applicare il template, avviare Host Wizard (in *Data collection* > *Hosts*):

- **Selezionare il template** ("Linux by Zabbix agent active")
- Creare un host per il server Linux e aggiungere un gruppo per esso. Questo host virtuale conterrà tutti i dati relativi al monitoraggio del server Linux.
- Configurare e installare l'agent. Per installare l'agent, eseguire sul server Linux il comando fornito. Questo installerà l'agent e configurerà la connessione dell'agent al server Zabbix con crittografia.

Se l'agent è già installato, è possibile saltare questo passaggio (fare clic su *Next*). Tuttavia, verrà richiesto di fornire l'identità della chiave precondivisa e la chiave precondivisa per connettersi all'agent.

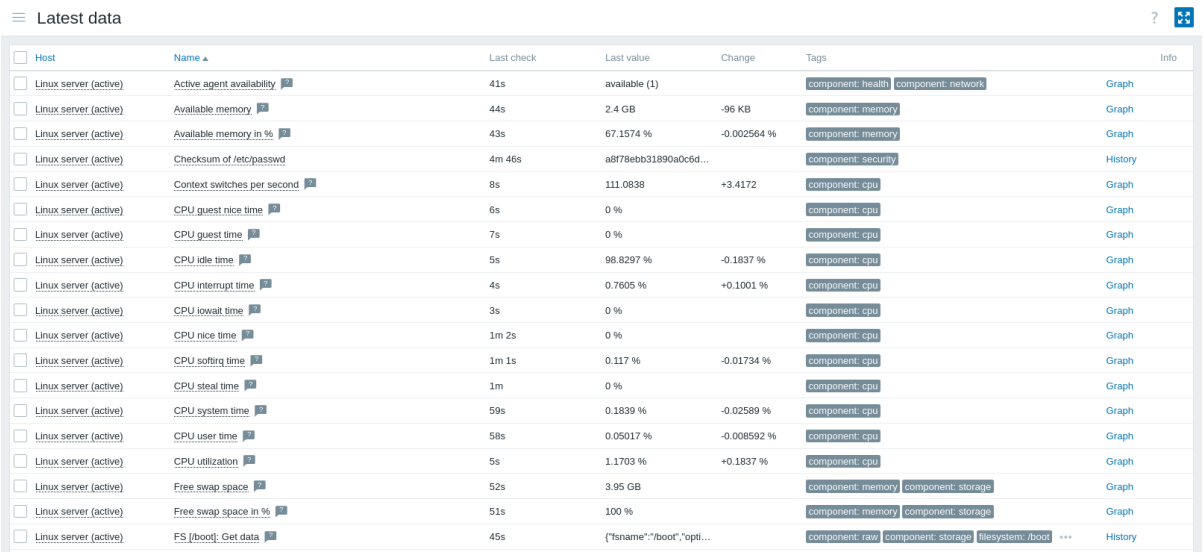
- Completare la configurazione dell'host. Immettere l'indirizzo IP e la porta dell'agent (server Linux).

Passaggio 1	Passaggio 2	Passaggio 3	Passaggio 4
			
Selezione del template.	Creazione di host/gruppo host.	Configurazione e installazione dell'agent sull'host.	Aggiunta dell'indirizzo IP/porta del server Linux come interfaccia agent.

È possibile lasciare le altre impostazioni ai valori predefiniti.

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

Visualizza gli ultimi dati



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, vedere Ricezione delle notifiche di problema.

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 gli ultimi dati.

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

Per monitorare il server MySQL verrà utilizzato un agent Zabbix. Le istruzioni per installare l'agent saranno fornite nel corso della procedura.

Per utilizzare questa guida, è necessario che il server Zabbix e l'interfaccia web siano installati. Devi aver effettuato l'accesso a Zabbix come utente predefinito *Admin*.

Creare un utente MySQL con privilegi limitati

Si consiglia di creare un utente MySQL con permessi limitati solo a quelli necessari per il monitoraggio di MySQL.

```
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.

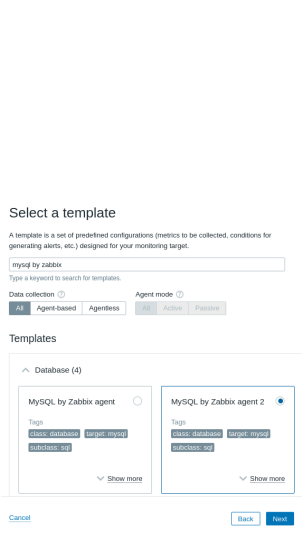
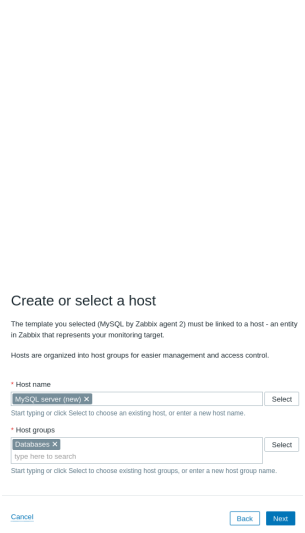
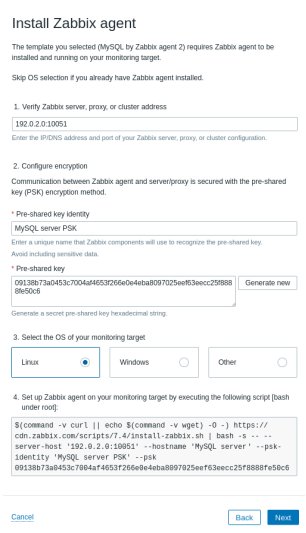
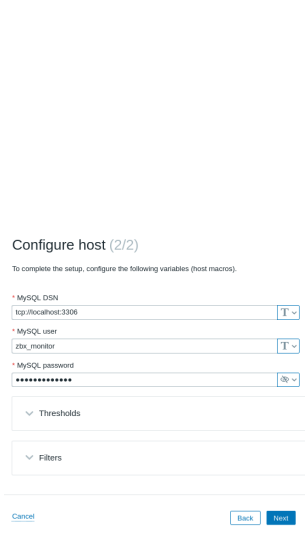
Per applicare il template, avviare il Host Wizard (in *Data collection* > *Hosts*):

- **Selezionare il template** ("MySQL by Zabbix agent 2")
- Creare un host server MySQL e aggiungere un gruppo per esso. Questo host virtuale conterrà tutti i dati relativi al monitoraggio del server MySQL.
- Configurare e installare agent 2 (Zabbix agent 2 fornisce il monitoraggio MySQL immediatamente).

Per installare l'agent, eseguire il comando fornito sulla macchina che ospita il server MySQL. Verrà installato l'agent e sarà configurata la connessione dell'agent al server Zabbix con crittografia.

Se agent 2 è già installato, è possibile saltare questo passaggio (fare clic su *Next*). Tuttavia, verrà richiesto di fornire l'identità della chiave precondivisa e la chiave precondivisa per connettersi all'agent.

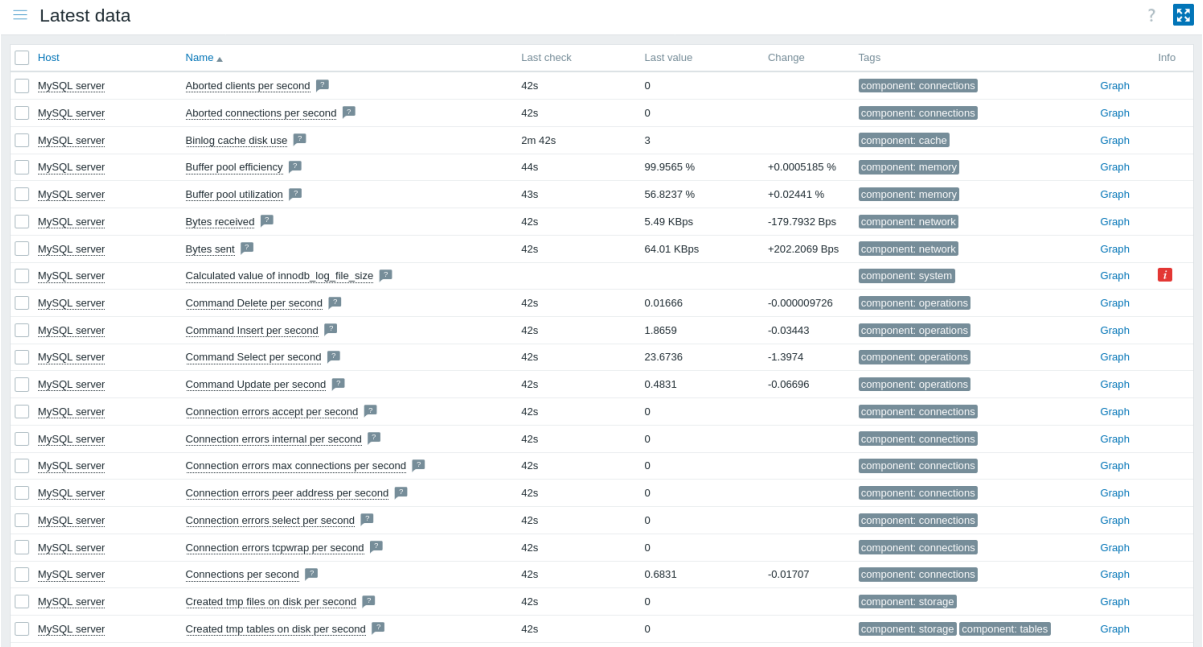
- Completare la configurazione dell'host. È importante notare che verrà richiesto di inserire il DSN del server MySQL, l'utente MySQL e la password. Fornire l'utente `zbx_monitor` con la password creata in precedenza.

Passo 1	Passo 2	Passo 3	Passo 4
			
Selezione del template.	Creazione di host/gruppo host.	Configurazione e installazione dell'agent sull'host.	Aggiunta di DNS del server MySQL, utente e password.

È possibile lasciare le altre impostazioni ai valori predefiniti.

Per completare l'Host Wizard, fare clic su *Create* e *Finish*.

Visualizza gli ultimi dati



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 tramite email, vedere Ricezione delle notifiche di problema.

3 Installazione: guide aggiuntive

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

Le guide seguenti descrivono metodi **aggiuntivi** o specifici per sistema per installare Zabbix o i suoi componenti:

- Installazione da container
- Installazione dai sorgenti
- Note sull'installazione da 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
- Compilazione di Zabbix agent su macOS
- Zabbix appliance

Installazione da container

Panoramica

Questa sezione descrive come distribuire Zabbix con Docker o Docker Compose.

Zabbix fornisce ufficialmente:

- Immagini Docker separate per ciascun componente di Zabbix, da eseguire come container portabili e autosufficienti.
- File Compose per definire ed eseguire componenti Zabbix multi-container in Docker.

Attention:
A partire da Zabbix 6.0, i trigger deterministici devono essere creati durante l'installazione. Se il binary logging è abilitato per MySQL/MariaDB, ciò richiede privilegi di superuser 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 Docker. In questo caso, mantenere in esecuzione solo il servizio SQL, arrestare soltanto 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.

File sorgente

I sorgenti dei file Docker sono archiviati nel repository ufficiale di Zabbix su GitHub, dove è possibile seguire le ultime modifiche ai file o fare un fork del progetto per creare le proprie immagini.

Docker

Zabbix fornisce immagini basate su una varietà di immagini OS di base. Per ottenere l'elenco delle immagini del sistema operativo di base supportate per uno specifico componente Zabbix, vedere la descrizione del componente in Docker Hub. Tutte le immagini Zabbix sono configurate per ricostruire le immagini più recenti se le immagini di base vengono aggiornate.

Installazione

Per ottenere l'immagine del componente Zabbix, eseguire:

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

Sostituire `zabbix/zabbix-server-mysql` con il nome del repository docker richiesto.

Questo comando scaricherà l'ultima versione stabile del componente Zabbix basata sul sistema operativo Alpine Linux. È possibile aggiungere dei tag al nome del repository per ottenere un'immagine basata su un altro sistema operativo oppure su una specifica versione major o minor di Zabbix.

I seguenti repository sono disponibili in Docker Hub:

Componente	Repository Docker
Zabbix agent	zabbix/zabbix-agent
Zabbix server	
con supporto MySQL	zabbix/zabbix-server-mysql
con supporto PostgreSQL	zabbix/zabbix-server-pgsql
Interfaccia web di Zabbix	
basata sul server web Apache2 con supporto MySQL	zabbix/zabbix-web-apache-mysql
basata sul server web Apache2 con supporto PostgreSQL	zabbix/zabbix-web-apache-pgsql
basata sul server web Nginx con supporto MySQL	zabbix/zabbix-web-nginx-mysql
basata sul server web Nginx con supporto PostgreSQL	zabbix/zabbix-web-nginx-pgsql
Zabbix proxy	
con supporto SQLite3	zabbix/zabbix-proxy-sqlite3
con supporto MySQL	zabbix/zabbix-proxy-mysql
Zabbix Java gateway	zabbix/zabbix-java-gateway

Note:
Il supporto per le trap SNMP è fornito in un repository separato zabbix/zabbix-snmptraps. Può essere collegato a Zabbix server e Zabbix proxy.

Tag

Le immagini ufficiali dei componenti Zabbix possono contenere i seguenti tag:

Tag	Descrizione	Esempio
latest	L'ultima versione stabile di un componente Zabbix basata sull'immagine Alpine Linux.	zabbix-agent:latest
<OS>-trunk	L'ultima build nightly della versione di Zabbix attualmente in fase di sviluppo su uno specifico sistema operativo. **<OS>** - il sistema operativo di base. Valori supportati: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (solo agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (solo agent); *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	L'ultima versione stabile di un componente Zabbix su uno specifico sistema operativo. **<OS>** - il sistema operativo di base. Valori supportati: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (solo agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (solo agent); *ubuntu* - Ubuntu	zabbix-agent:ol-latest
<OS>-X.X-latest	L'ultima versione secondaria di un componente Zabbix di una specifica versione principale e sistema operativo. **<OS>** - il sistema operativo di base. Valori supportati: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (solo agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (solo agent); *ubuntu* - Ubuntu **X.X** - la versione principale di Zabbix (ad es. *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	La versione secondaria specifica di un componente Zabbix di una specifica versione principale e sistema operativo. **<OS>** - il sistema operativo di base. Valori supportati: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (solo agent); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (solo agent); *ubuntu* - Ubuntu **X.X** - la versione principale di Zabbix (ad es. *6.0*, *7.4*, *8.0*). * - la versione secondaria di Zabbix	zabbix-agent:alpine-8.0.1

Configurazione iniziale

Dopo aver scaricato le immagini, avviare i container eseguendo il comando `docker run` seguito da argomenti aggiuntivi per specificare le variabili d'ambiente e/o i punti di mount richiesti. Di seguito sono riportati alcuni esempi di configurazione.

Note:
Per abilitare la comunicazione tra i componenti di Zabbix, alcune porte, come 10051/TCP per Zabbix server (trapper), 10050/TCP per Zabbix agent, 162/UDP per gli SNMP trap e 80/TCP per l'interfaccia web di Zabbix, saranno esposte a una macchina host. L'elenco completo delle porte predefinite utilizzate dai componenti di Zabbix è disponibile nella pagina Requisiti. Per Zabbix server e agent, la porta predefinita può essere modificata impostando la variabile d'ambiente ZBX_LISTENPORT environment variable.

Variabili d'ambiente

Tutte le immagini dei componenti Zabbix forniscono variabili d'ambiente per controllare la configurazione. Le variabili d'ambiente supportate sono elencate nel repository del componente.

Queste variabili d'ambiente corrispondono alle opzioni dei file di configurazione di Zabbix, ma con un metodo di denominazione diverso. Ad esempio, `ZBX_LOGSLOWQUERIES` equivale a `LogSlowQueries` nei file di configurazione di Zabbix server o Zabbix proxy.

Attention:
Alcune opzioni di configurazione (ad esempio `PIDFile` e `LogType`) non possono essere modificate.

Le seguenti variabili d'ambiente sono specifiche dei componenti Docker e non esistono nei file di configurazione di Zabbix:

Variabile	Componenti	Valore predefinito	Descrizione
`DB_SERVER_HOST`	Server Proxy Interfaccia web	`mysql-server` per MYSQL `postgres-server` per PostgreSQL	Nome IP o DNS del server MySQL o PostgreSQL.
`DB_SERVER_PORT`	Server Proxy Interfaccia web	3306 per MYSQL 5432 per PostgreSQL	Porta del server MySQL o PostgreSQL.
`MYSQL_USER`	Server Proxy Interfaccia web	`zabbix`	Utente del database MySQL.
`MYSQL_PASSWORD`	Server Proxy Interfaccia web	`zabbix`	Password del database MySQL.
`MYSQL_DATABASE`	Server Proxy Interfaccia web	`zabbix` per Zabbix server `zabbix_proxy` per Zabbix proxy	Nome del database Zabbix.
`POSTGRES_USER`	Server Interfaccia web	`zabbix`	Utente del database PostgreSQL.
`POSTGRES_PASSWORD`	Server Interfaccia web	`zabbix`	Password del database PostgreSQL.
`POSTGRES_DB`	Server Interfaccia web	`zabbix` per Zabbix server `zabbix_proxy` per Zabbix proxy	Nome del database Zabbix.
`PHP_TZ`	Interfaccia web	`Europe/Riga`	Fuso orario in formato PHP. L'elenco completo dei fusi orari supportati è disponibile su php.net.
`ZBX_SERVER_NAME`	Interfaccia web	`Zabbix Docker`	Nome visibile dell'installazione Zabbix sotto il logo Zabbix nel menu verticale dell'interfaccia web.
`ZBX_JAVAGATEWAY_ENABLE`	Server Proxy	`false`	Abilita la comunicazione con Zabbix Java gateway per raccogliere controlli relativi a Java.
`ZBX_ENABLE_SNMP_TRAPS`	Server Proxy	`false`	Abilita la funzionalità SNMP trap. Richiede un'istanza **zabbix-snmptraps** e il volume condiviso */var/lib/zabbix/snmptraps* per Zabbix server o Zabbix proxy.

Volumi

Le immagini consentono di montare volumi utilizzando i seguenti punti di mount:

Volume	Descrizione
Zabbix agent	
/etc/zabbix/zabbix_agentd.d	Il volume consente di includere file **.conf* ed estendere Zabbix agent utilizzando la funzionalità `UserParameter`
/var/lib/zabbix/modules	Il volume consente di caricare moduli aggiuntivi ed estendere Zabbix agent utilizzando la funzionalità LoadModule

Volume	Descrizione
/var/lib/zabbix/enc	Il volume viene utilizzato per memorizzare file relativi a TLS. I nomi di questi file sono specificati utilizzando le variabili d'ambiente ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE e ZBX_TLSPSKFILE
Zabbix server	
/usr/lib/zabbix/alertscripts	Il volume viene utilizzato per script di avviso personalizzati. Corrisponde al parametro AlertScriptsPath in zabbix_server.conf
/usr/lib/zabbix/externalscripts	Il volume viene utilizzato da controlli esterni. Corrisponde al parametro ExternalScripts in zabbix_server.conf
/var/lib/zabbix/modules	Il volume consente di caricare moduli aggiuntivi ed estendere Zabbix server utilizzando la funzionalità LoadModule
/var/lib/zabbix/enc	Il volume viene utilizzato per memorizzare file relativi a TLS. I nomi di questi file sono specificati utilizzando le variabili d'ambiente ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE e ZBX_TLSPSKFILE
/var/lib/zabbix/ssl/certs	Il volume viene utilizzato come posizione dei file di certificato client SSL per l'autenticazione del client. Corrisponde al parametro SSLCertLocation in zabbix_server.conf
/var/lib/zabbix/ssl/keys	Il volume viene utilizzato come posizione dei file di chiave privata SSL per l'autenticazione del client. Corrisponde al parametro SSLKeyLocation in zabbix_server.conf
/var/lib/zabbix/ssl/ssl_ca	Il volume viene utilizzato come posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server. Corrisponde al parametro SSLCALocation in zabbix_server.conf
/var/lib/zabbix/snmptraps	Il volume viene utilizzato come posizione del file snmptraps.log. Può essere condiviso dal container zabbix-snmptraps ed ereditato utilizzando l'opzione Docker volumes_from durante la creazione di una nuova istanza di Zabbix server. La funzionalità di elaborazione delle trap SNMP può essere abilitata utilizzando un volume condiviso e impostando la variabile d'ambiente ZBX_ENABLE_SNMP_TRAPS su 'true'
/var/lib/zabbix/mibs	Il volume consente di aggiungere nuovi file MIB. Non supporta sottodirectory, tutti i MIB devono essere collocati in /var/lib/zabbix/mibs
Zabbix proxy	
/usr/lib/zabbix/externalscripts	Il volume viene utilizzato da controlli esterni. Corrisponde al parametro ExternalScripts in zabbix_proxy.conf
/var/lib/zabbix/db_data	Il volume consente di memorizzare i file del database su dispositivi esterni. Supportato solo per Zabbix proxy con SQLite3
/var/lib/zabbix/modules	Il volume consente di caricare moduli aggiuntivi ed estendere Zabbix server utilizzando la funzionalità LoadModule
/var/lib/zabbix/enc	Il volume viene utilizzato per memorizzare file relativi a TLS. I nomi di questi file sono specificati utilizzando le variabili d'ambiente ZBX_TLSCAFILE, ZBX_TLSCRLFILE, ZBX_TLSKEY_FILE e ZBX_TLSPSKFILE
/var/lib/zabbix/ssl/certs	Il volume viene utilizzato come posizione dei file di certificato client SSL per l'autenticazione del client. Corrisponde al parametro SSLCertLocation in zabbix_proxy.conf
/var/lib/zabbix/ssl/keys	Il volume viene utilizzato come posizione dei file di chiave privata SSL per l'autenticazione del client. Corrisponde al parametro SSLKeyLocation in zabbix_proxy.conf
/var/lib/zabbix/ssl/ssl_ca	Il volume viene utilizzato come posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server. Corrisponde al parametro SSLCALocation in zabbix_proxy.conf
/var/lib/zabbix/snmptraps	Il volume viene utilizzato come posizione del file snmptraps.log. Può essere condiviso dal container zabbix-snmptraps ed ereditato utilizzando l'opzione Docker volumes_from durante la creazione di una nuova istanza di Zabbix server. La funzionalità di elaborazione delle trap SNMP può essere abilitata utilizzando un volume condiviso e impostando la variabile d'ambiente ZBX_ENABLE_SNMP_TRAPS su 'true'
/var/lib/zabbix/mibs	Il volume consente di aggiungere nuovi file MIB. Non supporta sottodirectory, tutti i MIB devono essere collocati in /var/lib/zabbix/mibs
Interfaccia web di Zabbix basata su web server Apache2	

Volume	Descrizione
/etc/ssl/apache2	Il volume consente di abilitare HTTPS per l'interfaccia web di Zabbix. Il volume deve contenere i due file `ssl.crt` e `ssl.key` preparati per le connessioni SSL di Apache2
Interfaccia web di Zabbix basata su web server Nginx	
/etc/ssl/nginx	Il volume consente di abilitare HTTPS per l'interfaccia web di Zabbix. Il volume deve contenere i due file `ssl.crt`, `ssl.key` e `dhparam.pem` preparati per le connessioni SSL di Nginx
Zabbix snmptraps	
/var/lib/zabbix/snmptraps	Il volume contiene il file di log `snmptraps.log` con il nome delle trap SNMP ricevute
/var/lib/zabbix/mibs	Il volume consente di aggiungere nuovi file MIB. Non supporta sottodirectory, tutti i MIB devono essere collocati in `/var/lib/zabbix/mibs`

Per ulteriori informazioni, vedere i repository ufficiali di Zabbix su Docker Hub.

Esempi

Esempio 1

L'esempio mostra come eseguire Zabbix server con supporto per il database MySQL, l'interfaccia web di Zabbix basata sul web server Nginx e Zabbix Java gateway.

1. Creare una rete dedicata ai container dei componenti Zabbix:

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

2. Avviare un'istanza vuota di server MySQL:

```
docker run --name mysql-server -t \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    --network=zabbix-net \
    --restart unless-stopped \
    -d mysql:8.4-oracle \
    --character-set-server=utf8 --collation-server=utf8_bin \
    --default-authentication-plugin=caching_sha2_password
```

3. Avviare un'istanza di 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
```

4. Avviare un'istanza di Zabbix server e collegarla all'istanza di server MySQL creata:

```
docker run --name zabbix-server-mysql -t \
    -e DB_SERVER_HOST="mysql-server" \
    -e MYSQL_DATABASE="zabbix" \
    -e MYSQL_USER="zabbix" \
    -e MYSQL_PASSWORD="zabbix_pwd" \
    -e MYSQL_ROOT_PASSWORD="root_pwd" \
    -e ZBX_JAVAGATEWAY="zabbix-java-gateway" \
    --network=zabbix-net \
```

```
    -p 10051:10051 \
    --restart unless-stopped \
    -d zabbix/zabbix-server-mysql:alpine-8.0-latest
```

5. Avviare l'interfaccia web di Zabbix e collegarla alle istanze create di server MySQL e Zabbix server:

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

Esempio 2

L'esempio mostra come eseguire Zabbix server con supporto per il database PostgreSQL, l'interfaccia web di Zabbix basata sul web server Nginx e la funzionalità SNMP trap.

1. Creare una rete dedicata ai container dei componenti Zabbix:

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

2. Avviare un'istanza vuota di server PostgreSQL:

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

3. Avviare un'istanza di Zabbix snmptraps:

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

4. Avviare un'istanza di Zabbix server e collegarla all'istanza di server PostgreSQL creata:

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

5. Avviare l'interfaccia web di Zabbix e collegarla alle istanze create di server PostgreSQL e Zabbix server:

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

```
    -e POSTGRES_DB="zabbix" \
    --network=zabbix-net \
    -p 443:8443 \
    -p 80:8080 \
    -v /etc/ssl/nginx:/etc/ssl/nginx:ro \
    --restart unless-stopped \
    -d zabbix/zabbix-web-nginx-pgsql:alpine-8.0-latest
```

Esempio 3

L'esempio mostra come eseguire Zabbix server con supporto per il database MySQL, l'interfaccia web di Zabbix basata sul web server Nginx e Zabbix Java gateway utilizzando podman su Red Hat 8.

1. Creare un nuovo pod con nome zabbix e porte esposte (interfaccia web, trapper di Zabbix server):

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

2. (opzionale) Avviare il container di Zabbix agent nella posizione del pod zabbix:

```
podman run --name zabbix-agent \
  -e ZBX_SERVER_HOST="127.0.0.1,localhost" \
  --restart=always \
  --pod=zabbix \
  -d registry.connect.redhat.com/zabbix/zabbix-agent-74:latest
```

3. Creare la directory ./mysql/ sull'host e avviare Oracle MySQL server 8.4:

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

4. Avviare il container di Zabbix server:

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

5. Avviare il container di Zabbix Java Gateway:

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

6. Avviare il container dell'interfaccia web di Zabbix:

```
podman run --name zabbix-web-mysql -t \
                -e ZBX_SERVER_HOST="127.0.0.1" \
                -e DB_SERVER_HOST="127.0.0.1" \
                -e MYSQL_DATABASE="zabbix" \
                -e MYSQL_USER="zabbix" \
                -e MYSQL_PASSWORD="zabbix_pwd" \
                -e MYSQL_ROOT_PASSWORD="root_pwd" \
```

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

Note:
Il pod `zabbix` espone la porta 80/TCP (HTTP) alla macchina host dalla porta 8080/TCP del container `zabbix-web-mysql`.

Docker Compose

In alternativa, Zabbix può essere installato utilizzando il plugin Docker Compose. I file Compose per definire ed eseguire componenti Zabbix multi-container sono disponibili nel repository Docker ufficiale di Zabbix su GitHub.

Attention:
I file Compose ufficiali di Zabbix supportano la versione 3 di Docker Compose.

Questi file Compose sono aggiunti come esempi; sono sovraccarichi. Ad esempio, contengono proxy con supporto sia per MySQL che per SQLite3.

Per ottenere i file Docker Compose forniti da Zabbix, clonare il repository:

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

Passare alla versione richiesta:

```
git checkout 8.0
```

Configurare i file Compose e creare e avviare i container:

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

Sostituire `docker-compose_v3_alpine_mysql_latest.yaml` nel comando sopra con il file di configurazione richiesto.

Sono disponibili le seguenti opzioni:

Nome file	Descrizione
`docker-compose_v3_alpine_mysql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Alpine Linux con supporto per database MySQL.
`docker-compose_v3_alpine_mysql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Alpine Linux con supporto per database MySQL.
`docker-compose_v3_alpine_pgsql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Alpine Linux con supporto per database PostgreSQL.
`docker-compose_v3_alpine_pgsql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Alpine Linux con supporto per database PostgreSQL.
`docker-compose_v3_ol_mysql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Oracle Linux con supporto per database MySQL.
`docker-compose_v3_ol_mysql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Oracle Linux con supporto per database MySQL.
`docker-compose_v3_ol_pgsql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Oracle Linux con supporto per database PostgreSQL.
`docker-compose_v3_ol_pgsql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Oracle Linux con supporto per database PostgreSQL.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Ubuntu 24.04 (noble) con supporto per database MySQL.
`docker-compose_v3_ubuntu_mysql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Ubuntu 24.04 (noble) con supporto per database MySQL.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	File Compose per l'ultima versione dei componenti Zabbix 8.0 su Ubuntu 24.04 (noble) con supporto per database PostgreSQL.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	File Compose per creare localmente l'ultima versione di Zabbix 8.0 ed esegue i componenti Zabbix su Ubuntu 24.04 (noble) con supporto per database PostgreSQL.

Archiviazione

I file Compose sono configurati per supportare l'archiviazione locale su una macchina host. Docker Compose creerà una directory `zbx_env` nella cartella contenente il file compose quando si eseguono i componenti Zabbix utilizzando il file compose. La directory conterrà la stessa struttura descritta nella sezione Volumes e una directory per l'archiviazione del database.

Sono inoltre presenti volumi in modalità di sola lettura per i file `/etc/localtime` e `/etc/timezone`.

Variabili d'ambiente

I file delle variabili hanno la seguente struttura di denominazione: `.env_<type of component>` e si trovano nella directory *env_vars*. Per i dettagli sulla denominazione delle variabili e sulla selezione disponibile, vedere variabili d'ambiente.

Esempi

Esempio 1

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

Il comando scaricherà le immagini Zabbix 8.0 più recenti per ciascun componente Zabbix e le eseguirà in modalità detach.

> **Attention:**
> Non dimenticare di scaricare i file `.env_<type of component>` dal repository ufficiale Zabbix su github.com insieme ai file compose.

Esempio 2

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

Il comando scaricherà l'immagine di base Ubuntu 24.04 (noble), quindi compilerà localmente i componenti Zabbix 8.0 e li eseguirà in modalità detach.

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 Configurare i 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 installa da Zabbix Git repository, 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. Solo un tipo di database può essere compilato con un processo server o proxy alla volta.

Per vedere tutte le opzioni di configurazione supportate, all'interno della directory dei sorgenti Zabbix estratti eseguire:

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

Per configurare i sorgenti per un Zabbix server e agent, è possibile eseguire qualcosa come:

```
./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 deve essere installata una versione Go supportata.

Note sulle opzioni di compilazione:

- `--enable-agent` - compila Zabbix agent, così come le utility 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.*`. 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 di installazione personalizzato di PCRE2.
- `--with-mysql=/path/to/mysql_config` - specifica il percorso di una particolare configurazione della libreria client MySQL. Utile quando sono installate più versioni di MySQL o MariaDB.
- `--enable-static` - collega staticamente le librerie (non supportato su Solaris). Usare questa opzione se si prevede di distribuire binari compilati a sistemi senza le librerie richieste. Non consigliato durante la compilazione di Zabbix server. Per compilare il server staticamente, è richiesta 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 bassi per lo stack dei thread).

Attention:
Se `./configure` non riesce a completarsi 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` rivela la causa effettiva:
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Vedere anche:

- Compilare Zabbix con supporto per la crittografia
- Problemi di compilazione noti

5 Compilare e installare tutto

Note:
Se si installa da repository Git di Zabbix, è necessario eseguire prima:

```
$ make dbschema
```

```
make install
```

Questo passaggio deve essere eseguito da un utente con permessi sufficienti (comunemente 'root' oppure utilizzando `sudo`).

L'esecuzione di `make install` installerà per impostazione predefinita i binari dei 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, utilizzare una chiave --prefix nel passaggio precedente di configurazione dei sorgenti, ad esempio --prefix=/home/zabbix. In questo caso i binari dei daemon verranno installati in <prefix>/sbin, mentre le utility in <prefix>/bin. Le pagine man verranno installate in <prefix>/share.

6 Rivedere e modificare i file di configurazione

- modificare il file di configurazione dell'agent Zabbix **/usr/local/etc/zabbix_agentd.conf**

È necessario configurare questo file per ogni host con zabbix_agentd installato.

Nel file è necessario specificare l'**indirizzo IP** del server Zabbix. Le connessioni da altri host saranno rifiutate.

- modificare il file di configurazione del server Zabbix **/usr/local/etc/zabbix_server.conf**

È necessario specificare il nome del database, l'utente e la password (se utilizzata).

Per il resto, i parametri predefiniti saranno adeguati 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 Zabbix (o del proxy).

- se è stato installato un proxy Zabbix, modificare il file di configurazione del proxy **/usr/local/etc/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 deve essere specificato il percorso completo del file del database; utente DB e password non sono richiesti.

7 Avviare i demoni

Eseguire zabbix_server sul lato server.

```
zabbix_server
```

Note:
Assicurarsi che il sistema consenta l'allocazione di 36 MB (o qualcosa in 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 web server che supporti PHP. L'installazione si effettua semplicemente copiando i file PHP dalla directory `ui` nella directory dei documenti HTML del web server.

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 utilizzare una sottodirectory invece della root HTML. Per creare una sottodirectory e copiarvi i file del frontend di Zabbix, eseguire i seguenti comandi, sostituendo `<htdocs>` con la directory effettiva:

```
mkdir <htdocs>/zabbix
cd ui
cp -a . <htdocs>/zabbix
```

Se si prevede di utilizzare una lingua diversa dall'inglese, consultare Installation of additional frontend languages 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 destinazioni non coperte 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. Potrebbe contenere requisiti specifici e istruzioni di installazione.

Per installare dai sorgenti, per prima cosa scaricare ed estrarre l'archivio dei sorgenti del plugin caricabile.

Per compilare il plugin, accedere alla directory del plugin estratto ed eseguire `make`:

```
make
```

Note:
Per compilare i plugin caricabili di Zabbix agent 2 deve essere installata una versione di Go supportata.

L'eseguibile del plugin può essere collocato ovunque, purché possa essere caricato da Zabbix agent 2. Specificare il percorso del file 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 maggiori dettagli sulla configurazione dei plugin, procedere alla configurazione.

I plugin caricabili forniti da Zabbix utilizzano semplici makefile con i seguenti target di build:

- `make` - compila il plugin
- `make clean` - elimina tutti i file creati durante la compilazione del plugin
- `make check` - esegue gli autotest (richiede una destinazione 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 dei sorgenti che include 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, per prima cosa 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 intero albero 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 proprio path.

```
make
```

Ora si dispone di un file zabbix-java-gateway-$VERSION.jar in src/zabbix_java/bin. Se si è a proprio agio nell'eseguire Java gateway da src/zabbix_java nella directory della distribuzione, si può procedere alle istruzioni per configurare ed eseguire Java gateway. In caso contrario, assicurarsi di disporre di privilegi sufficienti ed eseguire `make install`.

```
make install
```

Procedere alla configurazione per maggiori dettagli sulla configurazione e l'esecuzione di Java gateway.

Installazione del servizio web Zabbix

L'installazione del servizio web Zabbix è necessaria solo se si desidera utilizzare i report pianificati.

Per installare dai sorgenti, per prima cosa scaricare ed estrarre l'archivio dei sorgenti.

Per compilare il servizio web Zabbix, eseguire lo script `./configure` con l'opzione `--enable-webservice`.

Note:
Per compilare il servizio web Zabbix deve essere installata una versione di Go supportata.

Eseguire zabbix_web_service sulla macchina in cui è installato il servizio web:

```
zabbix_web_service
```

Per maggiori dettagli sulla configurazione della generazione dei report pianificati, procedere alla configurazione.

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 predefinite sui sistemi basati su Debian).

Dimensione dello stack troppo piccola su alcuni sistemi

Se Zabbix va in crash o si blocca a causa di overflow dello stack, è possibile aumentare la dimensione dello stack per thread utilizzando l'opzione `--with-stacksize` durante la configurazione dei sorgenti. Questo problema può verificarsi su sistemi con limiti predefiniti bassi per lo stack dei thread, in particolare durante il preprocessing, dove 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 controllare i limiti dello stack dei thread del sistema in fase di esecuzione utilizzando 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.

Seleziona il tuo sistema operativo e il componente di Zabbix per generare istruzioni di installazione adatte al tuo ambiente. Vedi anche le note sull'installazione dei pacchetti in questa pagina per importanti informazioni aggiuntive, nonché la sezione Installazione e configurazione.

I pacchetti sono disponibili per le seguenti distribuzioni Linux:

- Red Hat Enterprise Linux e i suoi derivati: 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 OS (in particolare le distribuzioni basate su Debian) forniscono i propri pacchetti Zabbix. Questi pacchetti **non** sono supportati da Zabbix e potrebbero essere obsoleti o non includere le funzionalità e le correzioni di bug più recenti. Si consiglia di utilizzare solo i pacchetti ufficiali del Repository ufficiale di Zabbix. Se in precedenza hai installato Zabbix dal repository del tuo sistema operativo, consulta i passaggi per l'aggiornamento dei pacchetti Zabbix dai repository del sistema operativo.

I pacchetti supportano database MySQL/PostgreSQL e web server Apache/Nginx. Tieni presente che Zabbix server e proxy non possono condividere lo stesso database; utilizza nomi di database diversi se entrambi sono installati sullo stesso host.

Se necessario, sono disponibili pacchetti separati per Zabbix agent/agent 2, Zabbix get e Zabbix sender nel Repository ufficiale di Zabbix.

Zabbix fornisce anche binari precompilati di Zabbix agent per sistemi operativi non Linux; vedi:

- Installazione dell'agent Windows da MSI
- Installazione dell'agent macOS da PKG
- Binari legacy (per sistemi meno recenti/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; vedere configurazione SELinux per le relative istruzioni di configurazione.
- Se si utilizza TimescaleDB, vedere la documentazione aggiuntiva su configurazione di TimescaleDB.
- Se si installa Zabbix Java gateway (per il monitoraggio delle applicazioni JMX), vedere le istruzioni di configurazione aggiuntive per sistemi basati su RHEL e sistemi basati su Debian.
- Per eseguire Zabbix agent come root, vedere Esecuzione di agent come root.

Le seguenti note si applicano a RHEL e ai suoi derivati:

- Se è stato abilitato il repository EPEL per EL9, che fornisce anch'esso pacchetti Zabbix, questo deve essere escluso dalla risoluzione dei pacchetti prima di installare i pacchetti Zabbix ufficiali; vedere Installazione accidentale dei pacchetti Zabbix di EPEL.
- Per installare i pacchetti Zabbix negli ambienti Red Hat UBI, vedere Pacchetti Zabbix per RHEL negli ambienti Red Hat UBI.
- Per utilizzare gli item ICMP ping, i pacchetti per `fping` sono disponibili anche nel repository ufficiale di Zabbix.

Configurazione di SELinux

Zabbix utilizza la comunicazione tra processi basata su socket. Nei sistemi in cui Security-Enhanced Linux (SELinux) è abilitato, potrebbe essere necessario aggiungere regole SELinux per consentire a Zabbix di creare/utilizzare socket di dominio UNIX nella directory SocketDir. I file socket sono utilizzati da Zabbix server (alerter, preprocessing, IPMI) e da 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 Zabbix frontend e server:

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 è accessibile tramite rete (incluso `localhost` per PostgreSQL), consentire anche a Zabbix frontend di connettersi al database:

```
setsebool -P httpd_can_network_connect_db on
```

Per RHEL precedenti alla versione 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
```

Facoltativamente, è possibile installare un pacchetto `zabbix-selinux-policy` predefinito da The Zabbix Official Repository. Questo pacchetto è fornito per tutte le versioni di OS supportate per semplificare il deployment di Zabbix ed evitare che gli utenti disattivino SELinux a causa della complessità della configurazione.

> **Attention:**
> Per la massima sicurezza, si consiglia di impostare configurazioni SELinux personalizzate.

Il pacchetto `zabbix-selinux-policy` contiene una policy SELinux di base, che consente a Zabbix di creare e utilizzare socket e abilita la connessione di HTTPd a PostgreSQL (utilizzato dal 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 la normale installazione o il normale funzionamento, ma sono utili per la risoluzione avanzata dei problemi.

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/stable/sles/15/x86_64/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/x86_64/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Una volta abilitato, installare i pacchetti:

- Su RHEL, installare un singolo pacchetto con le 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 seguenti servono per abilitare i repository delle release instabili di Zabbix (disabilitati per impostazione predefinita) utilizzati per i release candidate delle versioni minori di Zabbix.

Per prima cosa, installa o aggiorna all'ultima versione del pacchetto zabbix-release. Per abilitare i pacchetti rc sul tuo sistema, procedi come 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.

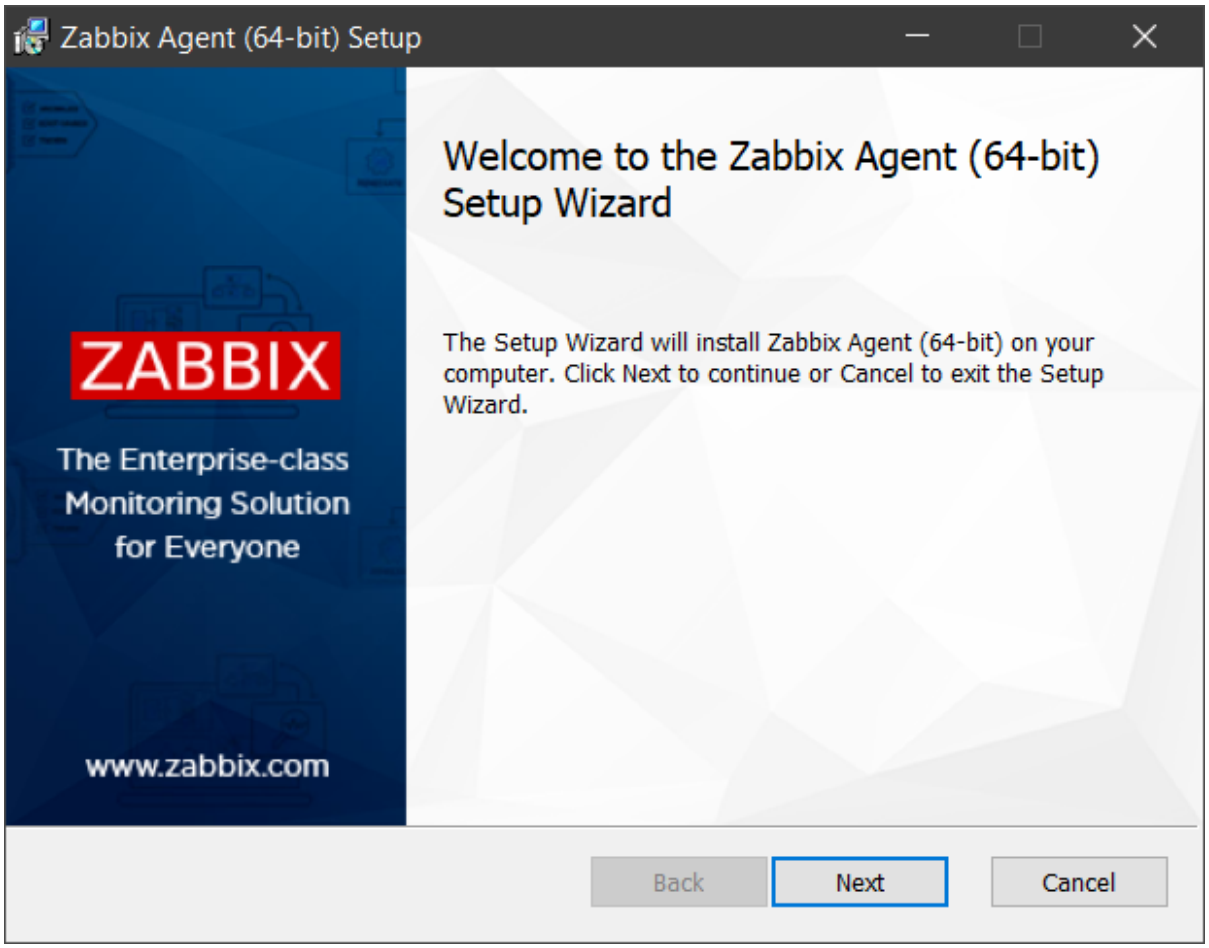
Sebbene l'installazione tramite pacchetti MSI sia pienamente supportata, si consiglia di installare almeno Microsoft .NET Framework 2 per una corretta gestione degli errori.

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 Setup Wizard

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:

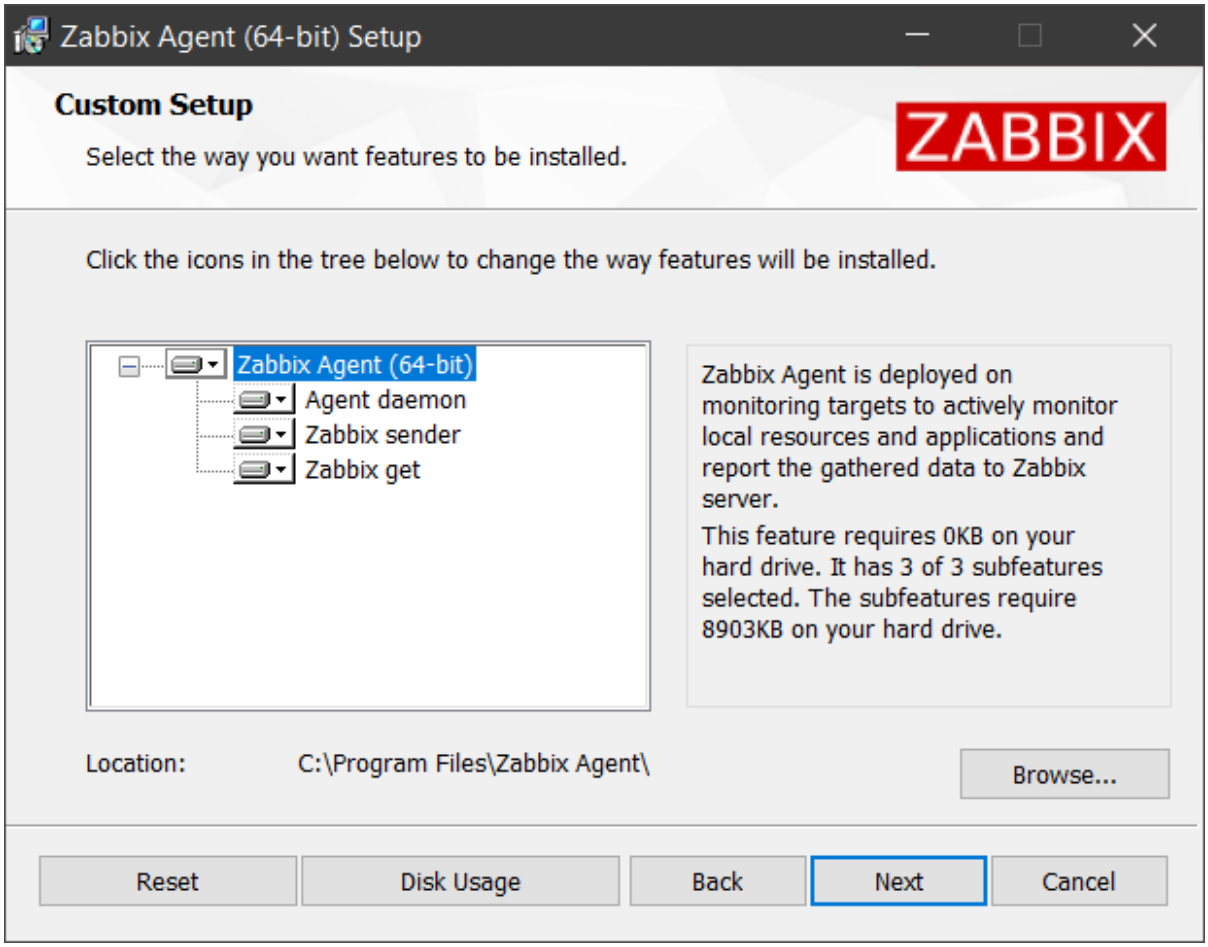


2. Accettare il Contratto di licenza con l'utente finale:



3. Selezionare i componenti di Zabbix (Agent daemon, Zabbix sender, Zabbix get) da installare:

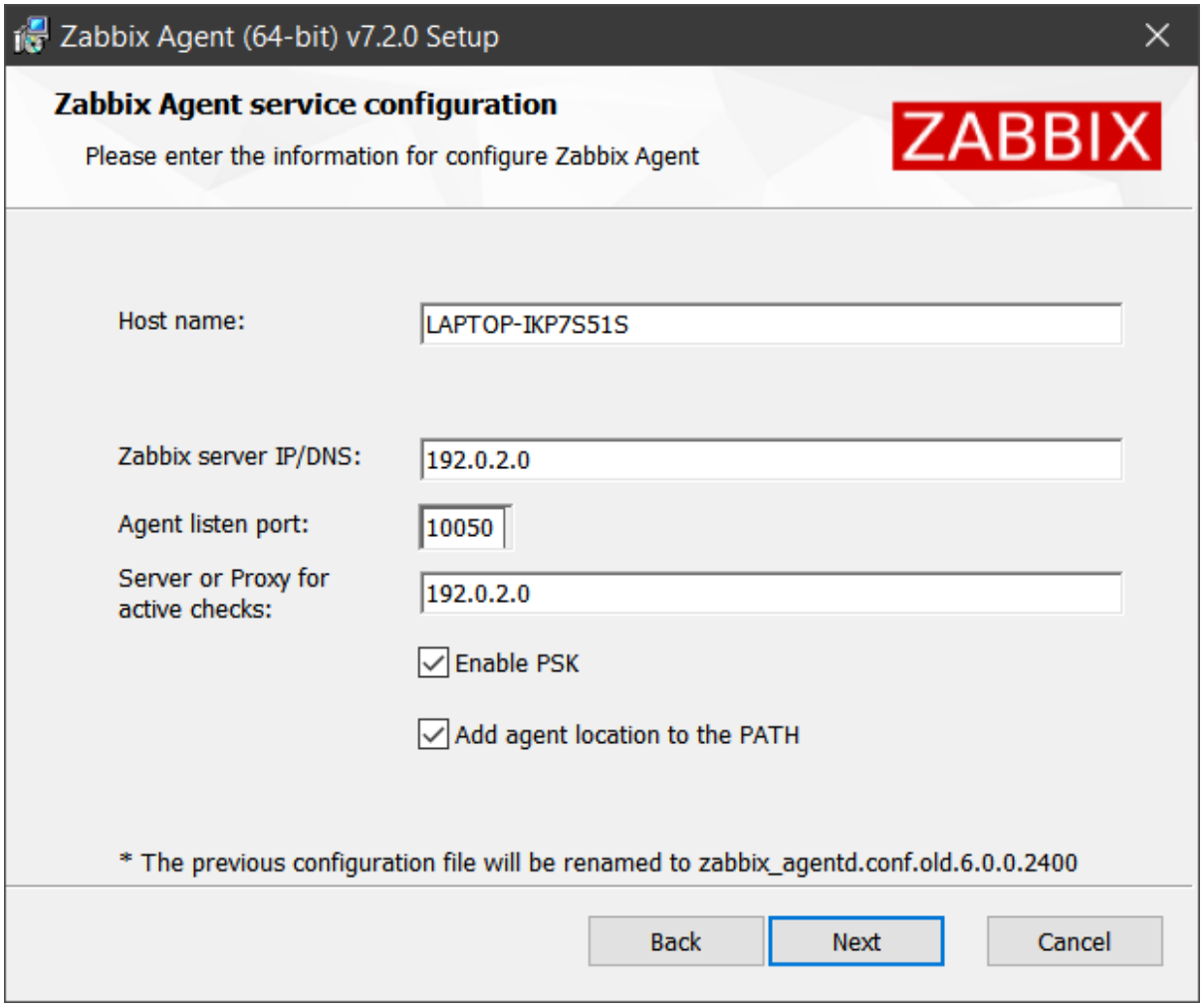
Attention:
Si consiglia di utilizzare il percorso di installazione predefinito fornito dal programma di installazione. L'utilizzo di un percorso personalizzato senza le autorizzazioni necessarie può compromettere la sicurezza dell'installazione.



4. Configurare i seguenti parametri. I relativi valori verranno impostati nel file di configurazione di Zabbix agent:

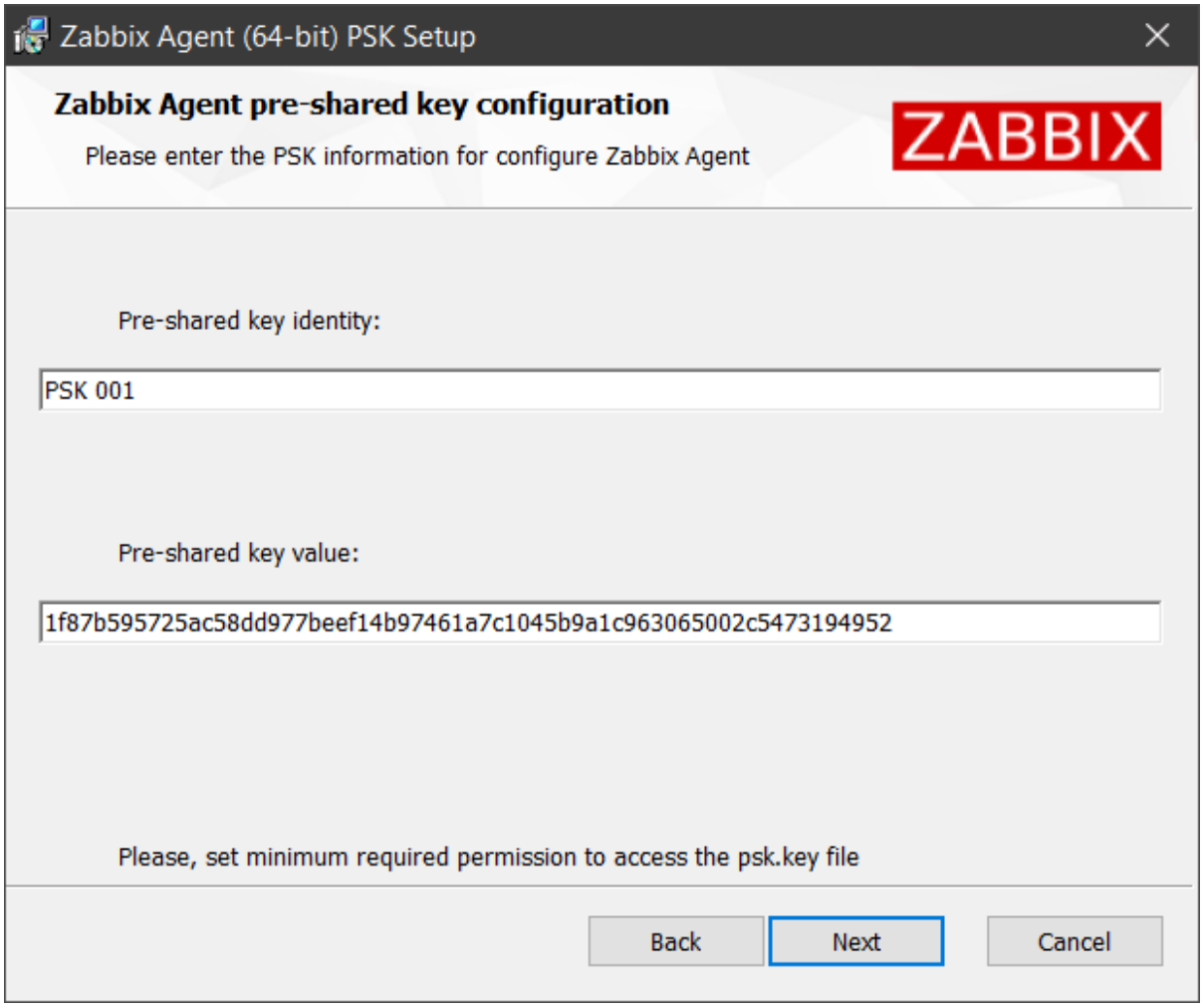
Parameter	Description
Host name	Il nome host della macchina su cui viene installato Zabbix agent. Imposta il parametro Hostname.
Zabbix server IP/DNS	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di server Zabbix o proxy Zabbix. Questo parametro è **obbligatorio**. Imposta il parametro Server.
Agent listen port	L'agent resterà in ascolto su questa porta per le connessioni dal server. Imposta il parametro ListenPort.
Server or Proxy for active checks	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere gli active checks. L'indirizzo del server/proxy è un indirizzo IP o un nome DNS con una porta facoltativa separata da due punti. Imposta il parametro ServerActive.
Enable PSK	Selezionare la casella di controllo per abilitare il supporto TLS utilizzando chiavi precondivise. Imposta i parametri TLSConnect e TLSAccept su `psk`.
Add agent location to the PATH	Selezionare la casella di controllo per aggiungere il percorso di Zabbix agent alla variabile di sistema PATH.

Note:
Se viene rilevato un Zabbix agent esistente, verranno visualizzati i valori dei parametri dal relativo file di configurazione. Inoltre, il file di configurazione esistente verrà rinominato durante l'installazione e ne verrà creato uno nuovo.



5. Configurare i parametri PSK se nel passaggio precedente è stata selezionata la casella di controllo *Enable PSK*. Anche questi parametri verranno impostati nel file di configurazione di Zabbix agent:

Parameter	Description
Pre-shared key identity	La stringa di identità della chiave precondivisa. Imposta il parametro TLSPSKIdentity.
Pre-shared key value	Il valore 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.

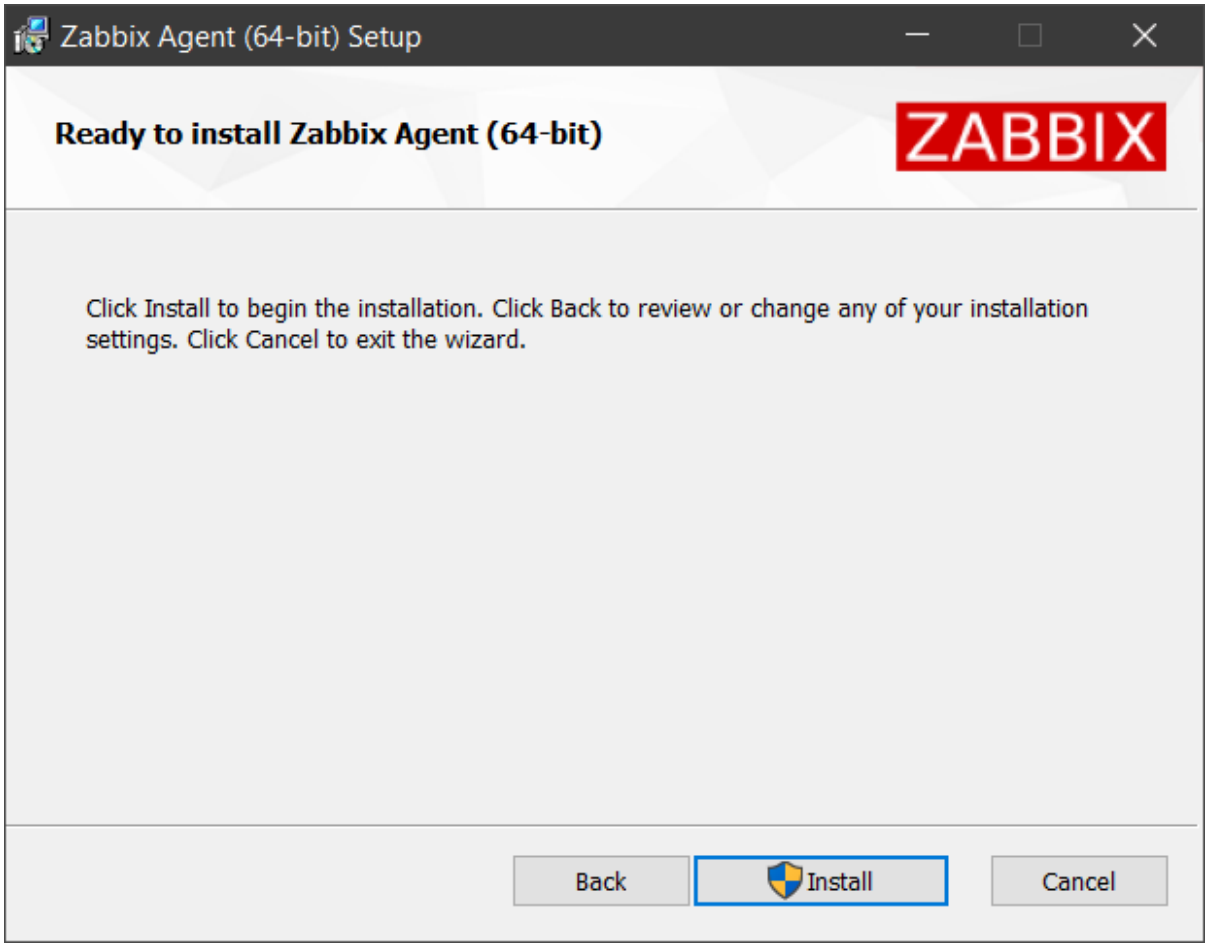


6. Fare clic su *Install* per avviare l'installazione.

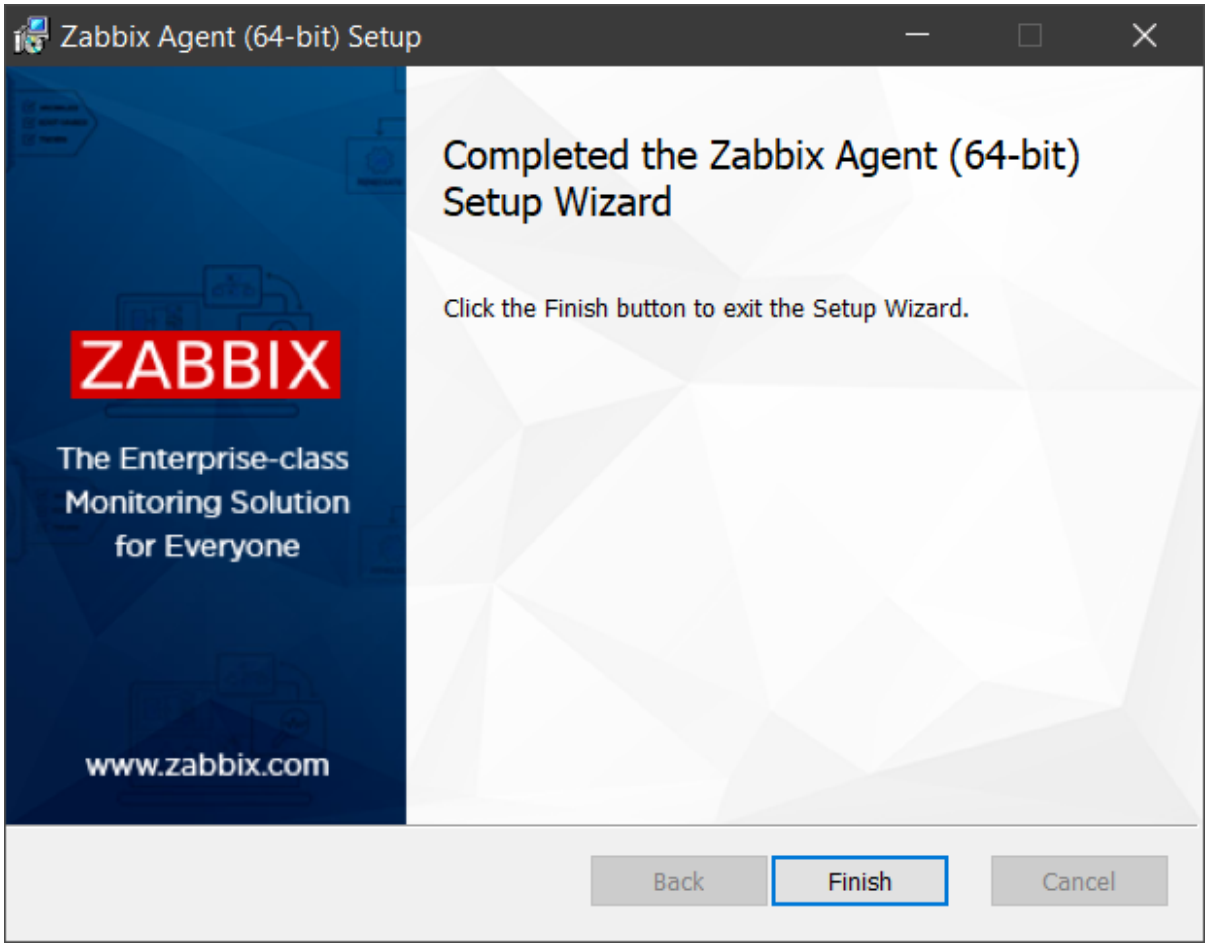
Tutti i componenti Zabbix selezionati e il file di configurazione di Zabbix agent verranno installati nel percorso specificato (predefinito: `C:\Program Files\Zabbix Agent`). Lo stesso vale per Zabbix agent 2, tranne per il fatto che i file di configurazione aggiuntivi per i suoi plugin integrati verranno 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 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, nel qual caso sarà necessario riavviare il sistema.



7. Fare clic sul pulsante *Finish* per uscire dal Setup Wizard.



Installazione dalla riga di comando

Zabbix agent può essere installato dalla 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 assistite 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).

Parametro	Descrizione
ADDDEFAULT	Un elenco di componenti separati da virgole da installare nella loro configurazione predefinita. Per ulteriori informazioni, vedere proprietà ADDDEFAULT. 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 proprietà ADDLOCAL. 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 dell'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 dell'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 Zabbix agent.
ENABLEPATH	Usare `ENABLEPATH=1` per aggiungere il percorso di Zabbix agent alla variabile PATH di sistema.
ENABLEPERSISTENTBUFFER	Solo Zabbix agent 2. Abilita l'uso dell'archiviazione persistente locale per gli item attivi.
HOSTINTERFACE	Parametro facoltativo che definisce l'interfaccia 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 host.
HOSTNAMEITEM	Parametro facoltativo che definisce un item usato per ottenere il nome host.
INCLUDE	Un elenco di singoli file separati da punto e virgola oppure di tutti i file in una directory da includere nel file di configurazione di Zabbix agent.
INSTALLFOLDER	Il percorso completo di una cartella in cui verranno installati i componenti 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 sui quali l'agent deve restare in ascolto.
LISTENPORT	L'agent resterà in ascolto su questa porta per 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 dell'agent presenti in questo file (limitati a quelli elencati in questa tabella) verranno scritti nel file di configurazione dell'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 mantenere il database SQLite.
PERSISTENTBUFFERPERIOD	Solo Zabbix agent 2. Il periodo di tempo per cui i dati devono essere conservati quando non c'è connessione al server o al proxy.
SERVER	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di server Zabbix o proxy Zabbix. Questo parametro è **obbligatorio**, tranne quando STARTAGENTS è impostato su 0.

Parametro	Descrizione
SERVERACTIVE	L'indirizzo del server/proxy Zabbix 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 nel Windows Firewall per Zabbix agent.
SOURCEIP	L'indirizzo IP sorgente per le connessioni in uscita verso server Zabbix o proxy Zabbix, oppure per effettuare 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 resterà in ascolto su alcuna porta TCP.
STARTUPTYPE	Tipo di avvio del servizio 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 in Windows Vista/Server 2008 e versioni successive); **manual** - avvia il servizio manualmente (da un utente o da 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 resterà in ascolto su questa porta per richieste di stato HTTP (http://localhost:<port>/status).
TIMEOUT	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con proxy o server Zabbix.
TLSACCEPT	Le connessioni in ingresso da accettare (usato per i controlli passivi). Se impostato su `psk`, anche TLSCONNECT verrà impostato su `psk` (se non specificato diversamente).
TLSCAFILE	Il percorso completo di un file contenente i certificati della CA di livello superiore per la verifica del certificato del peer.
TLSCERTFILE	Il percorso completo di un file contenente il certificato dell'agent o la catena di certificati.
TLSCONNECT	Come l'agent deve connettersi al server o proxy Zabbix (usato per i controlli attivi). Se impostato su `psk`, anche TLSACCEPT verrà impostato su `psk` (se non specificato diversamente).
TLSCRLFILE	Il percorso completo di un file contenente 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 sono impostati sia TLSPSKFILE sia TLSPSKVALUE, il valore di TLSPSKVALUE verrà scritto nel file specificato in TLSPSKFILE. Si **consiglia** 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 l'agent) possa leggerlo.
TLSPSKIDENTITY	La stringa identificativa della chiave precondivisa.
TLSPSKVALUE	Il valore stringa della chiave precondivisa. Se sono impostati sia TLSPSKFILE sia TLSPSKVALUE, 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 utilizzando chiavi precondivise.

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

L'esempio successivo installa una versione più recente di Zabbix agent e utilizza 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, utilizzare i segnaposto dei parametri (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, è possibile utilizzare 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 file di configurazione dell'agent appena creato. Per mantenere la configurazione esistente dell'agent, definire 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, controlla 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 dalla riga di comando

I plugin caricabili di Zabbix agent 2 possono essere installati dalla 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 del programma 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 nella loro configurazione predefinita. Per ulteriori informazioni, vedere proprietà ADDDEFAULT. 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 proprietà ADDLOCAL. Valori possibili: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Esempio: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Il percorso completo di una cartella in cui verranno installati i componenti 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 mostra 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, ma su una piattaforma a 32 bit può essere compilata solo la versione a 32 bit.

La compilazione di Zabbix agent 2 richiede:

- 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 modelli tramite 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, per installare 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` (oppure `C:\Zabbix\x86` per le compilazioni a 32 bit).

Compilazione di Zabbix agent 2 con vcpkg

Questa sezione contiene 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 *Desktop development with C++*, che include il gestore di pacchetti vcpkg.

2. Scaricare e installare Go (disponibile come programma di installazione MSI). Durante l'installazione, assicurarsi di specificare `C:\Zabbix\Go` come cartella di installazione.

3. Scaricare 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`

Quindi, estrarla in `C:\Zabbix\mingw64` (oppure `C:\Zabbix\mingw32` per build a 32 bit).

4. Inizializzare vcpkg e installare le dipendenze richieste per compilare Zabbix agent 2 (si noti che questa 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 build a 64 bit:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
vcpkg install --triplet x64-mingw-static --x-install-root=x64

#### Per build 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. Passare alla directory di build di Zabbix (`C:\Zabbix\zabbix-8.0.0\build\mingw`) e creare il seguente script `build.bat`:

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

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

7. Compilare Zabbix agent 2 eseguendo lo script:

```
build.bat
```

Dopo la compilazione, il file binario di Zabbix agent 2 si troverà in `C:\Zabbix\zabbix-8.0.0\bin\win64` (per build a 64 bit) oppure in `C:\Zabbix\zabbix-8.0.0\bin\win32` (per build 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`) e quindi avviare l'agent:

```
mkdir C:\Zabbix\agent2

#### Per build a 64 bit:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

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

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

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 corrispondente 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, controllare il relativo file README. Potrebbe contenere requisiti specifici e istruzioni di installazione.

2. Accedere 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 file 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 ovunque, purché possa essere caricato da Zabbix agent 2. Specificare il percorso del file 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 configurazione.

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 build

1. Scaricare e installare MSYS2 (disponibile come installer MSI). Durante l'installazione, assicurarsi di specificare `C:\Zabbix\msys64` come cartella di installazione.

2. Scaricare e installare Go (disponibile come installer MSI; vedere le versioni di Go attualmente supportate). Durante l'installazione, assicurarsi di specificare `C:\Zabbix\Go` come cartella di installazione.

3. Scaricare 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`

Quindi, estrarla in `C:\Zabbix\mingw64` (oppure `C:\Zabbix\mingw32` per build a 32 bit).

Installazione di OpenSSL

> **Note:**
> Per compilare Zabbix agent senza 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. Passare alla directory di OpenSSL estratta e creare il seguente script `build.sh`:

- Per 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 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 l'accesso in scrittura agli utenti non amministratori per la directory `C:\Zabbix\x64\OpenSSL-Win64-350-static`. In caso contrario, l'agent caricherà le impostazioni SSL da un percorso che può essere modificato da utenti senza privilegi, con conseguente potenziale vulnerabilità di sicurezza.

- L'opzione `no-shared` rende le librerie statiche OpenSSL libcrypto.lib e libssl.lib autosufficienti, in modo che i binari di Zabbix includano OpenSSL senza richiedere DLL esterne. Ciò significa che i binari di Zabbix possono essere copiati su altre macchine Windows senza le librerie OpenSSL; tuttavia, quando viene rilasciata una nuova versione OpenSSL con correzioni di bug, sarà necessario ricompilare Zabbix agent.
- Senza l'opzione `no-shared`, Zabbix si basa sulle DLL OpenSSL in fase di esecuzione. Ciò significa che gli aggiornamenti di OpenSSL potrebbero non richiedere la ricompilazione di Zabbix agent; tuttavia, quando lo si copia su altre macchine, sarà necessario copiare anche le DLL OpenSSL.

Per ulteriori informazioni su altre opzioni di configurazione di OpenSSL, fare riferimento alla documentazione di OpenSSL.

4. Configurare e installare OpenSSL eseguendo lo script (si noti che questa 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 grezzi (ad esempio, per applicare patch personalizzate o compilare a partire dal codice sorgente più recente), eseguire i seguenti comandi su una macchina **Linux** con Go installato (richiesto per configurare Zabbix agent 2):

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

Verrà così creato un archivio dei sorgenti, che potrà poi essere copiato su una macchina Windows.

2. Aprire il Prompt dei comandi con privilegi di amministratore. Quindi, passare alla directory di build di Zabbix e compilare Zabbix agent; assicurarsi di specificare correttamente le directory in cui sono installati OpenSSL e PCRE2:

- Per build 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 supporto TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Senza supporto TLS:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- Per build a 32 bit:

```
cd C:\Zabbix\zabbix-8.0.0\build\mingw
set PATH=C:\Zabbix\mingw32\bin;%PATH%

#### Con supporto TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2" OPENSSL="C:\Zabbix\x86\OpenSSL-Win64-350-static"

#### Senza supporto TLS:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Dopo la compilazione, il file 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 build 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 file binario zabbix_agent2.exe e i relativi file di configurazione in una cartella dedicata (ad esempio, C:\Zabbix\agent2) e quindi avviare l'agent:

```
mkdir C:\Zabbix\agent2
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\
copy C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.win.conf C:\Zabbix\agent2\
xcopy /E /I C:\Zabbix\zabbix-8.0.0\src\go\conf\zabbix_agent2.d C:\Zabbix\agent2\zabbix_agent2.d\

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

Se necessario, continuare con la compilazione dei plugin caricabili di Zabbix agent 2.

Compilazione di Zabbix agent su Windows

Panoramica

Questa pagina mostra come compilare Zabbix agent dai sorgenti su Windows 10 (64 bit).

Queste istruzioni si applicano alle versioni di Windows che supportano Visual Studio 2022.

La compilazione di Zabbix agent richiede:

- 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 modelli tramite espressioni regolari in Zabbix)

È possibile compilare Zabbix agent 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.

A seconda delle esigenze di monitoraggio, potrebbero essere necessarie librerie aggiuntive. Per ulteriori informazioni, vedere Requisiti.

Attention:
Prima di iniziare 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 di Zabbix agent con vcpkg

Questa sezione contiene 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 *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. Inizializzare vcpkg e installare le dipendenze necessarie per compilare Zabbix agent (si noti che questa 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. Passare 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:

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

:: Run the build process:
nmake -f Makefile CPU=AMD64 ^
    PCRE2INCDIR="%vcpkg%\include" ^
    PCRE2LIBDIR="%vcpkg%\lib" ^
    TLS=openssl ^
    TLSINCDIR="%vcpkg%\include" ^
    TLSLIBDIR="%vcpkg%\lib" ^
    LIBS="$(LIBS) Crypt32.lib" ^
    all
```

5. Compilare Zabbix agent eseguendo lo script:

```
build.bat
```

Dopo la compilazione, i file binari dei componenti Zabbix si troveranno in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Il file di configurazione di Zabbix agent si trova in `C:\Zabbix\zabbix-8.0.0\conf`.

Per eseguire l'agent, copiare `zabbix_agent.exe` e il relativo file di configurazione in una cartella dedicata (ad esempio, `C:\Zabbix\agent`) e quindi avviare l'agent:

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

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

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 build

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 manualmente l'agent:

- Compilatore C (Microsoft Visual C++)
- Strumento da riga di comando NMake
- Prompt dei comandi degli strumenti nativi x64

Installazione di OpenSSL

> **Note:**
> Per compilare Zabbix agent senza supporto TLS, procedere alla sezione Installazione di PCRE2.

1. Scaricare e installare Strawberry Perl (disponibile come programma di installazione MSI). Durante l'installazione, assicurarsi di 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; richiesto 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. Passare 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 di Zabbix agent su Windows (ad esempio, `C:\zabbix` o `C:\openssl-64bit`), assicurarsi di revocare l'accesso in scrittura a questa directory per gli utenti non amministratori. In caso contrario, l'agent caricherà le impostazioni SSL da un percorso che può essere modificato 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 di Zabbix includano OpenSSL senza richiedere DLL esterne. Ciò significa che i binari di Zabbix possono essere copiati su altre macchine Windows senza le librerie OpenSSL; tuttavia, quando viene rilasciata una nuova versione OpenSSL con correzioni di bug, sarà necessario ricompilare Zabbix agent.
- Senza l'opzione `no-shared`, Zabbix si basa sulle DLL OpenSSL in fase di esecuzione. Ciò significa che gli aggiornamenti di OpenSSL potrebbero non richiedere la ricompilazione di Zabbix agent; tuttavia, quando lo si copia su altre macchine, sarà necessario copiare anche le DLL OpenSSL.

Per ulteriori informazioni su altre opzioni di configurazione di OpenSSL, consultare la documentazione di OpenSSL.

6. Compilare OpenSSL ed eseguire i test (si noti che questa operazione potrebbe richiedere del tempo):

Attention:
Eseguire i test senza privilegi amministrativi; in caso contrario, ciò potrebbe portare a risultati imprevisti o a rischi per la sicurezza. Se alcuni test non riescono, 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, 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 di Zabbix agent

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 grezzi (ad esempio, per applicare patch personalizzate o compilare dal 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 --enable-agent --enable-ipv6 --prefix=`pwd`
make dist
```

Verrà così creato un archivio dei sorgenti, che potrà poi essere copiato su una macchina Windows.

2. Passare alla directory di build di Zabbix e compilare Zabbix agent (o altri componenti); assicurarsi di specificare correttamente le directory in cui sono installati OpenSSL e PCRE2:

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

#### With TLS support:
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

#### Without TLS support:
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
```

Dopo la compilazione, i file binari dei componenti di Zabbix si troveranno in `C:\Zabbix\zabbix-8.0.0\bin\win64`. Il file di configurazione di Zabbix agent si trova in `C:\Zabbix\zabbix-8.0.0\conf`.

Per eseguire l'agent, copiare `zabbix_agent.exe` e il relativo file di configurazione in una cartella dedicata (ad esempio, `C:\Zabbix\agent`) e quindi avviare l'agent:

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

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

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

Saranno necessari gli strumenti di sviluppo da riga di comando (Xcode non è richiesto), Automake, pkg-config e PCRE (v8.x) oppure PCRE2 (v10.x). Se si desidera compilare i binari dell'agent con TLS, saranno necessari anche OpenSSL oppure GnuTLS.

Per installare Automake e pkg-config, sarà necessario il gestore di pacchetti Homebrew da https://brew.sh/. Per installarlo, aprire il terminale ed eseguire il comando seguente:

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

Quindi installare 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 si intende eseguire i binari dell'agent su una macchina macOS che dispone già di queste librerie, è possibile utilizzare le librerie precompilate fornite da Homebrew. In genere si tratta di macchine macOS che utilizzano Homebrew per compilare i binari dell'agent Zabbix o per altri scopi.

Se i binari dell'agent verranno utilizzati su macchine macOS che non dispongono della versione condivisa delle librerie, sarà necessario compilare librerie statiche dai sorgenti e collegare a esse l'agent Zabbix.

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
```

4 Aggiornamento

Prima di eseguire l'aggiornamento, si consiglia vivamente di consultare tutte le note di aggiornamento pertinenti.

Sarà inoltre utile sapere quali componenti è necessario aggiornare.

Panoramica

Sono disponibili istruzioni dettagliate per l'aggiornamento per::

- Red Hat Enterprise Linux (utilizzando i pacchetti)
- Debian/Ubuntu (utilizzando i pacchetti)
- Aggiornamento tramite container
- Aggiornamento tramite sorgenti

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 TimescaleDB

Componenti da aggiornare

Si consiglia di aggiornare gli agent Zabbix, ma non è obbligatorio.

L'aggiornamento dei proxy Zabbix è fortemente consigliato. Zabbix server supporta pienamente i proxy che hanno la stessa versione principale del server. Zabbix server supporta anche proxy che **non sono più vecchi** della precedente versione LTS di Zabbix server, ma con funzionalità limitate (raccolta dati, esecuzione di comandi remoti, controlli immediati del valore degli item). Anche l'aggiornamento della configurazione è disabilitato e i proxy obsoleti funzioneranno solo con la vecchia configurazione.

> **Attention:**
> I proxy più vecchi della versione della precedente release LTS di Zabbix server o più nuovi della major version di Zabbix server non sono supportati. Zabbix server ignorerà i dati provenienti da proxy non supportati e tutta la comunicazione con Zabbix server non riuscirà con un avviso. Per ulteriori informazioni, vedere Compatibilità delle versioni.

Per ridurre al minimo il downtime e la perdita di dati durante l'aggiornamento, si consiglia di arrestare, aggiornare e avviare Zabbix server e quindi arrestare, aggiornare e avviare i proxy uno dopo l'altro. Durante il downtime del server, i proxy in esecuzione continueranno la raccolta dei dati. Una volta che il server sarà di nuovo operativo, i proxy obsoleti invieranno i dati al server più recente (anche se la configurazione del proxy non verrà aggiornata) e rimarranno parzialmente funzionali. Eventuali notifiche relative ai problemi verificatisi durante il downtime di Zabbix server verranno generate solo dopo l'avvio del server aggiornato.

Se Zabbix proxy viene avviato per la prima volta e il file del database SQLite è mancante, il proxy lo crea automaticamente.

Si noti che se Zabbix proxy utilizza SQLite3 e all'avvio rileva che la versione del file di database esistente è più vecchia di quella richiesta, **eliminerà automaticamente il file di database** e ne creerà uno nuovo. Pertanto, i dati storici memorizzati nel file di

database SQLite andranno persi. Se la versione di Zabbix proxy è più vecchia della versione del file di 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 dalle versioni **2.0**.x. Per l'aggiornamento da versioni precedenti, consultare la documentazione di Zabbix per la versione 2.0 e precedenti.

Note:
Si prega di notare che, dopo l'aggiornamento, alcune integrazioni di 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 di aggiornamento:

Aggiornamento da	Leggi le note di aggiornamento complete	Modifiche più importanti tra le versioni
7.4.x	Per: Zabbix 8.0	Versioni minime richieste del database aumentate. 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.
7.2.x	Per: Zabbix 7.4 Zabbix 8.0	Il supporto per la libreria PCRE è stato rimosso.
7.0.x	Per: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Il supporto per 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 controlli agent, HTTP agent e SNMP walk[oid]. Tabella database separata per i proxy. La posizione predefinita del file di configurazione dell'agent Windows è stata modificata. Oracle DB 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, Zabbix proxy con SQLite3 elimina automaticamente la vecchia versione del database (con tutto lo storico) 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 significativamente rielaborato. Durante l'aggiornamento devono essere creati trigger deterministici. Se la registrazione binaria è abilitata per MySQL/MariaDB, ciò richiede privilegi di superutente oppure l'impostazione della variabile/parametro di configurazione *log_bin_trust_function_creators = 1*. Vedere Script di creazione del database per istruzioni su come impostare la variabile.
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	Versioni minime richieste del database aumentate. Server/proxy non si avvierà se il database è obsoleto. Record del log di audit persi a causa della modifica della struttura del database.

Aggiornamento da	Leggi le note di aggiornamento complete	Modifiche più importanti tra le versioni
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	Versioni minime richieste del database aumentate. 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 stato modificato da MD5 a bcrypt.
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 per IBM DB2 è stato rimosso. La versione minima richiesta di PHP è stata aumentata da 5.4.0 a 7.2.0. Versioni minime richieste del database aumentate. Directory dei file PHP di Zabbix modificata.
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	Tipi di supporto Jabber, Ez Texting 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. I nuovi agent non potranno più funzionare con un server Zabbix meno recente.

Aggiornamento da	Leggi le note di aggiornamento complete	Modifiche più importanti tra le versioni
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 per il protocollo in testo semplice è stato rimosso e l'header è obbligatorio. Le versioni precedenti alla 1.4 degli agent Zabbix non sono più supportate. Il parametro Server nella configurazione del proxy passivo è ora obbligatorio.
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. Sono supportate le Perl Compatible Regular Expressions (PCRE) 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 parametri utente, comandi remoti e item system.run[] senza il flag 'nowait', nonché per gli script eseguiti da Zabbix server. Zabbix Java gateway deve essere aggiornato 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 della dimensione della tabella dello storico.

Aggiornamento da	Leggi le note di aggiornamento complete	Modifiche più importanti tra le versioni
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.
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 su 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 database MySQL sono richiesti il set di caratteri utf8 e la collation utf8_bin. Vedere script di creazione del database. È richiesta l'estensione PHP 'mysqli' al posto 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/
cp /etc/apache2/conf-enabled/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 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 anche 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 l'ultimo pacchetto di configurazione del repository per garantire la compatibilità con i pacchetti più recenti e includere eventuali patch di sicurezza o correzioni di bug recenti.

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 precedenti di Debian, 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 di Zabbix e, per aggiornare tali componenti dai pacchetti, valutare l'aggiornamento del sistema operativo. Per un elenco dei componenti inclusi, vedere Pacchetti Zabbix.

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 precedenti di Ubuntu, 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 di Zabbix e, per aggiornare tali componenti dai pacchetti, valutare l'aggiornamento del sistema operativo. Per un elenco dei componenti inclusi, vedere Pacchetti Zabbix.

Potrebbe essere visualizzata una richiesta relativa 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] ?
```

Immettere `Y` (o `I`) per installare la versione della configurazione del repository Zabbix fornita dal manutentore del pacchetto.

Quindi, aggiornare le informazioni del repository:

```
apt update
```

5 Aggiornare i 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 sta aggiornando il proxy, sostituire `server` con `proxy` nel comando.

- Se si sta aggiornando 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 visualizzato un prompt relativo alla configurazione di Zabbix server (o 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 corrente con quella nuova, quindi decidere se si desidera installare la versione del manutentore del pacchetto (`Y` o `I`).

Quindi, per aggiornare Zabbix frontend con Apache e riavviare Apache, eseguire:

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

6 Rivedere i parametri di configurazione dei componenti

Rivedere 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 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 Aggiornare il pacchetto di configurazione del repository

Prima di procedere con l'aggiornamento, aggiorna il pacchetto del repository attualmente in uso all'ultima versione 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
```

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 le sue derivate, sostituisci il link sopra con quello corretto dal repository Zabbix. Tieni presente, tuttavia, che i pacchetti per queste versioni potrebbero non includere tutti i componenti di Zabbix e, per aggiornare tali componenti dai pacchetti, valuta l'aggiornamento del tuo sistema operativo. Per un elenco dei componenti inclusi, vedi Pacchetti Zabbix.

Quindi, pulisci la cache del gestore di pacchetti `dnf` (inclusi header, metadati e file dei pacchetti scaricati durante installazioni o aggiornamenti precedenti):

```
dnf clean all
```

Alla successiva operazione `dnf`, `dnf` scaricherà metadati aggiornati dai repository, poiché i vecchi metadati sono stati rimossi.

Vedi anche: Problemi noti per aggiornare il pacchetto di configurazione del repository su RHEL.

5 Aggiornare i componenti di Zabbix

Per aggiornare i componenti di Zabbix, eseguire:

```
dnf install zabbix-server-mysql zabbix-web-mysql 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 sta aggiornando agent 2, sostituire `zabbix-agent` con `zabbix-agent2 zabbix-agent2-plugin-*` 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 di Zabbix con Apache e riavviare Apache, eseguire:

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

6 Rivedere i parametri di configurazione dei componenti

Rivedere 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 upgrade riuscito ai container di Zabbix **8.0**.x.

Sono disponibili insiemi separati di istruzioni per l'upgrade delle singole images dei componenti Zabbix e dei file Docker compose.

> **Warning:**
> Prima dell'upgrade, assicurarsi di leggere le **note di upgrade** pertinenti!

Attention:
Prima di avviare l'upgrade, verificare che gli utenti dispongano delle autorizzazioni necessarie sul database ai fini dell'upgrade.

Per gli upgrade da Zabbix 6.0 o versioni precedenti, durante l'upgrade dovranno essere creati trigger deterministici. Se il binary logging è abilitato per MySQL/MariaDB, ciò richiede privilegi di superuser 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 e 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'upgrade alla versione 8.0 potrebbe richiedere molto tempo.

Aggiornamento dell'immagine Zabbix

I passaggi elencati di seguito possono essere utilizzati per aggiornare qualsiasi componente Zabbix. Sostituire `zabbix-server-mysql` con il nome dell'immagine del componente richiesto.

1. Verificare la versione corrente dell'immagine:

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

2. Scaricare la versione desiderata dell'immagine, ad esempio:

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

`zabbix/zabbix-server-mysql:alpine-8.0-latest` scaricherà l'ultima minor release pubblicata di Zabbix server 8.0 con supporto MySQL basata su Alpine Linux. Sostituirla con il nome del repository Docker e la combinazione di tag necessari. Consultare Installation from containers per un elenco delle opzioni disponibili.

3. Arrestare il container:

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

4. Rimuovere il container:

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

5. Avviare il container aggiornato eseguendo il comando `docker run` seguito da argomenti aggiuntivi per specificare le environment variables e/o i mount points richiesti.

Esempi di configurazione

Zabbix server con MySQL:

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

Zabbix server con PostgreSQL:

```
docker run --name zabbix-server-pgsql -t \
    -e DB_SERVER_HOST="postgres-server" \
    -e POSTGRES_USER="zabbix" \
    -e POSTGRES_PASSWORD="zabbix_pwd" \
    -e POSTGRES_DB="zabbix" \
    -e ZBX_ENABLE_SNMP_TRAPS="true" \
    --network=zabbix-net \
    -p 10051:10051 \
```

```
    --volumes-from zabbix-snmptraps \
    --restart unless-stopped \
    -d zabbix/zabbix-server-pgsql:alpine-8.0-latest
```

Server MySQL:

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

Ulteriori esempi di configurazione, inclusi esempi per altri componenti Zabbix, sono disponibili nella pagina Installation from containers.

6. Verificare l'aggiornamento:

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

File Compose

Seguire le istruzioni di aggiornamento in questa sezione, se è stato installato Zabbix utilizzando il file compose.

1. Verificare la versione corrente dell'immagine:

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

2. Scaricare gli ultimi aggiornamenti dal repository GitHub e passare al branch richiesto:

```
git pull
git checkout 8.0
```

3. Avviare i componenti Zabbix utilizzando il nuovo file compose:

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

4. Verificare l'aggiornamento:

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

Per maggiori dettagli, inclusi gli elenchi delle variabili d'ambiente supportate e dei punti di mount dei volumi, vedere Installazione da container.

Aggiornamento dai sorgenti

Panoramica

Questa sezione fornisce i passaggi necessari per un aggiornamento riuscito da Zabbix **7.4**.x a Zabbix **8.0**.x utilizzando i sorgenti ufficiali di Zabbix.

> **Warning:**
> Prima dell'aggiornamento assicurarsi di leggere le **note di aggiornamento**!

Potresti anche voler consultare i requisiti per la versione 8.0.

> **Note:**
> Può essere utile eseguire due sessioni SSH parallele durante l'aggiornamento, eseguendo i passaggi dell'aggiornamento in una e monitorando i log del server/proxy nell'altra. Ad esempio, esegui `tail -f zabbix_server.log` oppure `tail -f zabbix_proxy.log` nella seconda sessione SSH per visualizzare in tempo reale le ultime voci del file di log e i possibili errori. Questo può essere fondamentale per gli ambienti di produzione.

Processo di aggiornamento del server

1 Arresta il server

Arresta il server Zabbix per assicurarti che non vengano inseriti nuovi dati nel database.

2 Esegui il backup del database Zabbix esistente

Questo è un passaggio molto importante. Assicurati di avere un backup del tuo database. Ti sarà utile se la procedura di aggiornamento fallisce (mancanza di spazio su disco, spegnimento, qualsiasi problema imprevisto).

3 Esegui il backup dei file di configurazione, dei file PHP e dei binari di Zabbix

Esegui 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 Avvia i nuovi binari di Zabbix

Avvia nuovi binari. Controlla i file di registro per vedere se i file binari sono stati avviati con successo.

Il server Zabbix aggiornerà automaticamente il database. All'avvio, Il server Zabbix riporta l'attuale (obbligatorio e facoltativo) e richiesto versioni della banca dati. Se la versione obbligatoria corrente è precedente alla versione richiesta, il server Zabbix esegue automaticamente il file richiesto patch di aggiornamento del database. Il livello iniziale e di avanzamento (percentuale) di l'aggiornamento del database viene scritto nel file di registro del server Zabbix. Quando il l'aggiornamento è completato, viene visualizzato un messaggio di "aggiornamento del database completamente completato". scritto nel file di registro. Se una delle patch di aggiornamento fallisce, Zabbix il server non si avvia. Anche il server Zabbix non si avvierà se il file current la versione obbligatoria del database è più recente di quella richiesta. Server Zabbix si avvierà solo se la versione attuale del database obbligatorio corrisponde a la versione obbligatoria richiesta.

```
8673:20161117:104750.259 versione corrente del database (obbligatorio/facoltativo): 03040000/03040000
8673:20161117:104750.259 versione obbligatoria richiesta: 03040000
```

Prima di avviare il server:

- Assicurati che l'utente del database disponga di autorizzazioni sufficienti (crea tabella, elimina tabella, crea indice, elimina indice)
- Assicurati di avere abbastanza spazio libero su disco.

7 Installare la nuova interfaccia web di Zabbix

La versione minima richiesta di PHP è 8.2.0. Aggiornare se necessario e seguire le istruzioni di installazione.

8 Cancella 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 Rivedere i parametri di configurazione del proxy

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.

5 Avvia il nuovo proxy Zabbix

Avvia il nuovo proxy Zabbix. Controlla 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 simile all'avvio del server Zabbix.

Processo di aggiornamento dell'agent

::: nota importante L'aggiornamento degli agent non è obbligatorio. È necessario aggiornare gli agent solo se necessario per accedere alle nuove funzionalità. :::

La procedura di aggiornamento descritta in questa sezione può essere utilizzata per aggiornare sia Zabbix agent che 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 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 Rivedere i parametri di configurazione dell'agent

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.

5 Avvia il nuovo Zabbix agent

Avvia il nuovo Zabbix agent. Controlla i file di log per verificare che l'agente sia stato avviato correttamente.

Aggiornamento tra versioni minori

Quando si esegue l'aggiornamento tra versioni minori di 8.0.x (ad esempio da 8.0.1 a 8.0.3) è necessario eseguire le stesse azioni per server/proxy/agent come durante l'aggiornamento tra versioni maggiori. L'unica differenza è che, durante l'aggiornamento tra versioni minori, 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:

```
\\\
```

Trig_228 - analogamente a Calc_228, il suo parametro contiene 228 barre rovesciate per **corrispondere** a 114 barre rovesciate reali del file /tmp/ONE:

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

```
\\\
```

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

Rimossa la validazione JSON e XML dagli item HTTP agent

La validazione JSON e XML è stata rimossa dal campo *Corpo della richiesta* negli item HTTP agent. Il campo non verifica più se il valore è un XML o JSON valido e la selezione di *Dati XML* come *Tipo di corpo della richiesta* non richiede più la libreria libxml2.

Link disabilitati nascosti negli host monitorati

I link *Grafici*, *Dashboard* e *Web* disabilitati non vengono più visualizzati nell'elenco degli host monitorati.

Tipo di dati JSON

Zabbix ora supporta JSON come tipo di dati per i valori degli item. Se si utilizza TimescaleDB, la nuova hypertable `history_json` (usata per memorizzare i valori JSON) deve essere configurata manualmente. Se si utilizza Elasticsearch, il valore predefinito del parametro di configurazione del server Zabbix `HistoryStorageTypes` ora include `json`.

Pulsante di opzione rimosso nel modulo Nuovo filtro tag

Il pulsante di opzione che richiedeva di scegliere tra *Tutti i tag* e *Elenco tag* durante la creazione di un nuovo filtro tag è stato rimosso.

Valori degli item visualizzati come HTML utilizzando iframe

I dati degli item nel widget Cronologia degli item, quando visualizzati come testo formattato in HTML, ora sono isolati in iframe.

Template compatibility with Host Wizard

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

Modifiche ai template

Questa pagina elenca tutte le modifiche ai template predefiniti forniti con Zabbix.

Durante l'aggiornamento di Zabbix, i template esistenti non vengono aggiornati automaticamente per evitare la sovrascrittura di modifiche personalizzate. Per aggiornare i template o aggiungerne di nuovi, vedere Aggiornamento dei template.

> **Note:**
> Si informa che a partire da Zabbix 6.0 tutti i template seguono un formato aggiornato, che potrebbe influire sull'importazione dei template precedenti alla versione 6.0. Per ulteriori informazioni, vedere Modifiche ai template in 6.0.

Modifiche nella versione 8.0.0

Note di aggiornamento per Zabbix 8.0.x

Questa pagina fornisce note di aggiornamento collettive per le release minori della versione principale di Zabbix.

Vedi anche le note di aggiornamento della versione principale.

Note di aggiornamento per 8.0.1 Questa versione non è ancora stata rilasciata.

Problemi noti

Vedi anche: Problemi di compilazione.

Aggiornamento

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).

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.

Timescale DB: utilizzo elevato della memoria con un numero elevato di partizioni

Le versioni di PostgreSQL 9.6-12 utilizzano troppa memoria durante l'aggiornamento di tabelle con un numero elevato di partizioni. Questo problema si manifesta quando Zabbix aggiorna i trend su sistemi con TimescaleDB se i trend sono suddivisi in chunk relativamente piccoli (ad esempio 1 giorno).
Ciò comporta la presenza di centinaia di chunk nelle tabelle dei trend con le impostazioni di housekeeping predefinite, una condizione in cui PostgreSQL rischia di esaurire la memoria.

Il problema è stato risolto a partire da Zabbix 5.0.1 per le nuove installazioni con TimescaleDB, ma se TimescaleDB è stato configurato con Zabbix prima di allora, consultare ZBX-16347 per le note sulla migrazione.

Timescale DB 2.5.0: il criterio di compressione può non riuscire sulle tabelle che contengono interi

Questo problema si manifesta quando viene utilizzato TimescaleDB 2.5.0/2.5.1.
È stato risolto a partire da TimescaleDB 2.5.2.

Per ulteriori informazioni, consultare TimescaleDB Issue #3773.

Connessione TLS al database con MariaDB

La connessione TLS al database non è supportata con l'opzione 'verify_ca' per il parameter 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 — host non attendibili possono mandare in crash OpenIPMI

Esiste 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 il crash della libreria OpenIPMI, il che a sua volta può causare la terminazione del processo Zabbix server 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 del 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 monitorare database Oracle usando 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 verrà restituito 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 casuale delle impostazioni locali del frontend

È stato osservato che le impostazioni locali del frontend possono cambiare senza una logica apparente, cioè alcune pagine (o parti di pagine) vengono visualizzate in una lingua, mentre altre pagine (o parti di pagine) in una lingua diversa. In genere il problema può comparire quando ci sono diversi utenti, alcuni dei quali usano una locale, mentre altri ne usano un'altra.

Una soluzione alternativa nota consiste nel disabilitare il multithreading in PHP e Apache.

Il problema è legato al modo in cui l'impostazione della locale funziona in PHP: le informazioni sulla locale vengono mantenute per processo, non per thread. Quindi, in un ambiente multithread, quando ci sono diversi progetti eseguiti dallo stesso processo Apache, è possibile che la locale venga modificata in un altro thread e che ciò cambi il modo in cui i dati possono essere elaborati nel thread di Zabbix.

Per ulteriori informazioni, vedere le segnalazioni di problema correlate:

- ZBX-10911 (Problema con il cambio casuale delle impostazioni locali del frontend)
- ZBX-16297 (Problema con l'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 di Zabbix agent su Windows

Le voci DNS inesistenti nel parametro `Server` del file di configurazione di Zabbix agent (zabbix_agentd.conf) possono aumentare il tempo di risposta di Zabbix agent su Windows. Questo accade perché il demone 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

Esistono alcuni problemi noti con l'esportazione/importazione YAML:

- I messaggi di errore non sono traducibili;
- JSON valido con estensione di file .yaml a volte non può essere importato;
- 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 riesce a 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 soluzioni alternative:

- 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 utilizzare PCRE2.
Pur offrendo numerosi vantaggi, il passaggio a PCRE2 può causare l'invalidità di alcuni pattern regexp PCRE esistenti o un comportamento differente.
In particolare, questo riguarda il pattern *^[\w-\.]*.
Per rendere nuovamente valida questa regexp senza alterarne la semantica, modificare l'espressione in *^[-\w\.]* .
Ciò accade perché PCRE2 tratta il trattino come un 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 è stato eseguito l'aggiornamento da una versione precedente di Zabbix con NGINX e durante l'aggiornamento non è stato adottato il nuovo file di configurazione NGINX.

Per risolvere il problema, è possibile scartare il vecchio file di configurazione, utilizzare 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, è possibile modificare manualmente un file di configurazione NGINX esistente (in genere, */etc/zabbix/nginx.conf*). Per farlo, aprire il file e individuare il seguente blocco:

```
location ~ /(api\/|conf[^\.]|include|locale|vendor) {
        deny            all;
        return          404;
}
```

Quindi, sostituire 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, 𝓍). 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.

Accesso agli elementi dell'interfaccia utente con MariaDB 10.5.1–10.5.9

L'accesso al frontend web di Zabbix con un ruolo diverso da Super Admin può comportare la visualizzazione del messaggio: "Si è verificato un errore di sistema. Contattare l'amministratore Zabbix.". Questo problema interessa le installazioni che utilizzano versioni di MariaDB dalla 10.5.1 alla 10.5.9.

Per evitare questo problema, aggiornare MariaDB a una versione successiva alla 10.5.9. Per maggiori dettagli, vedere ZBX-25746.

Analisi dell'utilizzo eccessivo della memoria con tcmalloc

Se sospetti che la tua installazione di Zabbix stia utilizzando troppa memoria, puoi usare la funzionalità di profiling della memoria di tcmalloc per analizzare il consumo di memoria del server/proxy Zabbix.

1. Durante l'installazione di 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 aggiuntive, mentre `-O0` disabilita le ottimizzazioni, che possono interferire con il profiling di tcmalloc.

2. Imposta le seguenti variabili d'ambiente prima di avviare Zabbix server. Queste variabili indicano a tcmalloc come tracciare e riportare l'utilizzo 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. Attiva un dump del profilo inviando il segnale 5 al processo di destinazione. Sostituisci 1234 con l'ID di processo (PID) effettivo:

```
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 le segnalazioni di problema correlate.
- GCC Option Summary sulle opzioni di compilazione (`-std=gnu99`, `-g`, `-O0`, ecc.).
- Documentazione di Gperftools Heap Profiler sulle variabili d'ambiente per il profiling 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.

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, assumendo 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 benchmark delle prestazioni del sistema Zabbix in un ambiente di staging o di sviluppo, in modo da comprendere pienamente 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] 1 metrica = 1 item + 1 trigger + 1 grafico
 [2] Esempio con istanze Amazon EC2 per uso generico, che utilizzano architettura ARM64 o x86_64; durante la valutazione e i test dell'installazione di Zabbix, prima dell'installazione nell'ambiente di produzione, deve essere selezionato un tipo di istanza appropriato, ad esempio ottimizzato per calcolo/memoria/storage.

Note:
La configurazione effettiva dipende in larga misura dal numero di item attivi e dagli intervalli di aggiornamento (per i dettagli, vedere la sezione dimensione del database di questa pagina). Per 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

Se indicato come obbligatorio, il software/la libreria richiesta è strettamente necessaria. Quelli opzionali sono necessari per supportare alcune funzioni specifiche.

Software	Stato obbligatorietà	Versioni supportate	Commenti
MySQL/Percona	Uno tra	8.4.0-9.5.X	Richiesto se MySQL (o Percona) viene utilizzato come database backend di Zabbix. È richiesto il motore InnoDB. Raccomandiamo di utilizzare la libreria C API (libmysqlclient) per compilare server/proxy.

Software	Stato obbligatorietà	Versioni supportate	Commenti
MariaDB		10.11.00-12.0.X	È richiesto il motore InnoDB. La versione raccomandata è 11.4. Raccomandiamo di utilizzare la libreria MariaDB Connector/C per compilare server/proxy. Vedi anche: Possibili deadlock con MariaDB e Accesso agli elementi dell'interfaccia utente con MariaDB 10.5.1–10.5.9.
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 (4MB è il valore predefinito), in modo che la quantità di memoria utilizzata dal database per una particolare operazione sia sufficiente e l'esecuzione delle query non richieda troppo tempo.
TimescaleDB per PostgreSQL		2.20.X-2.25.X	Richiesto se TimescaleDB viene utilizzato come estensione del database PostgreSQL. Assicurarsi di installare TimescaleDB Community Edition, che supporta la compressione. Si noti che PostgreSQL 15 è supportato a partire da TimescaleDB 2.10. Per dettagli sulla compatibilità delle versioni di PostgreSQL e TimescaleDB, è possibile fare riferimento anche alla documentazione di TimescaleDB.
SQLite	Opzionale	3.3.5-3.34.X	SQLite è supportato solo con i proxy Zabbix. Richiesto se SQLite viene utilizzato come database del proxy Zabbix.
Elasticsearch		7.X	Elasticsearch è supportato solo con i server Zabbix, esclusivamente per l'archiviazione dei dati storici. Il supporto di Elasticsearch è attualmente sperimentale. Vedi anche il software richiesto per server/proxy.
smartmontools		7.1 o successiva	Richiesto per Zabbix agent 2.
who			Richiesto per il plugin del conteggio 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`, come parte di q applets, sono richiesti per il plugin system.sw.packages su Gentoo Linux.

Note:
Sebbene Zabbix possa funzionare con database disponibili nei sistemi operativi, per ottenere la migliore esperienza consigliamo di utilizzare database installati dai repository ufficiali degli sviluppatori del database.

Frontend

La larghezza minima dello schermo supportata per il frontend di Zabbix è 1200px.

Se indicato come obbligatorio, il software/la libreria richiesta è strettamente necessaria. Quelli opzionali sono necessari per supportare alcune funzioni specifiche.

Software	Stato obbligatorietà	Versioni supportate	Commenti
PHP	Sì	8.2.0 - 8.4.X	
Apache	Uno tra	2.4 o successiva	
Nginx		1.20 o successiva	

Software	Stato obbligatorietà	Versioni supportate	Commenti
MySQL	Uno tra	Vedi Software esterno di terze parti dell'ambiente circostante	
PostgreSQL			
Estensioni PHP			
mysqli	Sì		Richiesta se MySQL è utilizzato come database backend di Zabbix.
pgsql			Richiesta se PostgreSQL è utilizzato come database backend di Zabbix.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); richiesta per il supporto degli script utente.
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*). La versione 2.3.0 o successiva può essere richiesta 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; richiesta per l'autenticazione LDAP.
openssl			php-openssl; richiesta per l'autenticazione SAML.
gettext			php-gettext (*--with-gettext*); richiesta per le traduzioni.
cURL		7.19.4 o successiva	php-curl; richiesta per Duo Universal Prompt MFA e per l'autenticazione SMTP.

Librerie frontend di terze parti fornite con Zabbix:

Libreria	Stato obbligatorietà	Versione fornita	Commenti
jQuery JavaScript Library	Sì	3.6.0	Libreria JavaScript che semplifica il processo di sviluppo cross-browser.
jQuery UI		1.12.1	Un insieme di interazioni dell'interfaccia utente, effetti, widget e temi costruiti su jQuery.
SAML PHP Toolkit		4.3.1	Un toolkit PHP che aggiunge il supporto per l'autenticazione SAML 2.0 per consentire l'accesso a Zabbix.
Symfony Yaml Component		5.1.0	Aggiunge il supporto per esportare e importare elementi di configurazione di Zabbix nel formato YAML.

Note:
Zabbix può funzionare anche con versioni precedenti di Apache, MySQL e PostgreSQL.

Attention:
Per font diversi da DejaVu 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 è 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 obbligatorietà	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 di unit testing PHP 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 richiesta è strettamente necessaria. Quelli opzionali sono necessari per supportare alcune funzioni specifiche.

Requirement	Mandatory status	Description
libpcre2	Sì	La libreria PCRE2 è richiesta per il supporto di Perl Compatible Regular Expression (PCRE). È supportata PCRE2 v10.x.
libevent		Richiesta per la comunicazione tra processi. Versione 2.0.10 o successiva.
libevent-pthreads		Richiesta per la comunicazione tra processi.
libpthread		Richiesta per il supporto di mutex e lock in lettura-scrittura (può far parte di libc).
libresolv		Richiesta per la risoluzione DNS (può far parte di libc).
libiconv		Richiesta per la codifica/convertizione del formato del testo (può far parte di libc). Obbligatoria per Zabbix server su Linux.
libz		Richiesta per il supporto della compressione.
libm		Libreria matematica. Richiesta solo da Zabbix server.
libmysqlclient	Una delle	Richiesta se viene utilizzato MySQL.
libmariadb		Richiesta se viene utilizzato MariaDB.
libpq5		Richiesta se viene utilizzato PostgreSQL; la versione di *libpq5* deve corrispondere o essere superiore alla versione del database PostgreSQL utilizzato.
libsqlite3		Richiesta se viene utilizzato SQLite. Richiesta solo per Zabbix proxy.
libOpenIPMI	No	Richiesta per il supporto IPMI. Richiesta solo da Zabbix server.
libssh2 or *libssh*		Richiesta per i controlli SSH. Versione 1.8.0 o successiva (libssh2); 0.9.0 o successiva (libssh).
libcurl		Richiesta 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). Per tutte le funzionalità è consigliata la versione 7.28.0 o successiva. 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		Richiesta per il monitoraggio VMware e il preprocessing XML XPath.

Requirement	Mandatory status	Description
net-snmp		Richiesta per il supporto SNMP. Versione 5.3.0 o successiva. Il supporto dei protocolli di crittografia avanzata (AES192/AES192C, AES256/AES256C) è disponibile a partire dalla libreria net-snmp 5.8; sui sistemi basati su RHEL 8+ si consiglia di utilizzare net-snmp 5.8.15 o successiva.
libunixodbc		Richiesta per il monitoraggio del database.
libgnutls or *libopenssl*		Richiesta quando si utilizza la crittografia. Versioni minime: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Richiesta per il supporto LDAP.
fping		Richiesto per gli item ICMP ping.
c-ares		Richiesto per la risoluzione DNS asincrona se Zabbix è configurato con l'opzione `--with-ares`. In caso contrario, verrà utilizzato *libevent*. Versione minima: 1.16.0. La cache delle query DNS (cache delle query abilitata per impostazione predefinita) richiede c-ares 1.26.0 o successiva.

Agent

Requisito	Stato obbligatorio	Descrizione
libpcre2	Sì	La libreria PCRE2 è richiesta per il supporto di Perl Compatible Regular Expression (PCRE). PCRE2 v10.x è supportata. Richiesta per il monitoraggio dei log. Richiesta anche su Windows.
libpthread		Richiesta per il supporto di mutex e blocchi lettura-scrittura (può essere parte di libc). Non richiesta su Windows.
libresolv		Richiesta per la risoluzione DNS (può essere parte di libc). Non richiesta su Windows.
libiconv		Richiesta per la codifica del testo/la conversione del formato in UTF-8 negli item log, contenuto file, file regex e regmatch (può essere parte di libc). Non richiesta su Windows.
libgnutls or *libopenssl*	No	Richiesta se si utilizza la crittografia. Versioni minime: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Su Microsoft Windows è richiesto OpenSSL 1.1.1 o successivo.
libldap		Richiesta se viene utilizzato 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 successiva (si raccomanda la 7.28.0 o successiva). Per utilizzare le funzionalità cURL aggiornate, riavviare Zabbix agent.
libmodbus		Richiesta solo se viene utilizzato il monitoraggio Modbus. Versione 3.0 o successiva.

Agent 2

Requisito	Obbligatorietà	Descrizione
Go	Sì	Richiesto per compilare Zabbix agent 2 e i relativi plugin dal codice sorgente. È supportato Go 1.24.10 o successivo. 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 - Supporto plugin - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Plugin di esempio
libpcre2	Sì	La libreria PCRE2 è richiesta per il supporto di Perl Compatible Regular Expression (PCRE). È supportato PCRE2 v10.x. Richiesta per il monitoraggio dei log. Richiesta anche su Windows.
libopenssl	No	Richiesto quando si utilizza la crittografia. OpenSSL 1.0.1 o successivo è richiesto sulle piattaforme UNIX. La libreria OpenSSL deve avere il supporto PSK abilitato. LibreSSL non è supportato. Sui sistemi Microsoft Windows è richiesto OpenSSL 1.1.1 o successivo.

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 hai ottenuto Zabbix dal repository dei sorgenti o da un archivio, le dipendenze necessarie sono già incluse nell'albero dei sorgenti.

Se hai ottenuto Zabbix dal pacchetto della tua distribuzione, le dipendenze necessarie sono già fornite dal sistema di pacchettizzazione.

In entrambi i casi sopra indicati, il software è pronto per essere utilizzato e non sono necessari download aggiuntivi.

Se tuttavia desideri fornire le tue versioni di queste dipendenze (ad esempio, se stai preparando un pacchetto per una distribuzione Linux), di seguito è riportato l'elenco delle versioni delle librerie con cui Java gateway è noto per funzionare. 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:

Library	Mandatory status	Bundled version	Comments
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		1.5.16	
logback-core		1.5.16	
slf4j-api		2.0.16	

Java gateway può essere compilato utilizzando Oracle Java oppure OpenJDK open source (versione 1.6 o successiva). I pacchetti forniti da Zabbix sono compilati utilizzando OpenJDK. La tabella seguente elenca i pacchetti OpenJDK utilizzati per compilare i pacchetti Zabbix per distribuzione:

Distribution	OpenJDK package
AlmaLinux 9	`java-11-openjdk-devel` (amd64: 11.0.19.0.7-4; arm64: 11.0.20.0.8-3)
AlmaLinux 8	`java-1.8.0-openjdk-devel` (amd64: 1.8.0.332.b09-2; arm64: 1.8.0.382.b05-2)
Amazon Linux 2023	`java-22-amazon-corretto-devel` (amd64, arm64: 22.0.2+9-1)

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

Numeri di porta predefiniti

Il seguente elenco delle porte aperte per componente è applicabile alla configurazione predefinita:

Componente Zabbix	Numero di porta	Protocollo	Tipo di connessione
Zabbix agent	10050	TCP	su richiesta
Zabbix agent 2	10050	TCP	su richiesta
Zabbix server	10051	TCP	su richiesta
Zabbix proxy	10051	TCP	su richiesta
Zabbix Java gateway	10052	TCP	su richiesta
Zabbix web service	10053	TCP	su richiesta
Zabbix frontend	80	HTTP	su richiesta
	443	HTTPS	su richiesta
Zabbix trapper	10051	TCP	su richiesta

Note:
I numeri di porta devono essere aperti nel firewall per abilitare le comunicazioni di Zabbix. Le connessioni TCP in uscita di solito non richiedono impostazioni esplicite del firewall.

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 Zabbix server 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 di housekeeper per lo storico

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 storico 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 e 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 degli item di testo/log è impossibile da prevedere con precisione, ma si possono stimare circa 500 byte per valore.

- Impostazione di housekeeper per i trend

Zabbix conserva un insieme di valori max/min/avg/count su 1 ora per ciascun item nella tabella **trends**. Questi dati vengono utilizzati per i trend e per i grafici 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 totale. 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, ovvero **11GB** per 5 anni.

- Impostazioni di housekeeper per gli eventi

Ogni evento di Zabbix richiede circa 250 byte di spazio su disco[1]. È difficile stimare il numero di eventi generati quotidianamente da Zabbix. Nello scenario peggiore, possiamo assumere che Zabbix generi un evento al secondo.

Per ogni evento ripristinato, viene creato un record event_recovery. Normalmente la maggior parte degli eventi verrà ripristinata, quindi possiamo assumere un record event_recovery per evento. Ciò significa ulteriori 80 byte per evento.

Facoltativamente, gli eventi possono avere tag; ogni record di tag richiede circa 100 byte di spazio su disco[1]. Il numero di tag per evento (#tags) dipende dalla configurazione. Quindi ciascun evento 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] Di più 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 per il sistema Zabbix:

Parameter	Formula for required disk space (in bytes)
Configurazione di Zabbix	Dimensione fissa. Normalmente 10MB o meno.
Storico	days*(items/refresh rate)*24*3600*bytes items : numero di item days : numero di giorni per cui conservare lo storico refresh rate : intervallo medio di aggiornamento degli item bytes : numero di byte necessari per conservare un singolo valore, dipende dal motore di database, normalmente ~90 byte.
Trend	days*(items/3600)*24*3600*bytes items : numero di item days : numero di giorni per cui conservare lo storico bytes : numero di byte necessari per conservare un singolo trend, dipende dal motore di database, normalmente ~90 byte.
Eventi	days*events*24*3600*bytes events : numero di eventi al secondo. Un (1) evento al secondo nello scenario peggiore. days : numero di giorni per cui conservare lo storico bytes : numero di byte necessari per conservare un singolo evento, dipende dal motore di database, normalmente ~330 + numero medio di tag per evento * 100 byte.

Quindi, lo spazio totale su disco richiesto può essere calcolato come:

Configurazione + Storico + Trend + Eventi

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 di 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.

Requisiti di rete

Il seguente elenco delle porte aperte per componente è applicabile alla configurazione predefinita.

Componenti	Porta
Frontend	http sulla 80, https sulla 443
Server	10051 (per l'uso con proxy/agent attivi)
Proxy attivo	10051
Proxy passivo	10051
Agent2	10050
Trapper	
Java gateway	10052
WebService	10053

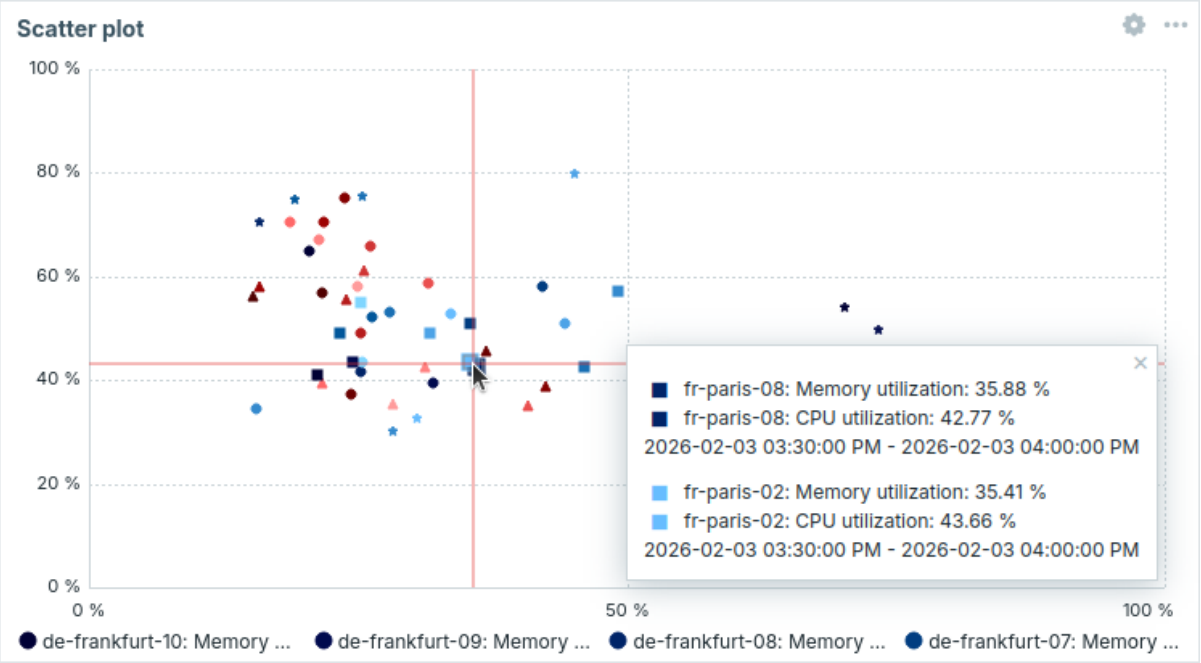
Note:
I numeri di porta devono essere aperti nel firewall per abilitare le comunicazioni esterne con Zabbix. Le connessioni TCP in uscita di solito non richiedono impostazioni esplicite del firewall.

2 Novità di Zabbix 8.0

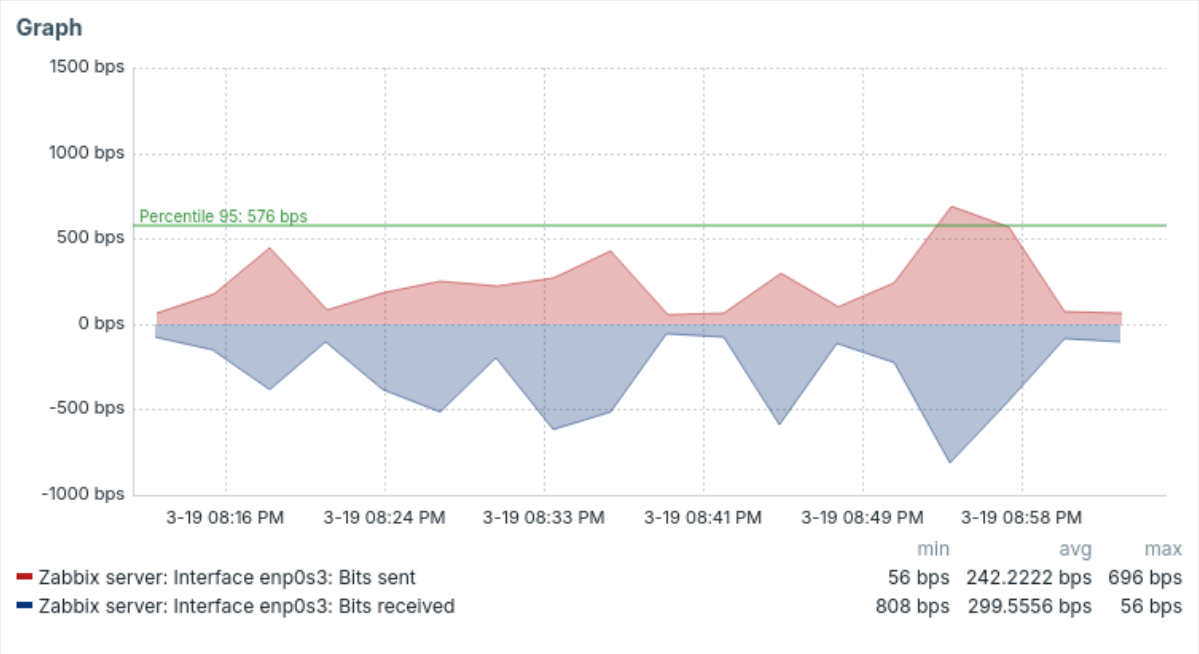
Zabbix 8.0.0 prosegue da Zabbix 7.4.0, aggiungendo nuove funzionalità e miglioramenti.

Vedere le modifiche incompatibili per questa versione.

Widget grafico a dispersione Il widget Scatter plot è stato aggiunto ai widget della dashboard. Visualizza la relazione tra due metriche tracciando singoli punti dati lungo un asse X e un asse Y. Questo aiuta a individuare schemi, cluster, correlazioni e valori anomali nel set di dati.



Inversione dell'asse Y per il widget Graph I graph nel widget Graph ora possono essere visualizzati con i valori dell'asse Y invertiti. Una nuova impostazione, *Invert values*, consente di moltiplicare per −1 i valori dell'asse Y senza alterare i dati originali.



Vedi anche: Miglioramenti del widget Graph.

Template

Nuovi template

- Aruba CX 8300s by SNMP, un template che fornisce il monitoraggio basato su SNMP per la serie di switch Aruba CX 8300.
- Il set di template AWS by HTTP è stato integrato con il template *AWS Backup Vault by HTTP*.
- Il set di template Azure by HTTP è stato integrato con il template *Azure Sentinel by HTTP*.
- Ciena 3906 by SNMP, un template per il monitoraggio dei dispositivi Ciena 3906.
- Cisco Secure Firewall Threat Defense by HTTP, un template che fornisce funzionalità di monitoraggio per i dispositivi Cisco Secure Firewall Threat Defense tramite REST API.
- Cradlepoint NCM v2 by HTTP, un set di template per il monitoraggio di Cradlepoint NCM v2 e dei relativi dispositivi tramite HTTP.
- Domain RDAP by HTTP, un template per il monitoraggio dei dati di registrazione dei domini tramite il protocollo RDAP.
- IBM Maximo Service Request, un template basato su webhook che consente di collegare Zabbix a IBM Maximo.
- MariaDB by ODBC, un template per il monitoraggio dei database MariaDB tramite ODBC.
- Microsoft Hyper-V Failover Cluster by SSH e Microsoft Hyper-V Standalone by SSH, template per il monitoraggio dei cluster Microsoft Hyper-V e degli host standalone tramite SSH.
- OpenAI Platform by HTTP, un template per il monitoraggio della piattaforma per sviluppatori di OpenAI.
- Percona by ODBC, un template per il monitoraggio dei database Percona tramite ODBC.
- Ribbon SBC Edge by HTTP, un template per il monitoraggio dei dispositivi Ribbon SBC Edge (in precedenza SWe Lite) tramite HTTP.
- Ribbon SBC SWe Core by HTTP, un template per il monitoraggio dei dispositivi SBC SWe Core tramite HTTP, e Ribbon SBC SWe CE by HTTP, un template per il monitoraggio delle istanze Ribbon SBC SWe Call Engine (CE) tramite HTTP.
- Stormshield SNS by SNMP, un template per il monitoraggio dei dispositivi Stormshield Network Security (SNS) tramite SNMP.
- VeloCloud SD-WAN Edge by HTTP, un template per il monitoraggio dei dispositivi VeloCloud SD-WAN Edge tramite HTTP.
- Vyatta Virtual Router by SNMP, un template per il monitoraggio del router virtuale Vyatta 1908e.

Template aggiornati

- Ciena 3906 by SNMP è stato aggiornato per includere item del filesystem e del carico della CPU.
- Il set di template GCP by HTTP è stato integrato con il template *GCP Cloud Run Service by HTTP*.
- GitHub organization by HTTP è stato aggiornato per includere item per il monitoraggio di Microsoft Copilot.
- Microsoft 365 reports by HTTP è stato aggiornato per includere item per il monitoraggio di Microsoft Copilot.
- MySQL by Zabbix agent, MySQL by Zabbix agent 2, MySQL by Zabbix agent active, MySQL by Zabbix agent 2 active e MySQL by ODBC sono stati aggiornati per supportare sia `SHOW SLAVE STATUS` (sintassi precedente) sia `SHOW REPLICA STATUS` (nuova sintassi).
- MySQL by ODBC è stato inoltre aggiornato con nuove metriche, regole di discovery per tabelle e repliche e una dashboard migliorata.
- PostgreSQL by ODBC è stato aggiornato con nuove metriche, regole di discovery e dashboard, insieme a miglioramenti generali delle prestazioni e dell'osservabilità.

- Proxmox VE by HTTP è stato aggiornato con la funzionalità LLD nidificata. Inoltre, il formato dell'unità per gli item che visualizzano percentuali è stato adattato per una maggiore chiarezza.
- VeloCloud SD-WAN by HTTP, in precedenza *VMWare SD-WAN VeloCloud by HTTP*, è stato rinominato e aggiornato per allinearsi alla più recente piattaforma VeloCloud SD-WAN.
- Vyatta Virtual Router by SNMP è stato aggiornato con nuovi item OID e miglioramenti alla dashboard.

Item

Tipo di dato JSON Zabbix ora supporta JSON come tipo di dato per i valori degli item.

In precedenza, i valori JSON venivano raccolti dagli item di testo e memorizzati come stringhe con un limite di 64KB. Ora Zabbix può memorizzare i valori JSON in modo nativo con un limite di 128MiB, oltre a rifiutare valori JSON non validi (ad esempio contenenti chiavi senza virgolette, virgole finali, parentesi non corrispondenti).

Il tipo di dato JSON è supportato da tutti i tipi di item e dai prototipi di item (tranne *Calculated*) ed è disponibile in esportazione dei dati in tempo reale e connector. I valori JSON possono essere memorizzati in tutti i database supportati e in Elasticsearch. Se utilizzi TimescaleDB, consulta le note di aggiornamento.

Tieni presente che gli item JSON non possono essere utilizzati nei trigger; tuttavia, puoi estrarre i campi JSON con item dipendenti che hanno un tipo di dato non JSON e usarli nei trigger.

Gli item che restituiscono una stringa JSON (`net.if.discovery`, `vfs.file.get`, ecc.) sono ancora item di testo; tuttavia, puoi cambiarli in JSON se necessario.

Per maggiori dettagli e per i limiti dei dati JSON, vedi la configurazione dell'item.

Parametro type per il rilevamento dei dischi S.M.A.R.T L'item smart.disk.discovery (plugin S.M.A.R.T. di Zabbix agent 2) ora accetta un parametro facoltativo **type** per specificare un valore da cercare durante la scansione dei dischi.

Plugin

Plugin Ceph Questo plugin ora funziona in due modalità:

- **native** - Questa modalità utilizza la libreria go-ceph per comunicare direttamente con il cluster Ceph usando l'API nativa di Ceph (protocollo msgr2). Questa è la modalità consigliata per le installazioni Ceph moderne, ma è supportata solo **su Linux** e a partire da Ceph 16.
- **restful** (deprecata) - Questa modalità utilizza l'API RESTful di Ceph per la comunicazione. È la modalità predefinita per garantire la retrocompatibilità, ma non funzionerà con Ceph versione 20 (Tentacle) o successive a causa della rimozione del modulo mgr/restful.

La modalità da utilizzare è determinata dal valore del parametro mode (native/restful):

- `Plugins.Ceph.Default.Mode=native` - imposta la modalità nativa per il plugin
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` - imposta la modalità nativa per la sessione denominata

Si noti che il set di credenziali utente differisce per ciascuna modalità e non sono compatibili tra loro. Il parametro `Plugins.Ceph.InsecureSkipVerify` viene ignorato in modalità nativa perché la sicurezza della connessione è definita lato cluster Ceph dal protocollo msgr2 (sicuro per impostazione predefinita).

Si noti che il plugin Ceph per Zabbix agent 2 è ora un plugin caricabile e richiede passaggi di installazione aggiuntivi. Ciò è dovuto al requisito del pacchetto librados (per la modalità nativa). Per i dettagli, vedere il readme del plugin Ceph.

Plugin MongoDB Il plugin MongoDB di Zabbix agent 2 offre diversi miglioramenti: - Supporto completo per l'analisi dei tradizionali URI MongoDB, con supporto per gli schemi `mongodb://` e `mongodb+srv://`. - Rilevamento automatico dei nodi del replica set MongoDB, migliorando la capacità di monitorare i cluster MongoDB. - Supporto per l'autenticazione x509, che consente connessioni sicure a MongoDB tramite certificati client.

Plugin Oracle Zabbix agent 2 plugin Oracle ora supporta connessioni crittografate ai database Oracle utilizzando il protocollo TCPS (TLS). Questo aggiunge la possibilità di monitorare le istanze Oracle tramite socket protetti con TLS, migliorando la sicurezza del monitoraggio remoto.

Plugin Redis — supporto TLS e convalida all'avvio È stato aggiunto il supporto TLS al plugin Redis per Zabbix agent 2.

È stata implementata la convalida della configurazione TLS del plugin all'avvio e sono stati migliorati i messaggi di convalida/errore. Una logica di configurazione non valida (ad esempio: l'uso del tipo di connessione `verify_full` senza specificare `TLSCAFile`) può impedire l'avvio di Zabbix agent 2.

Modalità di esecuzione di test per i plugin caricabili I plugin caricabili ora possono essere avviati in modalità di test utilizzando il flag `-t` (`--test`), passando una chiave item come argomento. In questa modalità, il plugin viene eseguito per scopi di debug e sviluppo e i file di configurazione del plugin vengono ignorati.

Low-level discovery

Casella di controllo Convert to JSON nei moduli di discovery È stata aggiunta una nuova casella di controllo *Convert to JSON* nel modulo della discovery rule e nel modulo del discovery prototype, visualizzata se nel menu a discesa *Type* è selezionato "HTTP agent". Questa opzione consente di contrassegnare i dati recuperati per la conversione automatica in JSON prima di un'ulteriore elaborazione.

Supporto delle macro per il low-level discovery annidato Le macro di low-level discovery sono ora supportate nelle regole di low-level discovery annidate, in:

- parametri di preprocessing JSONPath parameters
- campo JSONPath per le macro LLD personalizzate

Tag modificabili per i trigger creati dal low-level discovery I trigger creati dai prototipi di trigger possono ora avere tag aggiunti manualmente. I tag ereditati dai prototipi di trigger vengono ancora applicati automaticamente. I tag aggiunti manualmente possono essere modificati sui trigger rilevati e saranno inclusi negli array di tag degli eventi e disponibili per le funzionalità che riconoscono i tag, come filtri, widget della dashboard e macro di notifica.

Processi

Cache e riutilizzo dell'EngineID SNMPv3 Zabbix ora memorizza nella cache le mappature EngineID → IP di SNMPv3 e tenterà di riutilizzare gli EngineID memorizzati nella cache per i successivi controlli SNMPv3, riducendo il traffico di probing e migliorando le prestazioni del poller. Se un EngineID riutilizzato non risponde, il poller torna a eseguire un probe dell'EngineID e può rimuovere le voci obsolete dopo modifiche dell'interfaccia o errori persistenti.

Limitazione del proxy perfezionata durante il ripristino della cache della cronologia La logica di limitazione del proxy è stata perfezionata per migliorare la stabilità del server durante il ripristino della cache della cronologia. Quando l'utilizzo della cache della cronologia raggiunge la soglia di limitazione, il server continua a non accettare i dati del proxy come in precedenza. Quando l'utilizzo della cache scende al 60%, il server inizia a elaborare l'elenco di limitazione, ma può comunque rifiutare i caricamenti del proxy che contengono batch molto grandi (circa più di 10k record) finché la pressione sulla cache non diminuisce ulteriormente. Questa modifica riduce il rischio di ripetuti sovraccarichi della cache durante il ripristino del server.

Valore massimo del timeout aumentato per zabbix_get e zabbix_js Il valore massimo del parametro `timeout` delle utility da riga di comando zabbix_get e zabbix_js è stato aumentato a 600 secondi.

Housekeeping ottimizzato L'housekeeping è stato ottimizzato spostando la creazione dei task di housekeeper nei trigger del database. Quando item (incluse le regole di low-level discovery), trigger, servizi o regole di network discovery vengono eliminati, i trigger del database (invece di chiamate esplicite a livello applicativo) popolano ora la tabella `housekeeper` con i task di pulizia. Per i dettagli, vedere la procedura di housekeeping.

L'housekeeper ora rimuove anche gli eventi di network discovery generati da host o servizi rilevati che sono stati successivamente eliminati, nonché tutti gli eventi associati ai problemi dei trigger che sono stati successivamente eliminati (in precedenza venivano rimossi solo i problemi stessi; gli eventi associati venivano rimossi solo dopo la scadenza del *periodo di conservazione dei dati dei trigger* dell'housekeeper).

Chiusura manuale — tag del trigger ereditati dagli eventi di ripristino Gli eventi di ripristino creati dopo una chiusura manuale ereditano i tag del trigger oltre ai tag dell'item e dell'host. Questi tag sono presenti nell'array dei tag dell'evento e sono disponibili per le macro di notifica come `{EVENT.RECOVERY.TAGS}` e `{EVENT.RECOVERY.TAGSJSON}`.

Cache delle query DNS per i daemon Zabbix Zabbix server, Zabbix proxy e Zabbix agent hanno ricevuto il supporto per l'utilizzo del resolver c-ares per tutte le richieste DNS, fornendo la cache delle query DNS e un failover del resolver migliorato quando Zabbix è compilato con `--with-ares`. La cache delle query DNS richiede c-ares 1.26.0 o versione successiva.

Supporto per la compilazione con c-ares su Windows Zabbix agent ora può essere compilato con il resolver c-ares su Microsoft Windows. c-ares può essere installato tramite vcpkg e la compilazione dell'agent supporta sia `ARES=<prefisso vcpkg>` sia percorsi separati `ARESINCDIR`/`ARESLIBDIR` per le directory di inclusione e delle librerie.

Autenticazione

Certificati SAML importabili per il single sign-on Gli utenti Super admin possono ora importare certificati e chiavi private direttamente nel frontend per la configurazione di SAML. Sono stati aggiunti tre nuovi campi in *Administration* > *Authentication* > *SAML*:

- *Certificato IdP* - certificato X.509 presentato dal provider di identità
- *Certificato SP* - certificato del service provider utilizzato per gli scambi SAML
- *Chiave privata SP* - chiave privata corrispondente al certificato SP

Questi controlli consentono di aggiungere nuovi valori o modificare quelli esistenti dalla scheda delle impostazioni SAML. I certificati e le chiavi private vengono convalidati prima di essere salvati nel backend di archiviazione scelto; i valori non validi o malformati vengono rifiutati con un errore esplicativo.

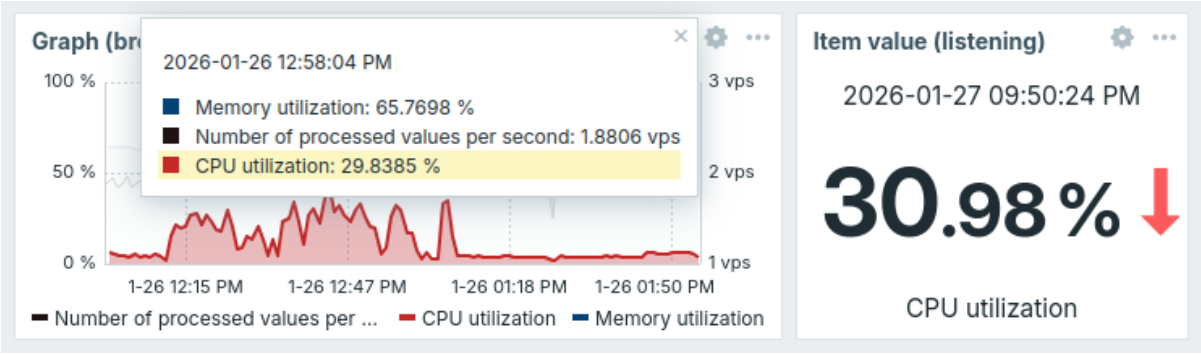
Widget

Raggruppamento dei marker host nella Geomap Il widget Geomap ora supporta un parametro *Clustering* per controllare come i marker host vicini vengono combinati in un singolo marker con un conteggio. Impostando il livello di zoom della mappa per il raggruppamento, è possibile mantenere leggibili le mappe di grandi dimensioni e conservare una visibilità precisa quando si aumenta lo zoom.

Colonne aggregate in Top items Il widget Top items ora può raggruppare i pattern degli item in un'unica colonna o riga aggregata, semplificando la presentazione di metriche combinate (ad esempio: il traffico totale in ingresso su tutte le interfacce di rete di un host). Sono state aggiunte tre nuove opzioni durante la configurazione delle colonne degli item: *Aggregate columns*, *Column aggregation function* e *Combined column name*.

Miglioramenti del widget Graph Il widget Graph offre diversi miglioramenti dell'usabilità:

- Gli item nel tooltip dei dati del grafico che appare passando il cursore sopra il grafico sono ordinati per valore in ordine decrescente.
- Passando il cursore sopra un item nel tooltip, il relativo grafico viene evidenziato mentre gli altri vengono attenuati.
- Selezionando un item nel tooltip, i suoi dati vengono trasmessi ad altri widget che supportano la ricezione di tali dati.
- Il limite al numero di item visualizzati nel tooltip è stato rimosso.
- Una nuova impostazione *Host names in labels* consente di scegliere se i nomi degli host vengono visualizzati nel tooltip e nella legenda del grafico.

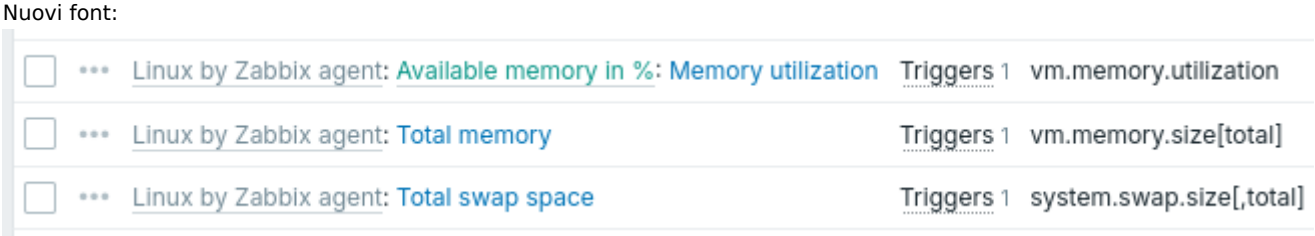


- Una nuova impostazione *Invert values* consente di capovolgere un grafico.

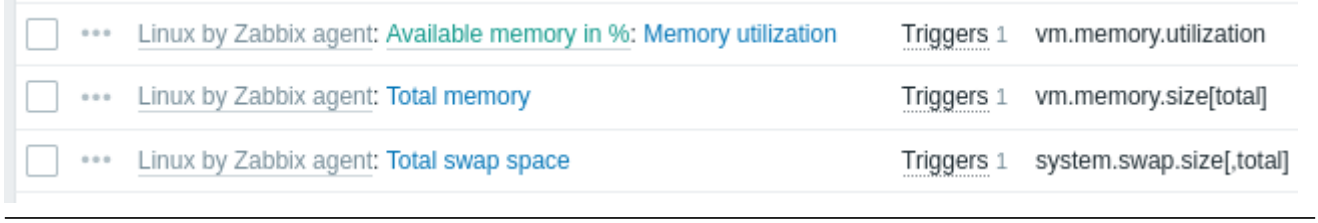
Frontend

Nuovi font e font incorporati Zabbix ora include nuovi font incorporati che migliorano la leggibilità, si caricano più velocemente e visualizzano lingue aggiuntive del frontend con un impatto minimo sul layout. Essendo incorporati, questi font garantiscono anche un aspetto coerente tra sistemi diversi.

Nuovi font:



Font legacy:



I nuovi font sono utilizzati in quasi tutti i temi. Se necessario, i temi *Blue (classic)* e *Dark (classic)*, aggiunti di recente, sono disponibili con i font legacy.

La famiglia di font per i contenuti monospaziati e i grafici rimane invariata.

Validazione inline I seguenti moduli nel frontend sono stati aggiunti all'insieme dei moduli che supportano la validazione inline:

- Token API
- Autenticazione
- Autoregistrazione
- Connettore
- Mappe geografiche
- Gruppo host
- Prototipi host
- Mappatura icone
- Immagini
- Manutenzione
- Tipo di supporto
- Notifiche
- Profilo
- Proxy
- Espressioni regolari
- Servizio
- SLA
- Gruppo template
- Aggiorna problema
- Macro utente globali
- Ruoli utente
- Utenti

Gli errori di input vengono visualizzati immediatamente dopo la compilazione dei campi, migliorando l'usabilità e riducendo gli errori di configurazione.

Moduli modali Nella configurazione del low-level discovery, il modulo di configurazione del prototipo di host viene ora aperto in una finestra modale (pop-up).

Tag ereditati visibili in host, template, item e trigger I tag ereditati ora vengono visualizzati e restituiti in modo coerente in template, host, item, scenari web e trigger. I tag ereditati dalle catene template/host sono visibili in *Monitoring > Latest data*. Il filtro per tag ereditati è disponibile ovunque sia supportato il filtro per tag, incluse le sezioni *Monitoring > Latest data* e *Data collection*, nonché tutti i widget della dashboard che consentono il filtro per tag di host, item, trigger o scenario web, così la selezione basata sui tag e il sottofiltro si comportano allo stesso modo indipendentemente da dove sia stato definito un tag.

La scheda *Tags* nei moduli di configurazione di template, host e prototipi di host ora espone un elemento radio per scegliere come vengono presentati i tag: i template mostrano *Template tags / Inherited and template tags*, mentre gli host e i prototipi di host mostrano *Host tags / Inherited and host tags*. I tag ereditati sono distinti visivamente da una nuova icona di documento contornata accanto all'etichetta del tag.

I widget Graph e Pie chart includono una nuova impostazione *Item tags*.

Posizionamento dei tooltip Ora i tooltip possono essere riposizionati trascinandoli in una nuova posizione. Questo vale, ad esempio, per il tooltip del Graph, così come per i tooltip con descrizioni nelle sezioni Latest Data o Problems.

Documentazione

Pagine di documentazione consolidate per le release minori La documentazione delle release per le versioni minori di una release principale di Zabbix sarà ora raccolta in singole pagine di documentazione rispettivamente per le nuove funzionalità e le note di aggiornamento.

Novità in Zabbix 8.0.x

Questa pagina fornisce informazioni aggregate sulle nuove funzionalità incluse nelle release minori della versione principale di Zabbix.

Vedere anche Novità della versione principale.

Per le modifiche ai template esistenti e le informazioni sui nuovi template che coprono vari dispositivi, servizi e soluzioni dei partner, vedere Modifiche ai template.

Novità in Zabbix 8.0.1 Questa versione non è ancora stata rilasciata.

3 Appliance Zabbix

Panoramica L'appliance Zabbix offre un modo per distribuire immediatamente Zabbix server e frontend, invece di configurarli manualmente o di riutilizzare un server esistente per Zabbix.

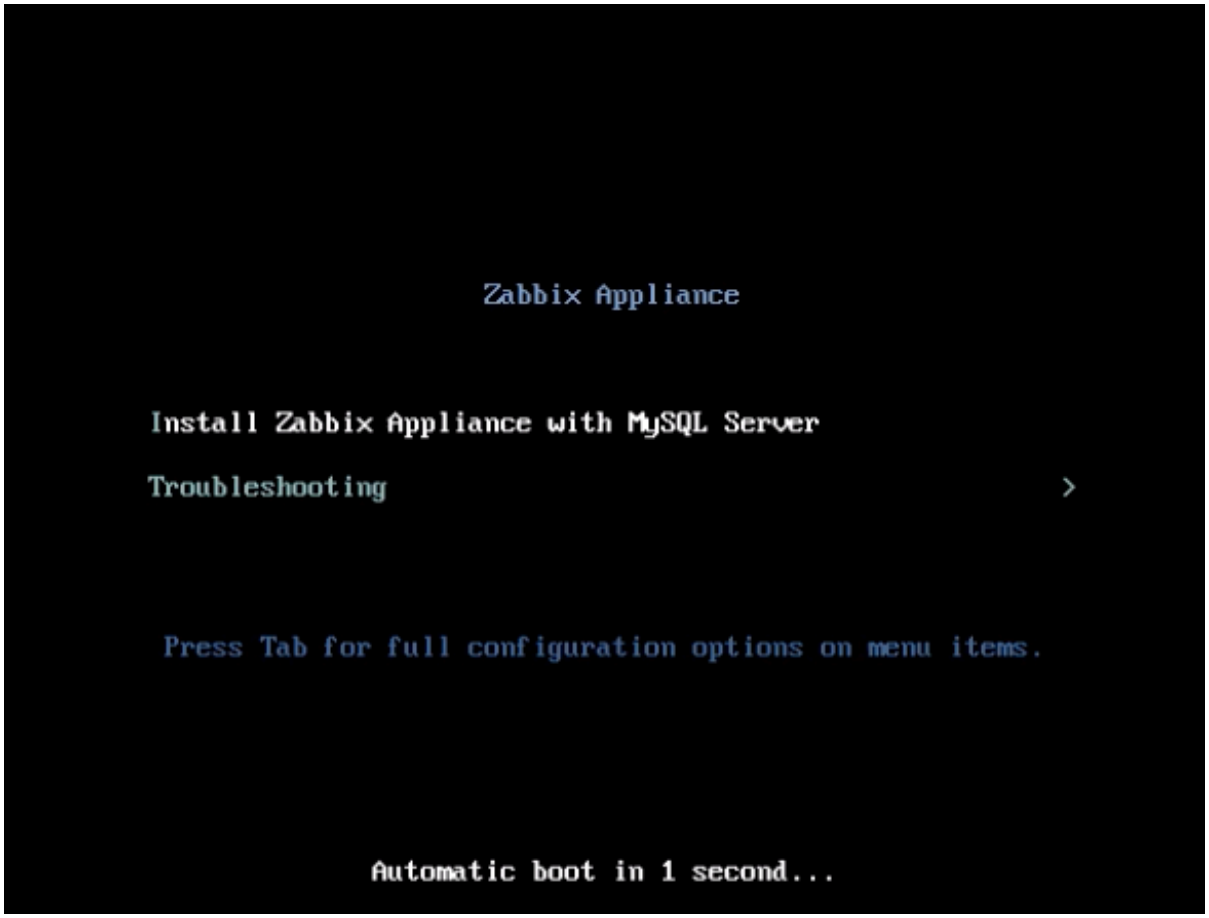
L'appliance è basata su AlmaLinux 8 (x86_64) e contiene un Zabbix server preconfigurato in esecuzione su MySQL e un frontend in esecuzione sul web server Nginx.

> **Attention:**
> Questa appliance è progettata per valutare Zabbix. Si sconsiglia di utilizzarla 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
- Open Virtualization Format (.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. Avviare la macchina virtuale dell'appliance dall'immagine scaricata.
2. Configurare le impostazioni di rete della macchina virtuale per consentire l'accesso da un browser sulla macchina host. Questo può essere ottenuto abilitando la *modalità Bridged*.
3. Accedere alla macchina virtuale utilizzando le credenziali di sistema predefinite.
4. Per recuperare l'indirizzo IP, eseguire il seguente comando sulla macchina virtuale:

```
ip addr show
```

5. Aprire un browser sulla macchina host e inserirvi l'indirizzo IP che l'appliance ha ricevuto tramite DHCP.
6. Accedere a Zabbix utilizzando le credenziali predefinite e iniziare 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

Zabbix frontend

- 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 in `/etc/zabbix/zabbix\_server.conf`

Zabbix frontend:

- Nome utente: zabbix_web
- Password: la password è memorizzata in `/etc/zabbix/web/zabbix.conf.php`

Per modificare la password di un utente del database, cambiarla 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 limitarlo, modificare `/etc/nginx/conf.d/zabbix.conf`. 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 utilizza DHCP per ottenere l'indirizzo IP. Per impostare un indirizzo IP statico:

- Accedere come utente root.
- Eseguire i seguenti comandi, sostituendo i valori con i propri indirizzi IP personalizzati:

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

In alternativa, è possibile aprire il file `/etc/NetworkManager/system-connections/eth0.nmconnection` e aggiungere le seguenti righe:

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

Dopo aver salvato il file modificato, eseguire il comando `systemctl restart network`.

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 si trovano in `/etc/zabbix`
- I file di log di Zabbix server, proxy e agent si trovano in `/var/log/zabbix`
- Zabbix frontend 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* sono utilizzabili direttamente nei prodotti VMware Player, Server e Workstation. Per l'uso in ESX, ESXi e vSphere devono essere convertite utilizzando VMware vCenter Converter (autenticazione richiesta per il download). Se si utilizza VMware vCenter Converter, si potrebbero riscontrare 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, una volta completata la 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)*, ciò potrebbe indicare che l'installazione è ancora in corso.



Se l'errore persiste dopo alcuni minuti di attesa, oppure se si osserva qualsiasi altro comportamento imprevisto, è probabile che il processo di installazione non sia stato completato correttamente. In questo caso, si consiglia di eliminare l'appliance corrente e distribuirla nuovamente seguendo le stesse istruzioni di installazione. Questo passaggio in genere risolve il problema.

Si noti che non è consigliabile tentare di correggere manualmente un'installazione danneggiata, poiché ciò potrebbe causare ulteriori complicazioni.

4 Processi di Zabbix

Utilizzare la barra laterale per accedere ai contenuti nella sezione dei processi di Zabbix.

1 Server

Panoramica

Zabbix server è il processo centrale del software Zabbix.

Il server esegue il polling e la ricezione dei dati, calcola i trigger, invia notifiche agli utenti. È il componente centrale a cui gli agent e i proxy Zabbix riportano i dati sulla disponibilità e sull'integrità dei sistemi. Il server può anche controllare da remoto i servizi di rete (come web server e mail server) utilizzando semplici controlli dei servizi.

Il server è il repository centrale in cui vengono memorizzati tutti i dati di configurazione, statistici e operativi, ed è l'entità di Zabbix che avviserà 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, web frontend e archiviazione del database.

Tutte le informazioni di configurazione di Zabbix sono memorizzate nel database, con cui interagiscono sia il server sia il web frontend. Ad esempio, quando si crea un nuovo item utilizzando il web frontend (o API), esso viene aggiunto alla tabella items nel database. Successivamente, circa una volta al minuto, Zabbix server interroga la tabella items per ottenere un elenco degli item attivi, che viene poi memorizzato in una cache all'interno di Zabbix server. Per questo motivo, possono essere necessari fino a due minuti prima che eventuali modifiche apportate nel frontend di Zabbix vengano visualizzate nella sezione dati più recenti.

Eseguire il 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
```

Analogamente, per arrestare/riavviare/visualizzare lo stato, utilizzare i seguenti comandi:

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

Avvio manuale

Se quanto sopra non funziona, è necessario avviarlo manualmente. Individuare il percorso del binario zabbix_server ed eseguire:

```
zabbix_server
```

Con Zabbix server è possibile utilizzare i seguenti parametri della riga di comando:

```
-c --config <file>              percorso del file di configurazione (il valore predefinito è /usr/local/et
-f --foreground                 esegui Zabbix server in primo piano
-R --runtime-control <option>   esegui funzioni amministrative
-T --test-config                convalida il file di configurazione ed esci
-h --help                       mostra questo aiuto
-V --version                    visualizza il numero di versione
```

Esempi di esecuzione di Zabbix server con parametri della riga di comando:

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

Controllo di runtime

Opzioni di controllo di runtime:

Option	Description	Target
config_cache_reload	Ricarica la cache di configurazione. Ignorato se la cache è attualmente in fase di caricamento.	
history_cache_clear=<**target**>	Cancella la cache della cronologia per l'item specificato dal suo 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 server.	**historycache** - statistiche della cache della cronologia **valuecache** - statistiche della cache dei valori **preprocessing** - statistiche del manager di preprocessing **alerting** - statistiche dell'alert manager **lld** - statistiche del manager LLD **locks** - elenco dei mutex (vuoto sui sistemi *BSD*) **connector** - statistiche per i connettori 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 suo nome o ID. Si noti che i nodi attivi/in standby non possono essere rimossi.	**target** - nome o ID del nodo (può essere ottenuto eseguendo ha_status)
ha_set_failover_delay=<**delay**>	Imposta il ritardo di failover dell'alta disponibilità (HA). I suffissi temporali sono supportati, ad esempio 10s, 1m.	
proxy_config_cache_reload[=<**target**>]	Ricarica la configurazione del proxy.	**target** - elenco di nomi di proxy separati da virgole Se target non è specificato, ricarica la configurazione per tutti i proxy
secrets_reload	Ricarica i segreti da Vault.	
service_cache_reload	Ricarica la cache del service manager.	
snmp_cache_reload	Ricarica la cache SNMP — cancella le proprietà del motore SNMP (engine time, engine boots, engine id, credenziali) per tutti gli host. Usare per forzare una cancellazione 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.	
trigger_housekeeper_execute	Avvia la procedura di housekeeping dei trigger. Ignorato 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 target non è specificato. Non supportato 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 di processo e numero (ad esempio, poller,3) **pid** - Identificatore di processo (da 1 a 65535). Per valori maggiori, specificare target come 'process type,N'.
log_level_decrease[=<**target**>]	Diminuisce il livello di log; influisce su tutti i processi se target non è specificato. Non supportato sui sistemi *BSD*.	

Option	Description	Target
prof_enable[=<**target**>]	Abilita la profilazione. Influisce su tutti i processi se target non è specificato. La profilazione abilitata fornisce dettagli di tutti i rwlock/mutex per nome funzione.	**process type** - Tutti i processi del tipo specificato (ad esempio history syncer) Tipi di processo supportati come target di 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, preprocessing worker, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector **process type,N** - Tipo di processo e numero (ad esempio, history syncer,1) **pid** - Identificatore di processo (da 1 a 65535). Per valori maggiori, specificare target come 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` possono essere usati con il tipo e il numero di processo (ad esempio, history syncer,1,processing) oppure con tutti i processi del tipo (ad esempio, history syncer,rwlock)
prof_disable[=<**target**>]	Disabilita la profilazione. Influisce su tutti i processi se target non è specificato.	**process type** - Tutti i processi del tipo specificato (ad esempio history syncer) Tipi di processo supportati come target di profilazione: vedere `prof_enable` **process type,N** - Tipo di processo e numero (ad esempio, history syncer,1) **pid** - Identificatore di processo (da 1 a 65535). Per valori maggiori, specificare target come 'process type,N'.

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 per 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:

```
### Raccoglie tutte le informazioni diagnostiche disponibili nel file di log del server:
zabbix_server -R diaginfo

### Raccoglie 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 UI di Zabbix, Zabbix ricarica automaticamente le nuove credenziali SNMPv3 per tale interfaccia nella maggior parte dei casi; utilizzare `-R snmp_cache_reload` solo se il polling continua a non riuscire dopo le modifiche alle credenziali (ad esempio, a causa di incongruenze di engineBoots/engineID o di dispositivi non RFC), oppure quando è necessario forzare una cancellazione globale della cache SNMP per la risoluzione dei problemi.

Esempio di utilizzo del controllo runtime per attivare 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
```

Utente del processo

Zabbix server è progettato per essere eseguito come utente non root. Verrà eseguito come qualsiasi utente non root con cui viene avviato. Quindi è possibile eseguire il server come qualsiasi utente non root senza alcun problema.

Se si tenta di eseguirlo come 'root', passerà a un utente 'zabbix' codificato in modo statico, che deve essere presente nel sistema. È possibile eseguire il server come 'root' solo se si modifica di conseguenza il parametro 'AllowRoot' nel file di configurazione del server.

Se Zabbix server e l'agent vengono eseguiti sulla stessa macchina, si consiglia di utilizzare un utente diverso per eseguire il server rispetto a quello usato per eseguire l'agent. Altrimenti, se entrambi vengono eseguiti con lo stesso utente, l'agent può accedere al file di configurazione del server e qualsiasi utente di livello Admin in Zabbix può recuperare abbastanza facilmente, ad esempio, la password del database.

File di configurazione

Per i dettagli 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 controlli passivi con un thread worker
- `alert manager` - gestore della coda degli avvisi
- `alert syncer` - processo di scrittura degli avvisi nel DB
- `alerter` - processo per l'invio delle notifiche
- `availability manager` - processo per gli aggiornamenti della disponibilità degli host
- `browser poller` - poller per i controlli degli item del 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 connettori
- `connector worker` - processo per la gestione delle richieste dal connector manager
- `discovery manager` - processo manager per il discovery dei dispositivi
- `discovery worker` - processo per la gestione delle attività di discovery dal 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` - processo di scrittura dello storico nel DB
- `housekeeper` - processo per la rimozione dei dati obsoleti (storico e trend degli item, sessioni utente, eventi, ecc.), nonché dei dati lasciati da oggetti eliminati
- `http agent poller` - processo poller asincrono per 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 proxy passivi
- `proxy group manager` - manager del bilanciamento del carico dei proxy e dell'alta disponibilità
- `report manager`- manager delle attività di generazione pianificata dei report
- `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 ricevendo informazioni su problemi, tag dei problemi e ripristino dei problemi da history syncer, task manager e alert manager
- `snmp poller` - processo poller asincrono per controlli SNMP con un thread worker (solo item `walk[OID]` e `get[OID]`)
- `snmp trapper` - trapper per trap SNMP
- `task manager` - processo per l'esecuzione remota di attività richieste da altri componenti (ad esempio, chiusura problema, riconoscimento problema, controllo immediato del valore di un item, funzionalità di comando remoto)
- `timer` - timer per l'elaborazione delle maintenance
- `trapper` - trapper per controlli attivi, trap, comunicazione proxy
- `trigger housekeeper` - processo per la rimozione di problemi ed eventi generati da trigger che nel frattempo sono stati eliminati
- `unreachable poller` - poller per dispositivi non raggiungibili
- `vmware collector` - collettore di 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 permessi di lettura-scrittura solo per il proprietario del file. Inoltre, il file è leggibile dal gruppo proprietario. Tutti gli altri permessi sono negati.

Vari tipi di processi del server Zabbix possono essere monitorati utilizzando l'**item** interno **zabbix[process,<type>,<mode>,<state>]** item.

Statistiche delle transazioni del history syncer

Il titolo del processo history syncer visualizza 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 della 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 host, ecc.);
- Tempo impiegato per scaricare i trend 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 e trend degli item, sessioni utente, eventi, ecc.), nonché i dati lasciati da oggetti eliminati. Viene eseguito in cicli, con frequenza e limite di eliminazione per ciclo determinati da

HousekeepingFrequency e MaxHousekeeperDelete. Tutti i dati non rimossi in un ciclo vengono riportati al ciclo successivo. L'housekeeping automatico può essere abilitato e configurato in *Administration* > *Housekeeping*.

Per rimuovere i dati lasciati da oggetti eliminati, il processo housekeeper si basa sui task della tabella housekeeper, che viene popolata ogni volta che un oggetto viene eliminato. Ad esempio, quando elimini un host, Zabbix elimina anche i suoi item, ma non il loro storico, i trend o i problemi. Invece, i trigger del database popolano la tabella housekeeper con task costituiti da questi campi:

- housekeeperid - ID del task
- object - tipo di oggetto (0 - item; 1 - trigger; 2 - servizio; 3 - host rilevato; 4 - servizio rilevato)
- objectid - ID dell'oggetto (aiuta l'housekeeper a trovare i dati correlati 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 senza verificare la presenza di dati correlati all'oggetto; tale verifica viene eseguita dal processo housekeeper.

Ogni task produce una o più operazioni di housekeeper che dipendono dal tipo di oggetto:

- Per gli item (incluse le regole LLD) - rimuove i dati da tutte le tabelle di storico e trend (history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint) che contengono valori per quegli item. Inoltre, controlla la tabella problems e rimuove gli eventi interni obsoleti associati a quegli item.
- Per i trigger - controlla le tabelle correlate agli eventi (problem, event_symptom, event_recovery, events) e rimuove gli eventi obsoleti associati a quei trigger, notificando inoltre il processo service manager degli eventi rimossi.

Note:
Un processo trigger housekeeper separato gestisce un'attività più specifica: rimuovere 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 causati da trigger che nel frattempo sono stati eliminati potrebbero comunque generare problemi di servizio e assegnarli ai servizi. Se la tua configurazione include molte regole di calcolo dello stato dei servizi basate su trigger rilevati/non più rilevati frequentemente, valuta di aumentare la frequenza della procedura di housekeeping modificando il parametro di configurazione del server ProblemHousekeepingFrequency.

- Per i servizi - controlla la tabella problems e rimuove gli eventi di servizio obsoleti, nonché i problemi di servizio obsoleti, risolvendoli quindi al momento dell'housekeeping.
- Per il rilevamento di rete - rimuove gli eventi di discovery obsoleti dalla tabella problems.

L'housekeeper rimuove solo gli eventi che non sono associati a problemi. Ad esempio, un evento obsoleto di problema/ripristino non verrà rimosso se è associato a un problema aperto. Quando l'housekeeper rimuove entità obsolete, rimuove prima i problemi e poi gli eventi.

Note:
Le tabelle che utilizzano la modalità partition (tabelle partizionate TimescaleDB) vengono saltate; vengono elaborate solo le tabelle che utilizzano 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 costantemente le prestazioni, la tolleranza ai guasti e la resilienza necessarie. Zabbix opera sulle versioni leader di mercato.

Zabbix server 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 di tipo Unix.

Impostazioni locali

Si noti che il server richiede una locale UTF-8 affinché alcuni item testuali possano essere interpretati correttamente. La maggior parte dei moderni sistemi di tipo Unix utilizza una locale UTF-8 come predefinita; tuttavia, esistono alcuni sistemi nei quali potrebbe essere necessario impostarla esplicitamente.

1 Alta disponibilità

Panoramica

L'alta disponibilità (HA) è in genere richiesta nelle infrastrutture critiche che possono permettersi virtualmente nessun tempo di inattività. Pertanto, per qualsiasi servizio che possa guastarsi, deve essere prevista un'opzione di failover che subentri nel caso in cui il servizio corrente fallisca.

Zabbix offre una soluzione di alta disponibilità **nativa** facile da configurare e che non richiede alcuna precedente esperienza con l'HA. L'HA nativa di Zabbix può essere utile come ulteriore livello di protezione contro guasti software/hardware del server Zabbix o per ridurre i tempi di inattività dovuti alla manutenzione.

Nella modalità di alta disponibilità di Zabbix, più server Zabbix vengono eseguiti come nodi in un cluster. Mentre un server Zabbix 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 al funzionamento standalone in qualsiasi momento.

Vedi anche: Dettagli di implementazione

Abilitazione dell'alta disponibilità

Avvio di Zabbix server come nodo del cluster

Per avviare un server Zabbix come nodo del cluster sono richiesti due parametri nella configurazione del server:

- **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 sarà indicato nelle configurazioni di agent e proxy. Se non si specifica HANodeName, il server verrà avviato in modalità standalone.

- **NodeAddress** deve essere specificato per ogni nodo.

Il parametro NodeAddress (indirizzo:porta) verrà utilizzato dal frontend di Zabbix per connettersi al nodo server attivo. NodeAddress deve corrispondere all'IP o al nome FQDN del rispettivo server Zabbix.

Riavviare tutti i server Zabbix dopo aver apportato modifiche ai file di configurazione. A questo punto verranno 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 di Zabbix server (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 l'indirizzo:porta del server Zabbix **non sia definito** nella configurazione del frontend (presente in `conf/zabbix.conf.php` nella directory dei file del frontend).

```
// 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 nodes 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 parameter Server del proxy, separati da una **virgola**.

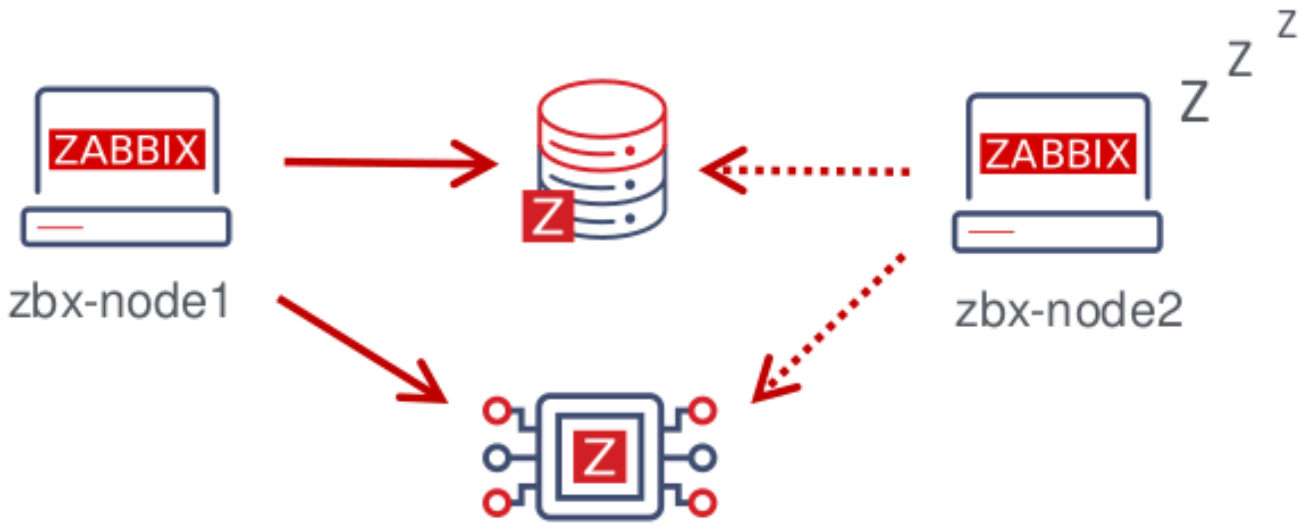
`Server=zabbix-node-01,zabbix-node-02`

Per un proxy attivo, i nomi dei nodi devono essere elencati nel parameter Server 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, 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. Si noti che per i controlli attivi i nodi devono essere separati da una virgola da 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 a un altro nodo se il nodo attivo si arresta. Deve esserci almeno un nodo nello stato standby affinché il failover possa avvenire.

Quanto sarà rapido il failover? Tutti i nodi aggiornano l'ora dell'ultimo accesso (e lo stato, se viene modificato) 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 poter aggiornare il proprio stato, i nodi standby attenderanno il **ritardo di failover** + 5 secondi prima di subentrare

Il ritardo di failover è configurabile, con un intervallo supportato compreso tra 10 secondi e 15 minuti (un minuto per impostazione predefinita). Per modificare il ritardo di failover, è possibile 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 runtime control:

- `ha_status` - registra lo stato del cluster HA nel log del server Zabbix (e su stdout)
- `ha_remove_node=target` - rimuove un nodo HA identificato dal suo <target> - nome o ID del nodo (nome/ID possono essere ottenuti dall'output dell'esecuzione di ha_status), ad esempio:

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

Si noti che i nodi attivi/in standby non possono essere rimossi.

- `ha_set_failover_delay=delay` - imposta il ritardo di failover HA (tra 10 secondi e 15 minuti; i suffissi temporali sono supportati, ad esempio 10s, 1m)

Lo stato del nodo può essere monitorato:

- in *Report* → *Informazioni di sistema*
- nel widget della dashboard *Informazioni di sistema*
- utilizzando l'opzione di runtime control `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:

- creare copie di backup dei file di configurazione
- arrestare i nodi standby
- rimuovere il parametro HANodeName dal server primario attivo
- riavviare il server primario (si avvierà in modalità standalone)

Aggiornamento del cluster HA

Per eseguire un aggiornamento di versione principale dei nodi HA:

- arrestare tutti i nodi;
- creare un backup completo del database;
- se il database utilizza la replica, assicurarsi che tutti i nodi siano sincronizzati e non presentino problemi. Non eseguire l'aggiornamento se la replica è interrotta.
- selezionare un singolo nodo che eseguirà l'aggiornamento del database, modificarne la configurazione in modalità standalone commentando HANodeName ed eseguire l'aggiornamento;
- assicurarsi che l'aggiornamento del database sia stato completato interamente (*Informazioni di sistema* dovrebbe indicare che Zabbix server è in esecuzione);
- riavviare il nodo in modalità HA;
- aggiornare e avviare il resto dei nodi (non è necessario modificarli in modalità standalone, poiché a questo punto il database è già stato aggiornato).

Per un aggiornamento di versione secondaria è sufficiente aggiornare il primo nodo, assicurarsi che sia stato aggiornato e sia in esecuzione, quindi avviare l'aggiornamento del 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 utilizzare la soluzione HA nativa di Zabbix oppure una soluzione HA di terze parti, a seconda di quale si adatti meglio ai requisiti di alta disponibilità del loro ambiente.

La soluzione è composta da più istanze o nodi di zabbix_server. Ogni nodo:

- è configurato separatamente

- utilizza lo stesso database
- può avere diverse modalità: active, standby, unavailable, stopped

Solo un nodo può essere active (in funzione) alla volta. Un nodo standby esegue un solo processo: l'HA manager. Un nodo standby non esegue raccolta dati, elaborazione o altre normali attività del server; non resta in ascolto sulle porte; ha un numero minimo di connessioni al database.

Sia i nodi active che quelli standby aggiornano il proprio ultimo orario di accesso ogni 5 secondi. Ogni nodo standby monitora l'ultimo orario di accesso del nodo active. Se l'ultimo orario di accesso del nodo active supera i secondi di 'failover delay', il nodo standby passa automaticamente a 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 orario 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 versioni secondarie di Zabbix.

2 Agent

Panoramica

Zabbix agent viene distribuito su un target di monitoraggio per monitorare attivamente risorse e applicazioni locali (dischi rigidi, memoria, statistiche del processore, ecc.).

L'agent raccoglie localmente le informazioni operative e riporta i dati a Zabbix server per un'ulteriore elaborazione. In caso di guasti (ad esempio un disco rigido che si riempie completamente o un processo di servizio arrestato in modo anomalo), Zabbix server può avvisare attivamente gli amministratori della macchina specifica che ha segnalato il guasto.

Gli agent Zabbix sono altamente efficienti grazie all'uso di chiamate di sistema native per raccogliere informazioni statistiche.

Controlli passivi e attivi

Gli agent Zabbix possono eseguire controlli passivi e attivi:

- Controlli passivi - l'agent Zabbix risponde a una richiesta del server Zabbix (o del proxy). Ad esempio, il server richiede dati (ad es. carico della CPU) e l'agent restituisce il risultato.
- Controlli attivi - l'agent Zabbix raccoglie e invia dati senza attendere una richiesta dal server Zabbix (o dal proxy). Per prima cosa, recupera dal server un elenco di item di monitoraggio (carico della CPU, memoria disponibile, ecc.), quindi raccoglie i dati richiesti e invia periodicamente i nuovi valori al server.

Il tipo di controllo dell'agent viene configurato selezionando il relativo tipo di item. L'agent Zabbix elabora item di tipo "Zabbix agent" o "Zabbix agent (active)".

Piattaforme supportate

Per le piattaforme supportate, fare riferimento alla pagina Requisiti.

Agente su sistemi UNIX-like

Zabbix agent sui sistemi UNIX-like viene eseguito sull'host monitorato.

Installazione

Zabbix agent può essere installato su 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 di 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:
Sono disponibili per il download binari precompilati di Zabbix agent per macOS, IBM AIX, FreeBSD, OpenBSD e Solaris. Per NetBSD e HP-UX sono disponibili binari legacy, compatibili con la versione corrente di Zabbix server/proxy.

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
```

Avvio manuale

È possibile avviare Zabbix agent individuando il binario `zabbix_agentd` ed eseguendolo direttamente; ad esempio:

```
zabbix_agentd
```

Agent su sistemi Windows

Zabbix agent su Windows viene eseguito come servizio di Windows.

Installazione

Zabbix agent può essere installato su Windows utilizzando uno dei seguenti metodi:

- Binari precompilati di Zabbix agent - scaricare il pacchetto di installazione MSI di Zabbix agent e seguire le istruzioni nella pagina Installazione di agent Windows da MSI.
- Sorgenti di Zabbix - scaricare i file sorgente e seguire le istruzioni nella pagina Compilazione di Zabbix agent su Windows.

Per ulteriori dettagli sull'installazione di Zabbix agent (da un archivio ZIP) come servizio Windows, vedere la pagina Zabbix agent su Microsoft Windows.

Opzioni

È possibile eseguire più istanze dell'agent su un host. Una singola istanza può usare il file di configurazione predefinito oppure un file di configurazione specificato nella riga di comando. In caso di istanze multiple, ogni istanza dell'agent deve avere il proprio file di configurazione (una delle istanze può usare il file di configurazione predefinito).

I seguenti parametri della riga di comando possono essere usati con Zabbix agent:

Parameter	Description
agent UNIX e Windows	
-c --config <config-file>	Percorso del file di configurazione. Puoi usare questa opzione per specificare un file di configurazione diverso da quello predefinito. Su UNIX, il valore predefinito è /usr/local/etc/zabbix_agentd.conf oppure quello impostato dalle variabili di compile-time ***--sysconfdir*** o ***--prefix*** Su Windows, il valore predefinito è C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Esegui Zabbix agent in foreground (predefinito: true).
-p --print	Stampa gli item conosciuti ed esce. *Nota*: per restituire anche i risultati dei parametri utente, devi specificare il file di configurazione (se non si trova nel percorso predefinito).
-t --test <item key>	Testa l'item specificato ed esce. *Nota*: per restituire anche i risultati dei parametri utente, devi specificare il file di configurazione (se non si trova nel percorso predefinito).
-T --test-config	Convalida il file di configurazione ed esce.
-h --help	Visualizza le informazioni di aiuto.
-V --version	Visualizza il numero di versione.
Solo agent UNIX	
-R --runtime-control <option>	Esegue funzioni amministrative. Vedi runtime control.
Solo agent Windows	
-m --multiple-agents	Usa più istanze di 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.

Parameter	Description
-S --startup-type <value>	Imposta il tipo di avvio del servizio Zabbix Windows agent. Valori consentiti: `automatic` - *(predefinito)* avvia il servizio automaticamente all'avvio di Windows; `delayed` - ritarda l'avvio del servizio finché i servizi avviati automaticamente non hanno completato l'avvio (disponibile su Windows Server 2008/Vista e versioni successive); `manual` - avvia il servizio manualmente (da un utente o da un'applicazione); `disabled` - disabilita il servizio, in modo che non possa essere avviato da un utente o da un'applicazione. Puoi usare 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 di tutti gli item integrati dell'agent con i valori
- test di un parametro utente con chiave "mysql.ping" definita nel file di configurazione specificato
- installazione di un servizio "Zabbix Agent" per Windows usando il percorso predefinito del file di configurazione C:\Program Files\Zabbix Agent\zabbix_agentd.conf
- installazione di un servizio "Zabbix Agent [Hostname]" per Windows usando il file di configurazione zabbix_agentd.conf situato nella stessa cartella dell'eseguibile dell'agent e rendendo univoco il nome del servizio estendendolo con il valore Hostname del file di configurazione
- modifica del tipo di avvio di un servizio "Zabbix Agent" per Windows già installato usando il file di configurazione zabbix_agentd.conf 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 di runtime

Con le opzioni di controllo di runtime 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) Vedi tutti i tipi di processo dell'agent. **tipo di processo,N** - tipo di processo e numero (ad esempio, listener,3) **pid** - identificatore di processo (da 1 a 65535). Per valori più grandi specificare la destinazione come 'process-type,N'.
log_level_decrease[=<target>]	Diminuisce 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
- diminuzione del livello di log di tutti i processi di controlli attivi

```
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 di runtime non è supportato su OpenBSD, NetBSD e Windows.

Tipi di processi dell'agent

- `active checks` - processo per l'esecuzione dei controlli attivi
- `collector` - processo per la raccolta dei dati
- `listener` - processo in ascolto dei controlli passivi

Il file di log dell'agent può essere utilizzato per osservare questi tipi di processi.

Il file di log dell'agent viene creato con permessi di lettura e scrittura solo per il proprietario del file. Inoltre, il file è leggibile dal gruppo proprietario. Tutti gli altri permessi sono negati.

Utente del processo

Zabbix agent su UNIX è progettato per essere eseguito come utente non root. Verrà eseguito come qualsiasi utente non root con cui viene avviato. Quindi è possibile eseguire agent come qualsiasi utente non root senza alcun problema.

Se si tenta di eseguirlo come 'root', passerà a un utente 'zabbix' codificato in modo statico, che deve essere presente nel sistema. È possibile eseguire agent come 'root' solo se si modifica di conseguenza il parametro 'AllowRoot' nel file di configurazione dell'agent.

File di configurazione

Per i dettagli sulla configurazione di Zabbix agent, vedere le opzioni del file di configurazione per zabbix_agentd o agent per 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 di tipo Unix ha una locale UTF-8 come predefinita; tuttavia, esistono alcuni sistemi in cui potrebbe essere necessario impostarla esplicitamente.

Codice di uscita

Zabbix agent restituisce 0 in caso di terminazione riuscita e 1 in caso di terminazione fallita.

3 Agent 2

Panoramica

Zabbix Agent 2 è una nuova generazione di Zabbix Agent, scritto 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 estensibile con plugin, che forniscono controlli semplici con codice minimo e supportano controlli complessi costituiti da script di lunga durata e raccolta dati autonoma con report periodici.
- Funzionare come sostituto di Zabbix Agent, supportando tutte le funzionalità precedenti.

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/di pianificazione e la concorrenza dei controlli all'interno di un server attivo.

> **Note:**
> Per impostazione predefinita, dopo un riavvio, Zabbix agent 2 pianificherà la prima raccolta dei dati per i controlli attivi in un momento casuale convenzionale entro l'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 di *Pianificazione*, impostare il parametro `ForceActiveChecksOnStart` (a livello globale) oppure `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (influisce solo sui controlli di plugin specifici) nel file di configurazione. Il parametro a livello di plugin, se impostato, sovrascriverà il parametro globale.

Concorrenza dei controlli

I controlli di plugin diversi possono essere eseguiti contemporaneamente. Il numero di controlli concorrenti all'interno di un plugin è limitato dall'impostazione della capacità del plugin. Ogni plugin può avere un'impostazione di capacità codificata rigidamente (1000 è il valore predefinito) che può essere ridotta utilizzando 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 sui 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 selezionato la versione di Zabbix, la distribuzione del sistema operativo e la versione del sistema operativo) 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 supporta la daemonizzazione su Linux.

L'agente può essere avviato eseguendo:

systemctl start zabbix-agent2

Per arrestare, riavviare o controllare lo stato di Zabbix Agent 2, utilizzare i seguenti comandi:

systemctl stop zabbix-agent2 systemctl restart zabbix-agent2 systemctl status zabbix-agent2

Avvio manuale

È possibile avviare l'agente Zabbix individuando il binario zabbix_agent2 ed eseguendolo direttamente; ad esempio:

`zabbix_agent2`

Agent 2 su sistemi Windows

Zabbix Agent 2 viene eseguito come processo autonomo; tuttavia, può anche essere eseguito come servizio Windows.

Installazione

Zabbix agent 2 può essere installato su Windows utilizzando uno dei seguenti metodi:

- Binari 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.

Per ulteriori dettagli sull'installazione di Zabbix agent 2 (da un archivio ZIP) come servizio Windows, vedere la pagina Zabbix agent su Microsoft Windows.

Opzioni

I seguenti parametri della riga di comando possono essere utilizzati con Zabbix agent 2:

Parametro	Descrizione
agent UNIX e Windows	

Parametro	Descrizione
-c --config <config-file>	Percorso del file di configurazione. È possibile usare questa opzione per specificare un file di configurazione diverso da quello predefinito. Su UNIX, il valore predefinito è /usr/local/etc/zabbix_agent2.conf oppure quello impostato dalle variabili di compilazione *--sysconfdir* o *--prefix* Su Windows, il valore predefinito è C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Esegui Zabbix agent in primo piano (predefinito: true).
-p --print	Stampa gli item conosciuti ed esce. *Nota*: per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nel percorso predefinito).
-t --test <item key>	Verifica l'item specificato ed esce. *Nota*: per restituire anche i risultati dei parametri utente, è necessario specificare il file di configurazione (se non si trova nel percorso predefinito).
-T --test-config	Convalida il file di configurazione ed esce.
-h --help	Stampa le informazioni di aiuto ed esce.
-v --verbose	Stampa informazioni di debug. Usare questa opzione con le opzioni -p e -t.
-V --version	Stampa la versione dell'agent e le informazioni sulla licenza.
-R --runtime-control <option>	Esegue funzioni amministrative. Vedere runtime control.
Solo agent Windows	
-m --multiple-agents	Usa più istanze di agent (con le opzioni -i, -d, -s, -x). Per distinguere i nomi dei servizi delle istanze, ogni nome di servizio includerà il valore Hostname del file di configurazione specificato.
-S --startup-type <value>	Imposta il tipo di avvio del servizio Zabbix Windows agent. Valori consentiti: `automatic` - *(predefinito)* avvia il servizio automaticamente all'avvio di Windows; `delayed` - ritarda l'avvio del servizio dopo che i servizi avviati automaticamente hanno completato l'avvio; `manual` - avvia il servizio manualmente (da un utente o da un'applicazione); `disabled` - disabilita il servizio, in modo che non possa essere avviato da un utente o da un'applicazione. È possibile usare 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:

- stampare tutti gli item integrati dell'agent con i valori
- verificare un parametro utente con chiave "mysql.ping" definita nel file di configurazione specificato
- installare un servizio "Zabbix Agent" per Windows usando il percorso predefinito del file di configurazione C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
- modificare il tipo di avvio di un servizio "Zabbix Agent" per Windows già installato usando il file di configurazione zabbix_agent2.conf situato nella stessa cartella dell'eseguibile dell'agent

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

Controllo runtime

Il controllo runtime fornisce alcune opzioni per il controllo remoto.

Opzione	Descrizione
log_level_increase	Aumenta il livello di log.
log_level_decrease	Diminuisce il livello di log.
metrics	Elenca le metriche disponibili.
version	Visualizza la versione dell'agent.
userparameter_reload	Ricarica i valori delle opzioni *UserParameter* e *Include* dal file di configurazione corrente.

Opzione	Descrizione
help	Visualizza le informazioni di aiuto sul controllo runtime.

Esempi:

- aumento del livello di log per agent 2
- stampa delle opzioni di controllo runtime

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

File di configurazione

I parametri di configurazione di agent 2 sono per lo più compatibili con Zabbix agent, con alcune eccezioni.

Nuovi parametri	Descrizione
ControlSocket	Il percorso del socket di controllo runtime. Agent 2 utilizza un socket di controllo per i comandi runtime.
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Questi parametri vengono utilizzati per configurare l'archiviazione persistente in agent 2 per gli item attivi.
ForceActiveChecksOnStart	Determina se l'agent debba eseguire i controlli attivi immediatamente 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 su cui agent 2 resterà in ascolto per le richieste di stato HTTP e per la visualizzazione di un elenco dei plugin configurati e di alcuni parametri interni
Parametri rimossi	**Descrizione**
AllowRoot, *User*	Non supportati perché la daemonizzazione non è supportata.
LoadModule, *LoadModulePath*	I moduli caricabili non sono supportati.
StartAgents	Questo parametro veniva utilizzato in Zabbix agent per aumentare la concorrenza dei controlli passivi o disabilitarli. In Agent 2, la concorrenza viene configurata a livello di plugin e può essere limitata da un'impostazione di capacità.

Per maggiori dettagli, vedere le opzioni del file di configurazione per zabbix_agent2.

Codici di uscita

L'agente Zabbix 2 può essere compilato anche con versioni precedenti di OpenSSL (1.0.1, 1.0.2).

In questo caso, Zabbix fornisce mutex per il blocco in OpenSSL. Se un blocco o uno sblocco di un mutex fallisce, viene visualizzato un messaggio di errore nel flusso di errori standard (STDERR) e l'Agent 2 termina con il codice di ritorno 2 o 3, rispettivamente.

Agent vs agent 2 comparison

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

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

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

See also:

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

Built-in plugins

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

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

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

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

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

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

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

Loadable plugins

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

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

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

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

Plugin name	Description	Supported item keys	Comments
Ceph	Ceph monitoring.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	See also Ceph plugin configuration parameters.
Ember+	Monitoring of Ember+.	ember.get	See also Ember+ plugin configuration parameters.
MongoDB	Monitoring of MongoDB servers and clusters (document-based, distributed database).	mongodb.cfg.discovery, mongodb.collection.stats, mongodb.collections.discovery, mongodb.collections.usage, mongodb.connpool.stats, mongodb.db.stats, mongodb.db.discovery, mongodb.jumbo_chunks.count, mongodb.oplog.stats, mongodb.ping, mongodb.rs.config, mongodb.rs.status, mongodb.server.status, mongodb.sh.discovery, mongodb.version	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session parameters. Currently TLS parameters cannot be passed as item key parameters. See also MongoDB plugin configuration parameters.
MSSQL	Monitoring of MSSQL database.	mssql.availability.group.get, mssql.custom.query, mssql.db.get, mssql.job.status.get, mssql.last.backup.get, mssql.local.db.get, mssql.mirroring.get, mssql.nonlocal.db.get, mssql.perfcounter.get, mssql.ping, mssql.quorum.get, mssql.quorum.member.get, mssql.replica.get, mssql.version	To configure an encrypted connection to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also MSSQL plugin configuration parameters.

Plugin name	Description	Supported item keys	Comments
NVIDIA GPU	Monitoring of NVIDIA GPU.	nvml.device.count, nvml.device.decoder.utilization, nvml.device.ecc.mode, nvml.device.encoder.stats.get, nvml.device.encoder.utilization, nvml.device.energy.consumption, nvml.device.errors.memory, nvml.device.errors.register, nvml.device.fan.speed.avg, nvml.device.get, nvml.device.graphics.frequency, nvml.device.memory.bar1.get, nvml.device.memory.fb.get, nvml.device.memory.frequency, nvml.device.pci.utilization, nvml.device.performance.state, nvml.device.power.limit, nvml.device.power.usage, nvml.device.serial, nvml.device.sm.frequency, nvml.device.temperature, nvml.device.utilization, nvml.device.video.frequency, nvml.system.driver.version, nvml.version	See also NVIDIA GPU plugin configuration parameters.
PostgreSQL	Monitoring of PostgreSQL and its forks.	pgsql.autovacuum.count, pgsql.archive, pgsql.bgwriter, pgsql.cache.hit, pgsql.connections, pgsql.custom.query, pgsql.dbstat, pgsql.dbstat.sum, pgsql.db.age, pgsql.db.bloating_tables, pgsql.db.discovery, pgsql.db.size, pgsql.locks, pgsql.oldest.xid, pgsql.ping, pgsql.queries, pgsql.replication.count, pgsql.replication.process, pgsql.replication.process.discovery, pgsql.replication.recovery_role, pgsql.replication.status, pgsql.replication_lag.b, pgsql.replication_lag.sec, pgsql.uptime, pgsql.version, pgsql.wal.stat	To configure encrypted connections to the database, specify TLS parameters in the agent configuration file as a named session or default parameters. Currently TLS parameters cannot be passed as item key parameters. See also PostgreSQL plugin configuration parameters.

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

4 Proxy

Panoramica

Zabbix proxy 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 temporaneamente in locale e quindi trasferiti al server Zabbix a cui il proxy appartiene.

La distribuzione di un proxy è facoltativa, ma può essere molto vantaggiosa per distribuire il carico di un singolo server Zabbix. Se solo i proxy raccolgono i dati, l'elaborazione sul server richiede meno CPU e I/O su disco.

Uno Zabbix proxy è la soluzione ideale per il monitoraggio centralizzato di sedi remote, filiali e reti senza amministratori locali.

Zabbix proxy richiede un database separato.

> **Attention:**
> Si noti che i database supportati con Zabbix proxy sono SQLite, MySQL e PostgreSQL.

Vedi anche: Utilizzo dei proxy in un ambiente distribuito

Eseguire il 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, è necessario avviarlo manualmente. Individuare il percorso del file binario zabbix_proxy ed eseguire:

```
zabbix_proxy
```

È possibile utilizzare i seguenti parametri della riga di comando con Zabbix proxy:

```
-c --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                convalida il file di configurazione ed esci
-h --help                       mostra questo aiuto
-V --version                    visualizza il numero di versione
```

Esempi di esecuzione di Zabbix proxy con parametri della riga di comando:

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

Controllo di runtime

Opzioni di controllo di 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 la volta successiva in cui il server si connetterà al proxy.	

Option	Description	Target
history_cache_clear=<**target**>	Cancella la cache della cronologia per l'item specificato dal suo 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 (vuoto sui sistemi *BSD*)
snmp_cache_reload	Ricarica la cache SNMP — cancella le proprietà del motore SNMP (engine time, engine boots, engine id, credenziali) per tutti gli host. Usare per forzare una cancellazione globale della cache durante la risoluzione di 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; influisce su tutti i processi se target non è specificato. Non supportato sui sistemi *BSD*.	**process type** - Tutti i processi del tipo specificato (ad esempio, poller) Vedi tutti i tipi di processo del proxy. **process type,N** - Tipo di processo e numero (ad esempio, poller,3) **pid** - Identificatore del processo (da 1 a 65535). Per valori più grandi specificare target come 'process type,N'.
log_level_decrease[=<**target**>]	Diminuisce il livello di log; influisce su tutti i processi se target non è specificato. Non supportato sui sistemi *BSD*.	
prof_enable[=<**target**>]	Abilita la profilazione. Influisce su tutti i processi se target non è specificato. La profilazione abilitata fornisce dettagli di tutti i rwlock/mutex per nome funzione.	**process type** - Tutti i processi del tipo specificato (ad esempio, history syncer) Vedi tutti i tipi di processo del proxy. **process type,N** - Tipo di processo e numero (ad esempio, history syncer,1) **pid** - Identificatore del processo (da 1 a 65535). Per valori più grandi specificare 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 del tipo (ad esempio, history syncer,rwlock)
prof_disable[=<**target**>]	Disabilita la profilazione. Influisce su tutti i processi se target non è specificato.	**process type** - Tutti i processi del tipo specificato (ad esempio, history syncer) Vedi tutti i tipi di processo del proxy. **process type,N** - Tipo di processo e numero (ad esempio, history syncer,1) **pid** - Identificatore del processo (da 1 a 65535). Per valori più grandi specificare target come 'process type,N'.

Esempio di utilizzo del controllo di 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 di runtime per cancellare 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 di runtime per raccogliere informazioni diagnostiche:

```
### Gather all available diagnostic information in the proxy log file:
zabbix_proxy -R diaginfo
```

```
### Gather history cache statistics in the proxy log file:
zabbix_proxy -R diaginfo=historycache
```

Esempio di utilizzo del controllo di runtime per ricaricare la cache SNMP:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Quando un'interfaccia SNMPv3 viene aggiornata tramite la UI di Zabbix, Zabbix nella maggior parte dei casi ricarica automaticamente le nuove credenziali SNMPv3 per quell'interfaccia; usare `-R snmp_cache_reload` solo se il polling continua a non riuscire dopo le modifiche alle credenziali (ad esempio, a causa di incoerenze di engineBoots/engineID o di dispositivi non RFC), oppure quando è necessario forzare una cancellazione globale della cache SNMP per la risoluzione dei problemi.

Esempio di utilizzo del controllo di runtime per attivare l'esecuzione dell'housekeeper:

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

Esempi di utilizzo del controllo di runtime per modificare il livello di log:

```
### Increase log level of all processes:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Increase log level of second poller process:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Increase log level of process with PID 1234:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Decrease log level of all http poller processes:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Utente del processo

Zabbix proxy è progettato per essere eseguito come utente non root. Verrà eseguito come qualsiasi utente non root con cui viene avviato. Quindi è possibile eseguire il proxy come qualsiasi utente non root senza alcun problema.

Se si tenta di eseguirlo come 'root', passerà a un utente 'zabbix' codificato in modo statico, che deve essere presente nel sistema. È possibile eseguire il proxy come 'root' solo se si modifica di conseguenza il parametro 'AllowRoot' nel file di configurazione del proxy.

File di configurazione

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 controlli passivi con un thread worker
- `availability manager` - processo per gli aggiornamenti della disponibilità degli host
- `browser poller` - poller per i controlli degli item del browser
- `configuration syncer` - processo per la gestione della cache in memoria dei dati di configurazione
- `data sender` - mittente dei dati del proxy
- `discovery manager` - processo manager per il rilevamento dei dispositivi
- `discovery worker` - processo per la gestione delle attività di discovery dal discovery manager
- `history syncer` - writer del DB della cronologia
- `housekeeper` - processo per la rimozione della cronologia obsoleta degli item
- `http agent poller` - processo poller asincrono per 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 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 preprocessing con thread worker di preprocessing
- `preprocessing worker` - thread per il preprocessing dei dati
- `self-monitoring` - processo per la raccolta delle statistiche interne del server

- `snmp poller` - processo poller asincrono per controlli SNMP con un thread worker (solo item `walk[OID]` e `get[OID]`)
- `snmp trapper` - trapper per trap SNMP
- `task manager` - processo per l'esecuzione remota di attività richieste da altri componenti (ad es. chiusura del problema, riconoscimento del problema, controllo immediato del valore dell'item, funzionalità dei comandi remoti)
- `trapper` - trapper per controlli attivi, trap, comunicazione del proxy
- `unreachable poller` - poller per dispositivi non raggiungibili
- `vmware collector` - collector 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 permessi di lettura e scrittura solo per il proprietario del file. Inoltre, il file è leggibile dal gruppo proprietario. Tutti gli altri permessi sono negati.

Vari tipi di processi del proxy Zabbix possono essere monitorati utilizzando l'item interno **zabbix[process,<type>,<mode>,<state>]** item.

Statistiche delle transazioni del history syncer

Il titolo del processo history syncer visualizza 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 degli item (stato, errori).

Procedura di housekeeping

Il proxy Zabbix dispone del processo `housekeeper` che rimuove la cronologia e i trend obsoleti degli item. Viene eseguito a cicli, con frequenza determinata da `HousekeepingFrequency` e con il limite di eliminazione per ciclo determinato da `ProxyLocalBuffer` e `ProxyOfflineBuffer`. A differenza della procedura di housekeeping del server Zabbix, 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 funziona sullo stesso elenco di piattaforme supportate di Zabbix server.

Buffer di memoria

Il buffer di memoria consente di memorizzare nuovi dati (valori degli item, network discovery, autoregistrazione degli host) nel buffer e di caricarli sullo Zabbix server senza accedere al database. Il buffer di memoria è stato introdotto per il proxy a partire da Zabbix 7.0.

Nelle installazioni precedenti a Zabbix 7.0, i dati raccolti venivano memorizzati nel database prima di essere caricati sullo Zabbix server. Per queste installazioni, questo rimane il comportamento predefinito dopo l'aggiornamento a Zabbix 7.0.

Per prestazioni ottimizzate, si consiglia di configurare l'uso del buffer di memoria sul proxy. Ciò è possibile modificando il valore di ProxyBufferMode da "disk" (valore predefinito hardcoded per le installazioni esistenti) a "hybrid" (consigliato) oppure "memory". È inoltre necessario impostare la dimensione del buffer di memoria (parametro ProxyMemoryBufferSize).

In modalità hybrid, il buffer è protetto dalla perdita di dati scaricando nel database i dati non inviati se il proxy viene arrestato, se il buffer è pieno o se 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 memory, verrà utilizzato il buffer di memoria, tuttavia non esiste alcuna protezione contro la perdita di dati. Se il proxy viene arrestato o la memoria si riempie eccessivamente, i dati non inviati verranno scartati.

La modalità hybrid (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.

Si noti che, in caso di configurazione in conflitto, il proxy stamperà un errore e non riuscirà ad avviarsi, 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 moderni sistemi di tipo Unix utilizza una locale UTF-8 come impostazione predefinita; tuttavia, esistono alcuni sistemi in cui potrebbe essere necessario configurarla esplicitamente.

Calcolo delle code durante la manutenzione

> **Attention:**
> Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; per i dettagli, vedere Calcolo delle code durante la manutenzione.

5 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 chiamato "Zabbix Java gateway". Zabbix Java gateway è un demone scritto in Java. Per ottenere il valore di uno specifico contatore JMX su un host, Zabbix server interroga Zabbix Java gateway, che utilizza la API di gestione JMX per interrogare da remoto l'applicazione di interesse. L'applicazione non richiede l'installazione di software aggiuntivo, deve solo essere avviata con l'opzione `-Dcom.sun.management.jmxremote` sulla riga di comando.

Java gateway accetta connessioni in ingresso da Zabbix server o proxy e può essere utilizzato solo come "proxy passivo". A differenza di Zabbix proxy, può anche essere utilizzato da Zabbix proxy (i proxy Zabbix non possono essere concatenati). L'accesso a ciascun Java gateway viene configurato direttamente nel file di configurazione di Zabbix server o proxy; pertanto, può essere configurato un solo Java gateway per ciascun Zabbix server o Zabbix proxy. Se un host avrà item di tipo **JMX agent** e item di altro tipo, solo gli item **JMX agent** verranno inoltrati a Java gateway per il recupero.

Quando un item deve essere aggiornato tramite Java gateway, Zabbix server o proxy si connetterà al Java gateway e richiederà il valore, che il Java gateway a sua volta recupera e restituisce al server o al proxy. Pertanto, il 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, il Java gateway avvia più thread, controllati dall'opzione **START_POLLERS**. Sul lato server, se una connessione richiede più di **Timeout** secondi, verrà terminata, ma il Java gateway potrebbe essere ancora occupato a recuperare il valore dal contatore JMX. Per risolvere questo problema, nel Java gateway è disponibile l'opzione **TIMEOUT**, che consente di impostare un timeout per le operazioni di rete JMX.

Zabbix server o proxy cercherà di raggruppare il più possibile le richieste verso un singolo target JMX (tenendo conto degli intervalli degli item) e di inviarle al Java gateway tramite un'unica connessione per ottenere prestazioni migliori.

Si consiglia di impostare **StartJavaPollers** a un valore minore o uguale a **START_POLLERS**, altrimenti potrebbero verificarsi situazioni in cui non sono disponibili thread nel Java gateway per gestire le richieste in arrivo; in tal caso 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 l'ultima versione di jboss-client.jar disponibile nella pagina di download di Wildfly.

2 Installazione dai pacchetti RHEL

Panoramica

Se installato da pacchetti, le seguenti informazioni ti aiuteranno a configurare lo Java gateway.

Configurazione e avvio di Java gateway

I parametri di configurazione di Zabbix Java gateway possono essere regolati nel file:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Per maggiori 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 versione 7:

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

Configurazione del server per l'uso con Java gateway

Con Java gateway attivo e 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 è necessario specificare i parametri di connessione nel file di configurazione del proxy.

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

Per impostazione predefinita, il server non avvia alcun processo relativo al monitoraggio JMX. Tuttavia, se si desidera utilizzarlo, è necessario specificare il numero di istanze pre-fork di Java poller. Questo si fa nello stesso modo in cui si specificano i normali poller e trapper.

```
StartJavaPollers=5
```

Non dimenticare di riavviare il server o il proxy una volta completata la configurazione.

Debug del Java gateway

Il file di log del Java gateway di Zabbix è:

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

Se si desidera aumentare il livello di logging, modificare il file:

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

e cambiare `level="info"` in "debug" o anche "trace" (per una risoluzione approfondita dei problemi):

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

</configuration>
```

Monitoraggio JMX

Vedi la pagina JMX monitoring per ulteriori dettagli.

3 Installazione 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 regolata nel file:

```
/etc/zabbix/zabbix_java_gateway.conf
```

Per maggiori 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:

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

Configurazione del server per l'utilizzo con Java gateway

Con Java gateway attivo e 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, è necessario specificare invece i parametri di connessione 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. Tuttavia, se si desidera utilizzarlo, è necessario specificare il numero di istanze pre-fork di Java poller. Questa operazione si esegue nello stesso modo in cui si specificano i normali poller e trapper.

```
StartJavaPollers=5
```

Non dimenticare di riavviare server o proxy una volta completata la configurazione.

Debug del Java gateway

Il file di log del Java gateway di Zabbix è:

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

Se si desidera aumentare il livello di logging, modificare il file:

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

e cambiare `level="info"` in ”debug” o anche ”trace” (per una risoluzione approfondita dei problemi):

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

</configuration>
```

Monitoraggio JMX

Vedi la pagina JMX monitoring per ulteriori dettagli.

Installazione dai sorgenti

Panoramica

Se installato dai sorgenti, le seguenti informazioni ti aiuteranno a configurare Zabbix Java gateway.

Panoramica dei file

Se hai ottenuto Java gateway dai sorgenti, dovresti ritrovarti con 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 vero e proprio.

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

Dipendenze di Java gateway: libreria 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 richiamato dagli script di avvio e arresto sopra indicati.

Configurazione ed esecuzione di Java gateway

Per impostazione predefinita, Java gateway resta in ascolto sulla porta 10052. Se si prevede di eseguire Java gateway su una porta diversa, è possibile specificarlo nello script settings.sh. Per informazioni su come specificare questa e altre opzioni, vedere la descrizione del file di configurazione di Java gateway.

Warning:
La porta 10052 non è registrata presso IANA.

Una volta definite le impostazioni desiderate, è possibile avviare Java gateway eseguendo lo script di avvio:

```
./startup.sh
```

Allo stesso modo, quando Java gateway non è più necessario, eseguire lo script di arresto per fermarlo:

```
./shutdown.sh
```

Si noti che, a differenza di server o proxy, Java gateway è leggero e non richiede un database.

Configurazione del server per l'utilizzo con Java gateway

Con Java gateway attivo e 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 è necessario specificare i parametri di connessione nel file di configurazione del proxy.

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

Per impostazione predefinita, il server non avvia alcun processo relativo al monitoraggio JMX. Tuttavia, se si desidera utilizzarlo, è necessario specificare il numero di istanze pre-fork dei Java poller. Questo si fa nello stesso modo in cui si specificano i normali poller e trapper.

```
StartJavaPollers=5
```

Non dimenticare di riavviare il server o il proxy una volta completata la configurazione.

Debug di Java gateway

Nel caso in cui si verifichino problemi con Java gateway oppure il messaggio di errore che vedi relativo 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 disporre di informazioni al 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 rilevate automaticamente. Quando hai terminato il debug, puoi riportare il livello di logging a "info".

Se desideri registrare i log in un file diverso o in un supporto completamente differente come un database, adatta il file logback.xml alle tue esigenze. Consulta il manuale di Logback per maggiori dettagli.

A volte, per scopi di debug, è utile avviare Java gateway come applicazione console anziché come demone. 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 sì che Logback utilizzi invece il file lib/logback-console.xml, che non solo registra sulla console, ma ha anche il livello di logging "debug" abilitato.

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. Consulta il manuale di SLF4J per maggiori dettagli.

Monitoraggio JMX

Vedi la pagina JMX monitoring per ulteriori dettagli.

6 Sender

Panoramica

Zabbix sender è un'utilità da riga di comando che può essere utilizzata per inviare dati sulle prestazioni al server Zabbix per l'elaborazione.

L'utilità viene solitamente usata in script utente a lunga esecuzione per l'invio periodico di dati di disponibilità e prestazioni.

Per inviare risultati direttamente al server Zabbix o al proxy, deve essere configurato un item di tipo trapper.

Vedere anche la libreria Python per Zabbix, che dispone di funzionalità integrate per agire come Zabbix sender.

Esecuzione di Zabbix sender

Un esempio di esecuzione di Zabbix UNIX sender:

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

dove:

- z - host del server Zabbix (può essere utilizzato anche l'indirizzo IP)
- s - nome tecnico dell'host monitorato (come registrato nel frontend di 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 maggiori informazioni, vedere la manpage 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 l'invio dei dati. Se l'invio a un indirizzo non riesce, il sender tenta l'invio agli altri indirizzi. Se l'invio di dati in batch a un indirizzo non riesce, i batch successivi non vengono inviati a questo indirizzo.

Zabbix sender accetta stringhe con codifica UTF-8 (sia per sistemi di tipo UNIX sia per Windows) senza 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 in rapida successione e li inviano al server tramite un'unica connessione. Un valore che non sia distante dal valore precedente più di 0,2 secondi può essere inserito nello stesso stack, ma il tempo massimo di polling rimane comunque di 1 secondo.

Note:
Zabbix sender terminerà 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 la low-level discovery:

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

Affinché questo funzioni, la regola di low-level discovery deve avere un tipo di item Zabbix trapper (in questo esempio, con chiave `trapper.discovery.item`).

7 Get

Panoramica

Zabbix get è un'utilità da riga di comando che può essere utilizzata per comunicare con Zabbix agent e recuperare le informazioni richieste dall'agent.

L'utilità viene solitamente utilizzata per la risoluzione dei problemi degli agent Zabbix.

Vedere anche la libreria Python per Zabbix, che dispone di funzionalità integrate per agire come Zabbix get.

Esecuzione di Zabbix get

Un esempio di esecuzione di Zabbix get in ambiente 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 item qui contiene uno spazio, quindi vengono utilizzate le virgolette per delimitare la chiave item per la shell. Le virgolette non fanno parte della chiave item; verranno rimosse dalla shell e non saranno passate a Zabbix agent.

Se una chiave item non è supportata, Zabbix get restituirà il codice di uscita 1.

Zabbix get accetta i seguenti parametri della 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 di origine
-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 utilizzato per comunicare con l'agent. Valori:
                                            auto - si connette utilizzando il protocollo JSON, in caso di
                                            json - si connette utilizzando il protocollo JSON
                                            plaintext - si connette utilizzando il protocollo plaintext in
-h --help                               Visualizza questo messaggio di aiuto
-V --version                            Visualizza il numero di versione

--tls-connect <value>                   Modalità di connessione all'agent. Valori:
                                            unencrypted - si connette senza crittografia (predefinito)
                                            psk - si connette utilizzando TLS e una chiave precondivisa
                                            cert - si connette utilizzando TLS e un certificato
--tls-ca-file <CA file>                 Percorso completo di un file contenente i certificati della/e CA d
--tls-crl-file <CRL file>               Percorso completo di un file contenente i certificati revocati
--tls-agent-cert-issuer <cert issuer>   Autorità emittente del certificato dell'agent consentita
--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, con distinzione tra maiuscole e minuscole, utiliz
--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 superiori) o stringa di
```

Vedere anche la manpage di Zabbix get per ulteriori informazioni.

Zabbix get in Windows può essere eseguito in modo simile:

```
zabbix_get.exe [options]
```

8 JS

Panoramica

zabbix_js è un'utilità da riga di comando che può essere utilizzata per testare script incorporati.

Questa utilità esegue uno script utente con un parametro di tipo stringa e ne stampa 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 codice 1.

Utilizzo

```
zabbix_js -s script-file -p input-param [-l log-level] [-t timeout]
zabbix_js -s script-file -i input-file [-l log-level] [-t timeout]
zabbix_js -h
zabbix_js -V
```

zabbix_js accetta i seguenti parametri della riga di comando:

```
-s, --script script-file          Specifica il nome del file dello script da eseguire. Se "-" è specificat
-i, --input input-file            Specifica il nome del file delle informazioni di input. Il contenuto ver
-p, --param input-param           Specifica il parametro di input: la variabile da passare allo script com
```

```
-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 (prede
-h, --help                        Visualizza le informazioni di aiuto.
-V, --version                     Visualizza 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`

9 Web service

Panoramica

Zabbix web service è un processo utilizzato per la comunicazione con web service esterni. Attualmente, il Zabbix web service viene utilizzato per generare e inviare report pianificati con l'intenzione di aggiungere ulteriori funzionalità in futuro.

Il server Zabbix si connette al web service tramite HTTP(S). Zabbix web service richiede che Google Chrome sia installato sullo stesso host; su alcune distribuzioni il servizio potrebbe funzionare anche con Chromium (vedere problemi noti).

Installazione

Il pacchetto ufficiale zabbix-web-service è disponibile nel repository di Zabbix.

Per compilare il servizio web di Zabbix dai sorgenti, specificare l'opzione di configurazione `--enable-webservice`.

Per configurare il servizio web di Zabbix, aggiornare i parametri del file di configurazione `zabbix_web_service.conf`.

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 comportare accessi non autorizzati.

5 Configurazione

Utilizzare la barra laterale per accedere ai contenuti della sezione Configurazione.

1 Configurazione di un modello

Panoramica

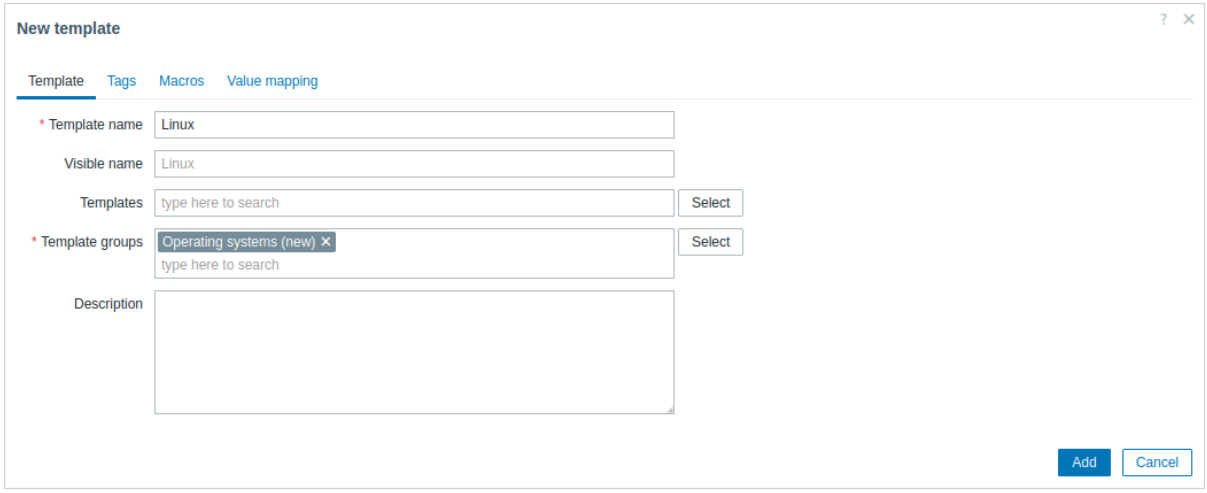
La configurazione di un template richiede innanzitutto la creazione del template definendone i parametri generali, quindi l'aggiunta di entità (item, trigger, grafici, ecc.).

Creazione di un template

Per creare un template, procedi come segue:

1. Vai su *Raccolta dati* > *Template*.
2. Fai clic su *Crea template*.
3. Modifica gli attributi del template.

La scheda **Template** contiene gli attributi generali del template.



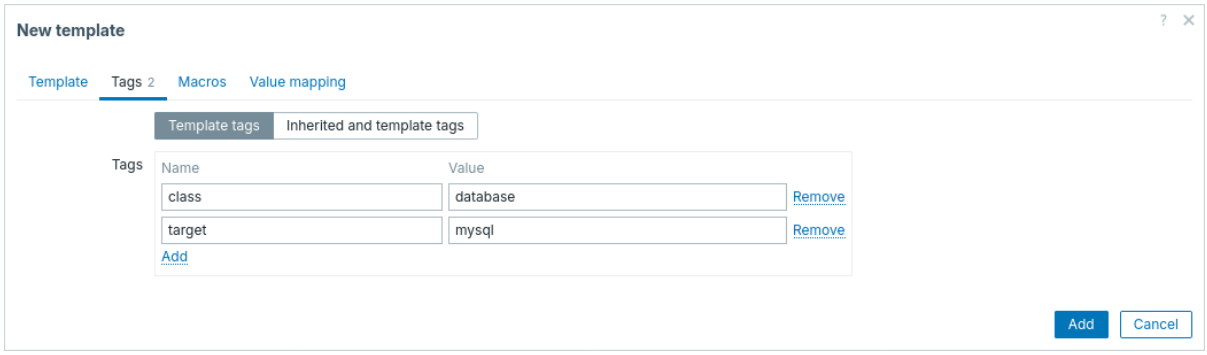
Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

Attributi del template:

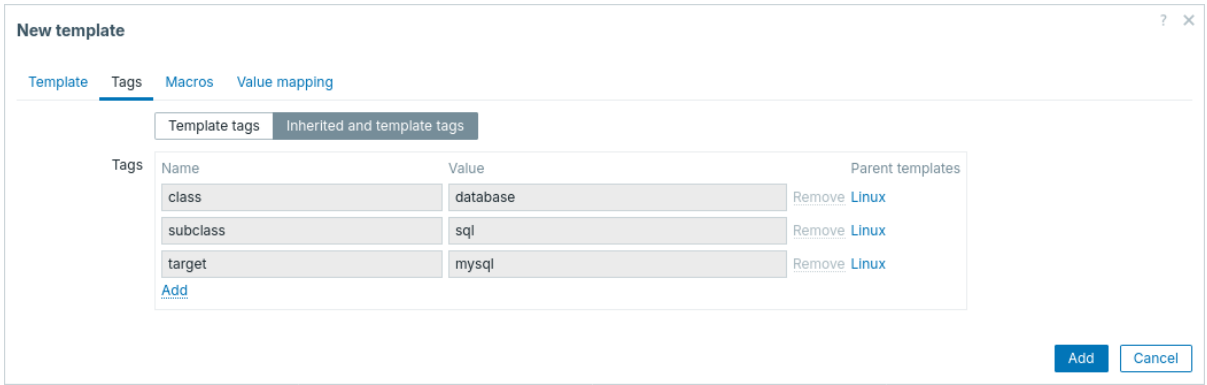
Parametro	Descrizione
Nome template	Nome univoco del template. Sono consentiti caratteri alfanumerici, spazi, punti, trattini e trattini bassi. Gli spazi iniziali e finali non sono consentiti.
Nome visibile	Se imposti questo nome, sarà quello visibile negli elenchi, nelle mappe, ecc.

Parametro	Descrizione
Template	Collega uno o più template a questo template. Tutte le entità (item, trigger, ecc.) saranno ereditate dai template collegati. Per collegare un nuovo template, digita il suo nome nel campo *Template*: verrà visualizzato un elenco a discesa dei template corrispondenti. In alternativa, fai clic su *Seleziona* per aprire il popup *Template*, che inizialmente non mostra alcun template. Per popolare l'elenco, digita il nome di un gruppo di template nel campo *Gruppo template* (e selezionane uno dall'elenco suggerito) oppure premi il pulsante *Seleziona* accanto al campo *Gruppo template* per aprire un popup *Gruppi template*. Una volta scelto un gruppo di template, il popup *Template* viene aggiornato per mostrare i template appartenenti a quel gruppo. I template selezionati verranno collegati quando la configurazione sarà salvata o aggiornata. Per scollegare un template, usa una delle due opzioni nel blocco *Template*: *Scollega* - scollega il template, ma ne conserva le entità (item, trigger, ecc.); *Scollega e cancella* - scollega il template e rimuove tutte le sue entità (item, trigger, ecc.).
Gruppi template	Gruppi template a cui appartiene il template.
Descrizione	Descrizione del template.
Fornitore e versione	Fornitore 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 può essere modificato 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 nuova versione principale di Zabbix). Ad esempio, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

La scheda **Tag** consente di definire tag a livello di template. Tutti i problemi dei host collegati a questo template saranno etichettati con i valori inseriti qui.



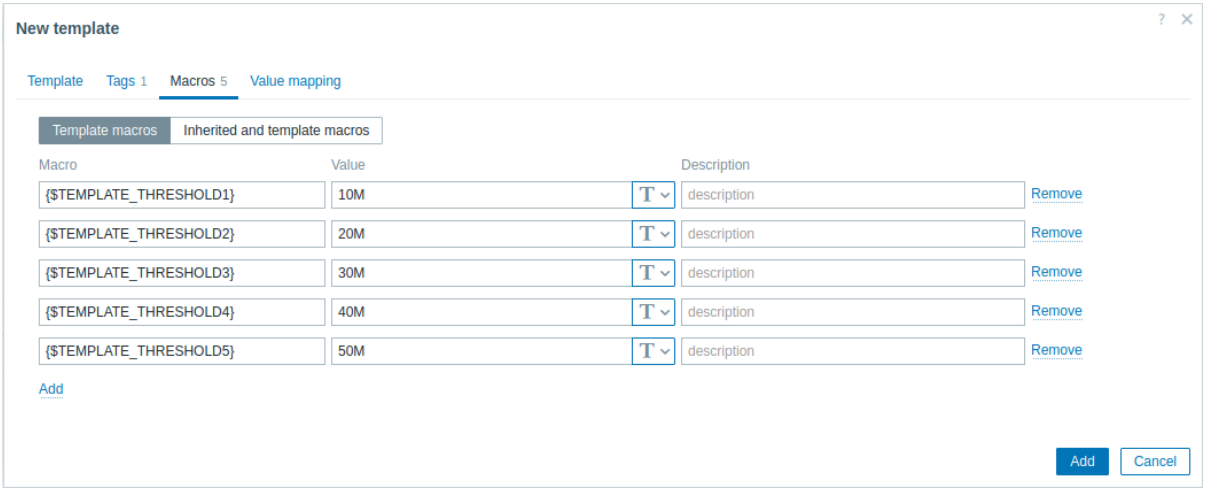
Se selezioni l'opzione *Tag ereditati e del template*, puoi anche visualizzare i tag dei template collegati che il template erediterà.



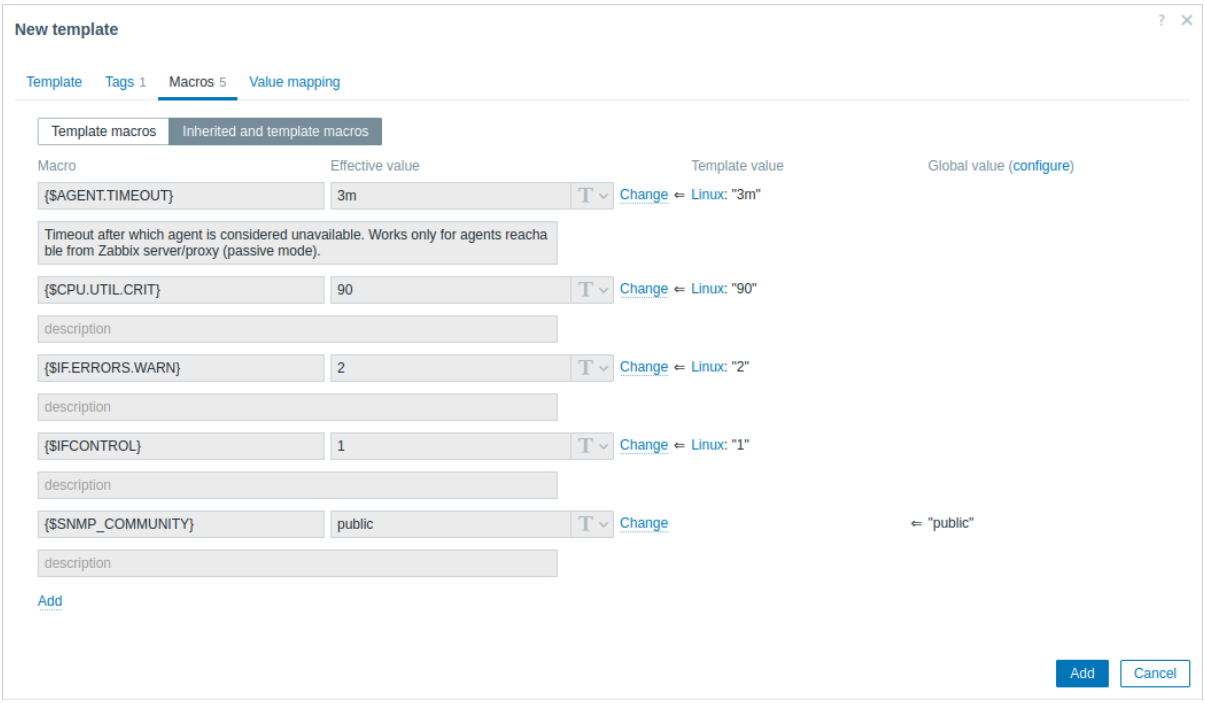
Per comodità, sono forniti collegamenti ai rispettivi template.

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}.

La scheda **Macro** consente di definire macro utente a livello di template come coppie nome-valore. Tieni presente che i valori delle macro possono essere mantenuti come testo normale, testo segreto o segreto Vault. È supportata anche l'aggiunta di una descrizione.



Se selezioni l'opzione *Macro ereditate e del template*, puoi anche visualizzare le macro dei template collegati e le macro globali che il template erediterà, nonché i valori a cui le macro verranno risolte.



Per comodità, sono forniti collegamenti ai rispettivi template, nonché un collegamento alla configurazione delle macro globali. È anche possibile modificare una macro di un template collegato o una macro globale a livello di template, creando di fatto una copia della macro nel template.

La scheda **Mappatura valori** consente di configurare una rappresentazione più comprensibile dei dati degli item nelle mappature dei valori.

Pulsanti:

Add	Aggiungi il template. Il template aggiunto dovrebbe comparire 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, ecc.) sia ereditate dai template collegati sia collegate direttamente al template corrente, ma **esclude** fornitore e versione del template corrente, in modo che il template clonato sia distinguibile dall'originale.
Delete	Elimina il template; le entità del template (item, trigger, ecc.) rimangono nei host collegati.
Delete and clear	Elimina il template e tutte le sue entità dai 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. I trigger e i grafici non possono essere aggiunti senza il corrispondente item.

Esistono due modi per aggiungere item al template:

1. Per creare nuovi item, seguire le linee guida per Creazione di un item.

2. Per aggiungere item esistenti al template:

- Andare in *Raccolta dati* > *Host* (oppure *Template*).
- Fare clic su *Item* nella riga dell'host/template richiesto.
- Selezionare le caselle di controllo degli item che si desidera aggiungere al template.
- Fare clic su *Copia* sotto l'elenco degli item.
- Selezionare il template (o il gruppo di template) in cui gli item devono essere copiati e fare clic su *Copia*.
  Tutti gli item selezionati dovrebbero essere copiati nel template.

L'aggiunta di trigger e grafici avviene in modo simile (rispettivamente dall'elenco dei trigger e dei grafici), tenendo ancora presente che possono essere aggiunti solo se prima vengono aggiunti gli item richiesti.

Aggiunta di dashboard

Per aggiungere dashboard a un template in *Raccolta dati* > *Template*, procedi come segue:

1. Fai clic su *Dashboard* nella riga del template.
2. Configura una dashboard seguendo le linee guida di configurazione delle dashboard.

Attention:
Quando si configurano widget in una dashboard di template (anziché in una dashboard globale), i parametri relativi all'host non sono disponibili e alcuni parametri hanno un'etichetta diversa. Questo perché le dashboard di template visualizzano dati solo dall'host a cui il template è collegato. Ad esempio, i parametri *Gruppi host*, *Escludi gruppi host* e *Host* nel widget *Problemi* non sono disponibili, il parametro *Gruppi host* nel widget *Disponibilità host* non è disponibile e il parametro *Mostra host in manutenzione* viene rinominato in *Mostra dati in manutenzione*, ecc. Per ulteriori informazioni sulla disponibilità dei parametri nei widget delle dashboard di template, consulta i parametri specifici per ciascun widget della dashboard.

Note:
Per i dettagli su come accedere alle dashboard host create dalle dashboard di template, consulta la sezione dashboard host.

Configurazione delle regole di low-level discovery

Vedere la sezione low-level discovery del manuale.

Aggiunta di scenari web

Per aggiungere scenari web a un template in *Raccolta dati* > *Template*, procedere come segue:

1. Fare clic su *Web* nella riga del template.
2. Configurare uno scenario web seguendo il metodo consueto per la configurazione degli scenari web.

2 Configurazione di un gruppo di template

Overview

Template groups are used for the logical grouping of templates and assigning user permissions to them.

Each template must have at least one template group assigned. A template may belong to multiple template groups, and each template group may contain multiple templates.

Note that in Zabbix, all permissions are based on groups: user groups, host groups, and template groups. So, even if a single user needs access to a single template, it is granted by adding the user to a user group that has permission to access the template group containing that template.

Configurazione

> **Attention:**
> Solo gli utenti Super admin possono creare gruppi di template.

Esistono due opzioni per creare un gruppo di template nel frontend di Zabbix.

Prima opzione:

- Andare su: *Raccolta dati → Gruppi di template*
- Fare clic su *Crea gruppo di template* nell'angolo superiore destro dello schermo
- Inserire il nome del gruppo nel modulo



Seconda opzione: durante la configurazione di un template, inserire un nome di gruppo non esistente nel campo di input *Gruppi di template*.

Una volta creato il gruppo di template, è possibile fare clic sul nome del template nell'elenco in *Raccolta dati → Gruppi di template* per modificare il nome del gruppo, clonare il gruppo o eliminare il gruppo.

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 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`.

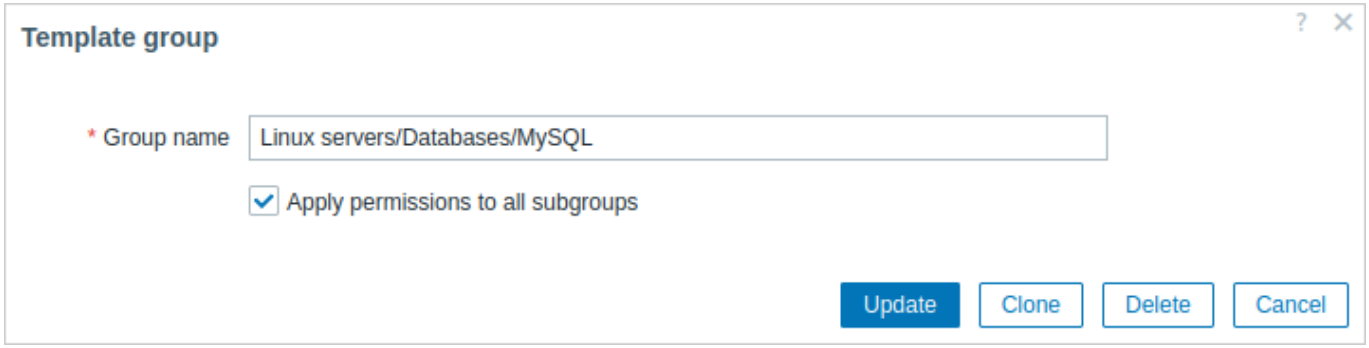
Quando si crea un sottogruppo, non è consentito utilizzare barre oblique iniziali o finali, né più barre oblique consecutive. L'escape di '/' non è supportato.

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 ai gruppi di template

- Quando si crea un sottogruppo in un gruppo di template padre esistente (ad esempio, creando `Linux servers/Databases` quando `Linux servers` esiste già), i permessi del gruppo utenti sul sottogruppo vengono ereditati dal gruppo padre.
- Quando si crea un gruppo di template padre per un sottogruppo esistente (ad esempio, creando `Linux servers` quando `Linux servers/Databases` esiste già), non vengono impostati permessi per il gruppo padre.

Quando si modifica un qualsiasi gruppo di template, è anche possibile impostare un'opzione aggiuntiva, *Applica i permessi a tutti i sottogruppi*.



Selezionando questa casella di controllo e facendo clic su *Aggiorna* verrà applicato lo stesso livello di permessi a tutti i sottogruppi attuali e futuri del gruppo di template in fase di modifica.

Quindi, se a uno qualsiasi dei gruppi utenti sono stati assegnati permessi diversi per i sottogruppi del gruppo di template in fase di modifica, selezionando la casella di controllo verranno concessi a tutti i sottogruppi attuali e futuri gli stessi permessi utente del gruppo in fase di 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.

3 Collegamento/scollegamento

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 in *Raccolta dati* → *Host*.
2. Fare clic sul host richiesto.
3. Iniziare a digitare il nome del template nel campo *Template*. Verrà visualizzato un elenco dei template corrispondenti; scorrere verso il basso per selezionare quello desiderato. In alternativa, è possibile 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 del host.

Il host avrà ora tutte le entità del template. Ciò include item, trigger, grafici, regole di low-level discovery, 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 item, non possono essere collegati a un singolo host da più template se usano chiavi item identiche.

Quando le entità (item, trigger, ecc.) vengono aggiunte dal template:

- le entità identiche già esistenti sul host vengono aggiornate come entità del template e **qualsiasi personalizzazione esistente dell'entità a livello di host andrà persa**;
- le entità del template vengono aggiunte;
- eventuali entità collegate direttamente che, prima del collegamento del template, esistevano solo sul host rimangono inalterate.

Negli elenchi, tutte le entità del template sono ora precedute dal nome del template, a indicare che appartengono a quel particolare template. Il nome del template stesso (in testo 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 HTTP agent, controlli semplici, controlli SSH e controlli Telnet, un'interfaccia host è facoltativa. Se, al momento del collegamento di un template, il host non ha un'interfaccia definita, questi item verranno aggiunti senza interfaccia. Se si aggiunge successivamente un'interfaccia host, questa non verrà assegnata automaticamente agli item già esistenti. Per assegnare in una sola volta la nuova interfaccia host a tutti gli item del template, scollegare il template dal host e quindi ricollegarlo. Per preservare la cronologia degli item utilizzare l'opzione *Scollega*, non usare *Scollega e cancella*.

Se un'entità non è preceduta dal nome del template, significa che esisteva già sul host e non è stata aggiunta dal template.

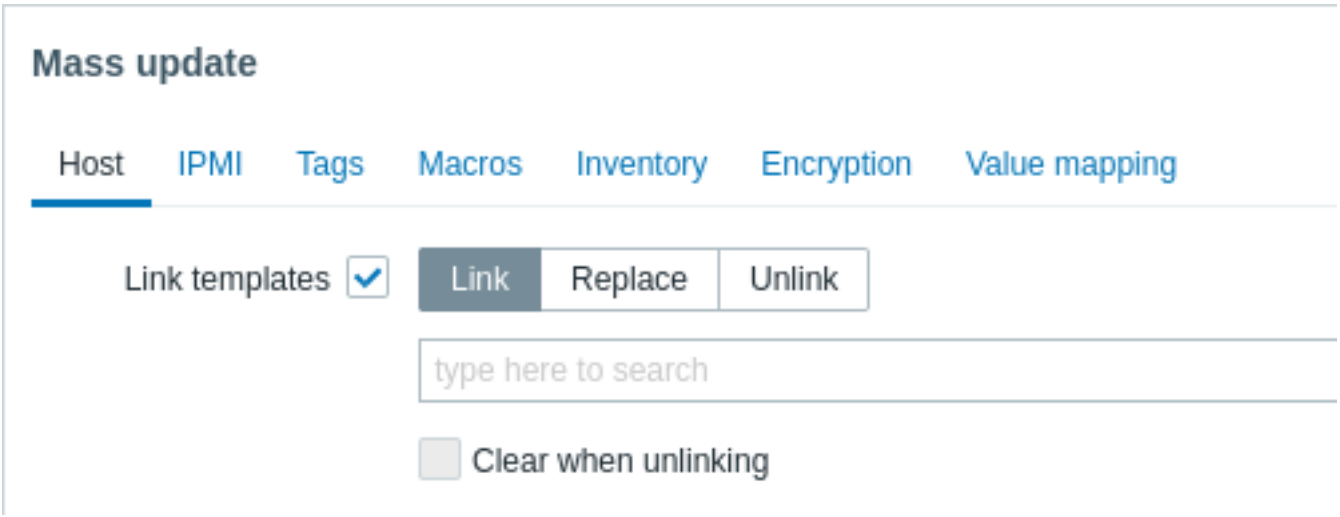
Criteri di unicità delle entità

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 massivo** sotto l'elenco e poi selezionare *Collega template*:



Per collegare template aggiuntivi, iniziare a digitare il nome del template nel campo con completamento automatico finché non appare un menu a discesa che propone i template corrispondenti. È sufficiente scorrere verso il basso per selezionare il template da collegare.

L'opzione *Sostituisci* consentirà di collegare un nuovo template scollegando qualsiasi template che era stato precedentemente collegato agli host. L'opzione *Scollega* consentirà di specificare quali template scollegare. L'opzione *Cancella durante lo scollegamento* consentirà non solo di scollegare eventuali template collegati in precedenza, ma anche di rimuovere tutte le entità ereditate da essi (item, trigger, ecc.).

Note:
Zabbix offre un ampio insieme di template predefiniti. È possibile usarli come riferimento, ma occorre fare attenzione a non utilizzarli invariati in produzione, poiché potrebbero contenere troppi item e raccogliere dati troppo frequentemente. Se si desidera usarli, è consigliabile perfezionarli in base alle reali esigenze.

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.

Scollegamento di un template

Per scollegare un template da un host, procedere come segue:

1. Andare su *Raccolta dati* → *Host*.
2. Fare clic sull'host richiesto e individuare il campo *Template*.
3. Fare clic su *Scollega* oppure *Scollega e cancella* accanto al template per scollegare il template.

4. Fare clic su *Aggiorna* nel modulo degli attributi dell'host.

La scelta dell'opzione *Scollega* rimuoverà semplicemente l'associazione con il template, lasciando però tutte le sue entità nell'host. Ciò include item, trigger, grafici, regole di low-level discovery e scenari web, ma esclude le dashboard. Si noti che anche le mappe dei valori e i tag ereditati dai template collegati verranno rimossi.

La scelta dell'opzione *Scollega e cancella* rimuoverà sia l'associazione con il template sia tutte le sue entità (item, trigger, ecc.).

4 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 annidati

Per collegare i template, è necessario prendere un template esistente (o crearne uno nuovo) e quindi:

1. Aprire il modulo di configurazione del template.
2. Individuare il campo *Templates*.
3. Fare clic su *Select* per aprire la finestra pop-up *Templates*.
4. Nella finestra pop-up, 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. Ciò include item, trigger, grafici, regole di low-level discovery e scenari web, ma esclude le dashboard. Tuttavia, le dashboard dei template collegati verranno comunque ereditate 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 alcuna 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.).

5 Aggiornamento di massa

Panoramica

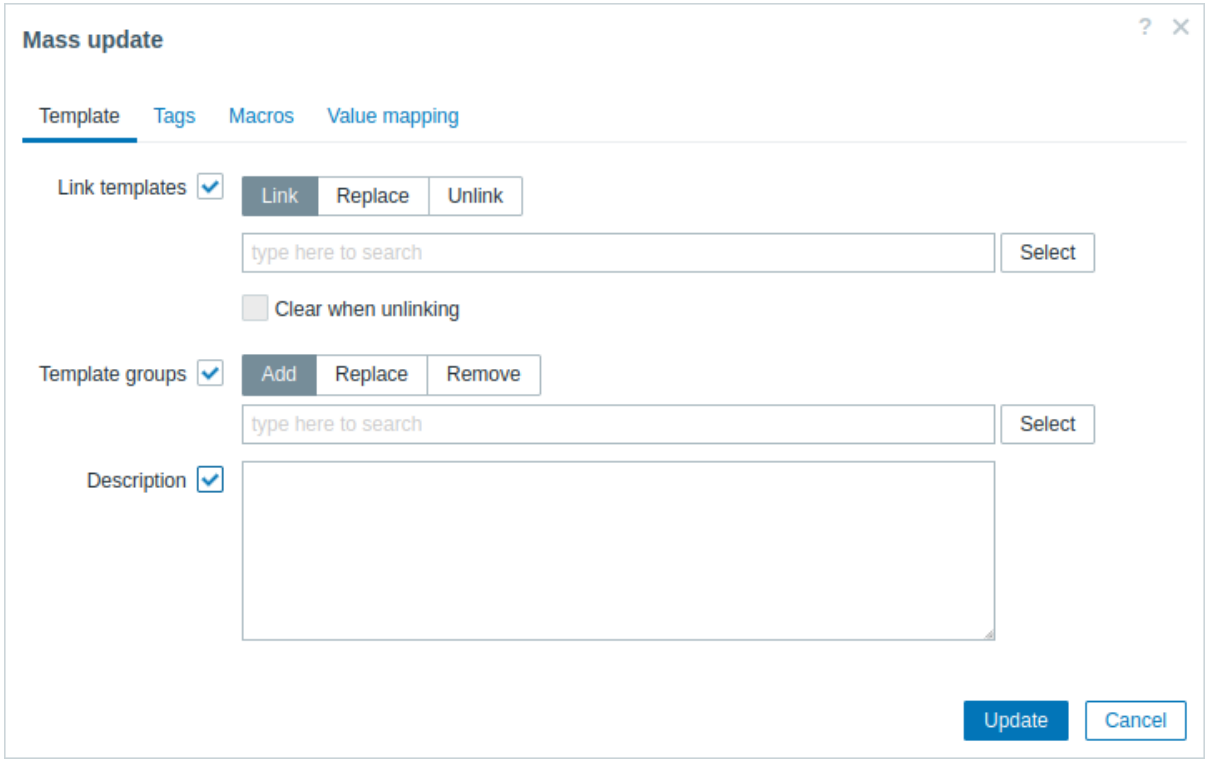
A volte può essere necessario modificare contemporaneamente un attributo per più template. Invece di aprire ogni singolo template per la modifica, è possibile utilizzare la funzione di aggiornamento di massa.

Utilizzo dell'aggiornamento di massa

Per aggiornare in massa alcuni template, procedere come segue:

1. Selezionare le caselle di controllo davanti 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*, *Tag*, *Macro* o *Mappatura dei valori*).
4. Selezionare le caselle di controllo degli attributi da aggiornare e inserire per essi un nuovo valore.

La scheda **Template** contiene le opzioni generali di aggiornamento massivo dei template.



Sono disponibili le seguenti opzioni selezionando il pulsante corrispondente per l'aggiornamento *Link templates*:

- *Link* - specifica quali template aggiuntivi collegare;
- *Replace* - specifica quali template collegare scollegando allo stesso tempo tutti i template collegati in precedenza;
- *Unlink* - specifica quali template scollegare.

Per specificare i template da collegare/scollegare, iniziare a digitare il nome del template nel campo di completamento automatico finché non appare un menu a discesa con i template corrispondenti. È sufficiente scorrere verso il basso per selezionare i template richiesti.

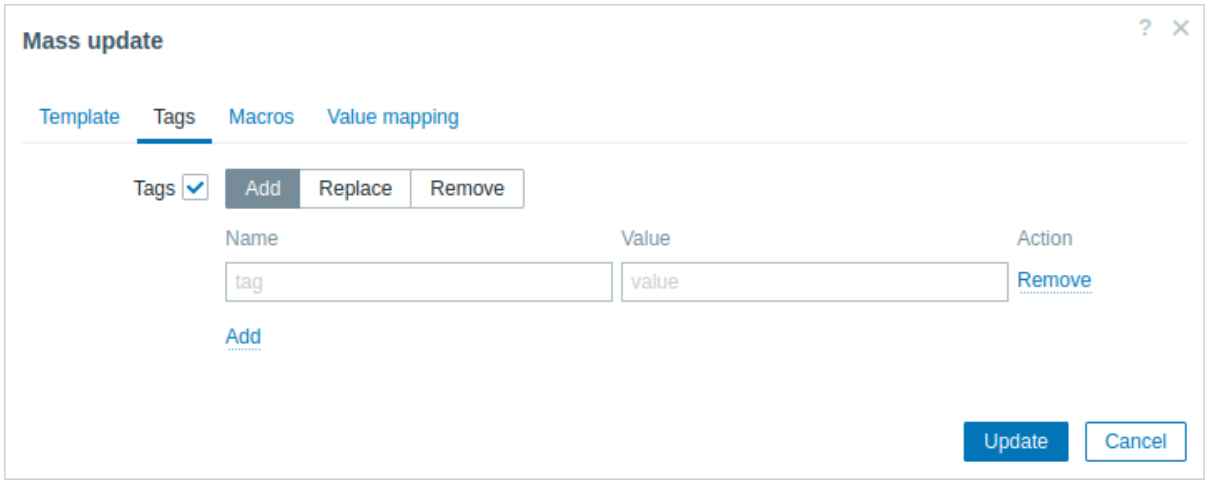
L'opzione *Clear when unlinking* consente di scollegare tutti i template collegati in precedenza, nonché di rimuovere tutti gli elementi ereditati da essi (item, trigger, grafici, ecc.).

Sono disponibili le seguenti opzioni selezionando il pulsante corrispondente per l'aggiornamento *Template groups*:

- *Add* - consente di specificare gruppi di template aggiuntivi tra quelli esistenti oppure di inserire gruppi di template completamente nuovi per i template;
- *Replace* - rimuoverà il template da tutti i gruppi di template esistenti e li sostituirà con quello/quelli specificati in questo campo (gruppi di template esistenti o nuovi);
- *Remove* - rimuoverà specifici gruppi di template dai template.

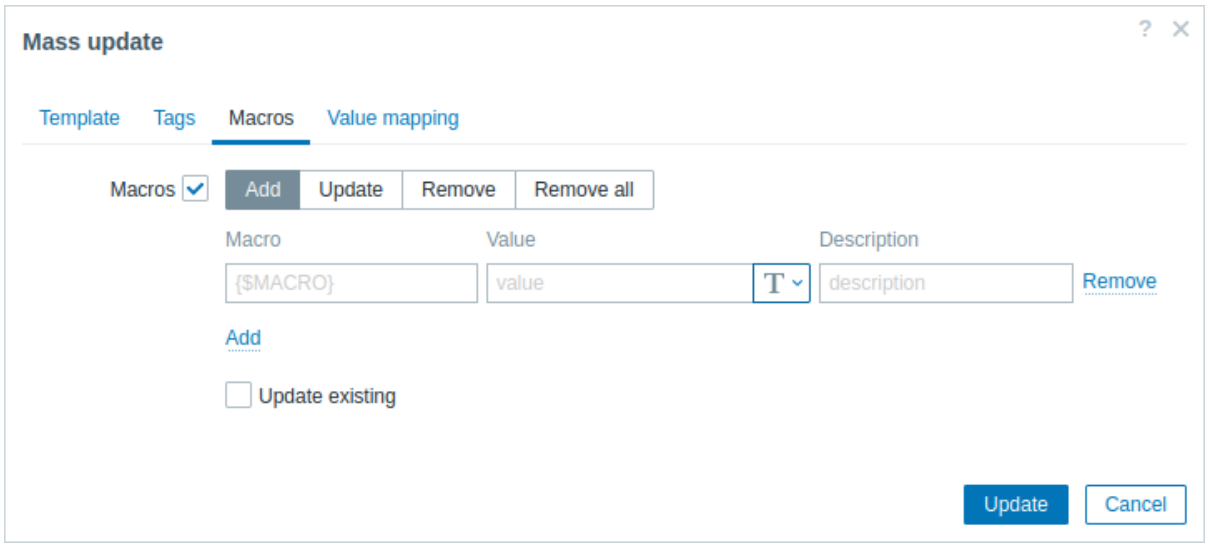
Questi campi supportano il completamento automatico: iniziando a digitare al loro interno viene visualizzato un menu a discesa con i gruppi di template corrispondenti. Se il gruppo di template è nuovo, anch'esso compare nel menu a discesa ed è indicato da *(new)* dopo la stringa. È sufficiente scorrere 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.

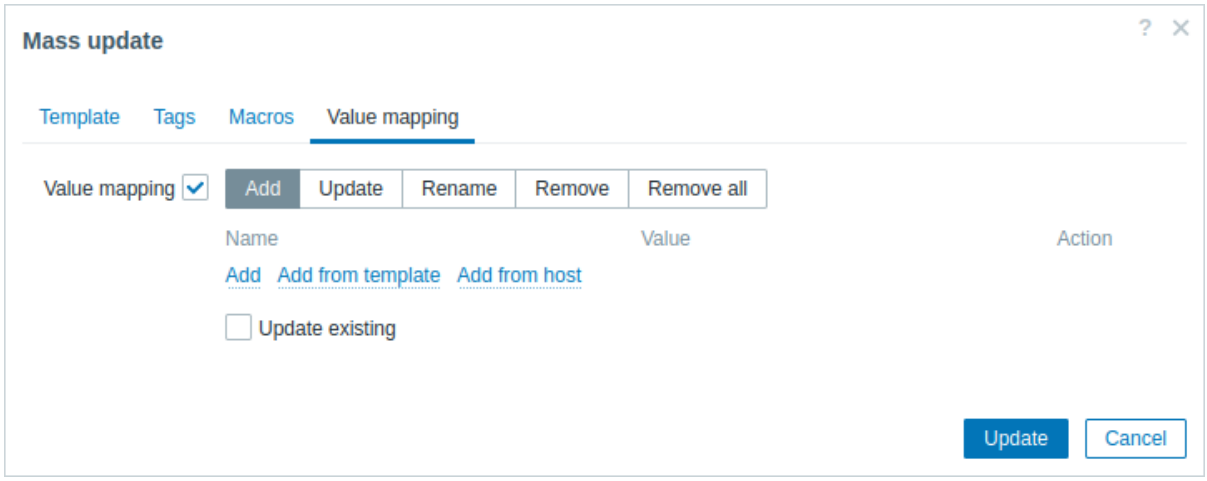
La scheda **Macro** consente di aggiornare in massa le macro a livello di template.



Sono disponibili le seguenti opzioni quando si seleziona il rispettivo pulsante per l'aggiornamento delle macro:

- *Aggiungi* - consente di specificare macro utente aggiuntive per i template. Se la casella di controllo *Aggiorna esistenti* è selezionata, il valore, il tipo e la descrizione del nome macro specificato verranno aggiornati. Se non è selezionata, se una macro con quel nome esiste già nel template o nei template, non verrà aggiornata.
- *Aggiorna* - sostituirà i valori, i tipi e le descrizioni delle macro specificate in questo elenco. Se la casella di controllo *Aggiungi mancanti* è selezionata, una macro che in precedenza non esisteva in un template verrà aggiunta come nuova macro. Se non è selezionata, verranno aggiornate solo le macro che esistono già in un template.
- *Rimuovi* - rimuoverà le macro specificate dai template. Se la casella *Tranne 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* - rimuoverà tutte le macro utente dai template. Se la casella di controllo *Confermo di 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 l'aggiornamento delle mappature dei valori sono disponibili pulsanti con le seguenti opzioni:

- *Aggiungi* - aggiunge mappature dei valori ai template. Se si seleziona *Aggiorna esistenti*, tutte le proprietà della mappatura dei valori con questo nome verranno aggiornate. Altrimenti, se una mappatura dei valori con quel nome esiste già, non verrà aggiornata.
- *Aggiorna* - aggiorna le mappature dei valori esistenti. Se si seleziona *Aggiungi mancanti*, una mappatura dei valori che in precedenza non esisteva in un template verrà aggiunta come nuova mappatura dei valori. Altrimenti verranno aggiornate solo le mappature dei valori già esistenti in un template.
- *Rinomina* - assegna un nuovo nome a una mappatura dei valori esistente.
- *Rimuovi* - rimuove le mappature dei valori specificate dai template. Se si seleziona *Tranne selezionati*, verranno rimosse tutte le mappature dei valori **tranne** quelle specificate.

- *Rimuovi tutto* - rimuove tutte le mappature dei valori dai template. Se la casella di controllo *Confermo di rimuovere tutte le mappature dei valori* non è selezionata, si aprirà una nuova finestra popup che chiederà di confermare la rimozione.

Per le operazioni di aggiunta/aggiornamento della mappatura dei valori sono disponibili le opzioni *Aggiungi da template* e *Aggiungi da host*.
Consentono di selezionare le mappature dei valori rispettivamente da un template o da un host.

Una volta completate tutte le modifiche richieste, fare clic su *Update*. Gli attributi saranno aggiornati di conseguenza per tutti i template selezionati.

1 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.

1 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 guida 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:

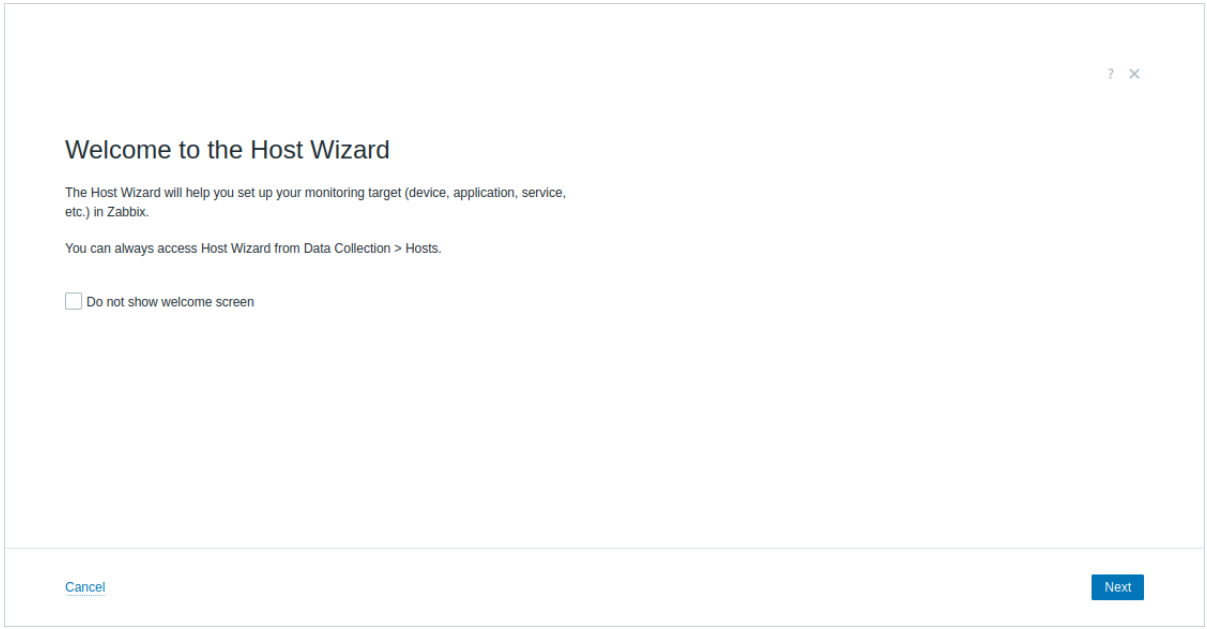
- Vai a: *Data collection* > *Hosts*
- Fai clic su *Host Wizard* nell'angolo in alto a destra dello schermo
- Segui le istruzioni nella procedura guidata

Puoi anche modificare un host esistente facendo clic sul suo nome (o sull'icona con i tre puntini accanto) in varie sezioni del frontend per accedere a Host Wizard dal menu host. Tieni presente che alcuni passaggi vengono saltati durante la modifica di un host esistente se questo include già la configurazione richiesta dal template selezionato.

In alternativa, puoi utilizzare l'approccio classico per la creazione e configurazione di un host.

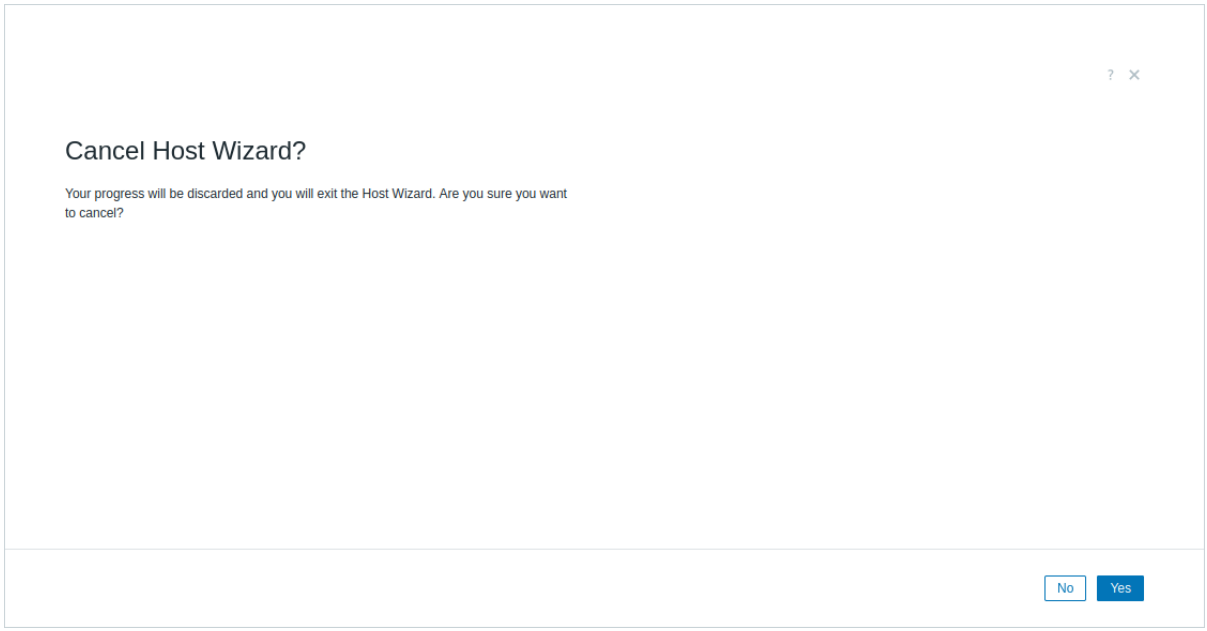
Benvenuti nella procedura guidata host

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 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 i progressi. Facendo clic su *No*, premendo ESC o facendo clic sull'icona di chiusura si tornerà all'ultimo passaggio.

Selezionare un template

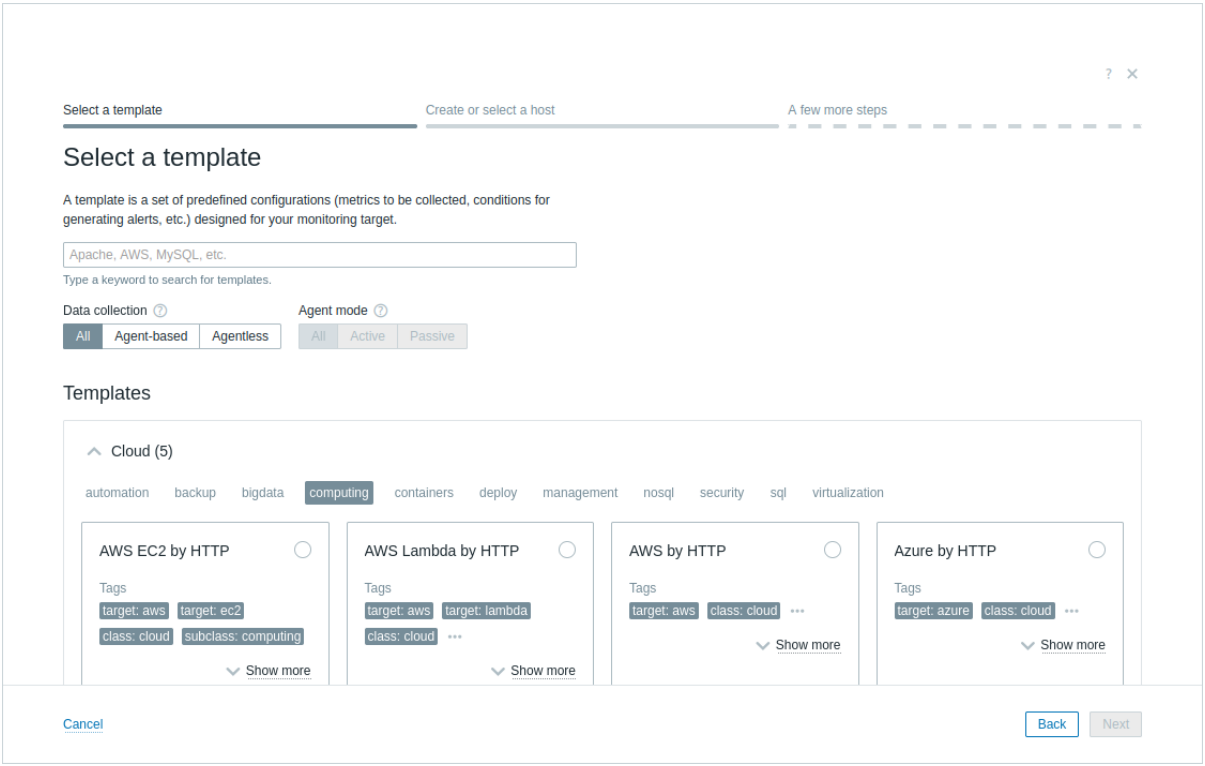
Il primo passo per configurare il target di monitoraggio consiste nel selezionare un template, ovvero un insieme di configurazioni predefinite (metriche da raccogliere, condizioni per generare avvisi, ecc.) progettate per il target di monitoraggio.

I template predefiniti di Zabbix offrono una varietà di configurazioni di monitoraggio già pronte per sistemi operativi, applicazioni, database, dispositivi di rete, servizi e altro ancora.

In questo passaggio è possibile:

- Sfogliare i template per classe (Cloud, Database, Network, ecc.) e sottoclasse (ad esempio, automation, backup), basate sui tag del template.
- Cercare i template tramite parole chiave nel nome del template oppure nei nomi e nei valori dei tag del template.
- Filtrare i template per metodo di raccolta dati (basato su agent o senza agent; i template basati su agent includono almeno un item Zabbix agent).
- Filtrare i template per modalità 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 agent passivi e attivi).

- Quando si utilizza l'Host Wizard per configurare un host esistente, è anche possibile filtrare i template già collegati all'host.



Attention:
Quando si seleziona un template, potrebbe essere visualizzato il seguente messaggio: *Some templates (n) are incompatible with the Host Wizard. See how to update them. Custom templates are not supported.* Ciò indica che alcuni template predefiniti non sono ancora compatibili con l'Host Wizard e devono essere aggiornati; vedere le istruzioni per l'aggiornamento dei template.

Il template selezionato determinerà i passaggi successivi nell'Host Wizard. Ad esempio, se si seleziona il template *MySQL by Zabbix agent*, che utilizza Zabbix agent per la raccolta dei dati, la procedura guidata accompagnerà l'utente nel 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'obiettivo del monitoraggio. Una volta collegato, l'host riceve tutte le entità del template, come item (metriche da raccogliere) e trigger (condizioni per generare avvisi), necessarie per il monitoraggio.

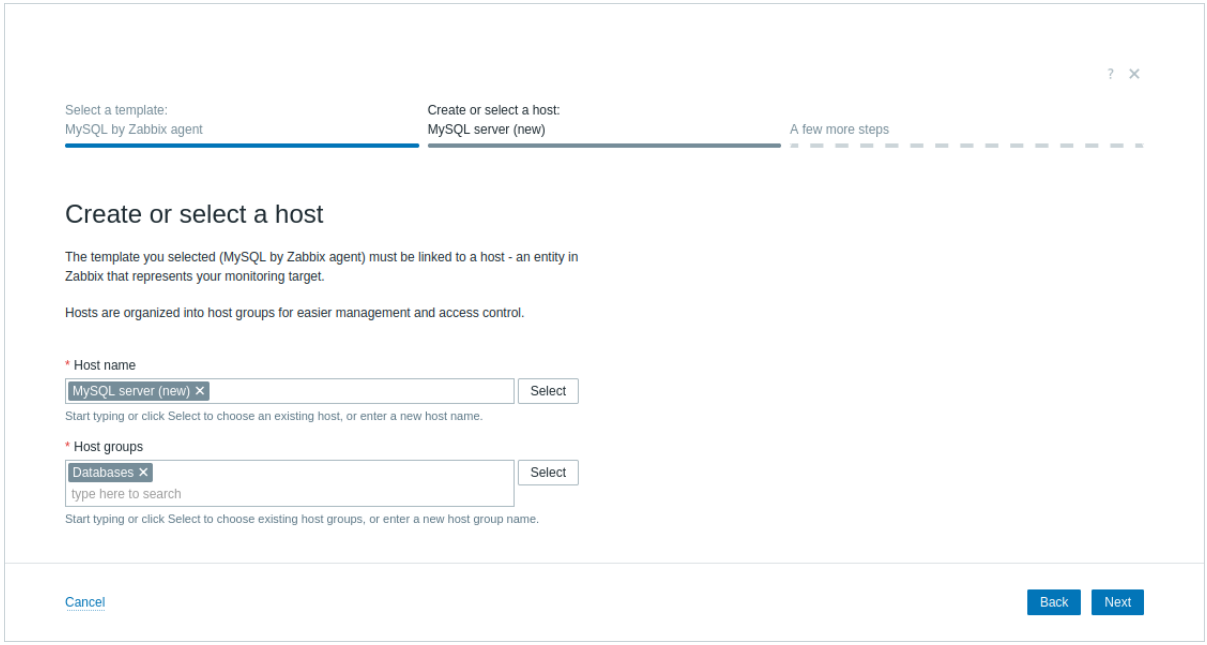
Ogni host deve inoltre appartenere ad almeno un gruppo di host, utilizzato per organizzare gli host e assegnare loro i permessi utente.

In questo passaggio, è possibile:

- Creare un nuovo host e un nuovo gruppo di host inserendo i nomi di entrambi.
- Creare un nuovo host inserendone il nome e selezionare un gruppo di host esistente, al quale verrà assegnato il nuovo host.
- Selezionare un host esistente; è inoltre possibile assegnare questo host a gruppi di host aggiuntivi senza rimuoverlo da alcun gruppo corrente.

Note:
La selezione di un host esistente non rimuove né sovrascrive la sua configurazione attuale, salvo diversa indicazione esplicita nei passaggi successivi. Gli host rilevati automaticamente 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 successivamente la gestione dei permessi.



Installare Zabbix agent

I template basati su agent (come *MySQL by Zabbix agent*) richiedono l'installazione di Zabbix agent o agent 2, un processo distribuito sulla destinazione di monitoraggio per monitorare attivamente le risorse locali e le applicazioni (se richiesto dal template).

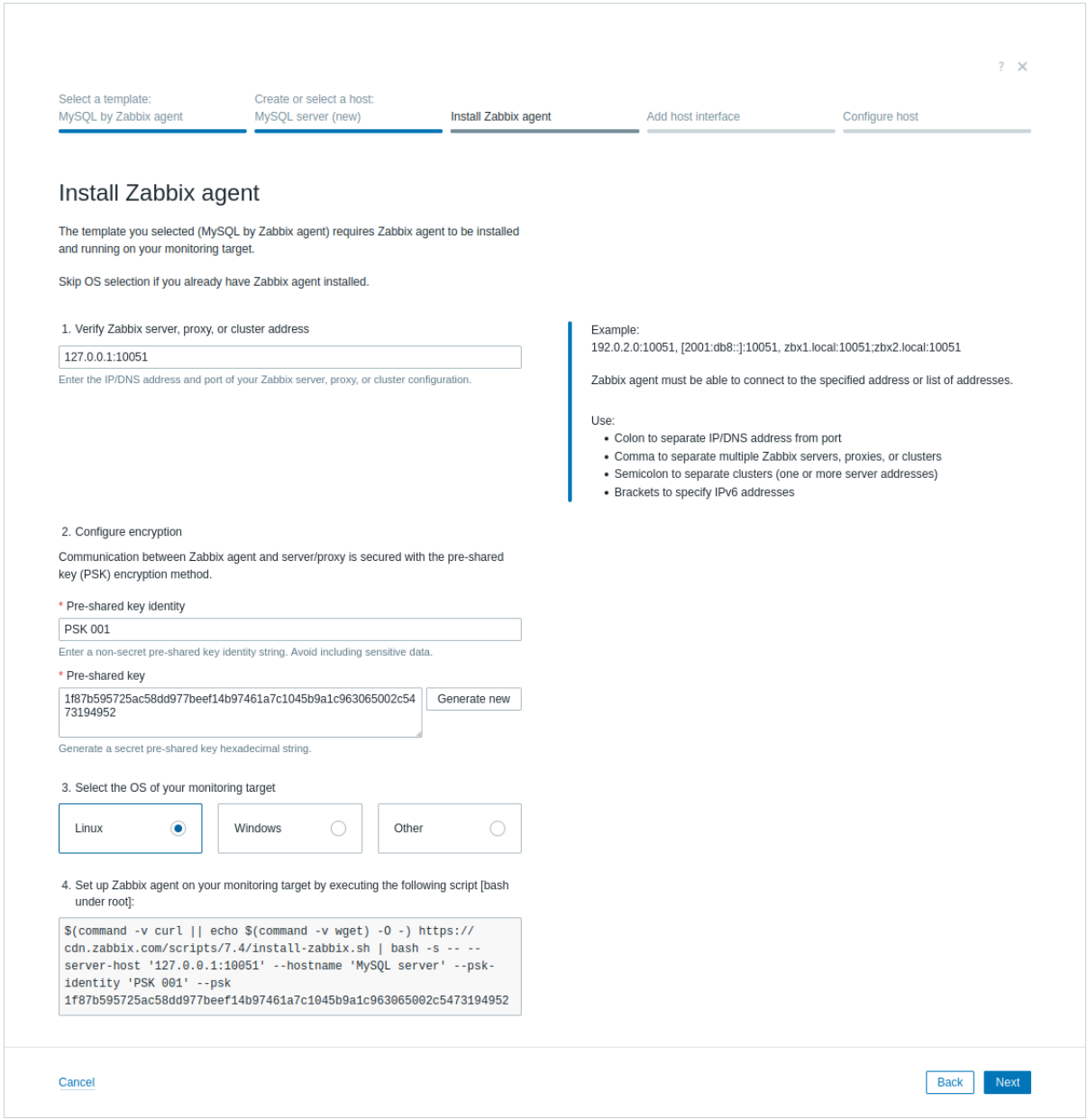
In questo passaggio:

- Verificare l'indirizzo del server, proxy o cluster Zabbix (ad esempio, 192.0.2.0:10051). Questo valore verrà utilizzato per configurare i parametri Server e ServerActive per Zabbix agent.
- Immettere una nuova identità della chiave precondivisa univoca e non segreta (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, *server MySQL*) 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 la destinazione di monitoraggio.
- Installare Zabbix agent su quel sistema eseguendo lo script fornito oppure seguendo le istruzioni di installazione.

Ad esempio, è possibile impostare *Pre-shared key identity* su *PSK 001* e generare una nuova *Pre-shared key*. Quindi, è possibile selezionare *Linux* come sistema operativo ed eseguire lo script fornito su quel sistema. Dopo aver installato Zabbix agent, tornare alla Procedura guidata host.

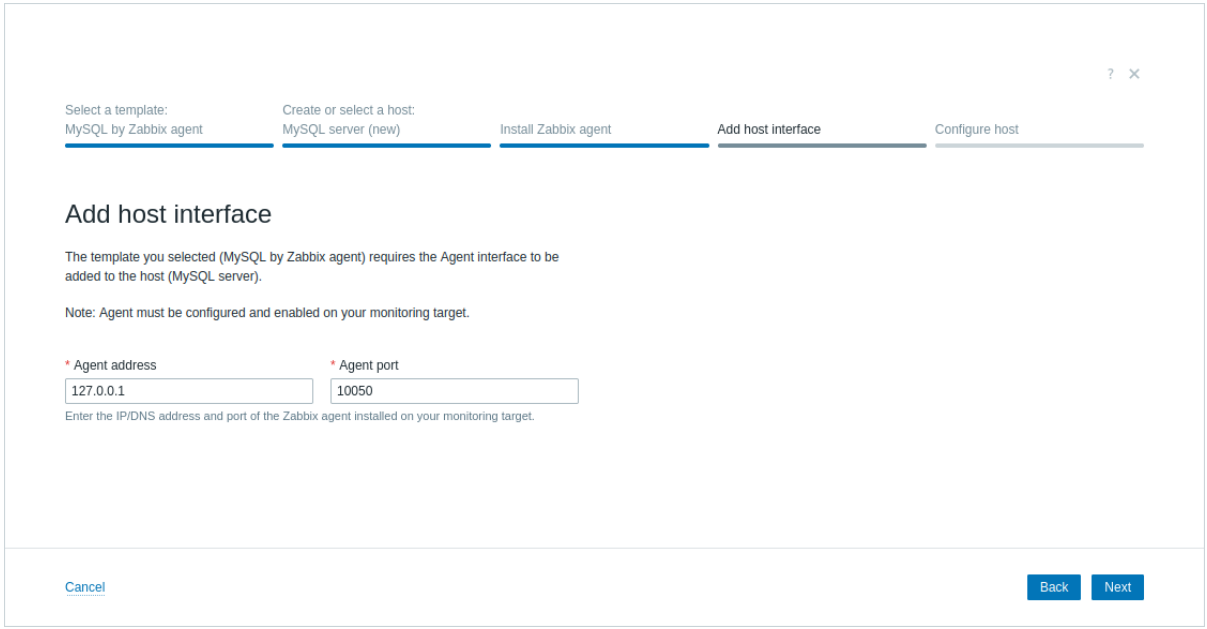


Aggiungi interfaccia host

L'interfaccia host definisce come il server Zabbix si connette tramite la rete alla destinazione di monitoraggio. Definisce i parametri di connessione, come indirizzo IP, nome DNS, porta e tipo di interfaccia (Agent, SNMP, JMX o IPMI), in base al metodo di raccolta dati richiesto dal template selezionato.

In questo passaggio, inserisci i dettagli dell'interfaccia host richiesti dal template selezionato.

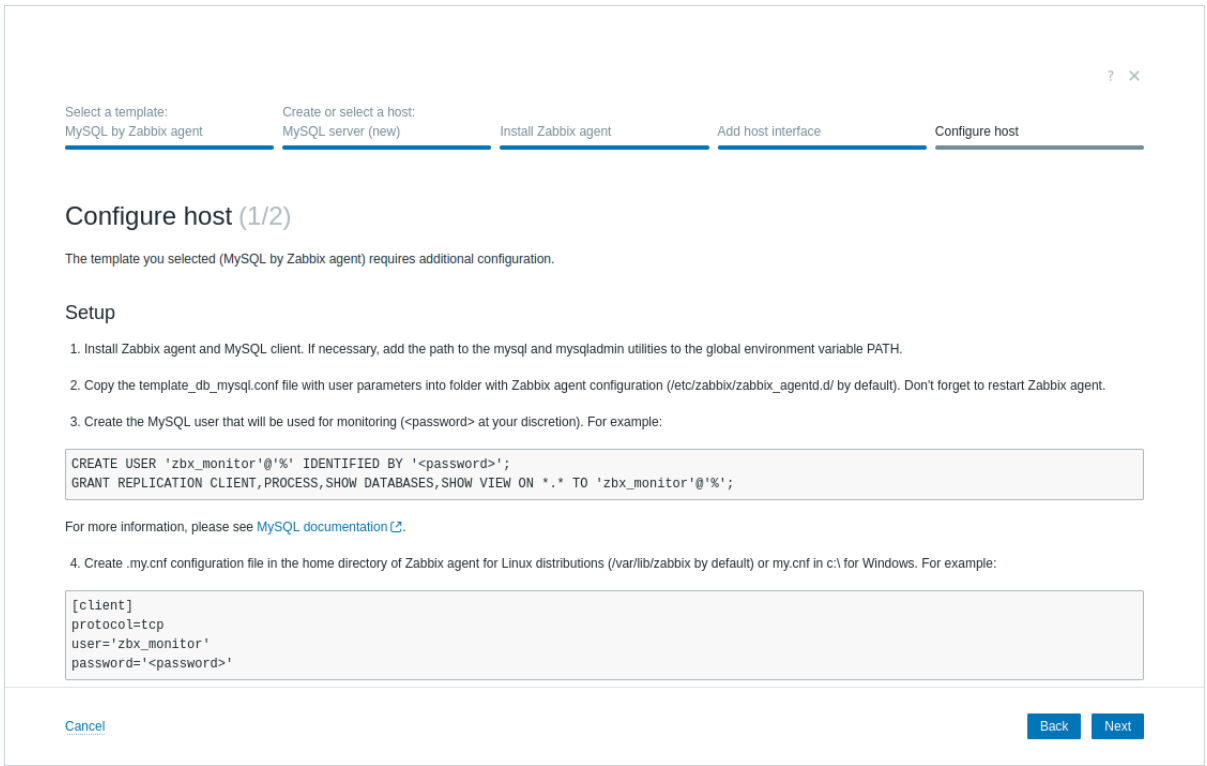
Ad esempio, puoi usare l'indirizzo agent predefinito (127.0.0.1) e la porta (10050) se il server Zabbix, Zabbix agent e il server MySQL sono tutti in esecuzione localmente sulla stessa macchina.



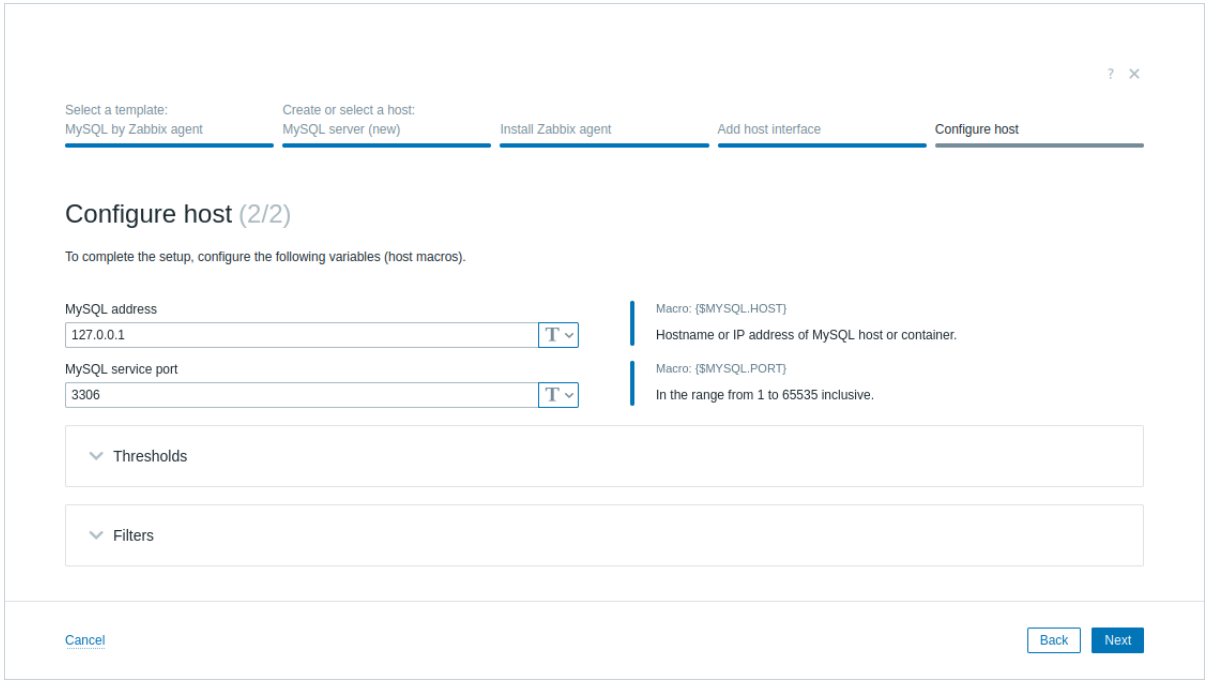
Configurare host

Alcuni template richiedono una configurazione aggiuntiva prima che l'host possa essere creato. Questa può includere:

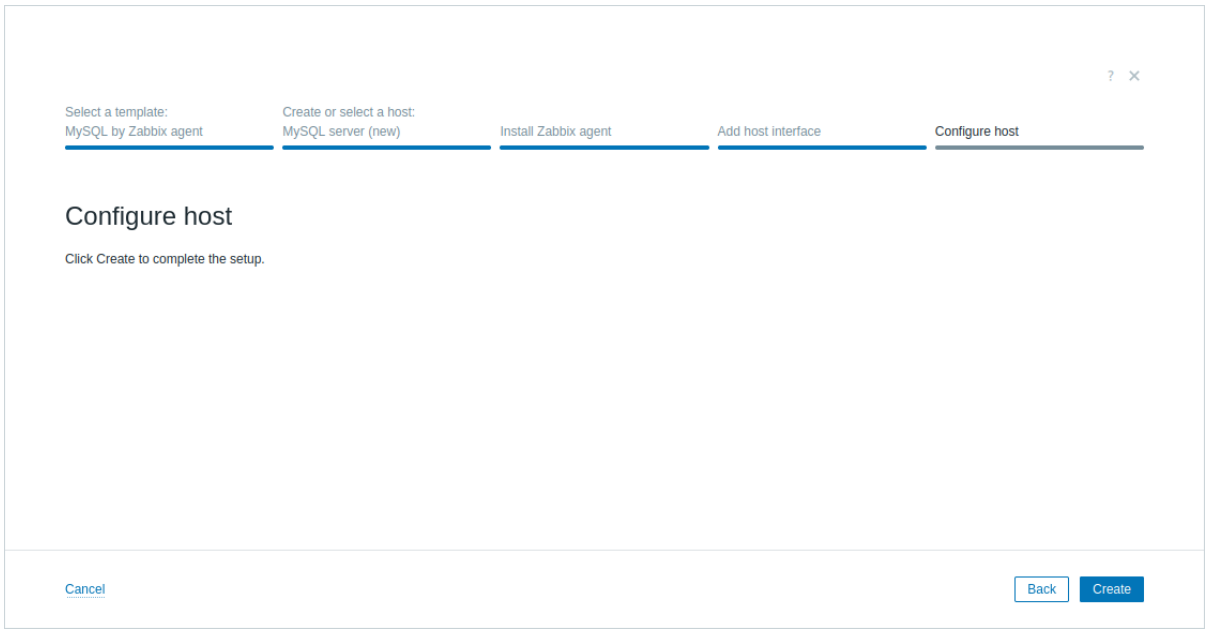
- Configurazione manuale del target di monitoraggio (ad esempio, abilitazione di servizi specifici, creazione di utenti di servizio o concessione di autorizzazioni):



- Definizione delle macro host, ovvero variabili che controllano il comportamento degli item, le impostazioni di connessione, le credenziali di autenticazione e altri parametri:



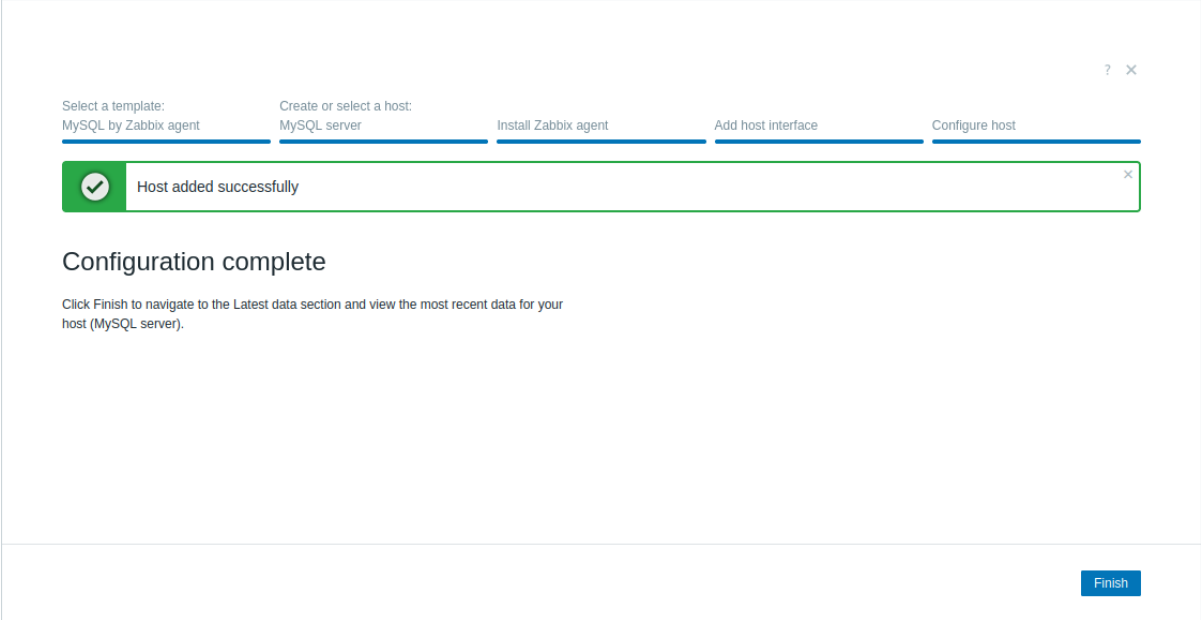
Dopo aver completato tutti i passaggi di configurazione richiesti, fare clic su *Create* per aggiungere l'host a Zabbix (oppure su *Update* per aggiornare la configurazione di un host esistente):



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.



2 Configurazione di un host

Panoramica

Per configurare un host nel frontend di Zabbix, procedi come segue:

- Vai a: *Raccolta dati* > *Host* oppure *Monitoraggio* > *Host*
- Fai clic su *Crea host* nell'angolo in alto a destra dello schermo (oppure sul nome dell'host per modificare un host esistente)
- Inserisci i parametri dell'host nel modulo

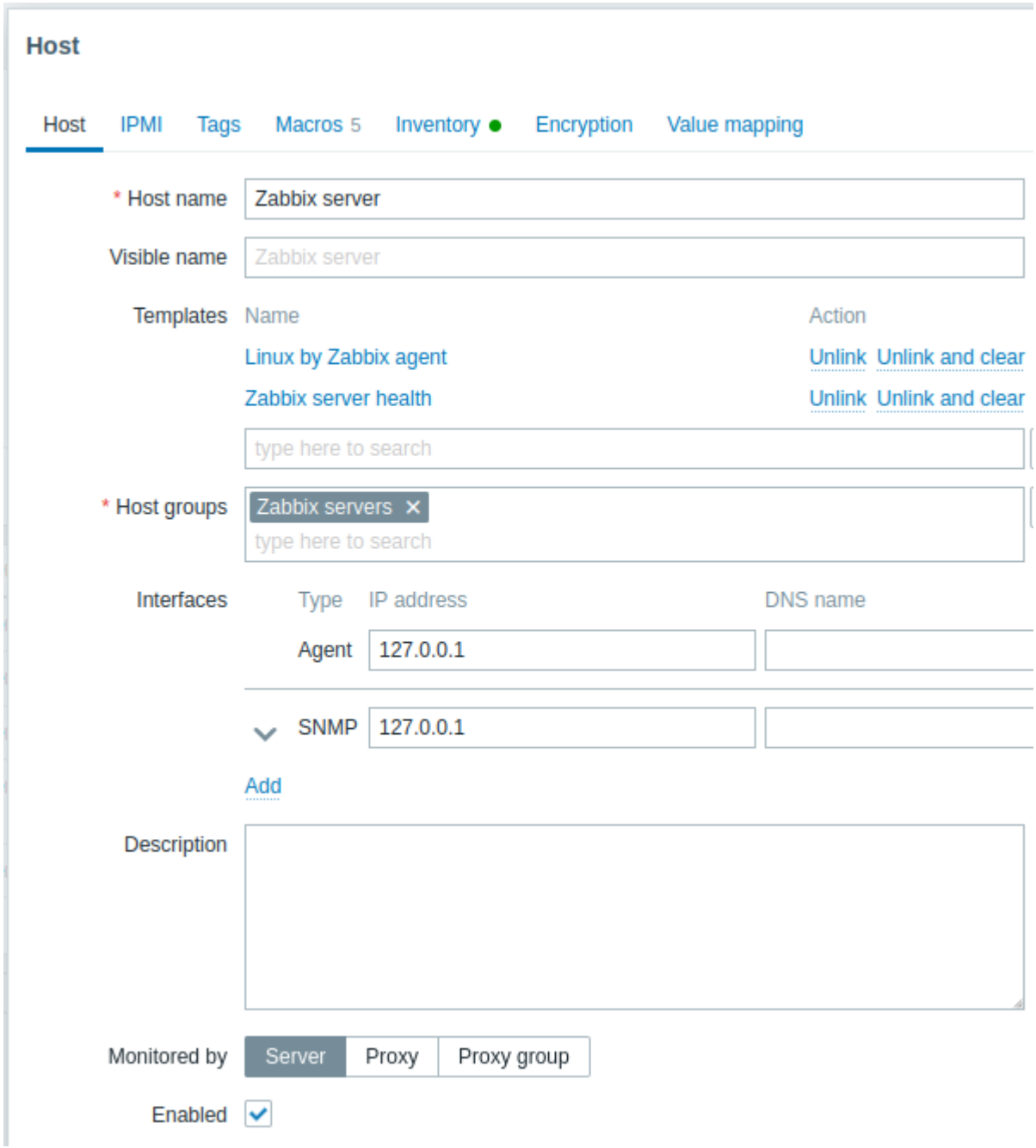
Puoi anche usare il pulsante *Clona* nel modulo di configurazione di un host esistente per creare un nuovo host. Questo host avrà tutte le proprietà dell'host esistente, inclusi i template collegati, le entità (item, trigger, ecc.) di tali template, nonché le entità collegate direttamente all'host esistente.

Tieni presente che, quando un host viene clonato, manterrà tutte le entità del template così come sono originariamente nel template. Eventuali modifiche a tali entità effettuate a livello dell'host esistente (come un intervallo dell'item modificato, un'espressione regolare modificata o prototipi aggiunti alla regola di low-level discovery) non verranno clonate nel nuovo host; saranno invece come nel template.

In alternativa, puoi usare 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 di input obbligatori sono contrassegnati con un asterisco rosso.

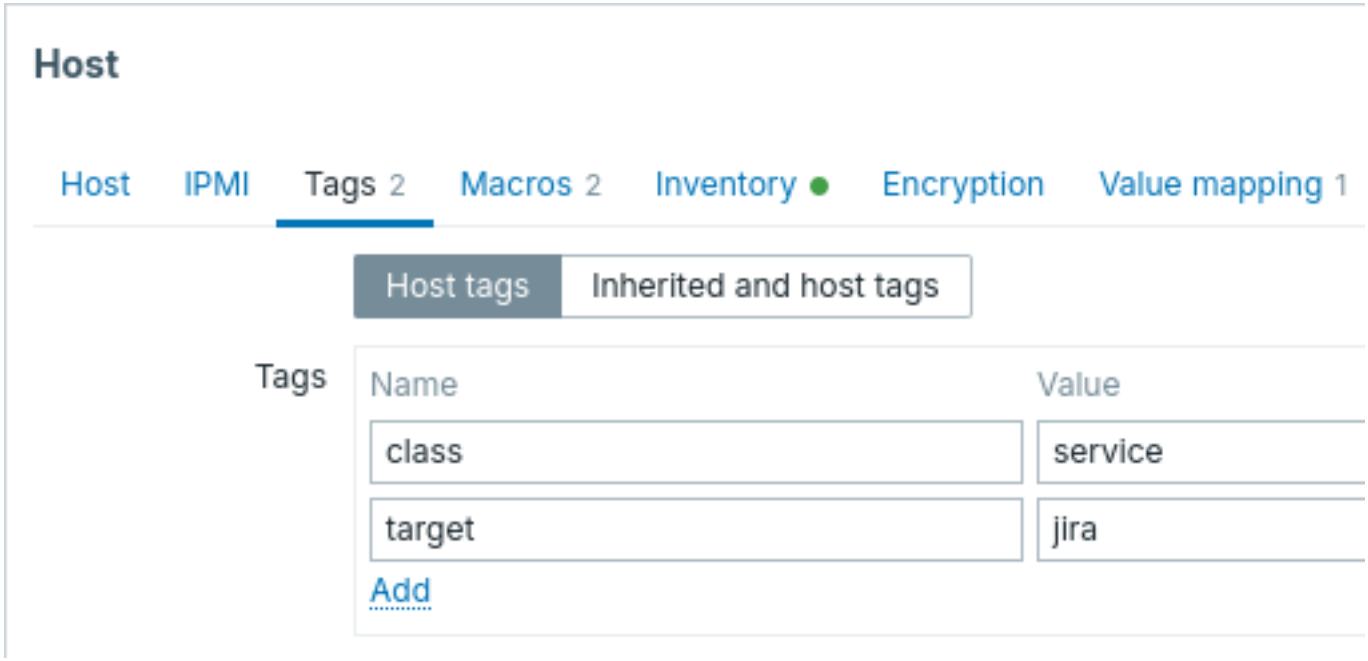
Parametro	Descrizione
Host name	Inserire un nome host univoco. Sono consentiti caratteri alfanumerici, spazi, punti, trattini e underscore. Tuttavia, gli spazi iniziali e finali non sono consentiti. *Nota:* con Zabbix agent in esecuzione sull'host che si sta configurando, il parametro *Hostname* del file di configurazione dell'agent deve avere lo stesso valore del nome host inserito qui. Il nome nel parametro è necessario per l'elaborazione dei controlli attivi.
Visible name	Inserire un nome visibile univoco per l'host. Se si imposta questo nome, sarà quello visualizzato negli elenchi, nelle mappe, ecc. invece del nome host tecnico. Questo attributo supporta UTF-8.

Parametro	Descrizione
Templates	Collegare i template all'host. Tutte le entità (item, trigger, ecc.) saranno ereditate dal template. Per collegare un nuovo template, iniziare a digitare il nome del template nel campo di input testuale. Verrà visualizzato un elenco di template corrispondenti; scorrere verso il basso per selezionare. In alternativa, è possibile fare clic su *Select* accanto al campo e selezionare i template dall'elenco in una finestra popup. Tutti i template selezionati saranno collegati all'host quando il modulo di configurazione dell'host verrà salvato o aggiornato. Per scollegare un template, utilizzare una delle due opzioni nel blocco *Linked templates*: *Unlink* - scollega il template, ma ne conserva le entità (item, trigger, ecc.); *Unlink and clear* - scollega il template e rimuove tutte le sue entità (item, trigger, ecc.). I nomi dei template elencati sono link cliccabili che portano al modulo di configurazione del template.
Host groups	Selezionare i gruppi host a cui appartiene l'host. Un host deve appartenere ad almeno un gruppo host. È possibile creare un nuovo gruppo e collegarlo all'host aggiungendo un nome di gruppo non esistente.
Interfaces	Per un host sono supportati diversi tipi di interfaccia host: *Agent*, *SNMP*, *JMX* e *IPMI*. Per impostazione predefinita non è definita alcuna interfaccia. Per aggiungere una nuova interfaccia, fare clic su *Add* nel blocco *Interfaces*, selezionare il tipo di interfaccia e inserire le informazioni *IP/DNS*, *Connect to* e *Port*. *Nota:* le interfacce utilizzate in qualsiasi item non possono essere rimosse e per esse il link *Remove* è disattivato. L'"IP" o il "DNS" di un'interfaccia SNMP viene utilizzato anche per le trap SNMP. Durante la corrispondenza, viene utilizzato solo l'"IP" o il "DNS" selezionato nell'interfaccia host. Vedere Configurazione del monitoraggio SNMP per ulteriori dettagli sulla configurazione di un'interfaccia SNMP (v1, v2 e v3).
IP address	Indirizzo IP dell'host (opzionale).
DNS name	Nome DNS dell'host (opzionale).
Connect to	Facendo clic sul pulsante corrispondente si indica a Zabbix server cosa utilizzare per recuperare i dati dagli agent: **IP** - Connessione all'indirizzo IP dell'host (consigliato) **DNS** - Connessione al nome DNS dell'host
Port	Numero di porta TCP/UDP. I valori predefiniti sono: 10050 per Zabbix agent, 161 per agent SNMP, 12345 per JMX e 623 per IPMI.
Default	Selezionare il pulsante di opzione per impostare l'interfaccia predefinita.
Description	Inserire la descrizione dell'host.
Monitored by	Selezionare se l'host è monitorato da: **Server** - l'host è monitorato da Zabbix server; **Proxy** - l'host è monitorato da un proxy standalone; **Proxy group** - l'host è monitorato da un gruppo di proxy.
Proxy	Viene visualizzato il nome del proxy assegnato (solo se Zabbix server ne ha assegnato uno dal gruppo di proxy selezionato). Questo campo viene visualizzato solo se l'host è monitorato da un gruppo di proxy.
Enabled	Quando la casella di controllo è selezionata, l'host è abilitato - pronto per il monitoraggio. Quando la casella di controllo non è selezionata, l'host è disabilitato - non monitorato: Per le richieste di dati passive avviate da Zabbix server/proxy (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 il periodo di tempo (circa 5 secondi) in cui Zabbix agent riceve 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 *Disabled*, il server/proxy rifiuterà i dati. Quando si disabilita l'host, i suoi item vengono immediatamente rimossi dalla cache della cronologia (tranne i loro ultimi valori, che vengono mantenuti per i log).

La scheda **IPMI** contiene gli attributi di gestione IPMI.

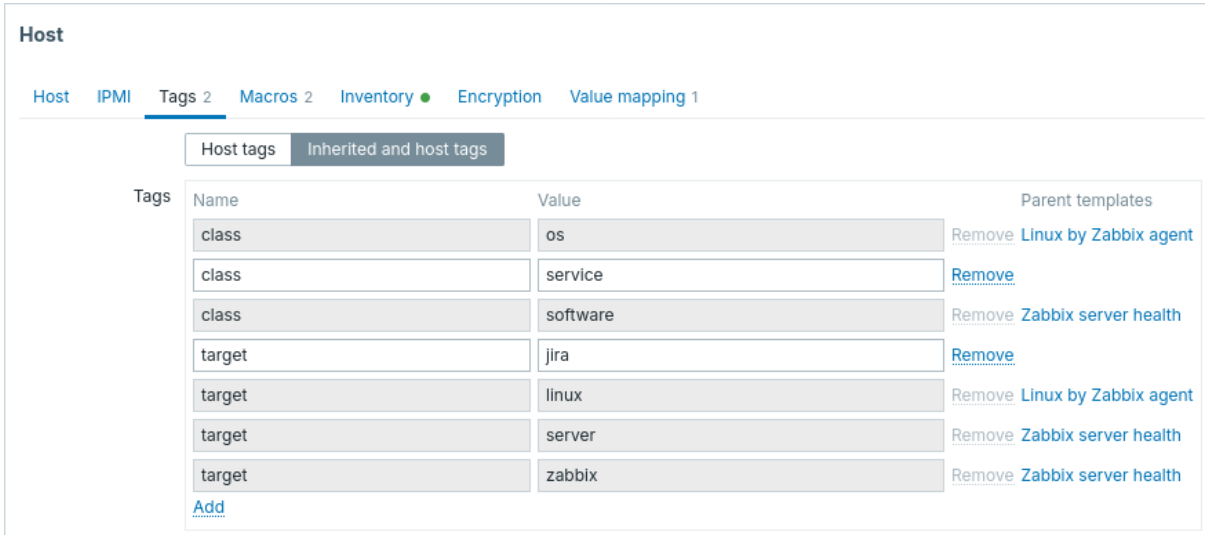
Parametro	Descrizione
Authentication algorithm	Selezionare l'algoritmo di autenticazione.
Privilege level	Selezionare il livello di privilegio.
Username	Nome utente per l'autenticazione. È possibile utilizzare macro utente.
Password	Password per l'autenticazione. È possibile utilizzare macro utente.

La scheda **Tags** consente di definire i tag a livello di host. Tutti i problemi di questo host saranno contrassegnati con i valori inseriti qui.



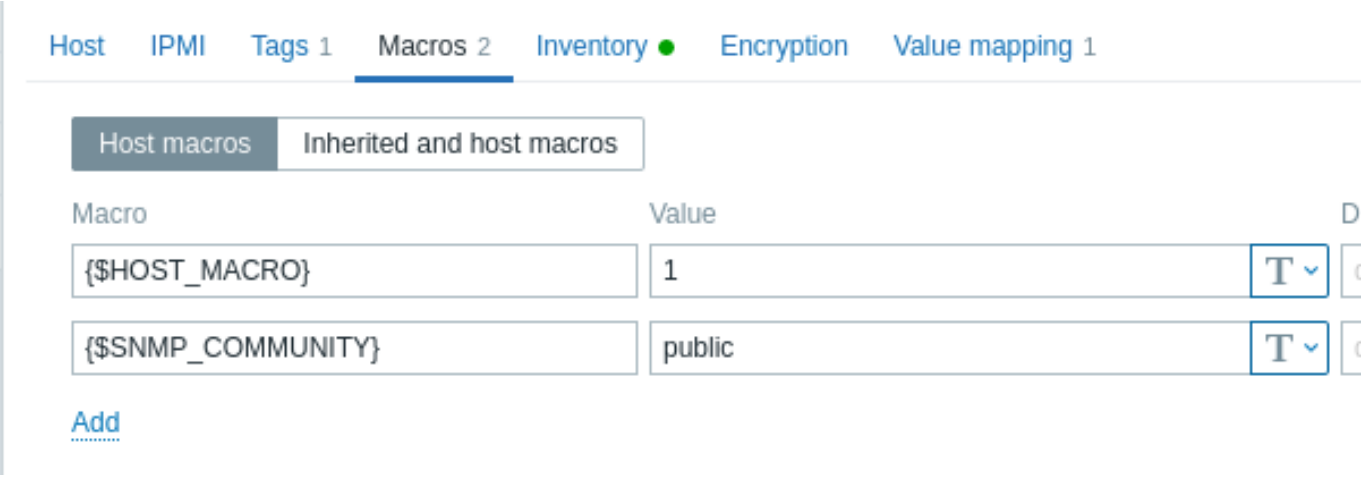
Nei tag sono supportate le macro utente, le macro {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID}.

Qui è anche possibile visualizzare i tag a livello di template selezionando l'opzione *Inherited and host tags*. Qui vengono visualizzati tutti i tag utente definiti per l'host, insieme alla loro origine.

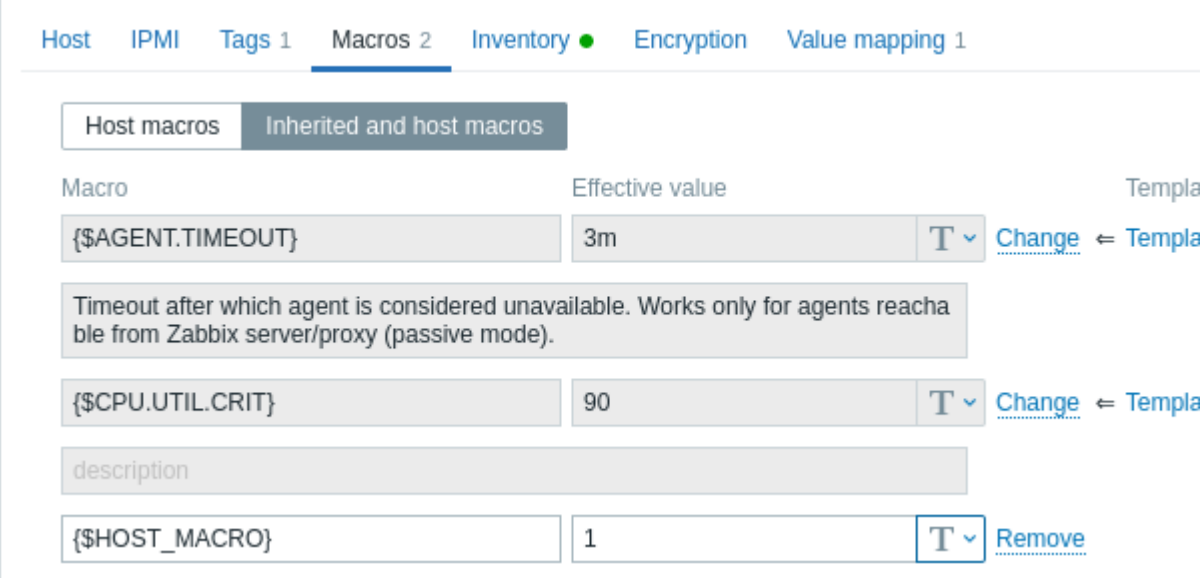


Per comodità, sono forniti link ai rispettivi template.

La scheda **Macros** consente di definire macro utente a livello di host come coppie nome-valore. Si noti che i valori delle macro possono essere mantenuti come testo normale, testo segreto o segreto Vault. È supportata anche l'aggiunta di una descrizione.

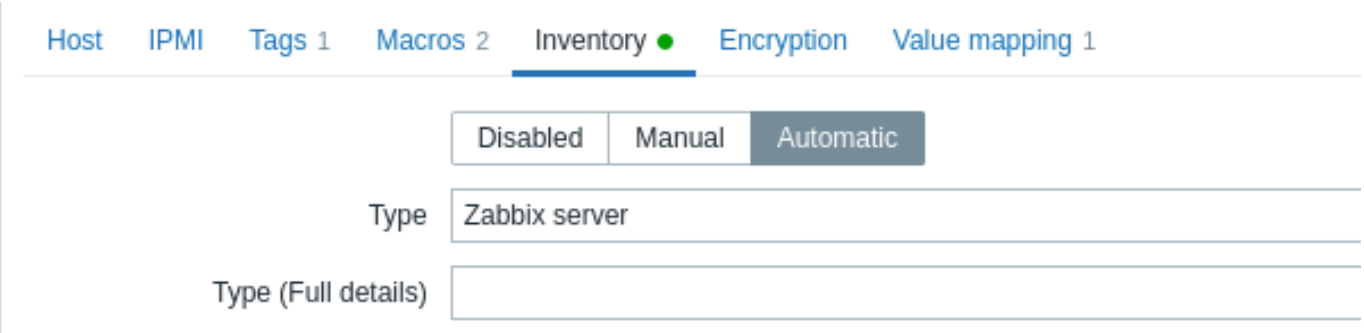


Qui è anche possibile visualizzare le macro utente a livello di template e globali selezionando l'opzione *Inherited and host macros*. Qui vengono visualizzate tutte le macro utente definite per l'host con il valore a cui vengono risolte, nonché la loro origine.



Per comodità, sono forniti link ai rispettivi template e alla configurazione delle macro globali. È anche possibile modificare una macro di template/globale a livello di host, creando di fatto una copia della macro sull'host.

La scheda **Inventory** consente di inserire manualmente le informazioni di inventory per l'host. È anche possibile scegliere di abilitare il popolamento *Automatic* dell'inventory oppure disabilitare il popolamento dell'inventory per questo host.



Se l'inventory è abilitato (manuale o automatico), viene visualizzato un punto verde accanto al nome della scheda.

Crittografia

La scheda **Crittografia** consente di richiedere connessioni crittografate con il host.

Parametro	Descrizione
Connessioni al host	Come Zabbix server o proxy si connette a Zabbix agent su un host: nessuna crittografia (predefinito), tramite PSK (chiave precondivisa) o certificato.

Parametro	Descrizione
Connessioni dal host	Seleziona quali tipi di connessioni sono consentiti dal host (cioè da Zabbix agent e Zabbix sender). È possibile selezionare contemporaneamente più tipi di connessione (utile per i test e per il passaggio a un altro tipo di connessione). Il valore predefinito è "Nessuna crittografia".
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 pensato per essere usato se l'installazione di Zabbix utilizza certificati provenienti da più CA. Se questo campo è vuoto, viene accettata qualsiasi CA.
Subject	Soggetto 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 solo un valore della stringa *Subject*. Se questo campo è vuoto, viene accettato qualsiasi certificato valido firmato dalla CA configurata.
PSK identity	Stringa identificativa della chiave precondivisa. Non inserire informazioni sensibili nell'identità PSK, poiché viene trasmessa in chiaro 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

Mappatura dei valori

La scheda **Mappatura dei valori** consente di configurare una rappresentazione dei dati degli item comprensibile per l'utente nelle mappature dei valori.

3 Configurazione di un gruppo host

Panoramica

I gruppi host vengono utilizzati per il raggruppamento logico degli host e per assegnare loro i permessi utente.

Ogni host deve avere 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 utenti, gruppi host e gruppi template. Quindi, anche se un singolo utente necessita dell'accesso a un singolo host, questo viene concesso aggiungendo l'utente a un gruppo utenti che dispone dei permessi per accedere al gruppo host contenente quell'host.

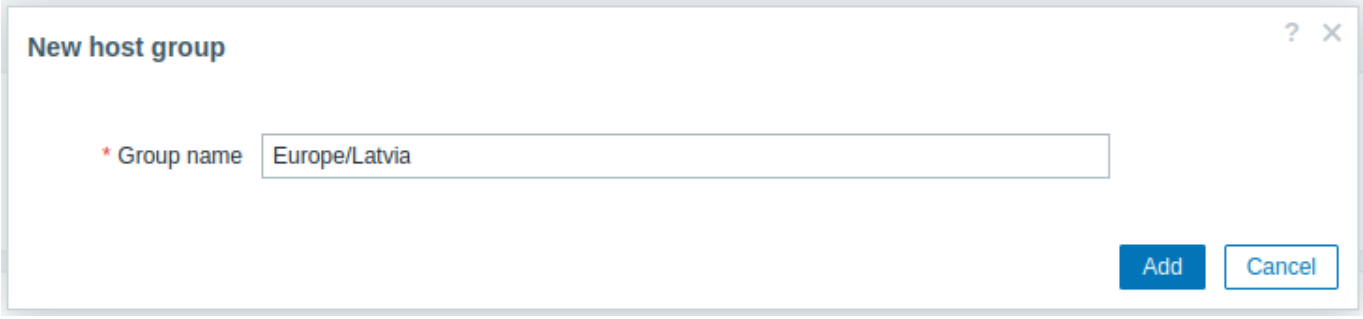
Configurazione

> **Attention:**
> Solo gli utenti Super admin possono creare gruppi host.

Esistono due opzioni per creare un gruppo host nel frontend di Zabbix.

Prima opzione:

- Vai a: *Raccolta dati* → *Gruppi host*
- Fai clic su *Crea gruppo host* nell'angolo superiore destro dello schermo
- Inserisci il nome del gruppo nel modulo



Seconda opzione: durante la configurazione di un host, inserisci un nome di gruppo non esistente nel campo di input *Gruppi host*.

Una volta creato il gruppo host, puoi fare clic sul nome del gruppo nell'elenco in *Raccolta dati* → *Gruppi host* per modificare il nome del gruppo, clonare il gruppo o eliminare il gruppo.

L'eliminazione di un gruppo host rimuove solo il gruppo logico, non gli host nel gruppo. Non è possibile eliminare un gruppo 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`.

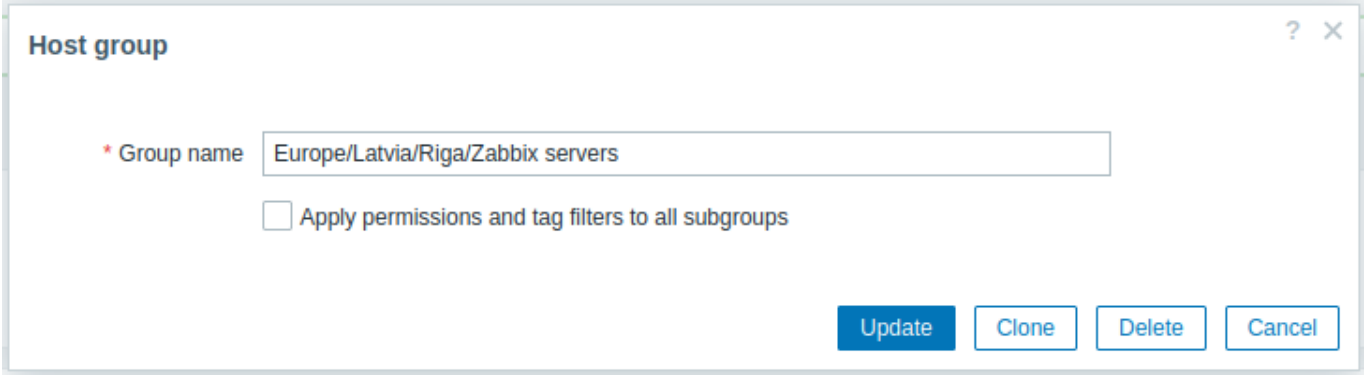
Quando si crea un sottogruppo, non è consentito utilizzare barre oblique iniziali o finali, né più barre oblique consecutive. L'escape di '/' non è supportato.

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 viene impostato alcun permesso per il padre.

Quando si modifica un qualsiasi gruppo host, è anche possibile impostare un'opzione aggiuntiva, *Applica permessi e filtri tag a tutti i sottogruppi*.



Selezionando questa casella e facendo clic su *Aggiorna* verranno applicati lo stesso livello di permessi e gli stessi filtri tag a tutti i sottogruppi attuali e futuri del gruppo host in fase di modifica.

Quindi, se a uno qualsiasi dei gruppi utenti sono stati assegnati permessi diversi per i sottogruppi del gruppo host in fase di modifica, selezionando la casella verranno concessi a tutti i sottogruppi attuali e futuri gli stessi permessi utente e gli stessi permessi basati sui tag del gruppo in fase di 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.

4 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 *Inventory* è possibile inserire dettagli quali il tipo di dispositivo, il numero di serie, la posizione, il responsabile, gli URL, ecc. - dati che andranno a popolare le informazioni di inventario.

Se nelle informazioni di inventario dell'host è incluso un URL e questo inizia con 'http' o 'https', verrà visualizzato come collegamento cliccabile nella sezione *Inventory*.

Modalità automatica

L'inventario host può anche essere popolato automaticamente. Affinché ciò funzioni, durante la configurazione di un host la modalità inventario nella scheda *Inventory* deve essere impostata su *Automatic*.

Successivamente è possibile configurare gli item host per popolare qualsiasi campo dell'inventario host con il relativo valore, indicando il campo di destinazione con il rispettivo attributo (denominato *Item will populate host inventory field*) 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 necessari i permessi di 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 inventario

La modalità inventario può essere selezionata nel modulo di configurazione del host.

La modalità inventario predefinita per i nuovi host viene selezionata in base all'impostazione *Default host inventory mode* in *Administration → General → Other*.

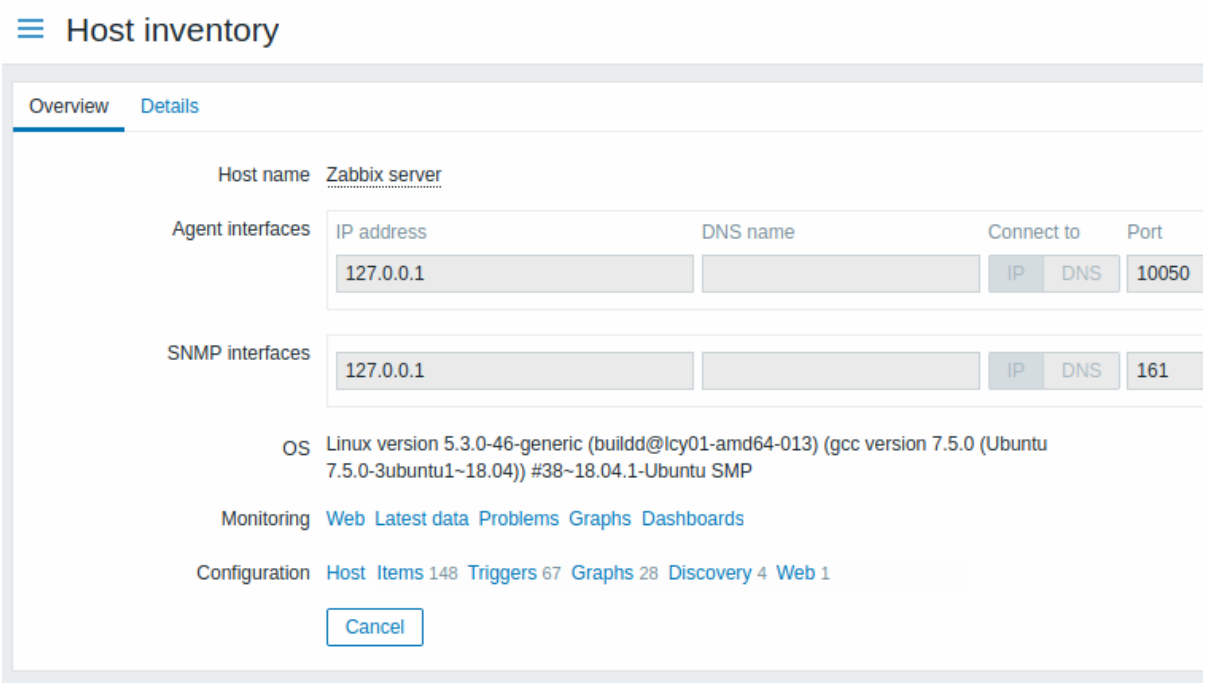
Per gli host aggiunti tramite individuazione di rete o azioni di autoregistrazione, è possibile definire un'operazione *Set host inventory mode* selezionando la modalità manuale o automatica. Questa operazione sovrascrive l'impostazione *Default host inventory mode*.

Panoramica dell'inventario

I dettagli di tutti i dati di inventario esistenti sono disponibili nel menu *Inventory*.

In *Inventory → Overview* è possibile ottenere il conteggio degli host in base a vari campi dell'inventario.

In *Inventory → Hosts* è possibile vedere tutti gli host che hanno informazioni di inventario. Facendo clic sul nome dell'host verranno mostrati 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 dell'inventario del sistema operativo dell'host (se definito).
Hardware	Campo dell'inventario dell'hardware dell'host (se definito).
Software	Campo dell'inventario del software dell'host (se definito).
Description	Descrizione dell'host.
Monitoring	Collegamenti alle sezioni di monitoraggio con i dati per questo host: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Collegamenti alle sezioni di configurazione per questo host: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Il numero di entità configurate è elencato dopo ciascun collegamento.

La scheda **Details** mostra tutti i campi dell'inventario che sono valorizzati (non vuoti).

Macro di inventario

Sono disponibili macro di inventario del host {INVENTORY.*} da utilizzare nelle notifiche, ad esempio:

"Il server in {INVENTORY.LOCATION1} ha un problema, il responsabile è {INVENTORY.CONTACT1}, numero di telefono {INVENTORY.POC.PRIMARY.PHONE.A1}."

Per maggiori dettagli, vedere la pagina Macro supportate.

5 Aggiornamento di massa

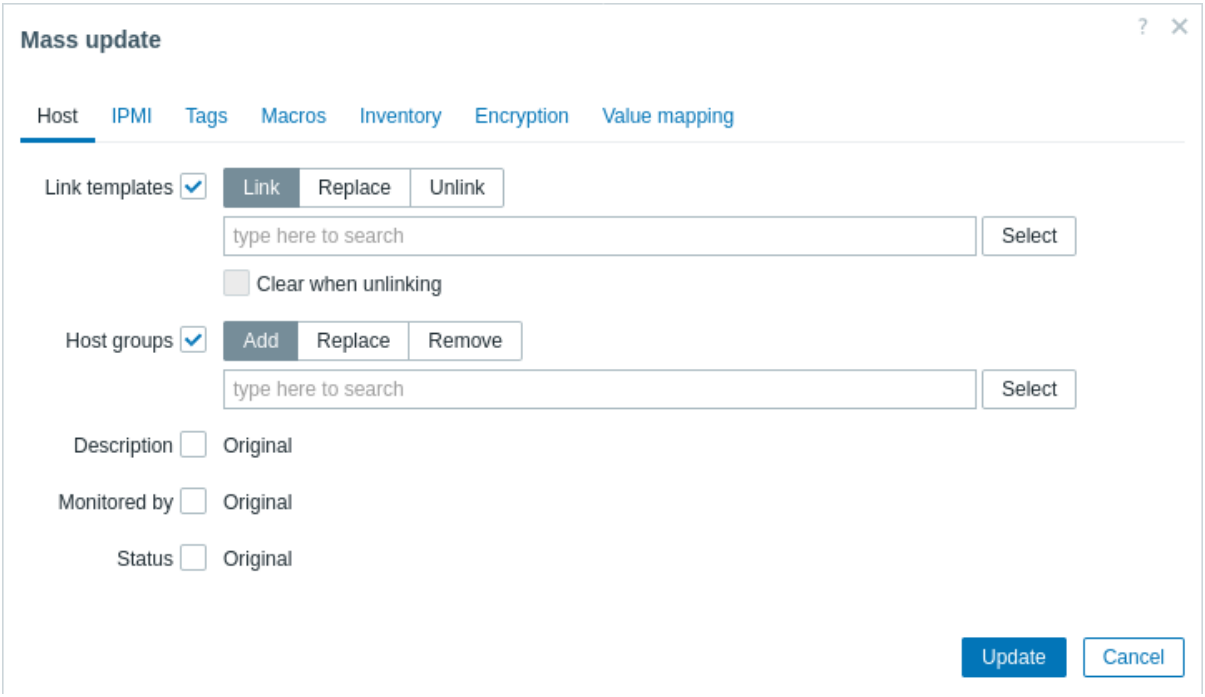
Panoramica

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 in massa alcuni host, procedere come segue:

- Selezionare le caselle di controllo davanti agli host che si desidera aggiornare nell'elenco degli host
- Fare clic su *Mass update* sotto l'elenco
- Passare alla scheda con gli attributi richiesti (*Host*, *IPMI*, *Tags*, *Macros*, *Inventory*, *Encryption* o *Value mapping*)
- Selezionare le caselle di controllo di qualsiasi attributo da aggiornare e inserire un nuovo valore per essi



Sono disponibili le seguenti opzioni quando si seleziona il pulsante corrispondente per l'aggiornamento del collegamento dei **template**:

- *Link* - specifica quali template aggiuntivi collegare
- *Replace* - specifica quali template collegare scollegando al contempo qualsiasi template che era stato collegato agli host in precedenza
- *Unlink* - specifica quali template scollegare

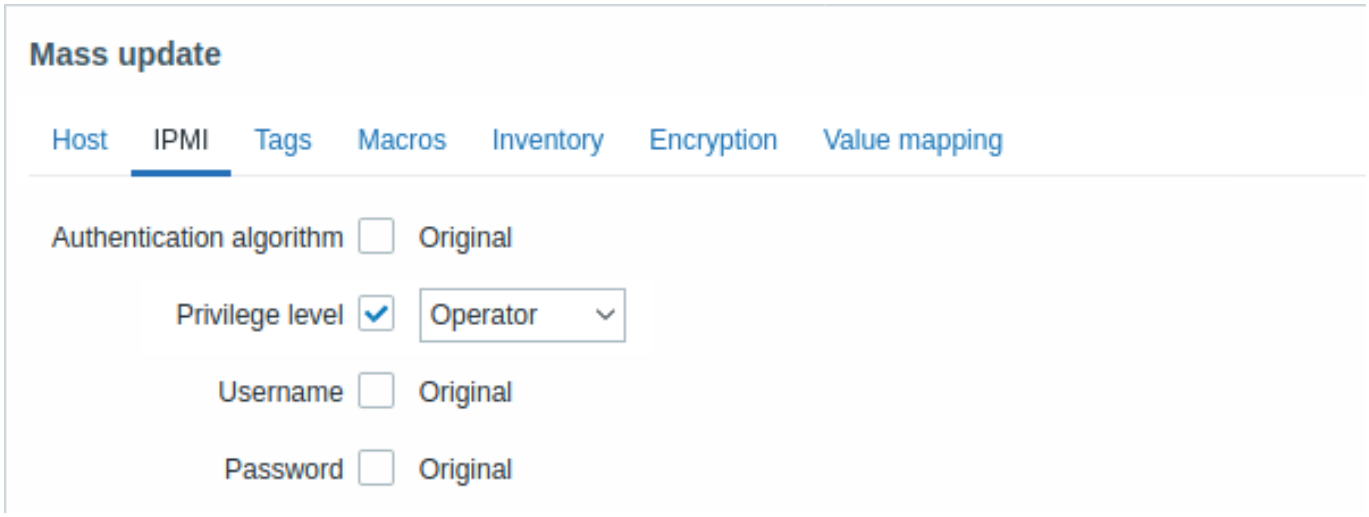
Per specificare i template da collegare/scollegare, iniziare a digitare il nome del template nel campo di completamento automatico finché non appare un menu a discesa con i template corrispondenti. È sufficiente scorrere verso il basso per selezionare il template richiesto.

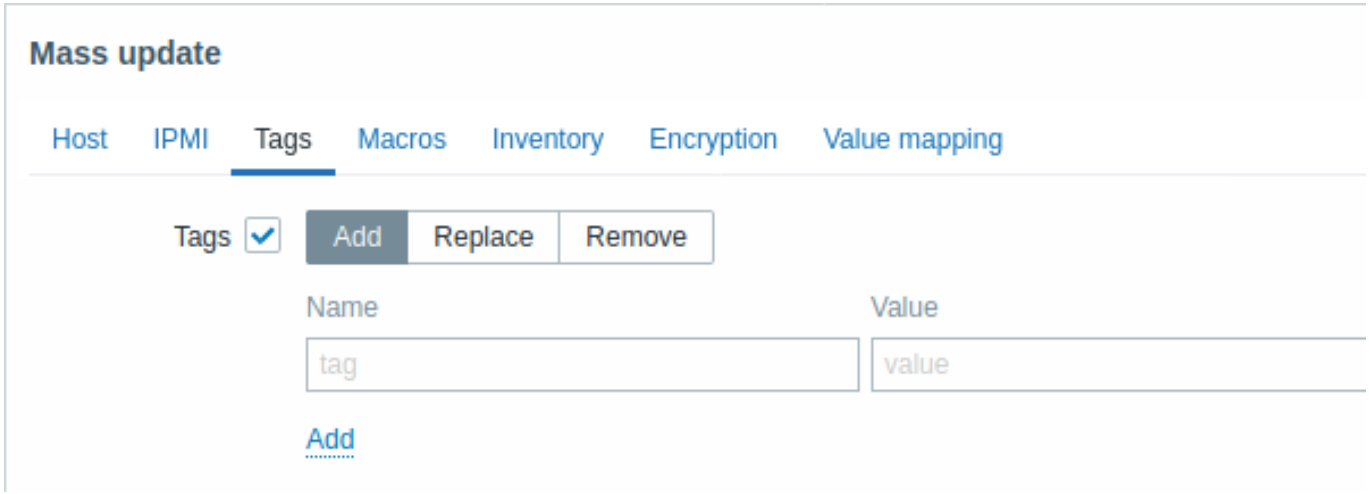
L'opzione *Clear when unlinking* consentirà non solo di scollegare eventuali template collegati in precedenza, ma anche di rimuovere tutti gli elementi ereditati da essi (item, trigger, ecc.).

Sono disponibili le seguenti opzioni quando si seleziona il pulsante corrispondente per l'aggiornamento del **gruppo host**:

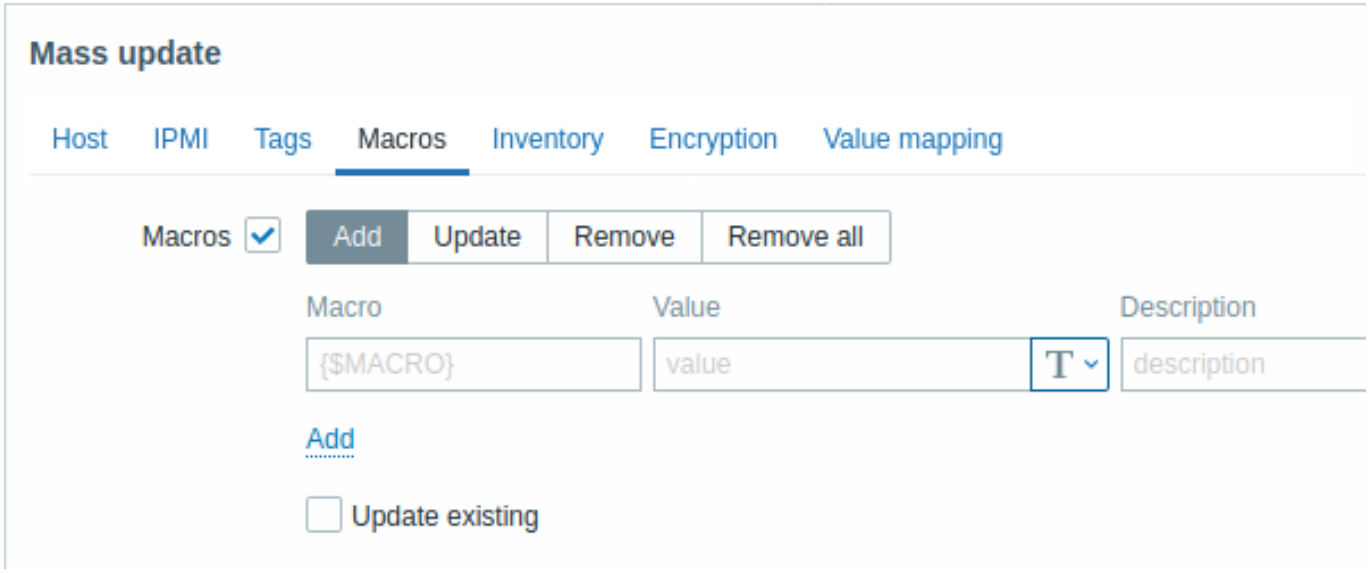
- *Add* - consente di specificare gruppi host aggiuntivi tra quelli esistenti oppure di inserire gruppi host completamente nuovi per gli host
- *Replace* - rimuoverà l'host da tutti i gruppi host esistenti e li sostituirà con quello/quelli specificati in questo campo (gruppi host esistenti o nuovi)
- *Remove* - rimuoverà gruppi host specifici dagli host

Questi campi supportano il completamento automatico: iniziando a digitare in essi viene proposto un menu a discesa con i gruppi host corrispondenti. Se il gruppo host è nuovo, apparirà anch'esso nel menu a discesa e sarà indicato da *(new)* dopo la stringa. È sufficiente scorrere verso il basso per selezionarlo.



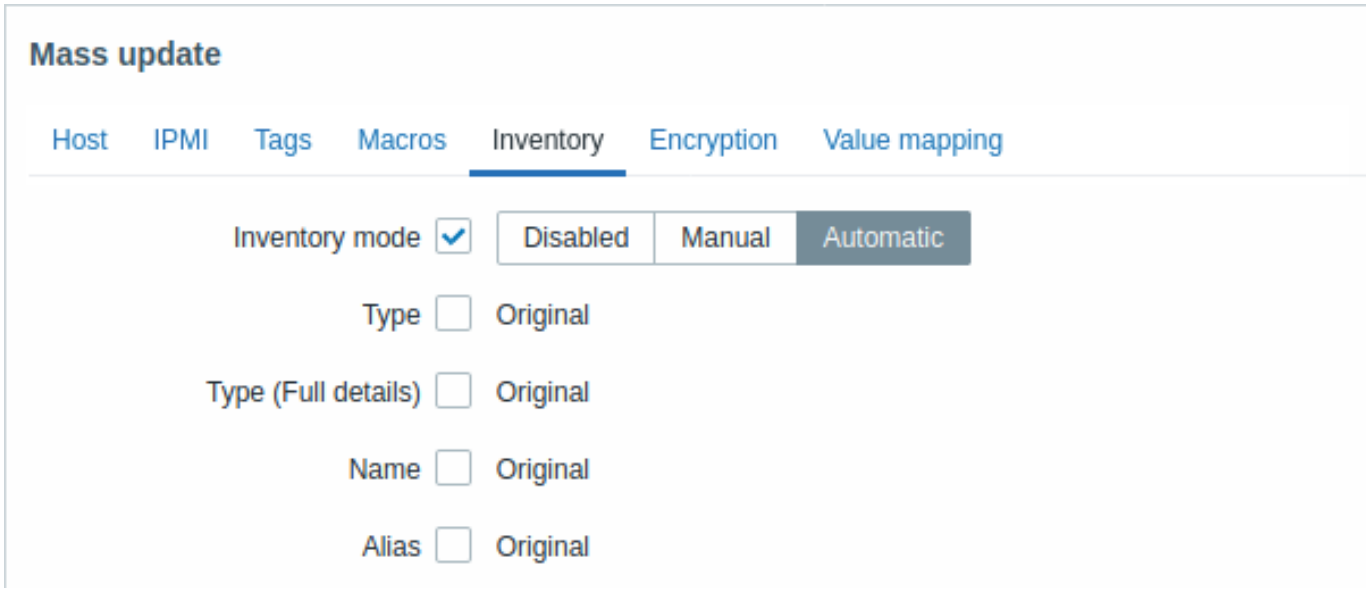


Sono supportate nelle tags le macro utente, le macro {INVENTORY.*}, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} e {HOST.ID}. Si noti che le tag con lo stesso nome ma valori diversi non sono considerate "duplicate" e possono essere aggiunte allo stesso host.

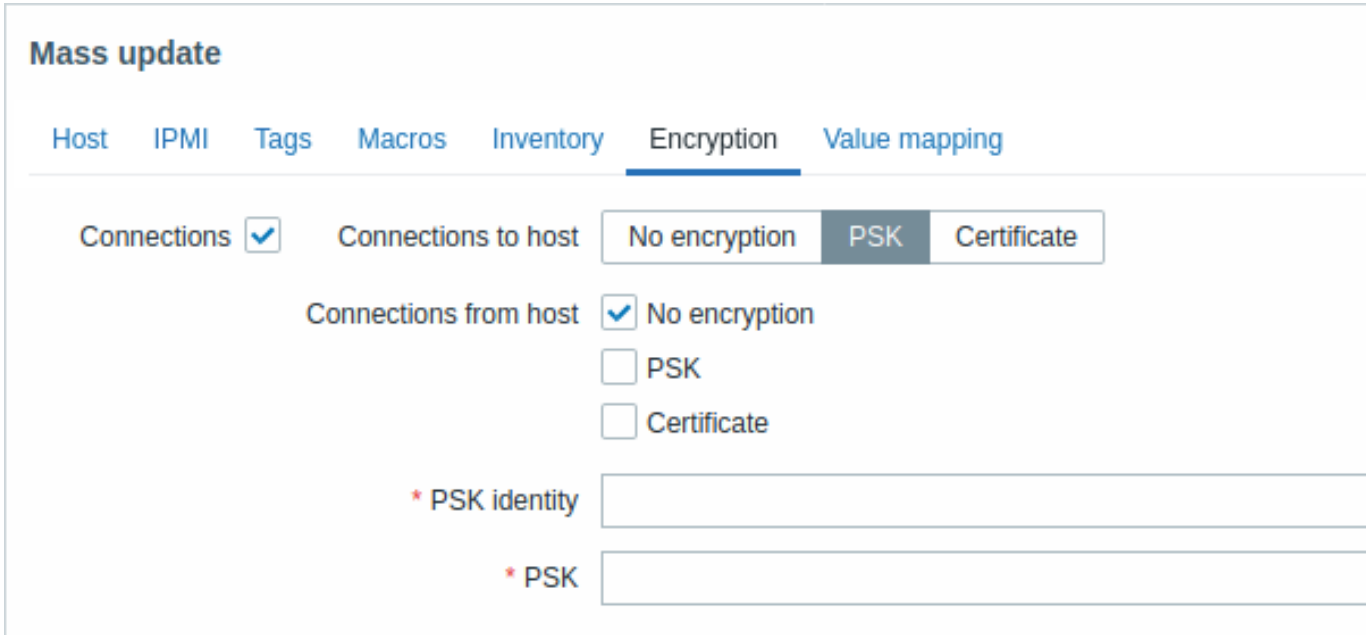


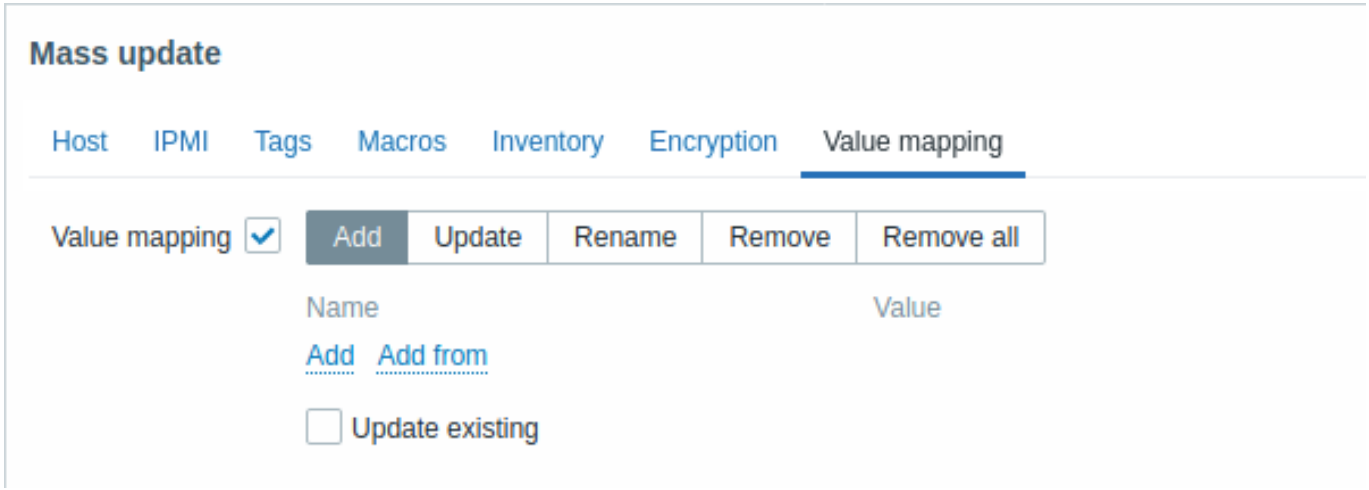
Sono disponibili le seguenti opzioni quando si seleziona il pulsante corrispondente per l'aggiornamento delle macro:

- *Add* - consente di specificare macro utente aggiuntive per gli host. Se la casella di controllo *Update existing* è selezionata, il valore, il tipo e la descrizione per il nome macro specificato verranno aggiornati. Se non è selezionata, se una macro con quel nome esiste già sull'host o sugli host, non verrà aggiornata.
- *Update* - sostituirà valori, tipi e descrizioni delle macro specificate in questo elenco. Se la casella di controllo *Add missing* è selezionata, una macro che in precedenza non esisteva su un host verrà aggiunta come nuova macro. Se non è selezionata, verranno aggiornate solo le macro che esistono già su un host.
- *Remove* - rimuoverà le macro specificate dagli host. Se la casella *Except selected* è selezionata, verranno rimosse tutte le macro tranne quelle specificate nell'elenco. Se non è selezionata, verranno rimosse solo le macro specificate nell'elenco.
- *Remove all* - rimuoverà tutte le macro utente dagli host. Se la casella *I confirm to remove all macros* non è selezionata, si aprirà una nuova finestra popup che chiederà di confermare la rimozione di tutte le macro.



Per poter aggiornare in massa i campi dell'inventario, la *Inventory mode* deve essere impostata su 'Manual' o 'Automatic'.





Per l'aggiornamento delle mappe di valori sono disponibili pulsanti con le seguenti opzioni:

- *Add* - aggiunge mappe di valori agli host. Se si seleziona *Update existing*, tutte le proprietà della mappa di valori con questo nome verranno aggiornate. In caso contrario, se una mappa di valori con quel nome esiste già, non verrà aggiornata.
- *Update* - aggiorna le mappe di valori esistenti. Se si seleziona *Add missing*, una mappa di valori che in precedenza non esisteva su un host verrà aggiunta come nuova mappa di valori. In caso contrario verranno aggiornate solo le mappe di valori che esistono già su un host.
- *Rename* - assegna un nuovo nome a una mappa di valori esistente
- *Remove* - rimuove le mappe di valori specificate dagli host. Se si seleziona *Except selected*, tutte le mappe di valori verranno rimosse **tranne** quelle specificate.
- *Remove all* - rimuove tutte le mappe di valori dagli host. Se la casella *I confirm to remove all value maps* non è selezionata, si aprirà una nuova finestra popup che chiederà di confermare la rimozione.

Una volta completate tutte le modifiche richieste, fare clic su *Update*. Gli attributi verranno aggiornati di conseguenza per tutti gli host selezionati.

2 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.

1 Creare un oggetto

Panoramica

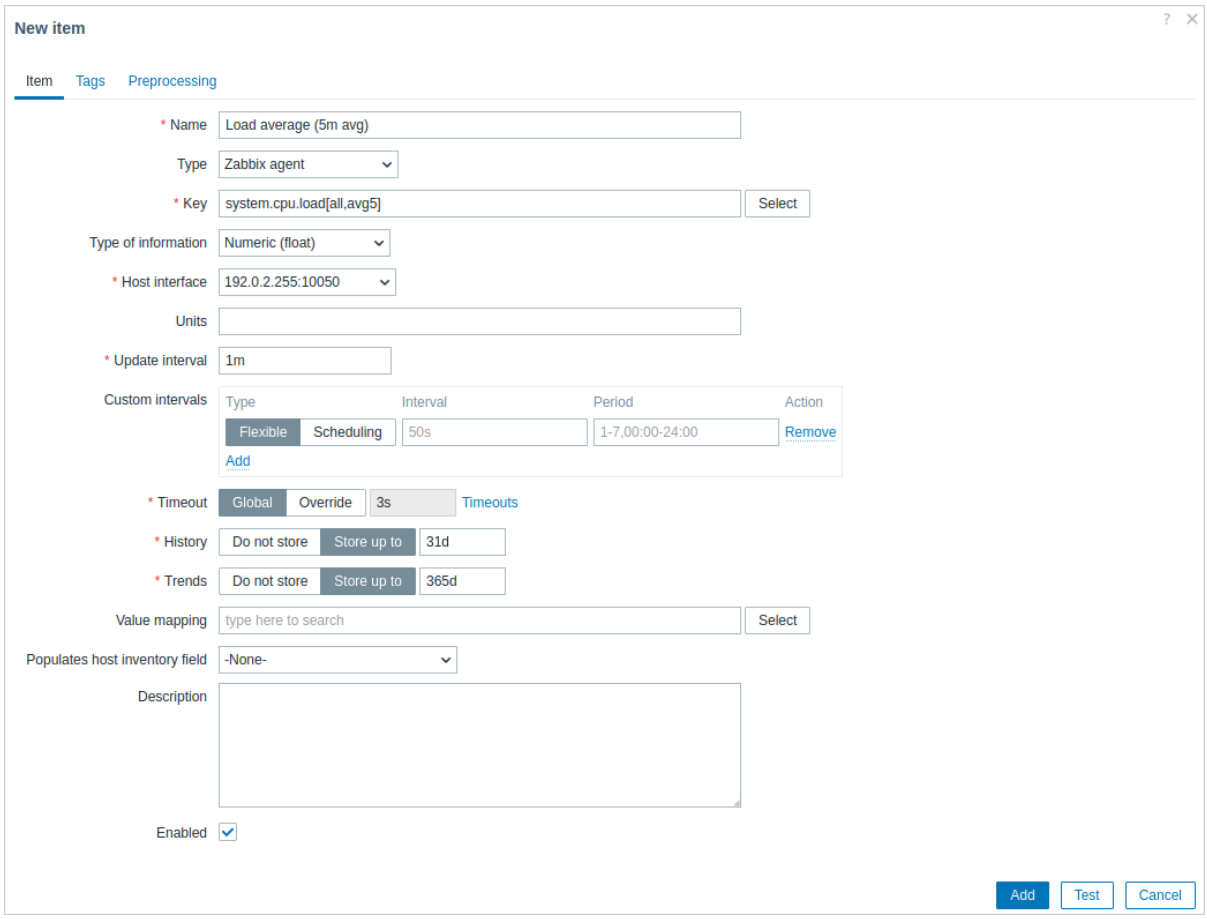
Per creare un item nel frontend di Zabbix, procedi come segue:

- Vai a: *Raccolta dati* > *Host*
- Fai clic su *Item* nella riga dell'host
- Fai clic su *Crea item* nell'angolo superiore destro dello schermo
- Inserisci i parametri dell'item nel modulo

Puoi anche creare un item aprendone uno esistente, premendo il pulsante *Clona* e quindi salvandolo con un nome diverso.

Configurazione

La scheda **Item** contiene gli attributi generali dell'item.



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

Parametro	Descrizione
Name	Nome dell'item. Sono supportate le macro utente.
Type	Tipo di item. Vedere le singole sezioni dei tipi di item.

Parametro	Descrizione
Key	Chiave dell'item (fino a 2048 caratteri). Le chiavi di item supportate sono disponibili nelle singole sezioni dei 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 corretto formato della chiave.
Type of information	Il tipo di dato utilizzato per validare il valore dell'item e memorizzarlo nel database, entrambi 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 di log (timestamp, origine, gravità, ID evento log); **Text** - testo lungo; **Binary** - numero binario (supportato solo per item dipendenti); **JSON** - dati JSON strutturati, memorizzati nativamente nel database (non supportati per gli item calcolati). Gli item che restituiscono valori con tipo di dato binario o JSON non sono supportati nelle formule degli item calcolati o nelle espressioni dei trigger. Vedere limiti dei dati dell'item per i limiti di archiviazione per tipo di dato e backend del database. Per gli item che restituiscono dati in un unico formato specifico, viene selezionato automaticamente un tipo di dato corrispondente; ad esempio, `system.cpu.load` e `system.cpu.util` restituiscono entrambi un numero in virgola mobile a 64 bit, quindi *Numeric (float)* viene selezionato automaticamente.
Host interface	Selezionare l'interfaccia host. Questo campo è disponibile quando si modifica un item a livello host.
Units	Se viene impostato un simbolo di unità, Zabbix applica una post-elaborazione al valore dell'item ricevuto e lo visualizza con il suffisso di unità specificato. Simboli di unità supportati con formattazione speciale (ed esempi di valore dell'item ricevuto → valore visualizzato): **B** - byte (1024 → 1 KB) **Bps** - byte al secondo (1024 → 1 KBps) **s** - secondi, visualizzati usando fino a tre delle più grandi unità di tempo non nulle (881764 → 10d 4h 56m) **uptime** - tempo trascorso in hh:mm:ss oppure N giorni, hh:mm:ss (881764 → 10 giorni, 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 dell'item ricevuto deve essere *Numeric (unsigned)*. Per altre unità (come Hz, W, ecc.), se il valore ricevuto supera 1000, viene diviso per 1000 e visualizzato con un prefisso corrispondente (5000 → 5 KHz, 881764 → 881.76 KW). Se il simbolo di 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 di unità e sulla conversione delle unità, vedere Suffissi dei valori degli item.
Update interval	Recupera un nuovo valore per questo item ogni N secondi. L'intervallo massimo di aggiornamento consentito è 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. Più macro in un campo o macro mescolate con testo non sono supportate. *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 durante la durata dell'intervallo personalizzato. *Nota* che il primo controllo dell'item dopo che l'item è diventato attivo o dopo una 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 pianificato o flessibile e che *Update interval* sia impostato su 0. Un item passivo esistente può essere interrogato immediatamente per ottenere un valore premendo il pulsante *Execute now*.

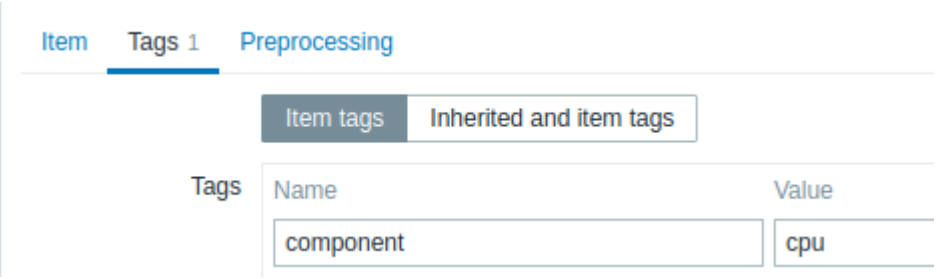
Parametro	Descrizione
Custom intervals	È possibile creare regole personalizzate per il controllo dell'item: **Flexible** - crea un'eccezione a *Update interval* (intervallo con frequenza diversa). **Scheduling** - crea una pianificazione di polling personalizzata. Per informazioni dettagliate vedere 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. Più macro in un campo o macro mescolate con testo non sono supportate.
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 globale/del proxy (visualizzato nel campo *Timeout* disattivato). **Override** - viene utilizzato un timeout personalizzato (impostato nel campo *Timeout*; intervallo consentito: 1 - 600s). Sono supportati i suffissi temporali, ad es. 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). Si noti che il collegamento *Timeouts* è visibile solo agli utenti di tipo *Super admin* con autorizzazioni per le sezioni del frontend *Administration* > *General* o *Administration* > *Proxies*.
History	Selezionare una delle seguenti opzioni: **Do not store** - la cronologia dell'item non viene memorizzata. Utile per gli item master se solo gli item dipendenti devono mantenere la cronologia. Questa impostazione non può essere sovrascritta dalle impostazioni globali dell'housekeeper. **Store up to** - specifica la durata di conservazione della cronologia dettagliata nel database (da 1 ora a 25 anni). I dati più vecchi verranno rimossi dall'housekeeper. Memorizzato 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 su di essa, viene visualizzato un messaggio di avviso, ad es. *Overridden by global housekeeper settings (1d)*. Si consiglia di mantenere i valori registrati per il minor tempo possibile per ridurre la dimensione della cronologia dei valori nel database. Invece di memorizzare una lunga cronologia di valori, è possibile conservare più a lungo i dati di trend. Vedere anche Cronologia e trend.
Trends	Selezionare una delle seguenti opzioni: **Do not store** - i trend non vengono memorizzati. Questa impostazione non può essere sovrascritta dalle impostazioni globali dell'housekeeper. **Store up to** - specifica la durata di conservazione della cronologia aggregata (min, max, media, conteggio orari) nel database (da 1 giorno a 25 anni). I dati più vecchi verranno rimossi dall'housekeeper. Memorizzato 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 info. Posizionando il mouse su di essa, viene visualizzato un messaggio di avviso, ad es. *Overridden by global housekeeper settings (7d)*. *Nota:* la conservazione dei trend non è disponibile per dati non numerici - character, log e text. Vedere anche Cronologia e trend.
Value mapping	Applica una mappatura dei valori a questo item. La mappatura dei valori non modifica i valori ricevuti, serve solo per la visualizzazione dei dati. Funziona con item *Numeric(unsigned)*, *Numeric(float)* e *Character*. Ad esempio, "Windows service states".

Parametro	Descrizione
Log time format	Disponibile solo per 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 a 0 nel tempo Unix, corrispondente al 1 gennaio 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 carattere 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 host che verrà popolato dal valore dell'item. Questo funzionerà se la compilazione automatica dell'inventario è abilitata per l'host e con item *Numeric(unsigned)*, *Numeric(float)*, *Character* e *Text*.
Description	Inserire una descrizione dell'item. Sono supportate le macro utente.
Enabled	Selezionare la casella per abilitare l'item in modo che venga elaborato. Quando si disabilita l'item, esso viene immediatamente rimosso dalla cache della cronologia (tranne il suo ultimo valore, che viene mantenuto per i log).
Latest data	Fare clic sul collegamento per visualizzare gli ultimi dati 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 host un item esistente a livello di template, un certo numero di campi è di sola lettura. È possibile utilizzare il collegamento nell'intestazione del modulo per passare al livello template e modificarli lì, tenendo presente che le modifiche a livello di template cambieranno l'item per tutti gli host a cui il template è collegato.

La scheda **Tags** consente di definire tag a livello di item.



Pre-elaborazione dei valori degli item

La scheda **Preprocessing** consente di definire regole di trasformazione per i valori ricevuti.

Test degli item

Attention:
Per eseguire il test degli item, assicurarsi che l'ora di sistema sul server e sul proxy sia sincronizzata. Nel caso in cui l'ora del server sia indietro, il test degli item potrebbe restituire il messaggio di errore "The task has been expired." L'impostazione di fusi orari diversi sul server e sul proxy, tuttavia, non influirà sul risultato del test.

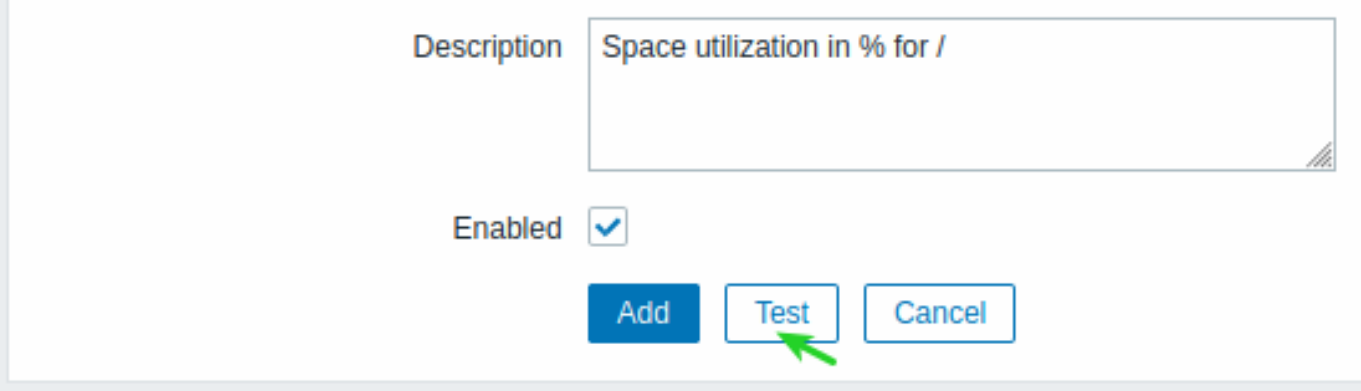
È possibile testare un item e, se configurato correttamente, ottenere in risposta un valore reale. Il test può essere eseguito anche prima che un item venga salvato.

Il test è disponibile per gli item di host e template, i prototipi di item e le regole di low-level discovery. Il test non è disponibile per gli item attivi.

Il test degli item è disponibile per i seguenti tipi di item passivi:

- Zabbix agent
- agent SNMP (v1, v2, v3)
- agent IPMI
- controlli SSH
- controlli Telnet
- agent JMX
- controlli semplici (eccetto gli item `icmpping*`, `vmware.*`)
- Zabbix internal
- item calcolati
- controlli esterni
- monitor database
- agent HTTP
- script
- browser

Per testare un item, fare clic sul pulsante *Test* nella parte inferiore del modulo di configurazione dell'item. Si noti che il pulsante *Test* sarà disabilitato per gli item che non possono essere testati (come i controlli attivi e i controlli semplici esclusi).

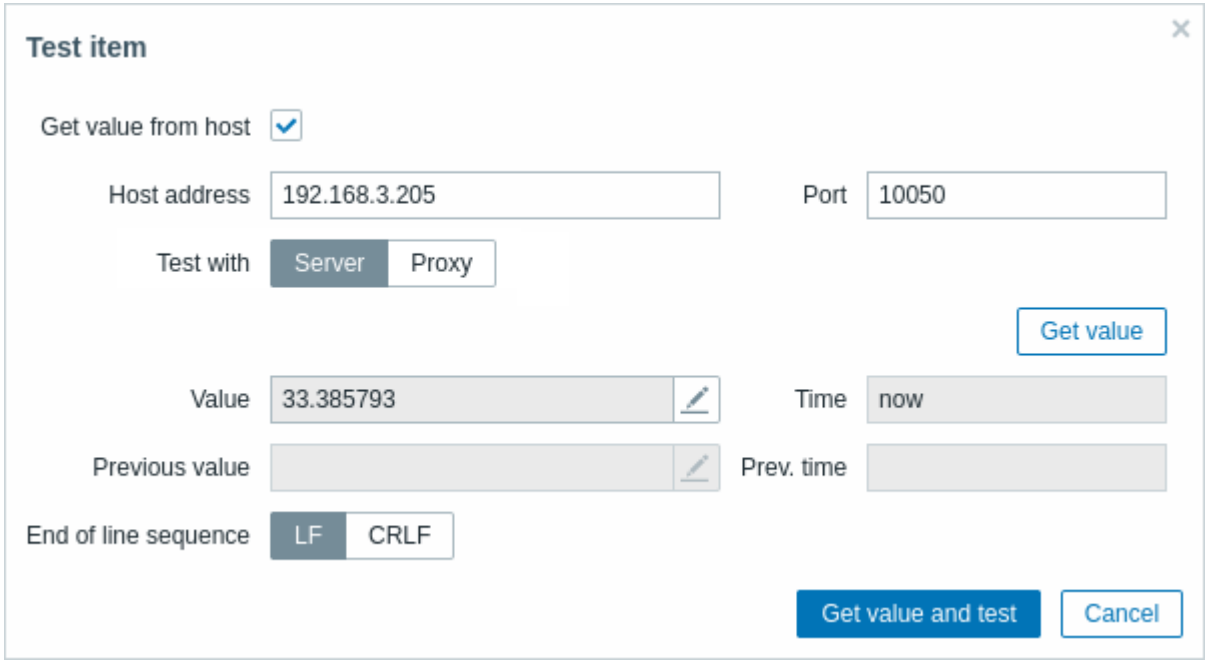


Il modulo di test dell'item contiene campi per i parametri host richiesti (indirizzo host, porta, test con server/proxy (nome proxy)) e dettagli specifici dell'item (come la community SNMPv2 o le credenziali di sicurezza SNMPv3). Questi campi sono sensibili al contesto:

- 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 di template.
- I valori delle macro in testo semplice vengono risolti.
- I campi in cui il valore (o parte del valore) è una macro segreta o 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 necessari nel contesto del tipo di item (ad esempio, il campo indirizzo host e il campo proxy sono disabilitati per gli item calcolati)

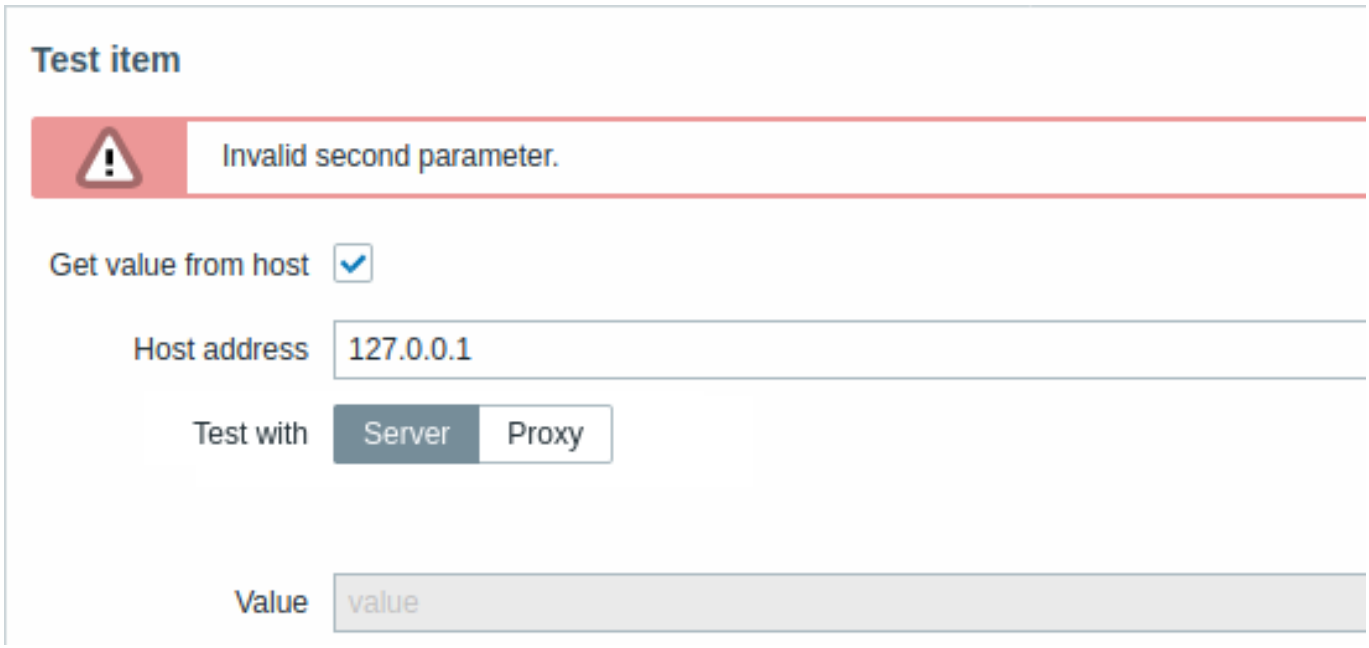
Per testare l'item, fare clic su *Get value*. Se il valore viene recuperato correttamente, riempirà il campo *Value*, spostando il valore corrente (se presente) nel campo *Previous value*, calcolando anche il campo *Prev. time*, cioè la differenza di tempo tra i due valori (clic), e tentando di rilevare una sequenza EOL e passare a CRLF se viene rilevato "\n\r" nel valore recuperato.

Fare 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; passarvi sopra con il puntatore per i dettagli. Se un valore viene troncato, potrebbe non superare la convalida del tipo di dato nei test consecutivi che confrontano valori grandi (ad esempio JSON); in tali casi viene visualizzato un messaggio di errore. Si noti 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 correttamente da un host può essere utilizzato anche per testare i passaggi di preprocessing.

Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire diverse operazioni.

Add	Aggiungi 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	Esegui immediatamente un controllo per un nuovo valore dell'item. Supportato solo per i controlli **passivi** (vedi maggiori dettagli). *Nota*: quando si esegue immediatamente il controllo di un valore, la cache di configurazione non viene aggiornata, pertanto il valore non rifletterà modifiche molto 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 approssimativo da -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	Tipo di informazione		
	Carattere	**Log**	**Testo**
MySQL	255 caratteri	65536 byte	65536 byte
PostgreSQL	255 caratteri	65536 caratteri	65536 caratteri
SQLite (solo Zabbix proxy)	255 caratteri	65536 caratteri	65536 caratteri

I valori binari sono limitati a 16MiB (16777216 byte). I valori che superano questo limite vengono scartati e nel frontend viene visualizzato un errore corrispondente.

I valori con tipo di dato JSON sono limitati a 128MiB (134217728 byte). I valori che superano questo limite vengono scartati e nel frontend viene visualizzato un errore corrispondente. Per MySQL/MariaDB, valutare la configurazione di partizioni giornaliere per la tabella `history_json` (utilizzata per memorizzare i valori JSON), poiché può crescere rapidamente e bloccare housekeeper per un periodo prolungato. Se il proprio caso d'uso prevede valori JSON di grandi dimensioni (1MiB o più), vedere come configurare il sistema per il supporto di valori JSON di grandi dimensioni.

Quando Zabbix scrive qualsiasi dato di item nel database, utilizza query INSERT e le registra nei log (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 e considerarlo 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 molteplici OID in un controllo SNMP).

Per molti tipi di item, è possibile impostare timeout degli item **flessibili**:

- 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 al contempo bassi i timeout per gli 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:

- agent Zabbix (sia controlli passivi che attivi)
- agent SNMP (eccetto i controlli SNMP legacy[1])
- Controllo semplice (eccetto `icmpping*` e gli item VMware[2])
- agent SSH
- agent Telnet
- Controllo esterno
- Monitor database
- Script
- agent HTTP
- 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 di timeout è specificato direttamente nella chiave dell'item; inoltre, si applica anche un timeout massimo hardcoded di 600 secondi. Gli item di monitoraggio VMware utilizzano il proprio parametro VMwareTimeout dalla configurazione di server o proxy.

Item non supportati

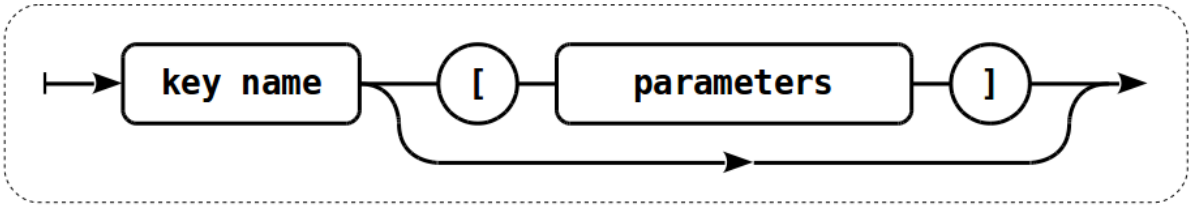
Gli item non supportati vengono segnalati con stato *Non supportato* e continuano a essere ricontrollati secondo il loro *Intervallo di aggiornamento* standard.

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).

1 Formato della chiave item

Il formato della chiave item, inclusi i parametri della chiave, deve seguire regole di sintassi. Le illustrazioni seguenti mostrano la sintassi supportata. Gli elementi e i caratteri consentiti in ciascun 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 item valida, si inizia specificando il nome della chiave; quindi è possibile scegliere se avere o meno parametri, come mostrato dalle due linee che possono essere seguite.

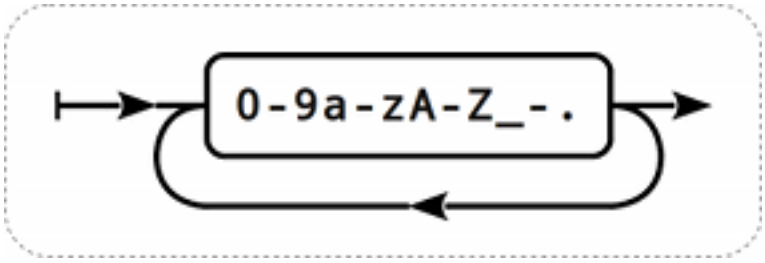
Nome della chiave

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_-.`

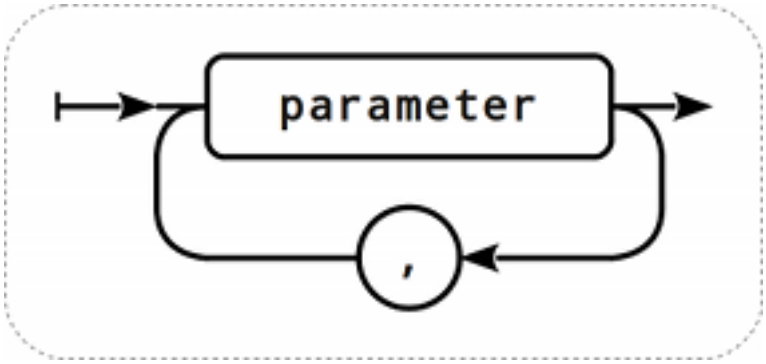
Il che significa:

- tutti i numeri;
- tutte le lettere minuscole;
- tutte le lettere maiuscole;
- trattino basso;
- trattino;
- punto.

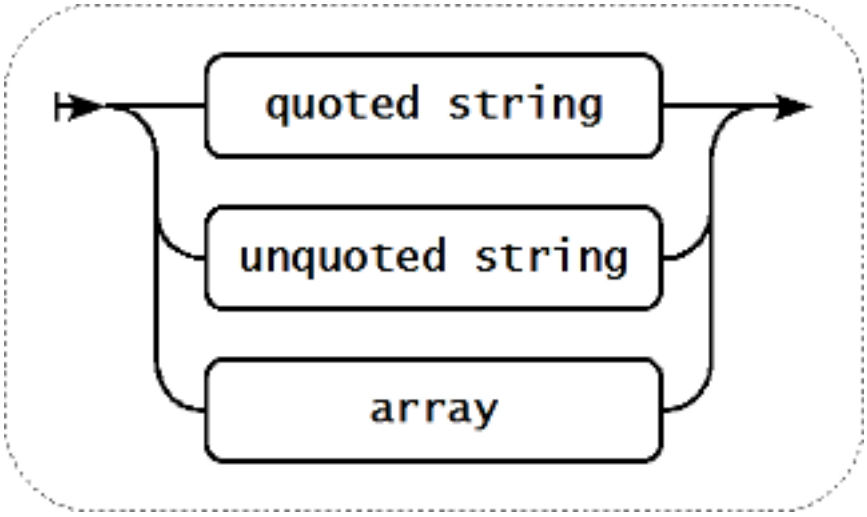


Parametri chiave

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.

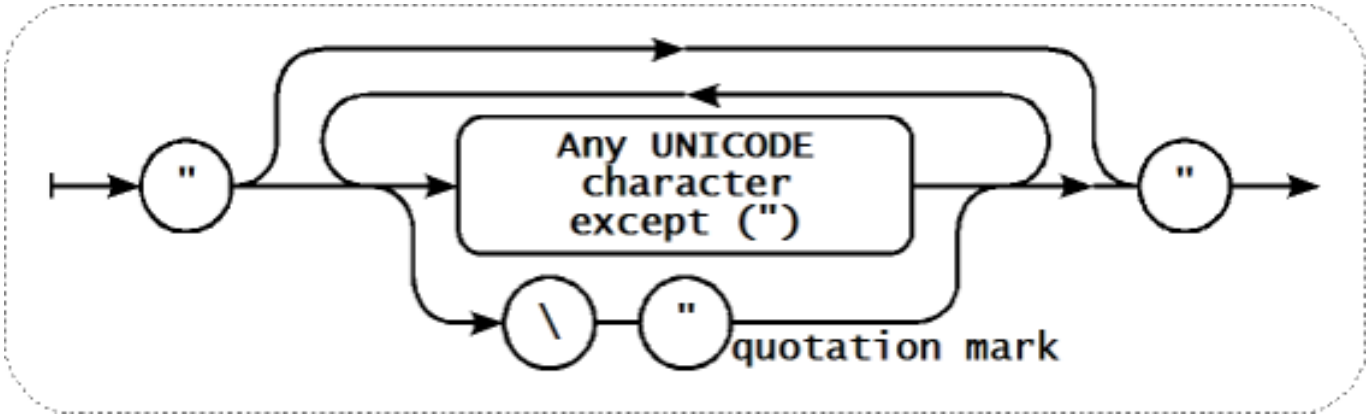


Il parametro può anche essere lasciato vuoto, utilizzando così il valore predefinito. In questo caso, deve essere aggiunto il numero appropriato di virgole se vengono specificati ulteriori parametri. Per esempio, la chiave item **icmpping[„200„500]** specifica che l'intervallo tra i singoli ping è di 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.

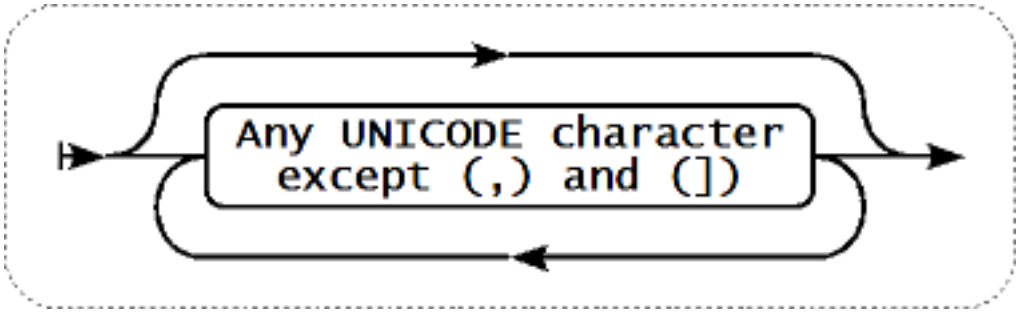


Warning:
Per racchiudere tra virgolette i parametri della chiave dell'item, utilizzare solo le virgolette doppie. Le virgolette singole non sono supportate.

Attention:
Gli array di parametri multilivello, ad esempio `[a,[b,[c,d]],e]`, non sono consentiti.

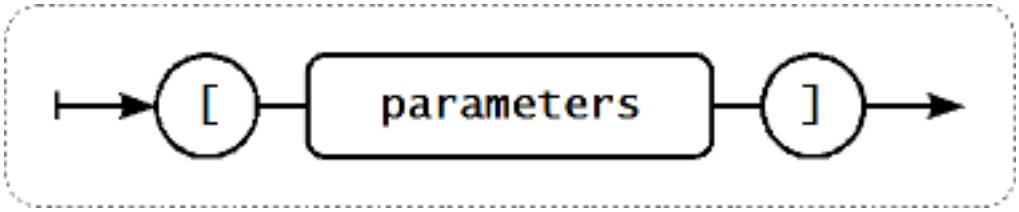
Parametro - stringa senza virgolette

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.



2 Intervalli personalizzati

Panoramica

È possibile creare regole personalizzate relative agli orari in cui un item viene controllato. I due metodi per farlo sono gli *Intervalli flessibili*, che consentono di ridefinire l'intervallo di aggiornamento predefinito, e la *Pianificazione*, tramite la quale il controllo di un item può essere eseguito in un momento specifico o in una sequenza di momenti specifici.

Intervalli flessibili

Gli intervalli flessibili consentono di ridefinire l'intervallo di aggiornamento predefinito per specifici periodi di tempo. 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 di tempo, ad esempio 30s, 1m, 2h, 1d.
- *Periodo* – il periodo di tempo in cui l'intervallo flessibile è attivo (vedere 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 tale configurazione non è consigliata). Se l'intervallo flessibile è inferiore al 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 riprende il polling in base all'*Intervallo di aggiornamento* predefinito una volta terminato il periodo. Esempi:

Interval	Period	Description
10	1-5,09:00-18:00	L'item verrà controllato ogni 10 secondi durante l'orario di lavoro.
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.
60	1-7,12:00-12:01	L'item verrà controllato alle 12:00 ogni giorno. Si noti che questo veniva utilizzato come soluzione alternativa per i controlli pianificati ed è consigliato utilizzare gli intervalli di pianificazione per tali controlli.

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 *<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 delle 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.

In Zabbix frontend, 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
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 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"
```

2 Preelaborazione del valore dell'articolo

Panoramica

Il preprocessing consente di applicare trasformazioni ai valori degli item ricevuti prima di salvarli nel database. Queste trasformazioni/fasi di preprocessing 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;
- applicazione di espressioni regolari per estrarre o modificare i valori;
- esecuzione di script personalizzati sui valori;
- scarto dei valori invariati per ottimizzare lo spazio di archiviazione nel database.

Per un item è possibile configurare una o più fasi di preprocessing. Queste fasi vengono eseguite nell'ordine in cui sono configurate.

Attention:
Se una fase di preprocessing non riesce, 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.

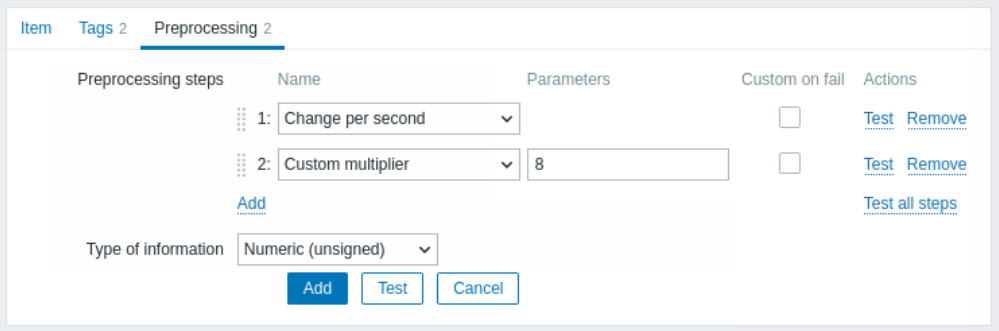
Tutti i valori passati al preprocessing vengono inizialmente trattati come stringhe. La conversione nel tipo di valore desiderato (come definito nella configurazione dell'item) viene eseguita alla fine della pipeline di preprocessing. Tuttavia, specifiche fasi di preprocessing possono attivare conversioni anticipate, se necessario. Per informazioni tecniche dettagliate, vedere Dettagli sul preprocessing.

Per assicurarti che la configurazione del preprocessing funzioni come previsto, puoi testarla.

Vedi anche: Esempi di preprocessing

Configurazione

I passaggi di preprocessing sono definiti nella scheda **Preprocessing** del modulo di configurazione dell'item.



Fare clic su *Aggiungi* per selezionare una trasformazione supportata.

Il campo *Tipo di informazione* viene visualizzato nella parte inferiore della scheda quando è definito almeno un passaggio di preprocessing. Se necessario, è possibile modificare il tipo di informazione senza uscire dalla scheda *Preprocessing*. Per una descrizione dettagliata dei parametri, vedere Creazione di un item.

Trasformazioni supportate

Tutte le trasformazioni supportate sono elencate di seguito. Fare clic sul nome della trasformazione per visualizzarne tutti i dettagli.

Nome	Descrizione	Tipo
Espressione regolare	Confronta il valore con l'espressione regolare e lo sostituisce con l'output richiesto.	Testo
Sostituisci	Trova la stringa di ricerca e la sostituisce con un'altra (o con nulla).	
Taglia	Rimuove i caratteri specificati dall'inizio e dalla fine del valore.	
Taglia a destra	Rimuove i caratteri specificati dalla fine del valore.	
Taglia a sinistra	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 in base al 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 ottenuto con SNMP get.	
Moltiplicatore personalizzato	Moltiplica il valore per il valore intero o in virgola mobile specificato.	Aritmetica
Variazione semplice	Calcola la differenza tra il valore corrente e quello precedente.	Variazione
Variazione al secondo	Calcola al secondo la velocità di variazione del valore (differenza tra il valore corrente e quello precedente).	
Booleano in decimale	Converte il valore dal formato booleano a quello decimale.	Sistemi numerici
Ottale in decimale	Converte il valore dal formato ottale a quello decimale.	
Esadecimale in decimale	Converte il valore dal formato esadecimale a quello decimale.	

Nome	Descrizione	Tipo
JavaScript	Inserire codice JavaScript.	Script personalizzati
Nell'intervallo	Definisce un intervallo entro cui un valore deve rientrare.	Convalida
Corrisponde all'espressione regolare	Specifica un'espressione regolare a cui un valore deve corrispondere.	
Non corrisponde all'espressione regolare	Specifica un'espressione regolare a cui un valore non deve corrispondere.	
Controlla errore in JSON	Controlla un messaggio di errore a livello applicativo individuato in JSONPath.	
Controlla errore in XML	Controlla un messaggio di errore a livello applicativo individuato in XPath.	
Controlla errore usando un'espressione regolare	Controlla un messaggio di errore a livello applicativo utilizzando un'espressione regolare.	
Controlla valore non supportato	Controlla se non è stato possibile recuperare alcun valore dell'item.	
Scarta se invariato	Scarta un valore se non è cambiato.	Limitazione della frequenza
Scarta se invariato con heartbeat	Scarta un valore se non è cambiato entro il periodo di tempo definito.	
Pattern Prometheus	Utilizza la seguente query per estrarre i dati richiesti dalle metriche Prometheus.	Prometheus
Prometheus in JSON	Converte le metriche Prometheus richieste in JSON.	

Si noti che per i passaggi di preprocessing *Variazione* e *Limitazione della frequenza*, Zabbix deve ricordare l'ultimo valore per calcolare/confrontare il nuovo valore come richiesto. Questi valori precedenti sono gestiti dal gestore del preprocessing. Se Zabbix server o proxy viene riavviato oppure viene apportata qualsiasi modifica ai passaggi di preprocessing, l'ultimo valore dell'item corrispondente viene reimpostato, con il seguente risultato:

- per i passaggi *Variazione semplice*, *Variazione al secondo* - il valore successivo verrà ignorato perché non esiste alcun valore precedente da cui calcolare la variazione;
- per i passaggi *Scarta se invariato*, *Scarta se invariato con heartbeat* - il valore successivo non verrà mai scartato, 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 l'N-esimo gruppo corrispondente. Una sequenza di escape \0 viene sostituita con il testo corrispondente.

Commenti:

- Se il valore di input non corrisponde, l'item diventerà non supportato;
- L'espressione regolare supporta l'estrazione di un massimo di 10 gruppi catturati con la sequenza \N;
- Se si seleziona la casella *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore nel passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.
- Fare riferimento alla sezione regular expressions per alcuni esempi esistenti.

Sostituisci

Trova la stringa di ricerca e sostituiscila con un'altra (oppure con nulla).

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 fatto di eliminare la stringa di ricerca quando viene trovata.

Commenti:

- Tutte le occorrenze della stringa di ricerca verranno sostituite;
- È possibile usare sequenze di escape per cercare o sostituire interruzioni di riga, ritorno a capo, tabulazioni e spazi "\n \r \t \s"; il backslash può essere sottoposto a escape come "\\" e le sequenze di escape possono essere sottoposte a escape come "\\n";
- L'escape di interruzioni di riga, ritorno a capo e tabulazioni viene eseguito automaticamente durante il 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 da dati XML utilizzando la funzionalità XPath.

Commenti:

- Affinché questa opzione funzioni, Zabbix server (o Zabbix proxy) deve essere compilato con il supporto libxml;
- I namespace non sono supportati;
- Se si seleziona la casella di controllo *Personalizza in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in una fase di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

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></doc
```

JSON Path

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: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Da CSV a JSON

Converte i dati di un file CSV nel formato JSON.

Per ulteriori informazioni, vedere: Preprocessing da CSV a JSON.

Da XML a JSON

Converti i dati in formato XML in JSON.

Per ulteriori informazioni, vedere: Regole di serializzazione.

Se si seleziona la casella di controllo *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in una fase di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Valore SNMP walk

Estrae il valore in base al nome OID/MIB specificato e applica le opzioni di formattazione:

- **Invariato** - restituisce la stringa esadecimale come stringa esadecimale non escape-ata (*nota* che gli hint di visualizzazione vengono comunque applicati);
- **UTF-8 da hex-STRING** - converte la stringa esadecimale in una stringa UTF-8;
- **MAC da hex-STRING** - valida la stringa esadecimale come indirizzo MAC e restituisce una stringa di indirizzo MAC corretta (dove ' ' viene sostituito con ':');
- **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 di controllo *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Conversione di SNMP walk in JSON

Converte i valori SNMP in JSON.

Specificare un nome di campo nel JSON e il corrispondente percorso OID SNMP. I valori dei campi verranno popolati con i valori presenti nel percorso OID SNMP specificato.

Commenti:

- Sono disponibili opzioni di formattazione dei valori simili a quelle del passaggio *Valore SNMP walk*;
- È possibile utilizzare questo passaggio di preprocessing per SNMP OID discovery;
- Se si seleziona la casella di controllo *Custom on fail*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore nel passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Valore ottenuto con SNMP

Applica le opzioni di formattazione al valore ottenuto con SNMP:

- **UTF-8 da Hex-STRING** - converte una stringa esadecimale in una stringa UTF-8;
- **MAC da Hex-STRING** - valida una stringa esadecimale come indirizzo MAC e restituisce una stringa MAC corretta (dove `' '` viene sostituito con `':'`);
- **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 selezioni la casella di controllo *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Moltiplicatore personalizzato

Moltiplica il valore per il valore intero o in virgola mobile specificato.

Commenti:

- Utilizzare questa opzione per convertire i valori ricevuti in KB, MBps, ecc. in B, Bps. Altrimenti, Zabbix non può impostare correttamente i prefissi (K, M, G, ecc.).
- Si noti che se il tipo di informazione dell'item è *Numeric (unsigned)*, i valori in ingresso con una parte frazionaria verranno troncati (cioè, '0.9' diventerà '0') prima che venga applicato il moltiplicatore personalizzato;
- Se si utilizza un moltiplicatore personalizzato o si memorizza il valore come *Change per second* per item con il tipo di informazione impostato su *Numeric (unsigned)* e il valore calcolato risultante è effettivamente un numero float, il valore calcolato viene comunque accettato come corretto troncando la parte decimale e memorizzando il valore come intero;
- Supportato: 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 in virgola mobile.
- Se si seleziona la casella *Custom on fail*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Variazione 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 variazione per item ("Simple change" oppure "Change per second");
- Se il valore corrente è inferiore al valore precedente, Zabbix scarta tale differenza (non memorizza nulla) e attende un altro valore;
- Se selezioni la casella *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

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 'Type of information' su *Numeric (float)*, anche se i valori grezzi in ingresso sono interi. Questo è particolarmente rilevante per i numeri piccoli, dove 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 usare *Numeric (unsigned)* e quindi troncare 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 ("Simple change" o "Change per second");
- Se il valore corrente è inferiore a quello precedente, Zabbix scarta tale differenza (non memorizza nulla) e attende un altro valore. Questo aiuta a gestire correttamente, per esempio, il wrapping (overflow) dei contatori SNMP a 32 bit.
- Se si seleziona la casella *Custom on fail*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Booleano in decimale

Converte il valore dal formato booleano a quello decimale.

Commenti:

- La rappresentazione testuale viene tradotta in 0 oppure 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 è considerato TRUE e zero è considerato FALSE.
- Se si seleziona la casella *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore nel passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Da ottale a decimale

Converte il valore dal formato ottale a quello decimale.

Se si seleziona la casella di controllo *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Da esadecimale a decimale

Converte il valore dal formato esadecimale a quello decimale.

Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

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 entro il quale un valore deve rientrare specificando i valori minimo/massimo (inclusi).

Commenti:

- Sono accettati valori numerici (incluso qualsiasi numero di cifre, parte decimale facoltativa e 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;
- Se si seleziona la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Corrisponde all'espressione regolare

Specificare un'espressione regolare a cui un valore deve corrispondere.

Se si seleziona la casella di controllo *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Non corrisponde all'espressione regolare

Specificare un'espressione regolare a cui un valore non deve corrispondere.

Se si seleziona la casella di controllo *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Controlla la presenza di errori in JSON

Controlla la presenza di un messaggio di errore a livello di applicazione situato in JSONPath. Interrompe l'elaborazione se l'operazione ha esito positivo e il messaggio non è vuoto; altrimenti, continua l'elaborazione con il valore presente prima di questo passaggio di preprocessing.

Commenti:

- Questi errori dei servizi esterni vengono riportati all'utente così come sono, senza aggiungere informazioni sul passaggio di preprocessing;
- Non verrà segnalato alcun errore in caso di mancata analisi di JSON non valido;
- Se selezioni la casella di controllo *Custom on fail*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Controllo di errori in XML

Controlla la presenza di un messaggio di errore a livello di applicazione situato in XPath. Interrompe l'elaborazione se l'operazione ha esito positivo e il messaggio non è vuoto; altrimenti, continua l'elaborazione con il valore precedente a questo passaggio di preprocessing.

Commenti:

- Questi errori dei servizi esterni vengono riportati all'utente così come sono, senza aggiungere informazioni sul passaggio di preprocessing;
- Non verrà segnalato alcun errore in caso di mancata analisi di XML non valido;
- Se si seleziona la casella di controllo *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore in un passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Controllo degli errori tramite un'espressione regolare

Controlla la presenza di un messaggio di errore a livello di applicazione utilizzando 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 preprocessing.

Parametri:

- **pattern** - l'espressione regolare;

- **output** - il modello 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.

Commenti:

- Questi errori dei servizi esterni vengono segnalati all'utente così come sono, senza aggiungere informazioni sul passaggio di preprocessing;
- Se si seleziona la casella di controllo *Custom on fail*, è possibile specificare opzioni personalizzate per la gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore nel passaggio di preprocessing, l'item non diventerà unsupported se è selezionata l'opzione per scartare il valore o impostare un valore specificato.

Controlla il valore non supportato

Verifica se non è stato possibile recuperare alcun valore dell'item. Specifica come deve essere gestito l'errore, in base all'ispezione del messaggio di errore restituito.

Parametri:

- **scope** - seleziona l'ambito di elaborazione dell'errore:
qualsiasi errore - qualsiasi errore;
errore corrisponde - solo l'errore che corrisponde all'espressione regolare specificata in *pattern*;
errore non corrisponde - 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 *qualsiasi errore*, questo campo non viene visualizzato. Se visualizzato, questo campo è obbligatorio.

Commenti:

- Normalmente, l'assenza/impossibilità di recuperare un valore comporterebbe che l'item diventi non supportato. Questo passaggio di preprocessing consente di modificare questo comportamento. Se selezioni la casella *Custom on fail* (sempre selezionata e disattivata per questo passaggio di preprocessing), è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato. In caso di errore nel passaggio di preprocessing, l'item non diventerà non supportato se è selezionata l'opzione per scartare il valore o impostare un valore specificato.
- Questo passaggio di preprocessing 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, vedi Esempi di preprocessing. Se c'è una mancata corrispondenza di tipo, l'item potrebbe comunque diventare non supportato dopo l'esecuzione di tutti i passaggi di preprocessing. Per verificare una mancata corrispondenza di tipo, puoi usare, ad esempio, il passaggio di preprocessing *Custom multiplier*; vedi Esempi di preprocessing.
- È supportata l'acquisizione dei gruppi dell'espressione regolare nei campi *Set value to* o *Set error to*. Usa \N (dove N=1…9) per recuperare l'N-esimo gruppo corrispondente; usa \0 per recuperare il testo corrispondente;
- Questi passaggi vengono sempre eseguiti come primi passaggi di preprocessing e, dopo il salvataggio delle modifiche all'item, vengono posizionati sopra tutti gli altri;
- Sono supportati più passaggi *Check for not supported value*, nell'ordine specificato. Un passaggio per *qualsiasi errore* verrà automaticamente posizionato come ultimo passaggio in 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 invariato 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 suffissi di tempo (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 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 presenti 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 throttling per item.

Pattern Prometheus

Utilizzare la seguente query per estrarre i dati richiesti dalle metriche Prometheus.

Per maggiori dettagli, vedere controlli Prometheus.

Da Prometheus a JSON

Converte le metriche Prometheus richieste in JSON.

Per maggiori dettagli, vedere controlli Prometheus.

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.

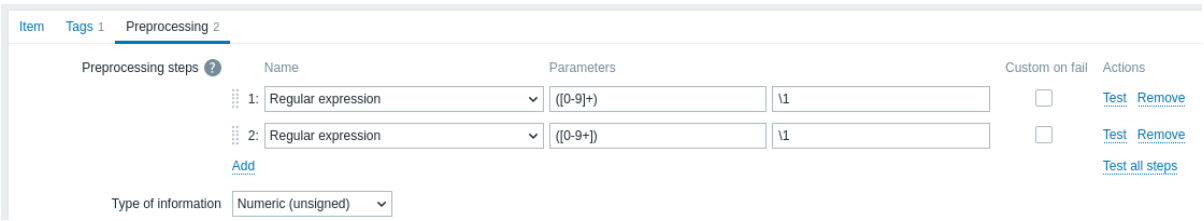
Test

Vedere test di preprocessing.

1 Test della pre-elaborazione

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 preprocessato.

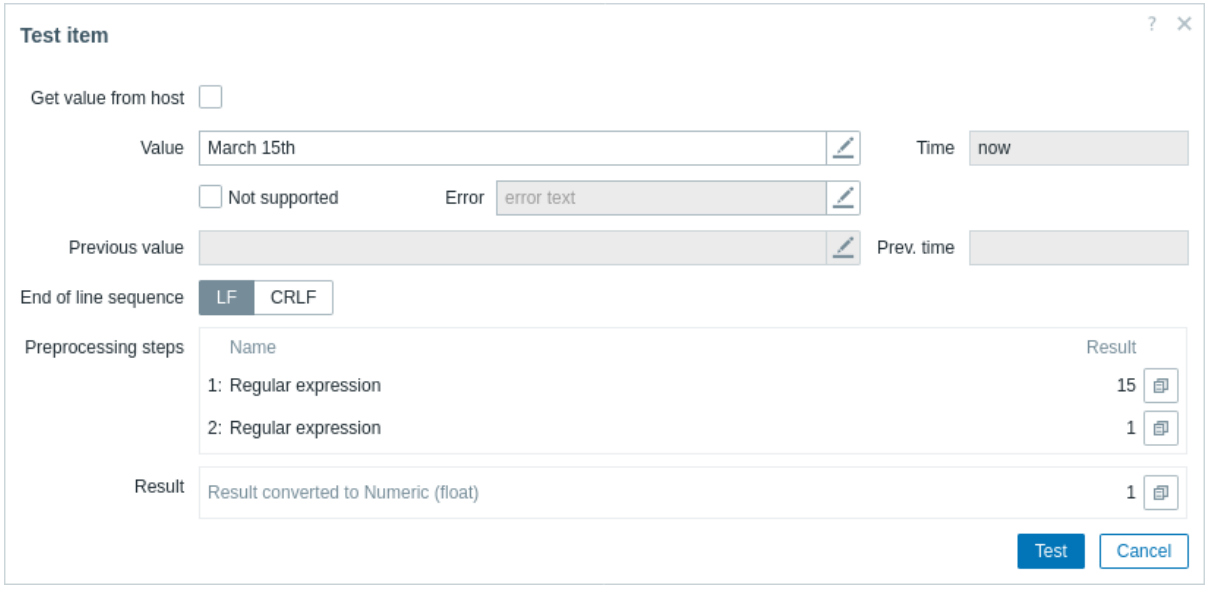


È possibile eseguire il test:

- 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. Facendo clic sul pulsante *Test* o *Test all steps* rispettivamente nel blocco Actions, si apre una finestra di test.

Test del valore ipotetico



Parametro	Descrizione
Ottieni valore da host	Se vuoi testare un valore ipotetico, lascia questa casella non selezionata. Vedi anche: Test del valore reale.
Valore	Inserisci il valore di input da testare. Facendo clic nel campo del parametro o sul pulsante visualizza/modifica si aprirà una finestra con area di testo per inserire il valore o il blocco di codice.
Non supportato	Seleziona questa casella per testare un valore non supportato. Questa opzione è utile per testare il passaggio di preprocessing *Verifica valore non supportato*.
Errore	Inserisci il testo dell'errore. Questo campo è abilitato quando *Ottieni valore da host* non è selezionato, ma *Non supportato* è selezionato. Se *Ottieni valore da host* è selezionato, questo campo viene compilato con il messaggio di errore effettivo (sola lettura) dall'host.
Ora	Viene visualizzata l'ora del valore di input: `now` (sola lettura).
Valore precedente	Inserisci un valore di input precedente con cui confrontare. Solo per i passaggi di preprocessing *Change* e *Throttling*.
Ora precedente	Inserisci l'ora del valore di input precedente con cui confrontare. Solo per i passaggi di preprocessing *Change* e *Throttling*. Il valore predefinito si basa sul valore del campo 'Intervallo di aggiornamento' dell'item (se '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`.
Macro	Se vengono utilizzate macro, queste sono elencate insieme ai rispettivi valori. I valori sono modificabili a scopo di test, ma le modifiche verranno salvate solo nel contesto del test.
Sequenza di fine riga	Seleziona la sequenza di fine riga per i valori di input multilinea: **LF** - sequenza LF (line feed) **CRLF** - sequenza CRLF (carriage-return line-feed).
Passaggi di preprocessing	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. Nota che i dati superiori a 512KB vengono comunque elaborati completamente da Zabbix server. Se il passaggio fallisce durante il 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 del preprocessing, che mostra quale azione è stata eseguita e quale risultato ha prodotto (errore o valore).

Parametro	Descrizione
Risultato	Il risultato finale del test dei passaggi di preprocessing viene visualizzato in tutti i casi in cui tutti i passaggi vengono testati insieme (quando fai clic sul pulsante *Test all steps*). Viene visualizzato anche 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. Nota che i dati superiori a 512KB vengono comunque elaborati completamente da Zabbix server.

Fai clic su *Test* per vedere il risultato dopo ogni 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.

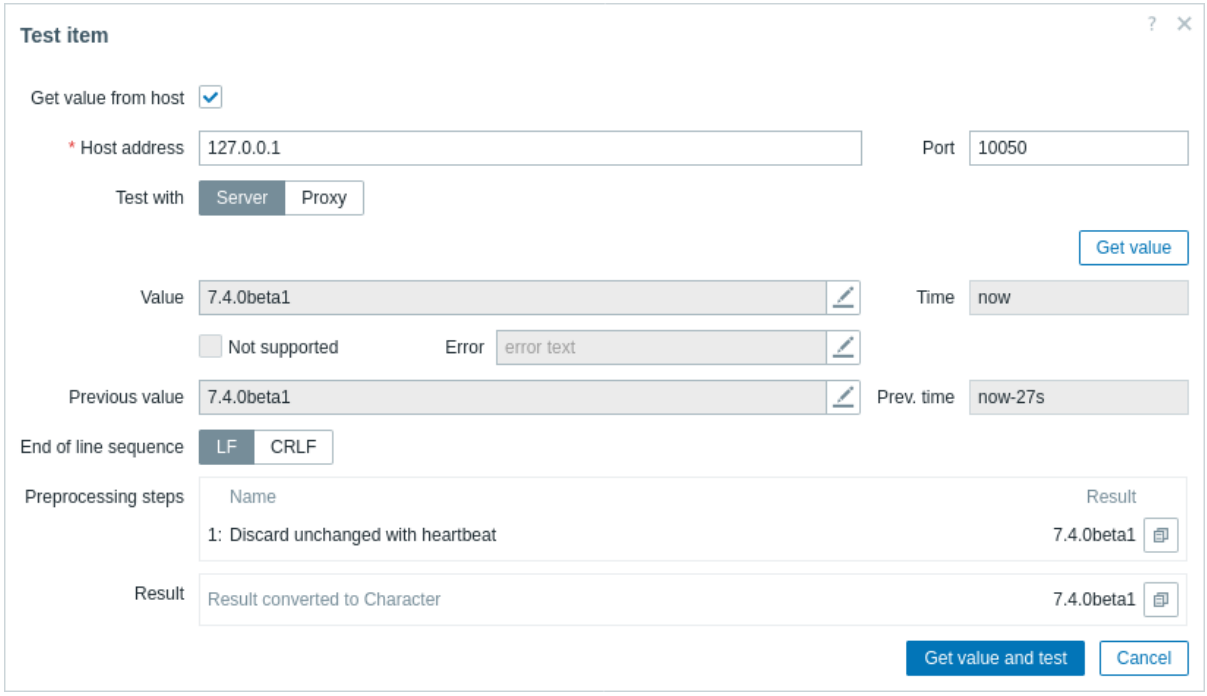
Il test viene eseguito da Zabbix server. Il frontend invia una richiesta corrispondente al server e attende il risultato. La richiesta contiene il valore di input e i passaggi di preprocessing (con le macro utente espanse). Per i passaggi *Change* e *Throttling*, è possibile specificare facoltativamente un valore precedente e la relativa ora. Il server risponde con i risultati per ciascun passaggio di preprocessing.

Tutti gli errori tecnici o gli errori di validazione dell'input vengono visualizzati nel riquadro degli errori nella parte superiore della finestra di test.

Test di un valore reale

Per testare la pre-elaborazione rispetto a un valore reale:

- Selezionare la casella *Get value from host*
- Inserire o verificare i parametri del host (indirizzo del host, porta, nome del proxy/nessun proxy) e i dettagli specifici dell'item (come la community SNMPv2 o le credenziali di sicurezza SNMPv3). Questi campi dipendono dal contesto:
  - I valori vengono precompilati quando possibile, ad esempio per gli item che richiedono un agent, prendendo le informazioni dall'interfaccia agent selezionata del host
  - I valori devono essere compilati manualmente per gli item del template
  - I valori delle macro in testo semplice vengono risolti
  - Quando il valore del campo (o parte del valore) è una macro segreta o Vault, il campo sarà vuoto e dovrà essere compilato manualmente. Se un parametro dell'item contiene un valore di macro segreta, viene visualizzato il seguente messaggio di avviso: "L'item contiene macro definite dall'utente con valori segreti. I valori di queste macro devono essere inseriti manualmente."
  - I campi sono disabilitati quando non necessari nel contesto del tipo di item (ad esempio, i campi indirizzo del host e proxy sono disabilitati per gli item calcolati)
- Fare clic su *Get value and test* per testare la pre-elaborazione



Se è stata specificata una mappatura dei valori nel modulo di configurazione dell'item (campo 'Show value'), la finestra di dialogo di 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:

Parameter	Description
Get value from host	Selezionare questa casella per ottenere un valore reale dal host.
Host address	Inserire l'indirizzo del host. Questo campo viene compilato automaticamente con l'indirizzo dell'interfaccia host dell'item.
Port	Inserire la porta del host. Questo campo viene compilato automaticamente con la porta dell'interfaccia host dell'item.
Additional fields for SNMP interfaces (SNMP version, SNMP community, Context name, etc.)	Vedere Configuring SNMP monitoring per ulteriori dettagli sulla configurazione di un'interfaccia SNMP (v1, v2 e v3). Questi campi vengono compilati automaticamente dall'interfaccia host dell'item.
Proxy	Specificare il proxy se il host è monitorato da un proxy. Questo campo viene compilato automaticamente con il proxy del host (se presente).
Value	Valore recuperato dal host. Facendo clic nel campo del parametro o sul pulsante di visualizzazione/modifica si aprirà una finestra con area di testo contenente il valore o il blocco di codice. I valori vengono troncati a una dimensione massima di 512KB e solo nel frontend. Se un risultato viene troncato, viene visualizzata un'icona di avviso. La descrizione dell'avviso viene mostrata al passaggio del mouse. Si noti che i dati superiori a 512KB vengono comunque elaborati completamente da Zabbix server.

Per il resto dei parametri, vedere Testing hypothetical value sopra.

2 Dettagli di preelaborazione

Panoramica

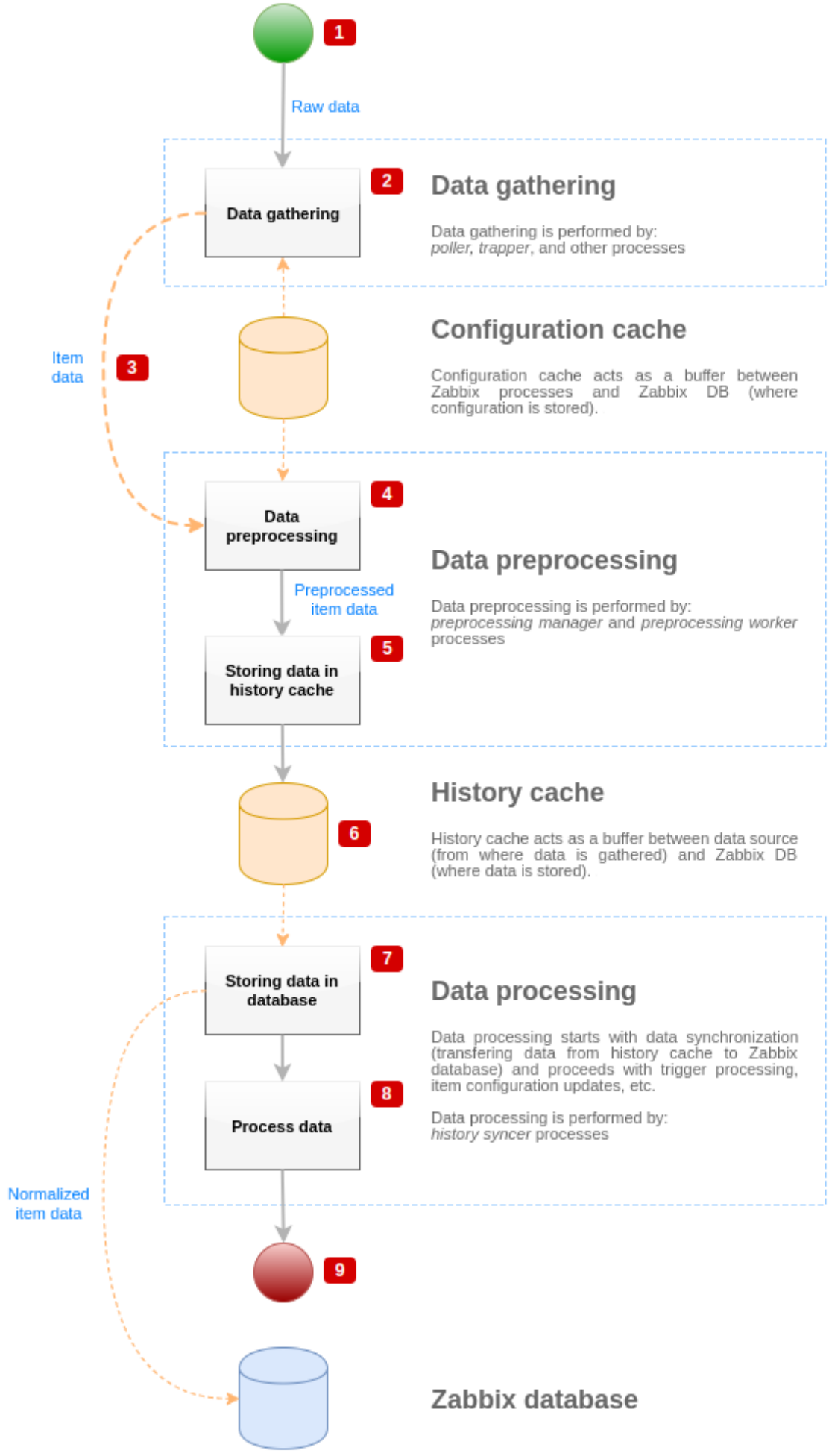
Questa sezione fornisce i dettagli del preprocessing dei valori degli item. Il preprocessing dei valori degli item consente di definire ed eseguire regole di trasformazione per i valori degli item ricevuti.

Il preprocessing è gestito dal processo di gestione del preprocessing insieme ai worker di preprocessing che eseguono i passaggi di preprocessing. Tutti i valori con preprocessing, ricevuti da diversi raccoglitori di dati, passano attraverso il gestore del preprocessing prima di essere aggiunti alla cache della cronologia. La comunicazione IPC basata su socket viene utilizzata tra i raccoglitori di

dati (poller, trapper, ecc.) e il processo di preprocessing. I passaggi di preprocessing vengono eseguiti da Zabbix server oppure da Zabbix proxy (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.

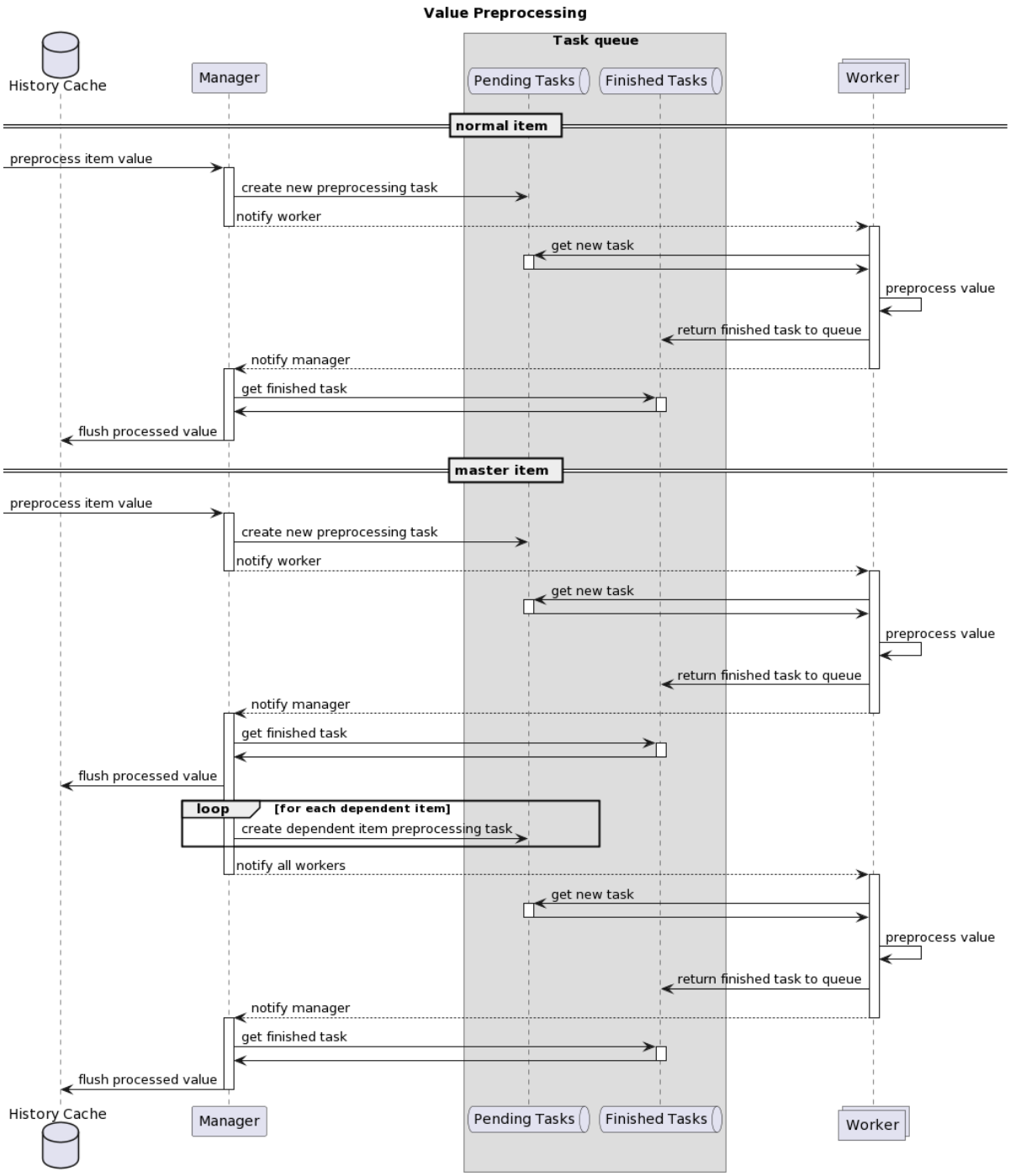
Note:
Un item può cambiare il proprio stato in NOT SUPPORTED se la normalizzazione dei dati fallisce (ad esempio, quando un valore testuale non può essere convertito in numero).

- 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 utilizzato per impostare il numero di istanze preavviate dei worker di preprocessing, che dovrebbe almeno corrispondere al numero di core CPU disponibili.

Se le attività di preprocessing non sono limitate dalla CPU e comportano frequenti richieste di rete, si consiglia di configurare worker aggiuntivi. Il numero ottimale di worker di preprocessing può essere determinato da molti fattori, tra cui il numero di item "preprocessabili" (item che richiedono l'esecuzione di uno qualsiasi dei passaggi di preprocessing), il numero di processi di raccolta dati, il numero medio di passaggi per il preprocessing degli item, ecc. Un numero insufficiente di worker può causare un elevato utilizzo della memoria. Per la risoluzione dei problemi relativi all'utilizzo eccessivo della memoria nella propria installazione di Zabbix, vedere Profiling excessive memory usage with tcmalloc.

Tuttavia, supponendo che non vi siano operazioni di preprocessing pesanti come il parsing di grandi blocchi XML/JSON, il numero di worker di preprocessing può corrispondere al numero totale di processi di raccolta dati. In questo modo, nella maggior parte dei casi (tranne quando i dati dal processo di raccolta arrivano in blocco) ci sarà almeno un worker di preprocessing libero per i dati raccolti.

Warning:
Troppi processi di raccolta dati (poller, unreachable poller, ODBC poller, HTTP poller, Java poller, pinger, trapper, proxy-poller), insieme a IPMI manager, SNMP trapper e worker di preprocessing, possono esaurire il limite di file descriptor per processo del 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 anche più tardi. Per evitare tali problemi, rivedere il file di configurazione di Zabbix server per ottimizzare il numero di controlli e processi concorrenti. Inoltre, se necessario, assicurarsi che il limite dei file descriptor sia impostato su un valore sufficientemente alto verificando 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.

3 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.

4 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:

Operando	Descrizione

`"<text>"`
`'<text>'`
	Costante di testo.
	
	Esempio:
	`'value: \\'1\\''`
	`"value: '1'"`

`<number>`	Costante numerica con supporto della notazione scientifica. Esempio: 123
`<jsonpath starting with $>`	Valore a cui si fa riferimento tramite il JSONPath dal nodo radice del documento di input; sono supportati solo percorsi definiti. Esempio: `$.object.name`
`<jsonpath starting with @>`	Valore a cui si fa riferimento tramite il JSONPath dall'oggetto/elemento corrente; sono supportati solo percorsi definiti. Esempio: `@.name`

Operatori supportati:

Operatore	Tipo	Descrizione	Risultato
`-`	Binario	Sottrazione	Numero
`+`	Binario	Addizione	Numero
`/`	Binario	Divisione	Numero
`*`	Binario	Moltiplicazione	Numero
`==`	Binario	Uguaglianza	Booleano (1/0)
`!=`	Binario	Disuguaglianza	Booleano (1/0)
	Binario	Minore di	Booleano (1/0)
`<=`	Binario	Minore o uguale a	Booleano (1/0)
`>`	Binario	Maggiore di	Booleano (1/0)
`>=`	Binario	Maggiore o uguale a	Booleano (1/0)
`=~`	Binario	Corrisponde all'espressione regolare	Booleano (1/0)
`!`	Unario	NOT booleano	Booleano (1/0)
`||`	Binario	OR booleano	Booleano (1/0)
`&&`	Binario	AND booleano	Booleano (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
`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": "riferimento",
      "author": "Nigel Rees",
      "title": "Detti del secolo",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "narrativa",
      "author": "Evelyn Waugh",
      "title": "Spada d'onore",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "narrativa",
      "author": "Herman Melville",
      "title": "Moby Dick",
```

```
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "narrativa",
      "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 assemblaggio 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 di pagine danneggiate dall'umidità",
          "price": 24.5
        },
        {
          "description": "Rilegatura di un libro strappato",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "narrativa",
    "no filters": "nessun \"filtro\""
  },
  "closed message": "Il negozio è chiuso",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	narrativa
`$.filters['no filters']`	definite	nessun ”filtro”
`$.filters`	definite	{ ”price”: 10, ”category”: ”narrativa”, ”no filters”: ”nessun \”filtro\”” }
`$.books[1].title`	definite	Spada d’onore
`$.books[-1].author`	definite	J. R. R. Tolkien
`$.books.length()`	definite	4
`$.tags[:]`	indefinite	[”a”, ”b”, ”c”, ”d”, ”e”]
`$.tags[2:]`	indefinite	[”c”, ”d”, ”e”]
`$.tags[:3]`	indefinite	[”a”, ”b”, ”c”]
`$.tags[1:4]`	indefinite	[”b”, ”c”, ”d”]
`$.tags[-2:]`	indefinite	[”d”, ”e”]
`$.tags[:-3]`	indefinite	[”a”, ”b”]
`$.tags[:-3].length()`	definite	2
`$.books[0, 2].title`	indefinite	[”Moby Dick”, ”Detti del secolo”]
`$.books[1]['author', "title"]`	indefinite	[”Spada d’onore”, ”Evelyn Waugh”]
`$..id`	indefinite	[1, 2, 3, 4]
`$.services..price`	indefinite	[154.99, 5, 46, 24.5, 99.49]
`$.books[?(@.id == 4 - 0.4 * 5)].title`	indefinite	[”Spada d’onore”] Nota: Questa query mostra che nelle query possono essere usate operazioni aritmetiche; può essere semplificata in `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Spada d’onore”, ”Il Signore degli Anelli”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Detti del secolo”, ”Moby Dick”, ”Il Signore degli Anelli”]
`$.books[?(@.id != 2)].title`	indefinite	[”Detti del secolo”, ”Moby Dick”, ”Il Signore degli Anelli”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Detti del secolo”, ”Spada d’onore”, ”Il Signore degli Anelli”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Il Signore degli Anelli”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Detti del secolo”, ”Il Signore degli Anelli”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Spada d’onore”, ”Il Signore degli Anelli”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Spada d’onore”,”Moby Dick”,”Il Signore degli Anelli”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Moby Dick”]

JSONPath	Type	Result

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”riferimento”,
		”author”: ”Nigel Rees”,
		”title”: ”Detti del secolo”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”narrativa”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Spada d’onore”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”narrativa”,
		”author”: ”Herman Melville”,
		”title”: ”Moby Dick”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”narrativa”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”Il Signore degli Anelli”,
		”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
`$.books[?(@.id == 2)].title.first()`	definite	Spada d’onore
`$..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 con valori booleani devono essere usate costanti testuali.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Nota: Nelle comparazioni con valori booleani devono essere usate costanti testuali.
`$.services[?(@.servicegroup=="1002")].description`	indefinite	[”Rilegatura di un libro strappato”]

1 Escape dei caratteri speciali dai valori delle macro LLD in JSONPath

Quando le macro di low-level discovery vengono utilizzate nel preprocessing JSONPath e i loro valori vengono risolti, si applicano le seguenti regole di escape dei caratteri speciali:

- solo i caratteri barra rovesciata (\) e virgolette doppie (") vengono considerati per l'escape;
- se il valore della macro risolta contiene questi caratteri, ciascuno di essi viene preceduto da una barra rovesciata;
- se sono già preceduti da una barra rovesciata, ciò non viene considerato come escape e sia la barra rovesciata sia il carattere speciale successivo vengono nuovamente sottoposti a escape.

Per esempio:

JSONPath	Valore macro LLD	Dopo la sostituzione
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]
	c:\temp	$.[?(@.value == "c:\\temp")]
	a\\b	$.[?(@.value == "a\\\\b")]

Quando viene utilizzata nell'espressione, la macro che può contenere caratteri speciali deve essere racchiusa tra virgolette doppie:

JSONPath	Valore macro LLD	Dopo la sostituzione	Risultato
$.[?(@.value == "{#MACRO}")]	special "value"	$.[?(@.value == "special \"value\"")]	OK
$.[?(@.value == {#MACRO})]		$.[?(@.value == special \"value\")]	**Espressione JSONPath non valida**

Quando viene utilizzata nel percorso, la macro che può contenere caratteri speciali deve essere racchiusa tra parentesi quadre **e** virgolette doppie:

JSONPath	Valore macro LLD	Dopo la sostituzione	Risultato
$.["{#MACRO}"].value	c:\temp	$.["c:\\temp"].value	OK
$.{#MACRO}.value		$.c:\\temp.value	**Espressione JSONPath non valida**

5 Preprocessing 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.

Pre-elaborazione JavaScript

La pre-elaborazione JavaScript viene eseguita invocando una funzione JavaScript con un singolo parametro 'value' e un corpo funzione fornito dall'utente. Il risultato del passaggio di pre-elaborazione è il valore restituito da questa funzione; per esempio, per eseguire la conversione da Fahrenheit a Celsius, inserire:

```
return (value - 32)  * 5 / 9
```

nei parametri di pre-elaborazione JavaScript, che verranno racchiusi dal server in una funzione JavaScript:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Il parametro di input 'value' viene sempre passato come stringa. Il valore restituito viene convertito automaticamente 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 produrrà un errore;

- la restituzione del valore null farà sì che il valore di input venga scartato, in modo simile alla pre-elaborazione 'Discard value' nell'azione 'Custom on fail'.

Gli errori possono essere restituiti lanciando valori/oggetti (normalmente stringhe oppure oggetti Error).

Per 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); se viene superato, verrà restituito un errore. Viene applicato un limite heap di 512 megabyte.

Il bytecode del passaggio di pre-elaborazione JavaScript viene memorizzato nella cache e riutilizzato quando il passaggio viene applicato la volta successiva. Qualsiasi modifica ai passaggi di pre-elaborazione dell'item comporterà l'azzeramento dello script in cache e la sua successiva ricompilazione.

Errori consecutivi in fase di esecuzione (3 di seguito) causeranno la reinizializzazione del motore per ridurre la possibilità che uno script comprometta l'ambiente di esecuzione degli script successivi (questa azione viene registrata con DebugLevel 4 e superiori).

La pre-elaborazione JavaScript è implementata con il motore JavaScript Duktape.

Vedi anche: Oggetti JavaScript aggiuntivi e funzioni globali

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
    }
}
```

1 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 che 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.

Method	Description
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 (caratteri speciali e nomi delle intestazioni inclusi).
clearHeader()	Cancella l'intestazione HTTP. Se non è impostato alcun campo di intestazione, HttpRequest imposterà Content-Type su application/json se i dati inviati sono in formato JSON; altrimenti 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 payload *data* facoltativo e restituisce la risposta.
delete(url, data)	Invia una richiesta HTTP DELETE all'URL con payload *data* facoltativo e restituisce la risposta.
getHeaders(<asArray>)	Restituisce l'oggetto dei campi di intestazione HTTP ricevuti. Il parametro asArray può essere impostato su "true" (ad esempio, getHeaders(true)), "false" oppure può non essere definito. Se impostato su "true", i valori dei campi di intestazione HTTP ricevuti verranno restituiti come array; questo dovrebbe essere usato 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 di intestazione HTTP ricevuti verranno restituiti come stringhe.
get(url, data)	Invia una richiesta HTTP GET all'URL con 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 payload *data* facoltativo e restituisce la risposta.
put(url, data)	Invia una richiesta HTTP PUT all'URL con payload *data* facoltativo e restituisce la risposta.
post(url, data)	Invia una richiesta HTTP POST all'URL con 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 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').

2 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.
`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`.

Method	Description
`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 creato 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). Parametri: `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). Parametri: `name` - (string) Nome della proprietà.
`getText()`	Restituisce una stringa con il valore testuale del testo dell'elemento.
`click()`	Fa clic su un elemento.
`clear()`	Cancella il contenuto di un elemento modificabile.
`sendKeys(keys)`	Invia tasti. Parametri: `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.

6 Preelaborazione da CSV a JSON

Panoramica

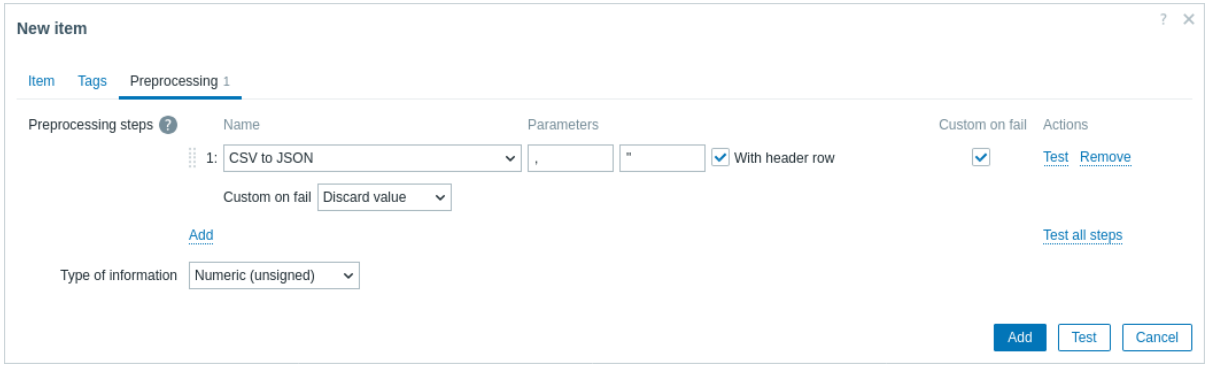
In questo passaggio di preprocessing è possibile convertire i dati di un file CSV nel formato JSON. È supportato in:

- item (prototipi di item)
- regole di low-level discovery

Configuration

To configure a CSV to JSON preprocessing step:

- Go to the Preprocessing tab in item/discovery rule configuration
- Click on *Add*
- Select the *CSV to JSON* option



The first parameter allows to set a custom delimiter. Note that if the first line of CSV input starts with "Sep=" and is followed by a single UTF-8 character then that character will be used as the delimiter in case the first parameter is not set. If the first parameter is not set and a delimiter is not retrieved from the "Sep=" line, then a comma is used as a separator.

The second optional parameter allows to set a quotation symbol.

If the *With header row* checkbox is marked, the header line values will be interpreted as column names (see Header processing for more information).

If the *Custom on fail* checkbox is marked, the item will not become unsupported in case of a failed preprocessing step. Additionally custom error handling options may be set: discard the value, set a specified value or set a specified error message.

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"
   }
]
```

3 Tipi di oggetti

Panoramica

I tipi di item coprono vari metodi di acquisizione dei dati dal sistema. Ogni tipo di item dispone di un proprio insieme di chiavi item supportate e parametri richiesti.

I seguenti tipi di item sono attualmente offerti da Zabbix:

- Controlli Zabbix agent
- Controlli SNMP agent
- Trap SNMP
- Controlli IPMI
- Controlli semplici
  - Monitoraggio VMware
- Monitoraggio dei file di log
- Item calcolati
  - Calcoli aggregati
- Controlli interni di Zabbix
- Controlli SSH
- Controlli Telnet
- Controlli esterni
- Item trapper
- Monitoraggio JMX
- Controlli ODBC
- Item dipendenti
- Controlli HTTP
- Controlli Prometheus
- Item script
- Item browser

I dettagli per tutti i tipi di item sono inclusi nelle sottopagine di questa sezione. Sebbene i tipi di item offrano molte opzioni per la raccolta dei dati, esistono ulteriori possibilità tramite parametri utente o moduli caricabili.

Alcuni controlli vengono eseguiti solo da Zabbix server (come monitoraggio senza agent), mentre altri richiedono Zabbix agent o persino Zabbix Java gateway (con il monitoraggio JMX).

Attention:
Se un particolare 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 possono essere impostate 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 da collegare.

Tutti gli items che restituiscono testo (tipi di informazione character, log, text) possono anche restituire solo spazi vuoti (ove applicabile), impostando il valore restituito su una stringa vuota (supportato dalla versione 2.0).

1 Agente Zabbix

Panoramica

Questa sezione fornisce dettagli sulle chiavi degli item che utilizzano la comunicazione con Zabbix agent per la raccolta dei dati.

Esistono controlli agent passivi e attivi. Quando si configura un item, è possibile selezionare il tipo richiesto:

- *Zabbix agent* - per i controlli passivi
- *Zabbix agent (active)* - per i controlli attivi

Si noti che tutte le chiavi degli item supportate da Zabbix agent sono supportate anche dalla nuova generazione di Zabbix agent 2. Vedere le chiavi degli item aggiuntive che è possibile utilizzare solo con agent 2.

Chiavi item supportate

Le chiavi item che è possibile utilizzare con Zabbix agent sono elencate di seguito.

Le chiavi item sono elencate senza parametri e informazioni aggiuntive. Fare clic sulla chiave item per visualizzare tutti i dettagli.

Chiave 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 log
log.count	Il conteggio delle righe corrispondenti in un file di log monitorato.	
logrt	Il monitoraggio di un file di log sottoposto a rotazione.	
logrt.count	Il conteggio delle righe corrispondenti in un file di log monitorato sottoposto a rotazione.	
modbus.get	Legge dati Modbus.	Modbus
net.dns	Controlla lo stato di un servizio DNS.	Rete
net.dns.perf	Controlla 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.in	Le statistiche del traffico in ingresso su un'interfaccia di rete.	
net.if.out	Le statistiche del traffico in uscita su un'interfaccia di rete.	
net.if.total	La somma delle statistiche del traffico in ingresso e in uscita su un'interfaccia di rete.	
net.tcp.listen	Controlla se questa porta TCP è nello stato LISTEN.	
net.tcp.port	Controlla se è possibile stabilire una connessione TCP alla porta specificata.	
net.tcp.service	Controlla se un servizio è in esecuzione e accetta connessioni TCP.	
net.tcp.service.perf	Controlla le prestazioni di un servizio TCP.	
net.tcp.socket.count	Restituisce il numero di socket TCP che corrispondono ai parametri.	
net.udp.listen	Controlla se questa porta UDP è nello stato LISTEN.	
net.udp.service	Controlla se un servizio è in esecuzione e risponde alle richieste UDP.	
net.udp.service.perf	Controlla 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.	

Chiave item	Descrizione	Gruppo di item
proc.mem	La memoria utilizzata dal processo in byte.	
proc.num	Il numero di processi.	
sensor	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 conteggio dei cambi di contesto.	
system.cpu.util	La percentuale di utilizzo della CPU.	
system.hostname	Il nome del host del sistema.	
system.hw.chassis	Le informazioni sul telaio.	
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 sul 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 minore e maggiore, ecc.).	
system.sw.packages	L'elenco dei pacchetti installati.	
system.sw.packages.get	Un elenco dettagliato dei pacchetti installati.	
system.swap.in	Le statistiche di swap-in (dal dispositivo alla memoria).	
system.swap.out	Le statistiche di 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.read	Le statistiche di lettura del disco.	
vfs.dev.write	Le statistiche di scrittura del disco.	
vfs.dir.count	Il conteggio delle 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 con l'algoritmo UNIX cksum.	
vfs.file.contents	Recupera il contenuto di un file.	
vfs.file.exists	Controlla 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 nel file.	
vfs.file.regmatch	Trova una stringa nel file.	
vfs.file.size	La dimensione del file.	
vfs.file.time	Le informazioni temporali del file.	
vfs.fs.discovery	L'elenco dei filesystem montati con il relativo tipo e le opzioni di montaggio.	
vfs.fs.get	L'elenco dei filesystem montati con il relativo tipo, lo spazio disco disponibile, le statistiche inode e le opzioni di montaggio.	
vfs.fs.inode	Il numero o la percentuale di inode.	
vfs.fs.size	Lo spazio disco in byte o in percentuale del totale.	
vm.memory.size	La dimensione della memoria in byte o in percentuale del totale.	Memoria virtuale
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 del host dell'agent.	Zabbix
agent.hostname	Il nome del host dell'agent.	

Chiave item	Descrizione	Gruppo di item
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 un insieme di metriche interne di Zabbix server o proxy. Se sono specificati `ip` e `port`, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.	
zabbix.stats	Restituisce il numero di item monitorati nella coda che sono in ritardo su Zabbix server o proxy. Se sono specificati `ip` e `port`, le metriche vengono recuperate dall'istanza remota; altrimenti, 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;
- **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** - valori possibili: *all* (predefinito) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati);
- **output** - un template facoltativo di 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 l'N-esimo gruppo corrispondente (oppure 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 meno recenti per fare in modo che vengano analizzate le righe più recenti entro "maxdelay" secondi. 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 tempo 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 controllo attivo.
- 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 globale, tranne 'Result is TRUE', 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 di inviarlo 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 numero di righe corrispondenti in un file di log monitorato.
 Valore restituito: *Integer*.
 Vedi 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) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati).
- **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 meno recenti per fare in modo che vengano analizzate le righe più recenti entro "maxdelay" secondi. 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 tempo 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. Vedi 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 è mancante 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 sottoposto a rotazione.
 Valore restituito: *Log*.
 Vedi piattaforme supportate.

Parametri:

- **file regexp** - il percorso assoluto del file, con il nome del file specificato usando una espressione regolare. Nota che l'espressione regolare si applica solo al nome del file e non deve necessariamente corrispondere all'intero nome (ad esempio, /path/to/agent corrisponderà a zabbix_agentd.log).
- **regexp** - una espressione regolare che descrive il modello di contenuto richiesto.
- **encoding** - l'identificatore della code page.
- **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** - valori possibili: *all* (predefinito) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati).

- **output** - un modello facoltativo di 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 l'N-esimo gruppo corrispondente (oppure con una stringa vuota se N supera il numero di gruppi catturati).
- **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 meno recenti per fare in modo che vengano analizzate le righe più recenti entro "maxdelay" secondi. Leggi le note su maxdelay prima di usarlo!
- **options** - il tipo di rotazione del file di log e altre opzioni. Valori possibili:
rotate (predefinito),
copytruncate - nota che *copytruncate* non può essere usato insieme a *maxdelay*. In questo caso *maxdelay* deve essere 0 oppure non specificato; vedi le note su copytruncate,
mtime-reread - record non univoci, rilettura se cambiano l'ora di modifica o la dimensione (usato per impostazione predefinita se nessuna opzione `mtime-*` è impostata esplicitamente),
mtime-noreread - record non univoci, rilettura solo se cambia la dimensione (ignora il cambiamento dell'ora 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. Vedi anche le note aggiuntive sui file persistenti.

Commenti:

- L'item deve essere configurato come controllo attivo.
- La rotazione dei log si basa sull'ora dell'ultima modifica dei file.
- Nota che logrt è progettato per funzionare con un solo file di log attualmente attivo, con diversi altri file inattivi corrispondenti già ruotati. Se, ad esempio, una directory contiene molti file di log attivi, dovrebbe essere creato un item logrt separato per ciascuno di essi. Altrimenti, se un item logrt rileva troppi file, ciò può portare all'esaurimento della memoria e al crash del monitoraggio.
- Se `output` viene lasciato vuoto, viene restituita l'intera riga contenente il testo corrispondente. Nota che tutti i tipi di espressione regolare globale, eccetto 'Result is TRUE', 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 dei 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 normali possono essere tollerate in altre posizioni, tranne in quella che separa il percorso della directory dei log e l'espressione regolare del file di log (vedi gli esempi sotto).
  - Sugli altri sistemi, deve essere una barra normale (/).

Esempi per Windows:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #questo item raccoglierà dati dai file di lo
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #questo item raccoglierà da
```

Esempi per altri sistemi:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #questo item corrisponderà a un file come "logfile1"
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #questo item raccoglierà dati d
```

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] #questo i
```

Esempio di utilizzo del parametro `output` per riscrivere un record di log prima di inviarlo al 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 soggetto a rotazione.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Parametri:

- **file regexp** - il percorso assoluto del file, con il nome del file specificato usando un'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** - un'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) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati).

- **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 meno recenti per fare in modo che vengano analizzate le righe più recenti 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 oppure non specificato; vedere le note su copytruncate,
mtime-reread - record non univoci, rilettura se l'ora di modifica o la dimensione cambiano (predefinito),
mtime-noreread - record non univoci, rilettura solo se cambia la dimensione (ignora il cambiamento dell'ora 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.
- Le righe corrispondenti vengono conteggiate nelle nuove righe dall'ultimo controllo del log da parte dell'agent e dipendono quindi dall'intervallo di aggiornamento dell'item.
- La rotazione del log si basa sull'ora dell'ultima 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 normali possono essere tollerate in altre posizioni, tranne in 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 normale (/).

Esempi per Windows:

```
logrt.count["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will count the number of ma
logrt.count["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #this item will count
```

Esempi per altri sistemi:

```
logrt.count["/home/zabbix/logs/.*","^logfile[0-9]{1,3}$",,100] #this item will count the number of matches
logrt.count["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #this item will count the
```

modbus.get[endpoint,<slave id>,<function>,<address>,<count>,<type>,<endianness>,<offset>]

 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 dell'endianness;
- **offset** - il numero di registri, a partire da 'address', i cui risultati verranno scartati.

In 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 non è riuscita (il server DNS non ha risposto o ha restituito un errore); 1 - la risoluzione DNS è riuscita.
 Vedere 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 utilizzato 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 che 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[198.51.100.1,example.com,MX,2,1]

net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 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 se non si utilizza 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 se non si utilizza Zabbix agent 2) - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** (ignorato su Windows se non si utilizza Zabbix agent 2) - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo utilizzato 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 inverso che non inverso (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.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.8.0.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 granulare del monitoraggio DNS. Valore restituito: una stringa di caratteri con il tipo di informazione richiesto.
 Vedere le piattaforme supportate.

Parametri:

- **ip** (ignorato su Windows se non si utilizza 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 se non si utilizza Zabbix agent 2) - il timeout della richiesta in secondi (il valore predefinito è 1 secondo);
- **count** (ignorato su Windows se non si utilizza Zabbix agent 2) - il numero di tentativi per la richiesta (il valore predefinito è 2);
- **protocol** - il protocollo utilizzato 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 la low-level discovery.
 Valore restituito: *stringa JSON*.
 Piattaforme supportate: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

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 utilizzare questa chiave con il passaggio di preprocessing *Change per second* per ottenere le statistiche dei 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 di 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 i 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 portante rilevate dal driver del dispositivo
compressed - numero di pacchetti compressi trasmessi dal driver del dispositivo

Commenti:

- È possibile usare questa chiave con il passaggio di preprocessing *Change per second* per ottenere le statistiche dei 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 di 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 utilizzare questa chiave con il passaggio di preprocessing *Change per second* per ottenere le statistiche dei byte al secondo.
- La modalità *dropped* è supportata solo su Linux, HP-UX. I pacchetti scartati sono supportati solo se sia `net.if.in` che `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]

 Controlla 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

Sui 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 riuscita.
 Vedere piattaforme supportate.

Parametri:

- **ip** - l'indirizzo IP o il nome DNS (il valore predefinito è 127.0.0.1);
- **port** - il numero di porta.

Commenti:

- Per un semplice test delle prestazioni TCP utilizzare `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Questi controlli possono generare messaggi aggiuntivi nei file di log dei demoni di sistema (di solito vengono registrate le sessioni SMTP e SSH).

Esempio:

```
net.tcp.port[,80] #questo item può essere utilizzato per verificare la disponibilità del server web in ese
```

net.tcp.service[service,<ip>,<port>]

 Verifica se un servizio è in esecuzione e accetta connessioni TCP.
 Valori restituiti: 0 - il servizio non è attivo; 1 - il servizio è in esecuzione.
 Vedere 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 utilizzato il numero di porta standard del servizio).

Commenti:

- 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.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 login (':' alla fine).

Esempio:

```
net.tcp.service[ftp,,45] #questo item può essere utilizzato per testare la disponibilità del server FTP su
```

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 (il valore predefinito è 127.0.0.1);
- **port** - il numero di porta (per impostazione predefinita viene utilizzato il numero di porta standard del servizio).

Commenti:

- 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.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] #questo item può essere utilizzato per testare la velocità della risposta inizia
```

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 disponibile; 1 - il servizio è in esecuzione.
 Vedere piattaforme supportate.

Parametri:

- **service** - *ntp* (vedere 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 utilizzato 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 disponibile; 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 utilizzato 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 della CPU del processo.
 Valore restituito: *Float*.
 Piattaforme supportate: Linux, Solaris[6].

Parametri:

- **name** - il nome del processo (il valore predefinito è *tutti i processi*);
- **user** - il nome dell'utente (il valore predefinito è *tutti gli utenti*);
- **type** - il tipo di utilizzo della CPU: *total* (predefinito), *user* oppure *system*;
- **cmdline** - filtro per riga di comando (è una espressione regolare);
- **mode** - la modalità di raccolta dei dati: *avg1* (predefinito), *avg5* oppure *avg15*;
- **zone** - la zona di destinazione: *current* (predefinito) oppure *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 della CPU di un processo che usa completamente due core è del 200%.
- I dati di utilizzo della 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 a cui non si è avuto accesso nelle ultime 24 ore vengono rimosse dal collector.

- Quando il parametro zone è impostato su *current* (o sul valore predefinito), nel caso in cui l'agent sia stato compilato su un Solaris senza supporto per le zone ma sia in esecuzione 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, in questo caso *all* è supportato.

Esempi:

```
proc.cpu.util[,root] #utilizzo della CPU di tutti i processi in esecuzione con l'utente "root"
proc.cpu.util[zabbix_server,zabbix] #utilizzo della CPU di tutti i processi zabbix_server in esecuzione co
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 L'elenco dei processi del sistema operativo e dei relativi parametri. Può essere utilizzato per il low-level discovery.
 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 per riga di comando (è una espressione regolare). Questo parametro non è supportato in Windows; sulle altre piattaforme non è supportato se mode è impostato su 'summary'.
- **mode** - valori possibili:
process (predefinito), *thread* (non supportato per NetBSD), *summary*. Vedere l'elenco dei parametri di 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, errore di chiamata di sistema), verrà restituito −1.
- Vedere le note sulla selezione dei processi con i parametri name e cmdline (specifico per Linux).

Esempi:

```
proc.get[zabbix_server,zabbix,,process] #list of all zabbix_server processes running under the zabbix user
proc.get[java,,,thread] #list of all Java processes, returns one entry per thread
proc.get[,zabbix,,summary] #combined data for processes of each type running under the zabbix user, return
```

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 (il valore predefinito è *tutti i processi*);
- **user** - il nome utente (il valore predefinito è *tutti gli utenti*);
- **mode** - valori possibili: *avg*, *max*, *min* o *sum* (predefinito);
- **cmdline** - filtro per 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.

Vedere le note sulla selezione dei processi con i parametri name e cmdline (specifico per Linux).

Quando questo item viene richiamato dalla riga di comando e contiene un parametro della riga di comando (ad esempio utilizzando la modalità di test dell'agent: zabbix_agentd −t proc.mem[,,,apache2]), verrà conteggiato un processo aggiuntivo, poiché l'agent conterà sé stesso.

Esempi:

```
proc.mem[,root] #la memoria utilizzata da tutti i processi eseguiti con l'utente "root"
proc.mem[zabbix_server,zabbix] #la memoria utilizzata da tutti i processi zabbix_server eseguiti con l'ute
proc.mem[,oracle,max,oracleZABBIX] #la memoria utilizzata dal processo che consuma più memoria eseguito co
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Il numero di processi.
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parametri:

- **name** - il nome del processo (il valore predefinito è *tutti i processi*);
- **user** - il nome utente (il valore predefinito è *tutti gli utenti*);

- **state** - valori possibili:
all (predefinito),
disk - sospensione non interrompibile,
run - in esecuzione,
sleep - sospensione interrompibile,
trace - arrestato,
zomb - zombie;
- **cmdline** - filtro per 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 in più, poiché l'agent conterà anche sé stesso.
- Impostando il parametro `zone` su `current` (o lasciando il valore predefinito), nel caso in cui l'agent sia stato compilato su un Solaris senza supporto per le zone ma sia in esecuzione 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, in questo caso *all* è supportato.
- Vedere le note sulla selezione dei processi con i parametri `name` e `cmdline` (specifico per Linux).

Esempi:

```
proc.num[,mysql] #il numero di processi in esecuzione con l'utente mysql
proc.num[apache2,www-data] #il numero di processi apache2 in esecuzione con l'utente www-data
proc.num[,oracle,sleep,oracleZABBIX] #il numero di processi in stato sleep in esecuzione con l'utente Orac
```

sensor[device,sensor,<mode>]

 Lettura del 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* oppure *min*.

Commenti:

- Legge /proc/sys/dev/sensors su Linux 2.4.
- Legge /sys/class/hwmon su Linux 2.6+. Vedere una descrizione più dettagliata dell'item sensor su Linux.
- Legge l'MIB *hw.sensors* su OpenBSD.
- Su Linux 2.4, se si specifica mode, allora device e sensor vengono trattati come espressioni regolari.
- Su Linux 2.6+ e OpenBSD, oppure se mode viene omesso, device e sensor vengono trattati come valori letterali.

Esempi:

```
sensor[w83781d-i2c-0-2d,temp1]
sensor[cpu0,temp0] #the temperature of one CPU
sensor["cpu[0-2]$",temp,avg] #the average temperature of the first three CPUs
```

system.boottime

 L'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 del host del sistema.
 Valore restituito: *String*.
 Vedere le 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) oppure *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>]

 L'elenco degli indirizzi MAC.
 Valore restituito: *String*.
 Piattaforme supportate: Linux.

Parametri:

- **interface** - *all* (predefinito) oppure un'espressione regolare;
- **format** - *full* (predefinito) oppure *short*

Commenti:

- Elenca gli indirizzi MAC delle interfacce il cui nome corrisponde all'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 oppure *local* - l'ora nel formato 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Deve essere utilizzato come controllo passivo per Zabbix agent; in Zabbix agent 2 può essere configurato come controllo attivo.

Esempio:

```
system.localtime[local] #creare un item utilizzando questa chiave e quindi usarlo per visualizzare l'ora d
```

system.run[command,<mode>]

 Esegue il comando specificato sul host.
 Valore restituito: risultato *Text* del comando oppure 1 - con `mode` impostato su *nowait* (indipendentemente dal risultato del comando).
 Vedere le piattaforme supportate.

Parametri:

- **command** - comando da eseguire;
- **mode** - valori possibili: *wait* - attende la fine dell'esecuzione (predefinito) oppure *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 del tipo di dato `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.
- Vedere 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 di 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 di file al secondo (float)
fo - page-out di file al secondo (float)
pi - pagine trasferite in memoria dallo spazio di paging (float)
po - pagine trasferite nello spazio di paging (float)
fr - pagine liberate (sostituzione di pagina) (float)
sr - pagine analizzate dall'algoritmo di sostituzione delle pagine (float)
- **faults,<type>** - frequenza di trap e interrupt:
in - interrupt del dispositivo (float)
sy - chiamate di sistema (float)
cs - cambi di contesto dei thread del kernel (float)
- **cpu,<type>** - dettaglio percentuale dell'utilizzo del tempo del processore:
us - tempo utente (float)
sy - tempo di sistema (float)
id - tempo inattivo (float)
wa - tempo inattivo durante il quale il sistema aveva richieste di 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 dei processori logici verificatasi 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) verso l'unità in byte al secondo (integer)
tps - indica il numero di trasferimenti al secondo emessi verso il disco/nastro fisico (float)

Commenti:

- Tenere presenti le seguenti limitazioni di 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 (percento));
 `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>]

 Le informazioni sul sistema operativo.
 Valore restituito: *String*.
 Piattaforme supportate: Linux, Windows.

Parametro:

- **info** - valori possibili: *full* (predefinito), *short* oppure *name*

Le informazioni 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>]

 L'elenco dei pacchetti installati.
 Valore restituito: *Testo*.
 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):
  - dpkg (dpkg --get-selections)
  - pkgtool (ls /var/log/packages)
  - rpm (rpm -qa)
  - pacman (pacman -Q)
portage
- Se `format` è specificato come *full*, i pacchetti vengono raggruppati per gestore di pacchetti (ogni gestore su una riga separata che inizia con il suo nome tra parentesi quadre).
- Se `format` è specificato come *short*, i pacchetti non vengono raggruppati e sono elencati su una singola 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 una 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 (vedere maggiori dettagli).

system.swap.in[<device>,<type>]

 Le statistiche di swap-in (dal dispositivo alla memoria).
 Valore restituito: *Intero*.
 Piattaforme supportate: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - specifica il dispositivo utilizzato per lo swapping (solo Linux) oppure *all* (predefinito);
- **type** - valori possibili: *count* (numero di swap-in, predefinito sulle piattaforme non Linux), *sectors* (settori trasferiti in swap in ingresso) oppure *pages* (pagine trasferite in swap in ingresso, 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)

- Si noti che *pages* funzionerà solo se il dispositivo non è stato specificato.
- 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 utilizzato per lo swapping (solo Linux) oppure *all* (predefinito);
- **type** - valori possibili: *count* (numero di swap-out, predefinito sulle piattaforme non Linux), *sectors* (settori trasferiti in swap), oppure *pages* (pagine trasferite 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)
- Si noti che *pages* funzionerà solo se device non è stato specificato.
- 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 del totale.
 Valore restituito: *Integer* - per i byte; *Float* - per la percentuale.
 Piattaforme supportate: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parametri:

- **device** - specifica il dispositivo utilizzato per lo swapping (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 utilizzato, in percentuale), *total* (spazio di swap totale) oppure *used* (spazio di swap utilizzato).

Commenti:

- Si noti che *pfree* e *pused* non sono supportati su Windows se la dimensione dello swap è 0.
- Se il dispositivo non è specificato, Zabbix agent terrà conto solo dei dispositivi di swap (file), mentre la memoria fisica verrà ignorata. Ad esempio, sui sistemi Solaris il comando `swap -s` include una parte della memoria fisica e i 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.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* oppure *bps* (*sps*, *ops*, *bps* indicano rispettivamente: settori, operazioni, byte al secondo);
- **mode** - valori possibili: *avg1* (media di un minuto, predefinito), *avg5* oppure *avg15*. Questo parametro è supportato solo con `type` in: 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 di dispositivo relativi (ad esempio `sda`) così come il prefisso opzionale `/dev/` (ad esempio `/dev/sda`).
- I volumi logici LVM sono supportati.
- 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 individuali 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* indicano rispettivamente: settori, operazioni, byte al secondo);
- **mode** - valori possibili: *avg1* (media di un minuto, predefinito), *avg5* o *avg15*. Questo parametro è supportato solo con `type` in: 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 utilizzare nomi di dispositivo relativi (ad esempio, `sda`) così come il prefisso opzionale `/dev/` (ad esempio, `/dev/sda`).
- I volumi logici LVM sono supportati.
- 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 individuali 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 conteggio delle voci della directory.
 Valore restituito: *Integer*.
 Vedere le piattaforme supportate.

Parametri:

- **dir** - il percorso assoluto della directory;
- **regex incl** - una espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da includere; include tutto se vuoto (valore predefinito);
- **regex excl** - una 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 regolare, *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), cioè "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 conteggiate.
- **max depth** - la profondità massima delle sottodirectory da attraversare:
-1 (predefinito) - illimitata,
0 - nessuna discesa nelle sottodirectory.
- **min size** - la dimensione minima (in byte) affinché un file venga conteggiato. I file più piccoli non verranno conteggiati. È possibile usare i suffissi di memoria.
- **max size** - la dimensione massima (in byte) affinché un file venga conteggiato. I file più grandi non verranno conteggiati. È possibile usare i suffissi di memoria.
- **min age** - l'età minima (in secondi) della voce della directory affinché venga conteggiata. Le voci più recenti non verranno conteggiate. È possibile usare i suffissi di tempo.
- **max age** - l'età massima (in secondi) della voce della directory affinché venga conteggiata. Le voci di questa età o più vecchie non verranno conteggiate (tempo di modifica). È possibile usare i suffissi di tempo.
- **regex excl dir** - una 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%, $HOME e %TEMP%, non sono supportate.
- Le pseudo-directory "." e ".." non vengono mai conteggiate.
- I collegamenti simbolici non vengono mai seguiti durante l'attraversamento delle directory.
- Sia `regex incl` sia `regex excl` vengono applicati a file e directory durante il calcolo del conteggio delle voci, ma vengono ignorati nella selezione delle sottodirectory da attraversare (se `regex incl` è "(?i)^.+\.zip$" e `max depth` non è impostato, allora 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 (3 sec). Poiché l'attraversamento di directory di grandi dimensioni può richiedere più tempo, non verrà restituito alcun dato e l'item diventerà unsupported. Non verrà restituito un conteggio parziale.
- Quando si filtra per dimensione, solo i file regolari hanno dimensioni significative. In Linux e BSD, anche le directory hanno dimensioni diverse da zero (tipicamente alcuni KB). I dispositivi hanno dimensione zero; ad esempio, la dimensione di **/dev/sda1** non riflette la dimensione della rispettiva partizione. Pertanto, quando si usano <`min_size`> e <`max_size`>, si consiglia di specificare <`types_incl`> come "*file*", per evitare risultati inattesi.

Esempi:

```
vfs.dir.count[/dev] #monitors the number of devices in /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #monitors the number of files in a temporary directo
```

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 le piattaforme supportate.

Parametri:

- **dir** - il percorso assoluto della directory;
- **regex incl** - una espressione regolare che descrive il modello del nome dell'entità (file, directory, collegamento simbolico) da includere; include tutto se vuoto (valore predefinito);
- **regex excl** - una 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 di directory da elencare, valori possibili: *file* - file regolare, *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), cioè "file,dir,sym,sock,bdev,cdev,fifo". Più tipi devono essere separati da virgole e racchiusi tra virgolette.
- **types excl** - i tipi di voci di directory (vedere `types incl`) da NON elencare. Se un tipo di voce è presente sia in `types incl` che in `types excl`, le voci di directory di questo tipo NON vengono elencate.
- **max depth** - la profondità massima delle sottodirectory da attraversare:
-1 (predefinito) - illimitata,
0 - nessuna discesa nelle sottodirectory.
- **min size** - la dimensione minima (in byte) affinché un file venga elencato. I file più piccoli non verranno elencati. È possibile usare i suffissi di memoria.
- **max size** - la dimensione massima (in byte) affinché un file venga elencato. I file più grandi non verranno elencati. È possibile usare i suffissi di memoria.
- **min age** - l'età minima (in secondi) della voce di directory da elencare. Le voci più recenti non verranno elencate. È possibile usare i suffissi di tempo.
- **max age** - l'età massima (in secondi) della voce di directory da elencare. Le voci così vecchie e più vecchie non verranno elencate (ora di modifica). È possibile usare i suffissi di tempo.
- **regex excl dir** - una 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%, $HOME e %TEMP%, non sono supportate.
- Le pseudo-directory "." e ".." non vengono mai elencate.
- I collegamenti simbolici non vengono mai seguiti durante l'attraversamento delle directory.
- Sia `regex incl` che `regex excl` vengono applicati a file e directory durante la generazione dell'elenco delle voci, ma vengono ignorati nella selezione delle sottodirectory da attraversare (se `regex incl` è "(?i)^.+\.zip$" e `max depth` non è impostato, allora 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 verrà restituito alcun dato e l'item diventerà unsupported. Non verrà restituito un elenco parziale.
- Quando si filtra per dimensione, solo i file regolari hanno dimensioni significative. In Linux e BSD, anche le directory hanno dimensioni diverse da zero (tipicamente alcuni KB). I dispositivi hanno dimensione zero, ad esempio la dimensione di **/dev/sda1** non riflette la dimensione della partizione corrispondente. Pertanto, quando si usano `min size` e `max size`, si consiglia di specificare `types incl` come "*file*", per evitare risultati inattesi.

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 di tipo UNIX.

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);
- **mode** - valori possibili: *apparent* (predefinito) - ottiene le dimensioni apparenti dei file anziché l'utilizzo del disco (si comporta come `du -sb dir`), *disk* - ottiene l'utilizzo del disco (si comporta come `du -s -B1 dir`). A differenza del comando du, l'item vfs.dir.size considera i file nascosti nel calcolo della 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** - non scende nelle sottodirectory.
- **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.
- Vengono calcolate solo le directory con almeno il permesso di lettura per l'utente *zabbix*. Per le directory con solo 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 del file dipende dal supporto per file di grandi dimensioni.

Esempi:

```
vfs.dir.size[/tmp,log] #calculates the size of all files in /tmp containing 'log' in their names
vfs.dir.size[/tmp,log,^.+\.old$] #calculates the size of all files in /tmp containing 'log' in their names
```

vfs.file.cksum[file,<mode>]

 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* oppure *sha256*.

Commenti:

- In 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 (crc32/md5/sha256 rispettivamente):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Recupera il contenuto di un file[7].
 Valore restituito: *Testo*.
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **encoding** - l'identificatore della code page.

Commenti:

- In 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.
- Se il file è vuoto o contiene solo caratteri LF/CR, viene restituita una stringa vuota.
- Il byte order mark (BOM) è escluso dall'output.

Esempio:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Controlla se il file esiste.
 Valore restituito: 0 - non trovato; 1 - esiste un file del tipo specificato.
 Vedere le 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:

- In Windows, questo item segue i percorsi UNC, che possono rappresentare un rischio per la sicurezza.
- Più tipi devono essere separati da una virgola e l'intero insieme deve essere racchiuso tra virgolette "".
- Se lo stesso tipo è presente sia in <types_incl> che in <types_excl>, i file di questo tipo vengono esclusi.
- Il limite della 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.getfile

 Restituisce informazioni su un file.
 Valore restituito: *stringa JSON*.
 Vedere le piattaforme supportate.

Parametro:

- **file** - il percorso completo del file

Commenti:

- In Windows, questo item segue i percorsi UNC, il che può rappresentare un rischio per la sicurezza.
- Tipi di file supportati sui sistemi di tipo 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] #restituisce un JSON con informazioni sul file /etc/passwd (tipo, utente, permes
```

vfs.file.md5sumfile

 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:

- In 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*.
 Vedere le 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 comportare un rischio per la sicurezza.
- Il limite della 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.permissionsfile

 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 come specificato dal parametro facoltativo `output`.
 Vedi le 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 in cui effettuare la ricerca (per impostazione predefinita, la prima riga del file);
- **end line** - il numero dell'ultima riga in cui effettuare la ricerca (per impostazione predefinita, l'ultima riga del file);
- **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 l'N-esimo gruppo corrispondente (oppure con una stringa vuota se N supera il numero di gruppi catturati).

Commenti:

- In 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.
- Viene restituita solo la prima riga corrispondente.
- Se nessuna riga corrisponde all'espressione, viene restituita una stringa vuota.
- 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 modello richiesto;
- **encoding** - l'identificatore della code page;
- **start line** - il numero della prima riga in cui cercare (per impostazione predefinita, la prima riga del file);
- **end line** - il numero dell'ultima riga in cui cercare (per impostazione predefinita, l'ultima riga del file).

Commenti:

- In 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.
- 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*.
 Vedere le piattaforme supportate.

Parametri:

- **file** - il percorso completo del file;
- **mode** - valori possibili: *bytes* (predefinito) oppure *lines* (anche le righe vuote vengono conteggiate).

Commenti:

- In Windows, questo item segue i percorsi UNC, che possono 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>]

 Le informazioni temporali 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,
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:

- In 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.

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

 L'elenco dei file system montati con il loro tipo, lo spazio su disco disponibile, le statistiche degli inode e le opzioni di montaggio. 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.

Commenti:

- Vengono riportati anche i file system con un numero di inode pari a zero, come può accadere per i file system con inode dinamici (ad esempio btrfs).
- Vedi anche: Discovery dei file system montati.

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>]

 Ottiene il contenuto di una pagina web.
 Valore restituito: codice sorgente della pagina web come testo (incluse le intestazioni).
 Vedere le 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 considerato come *http*. Se viene specificato un URL, `path` e `port` devono essere vuoti. La specifica di nome utente/password durante la connessione a server che richiedono 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)

Commenti:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Ma per un indirizzo IPv6, Zabbix agent deve essere compilato con il supporto IPv6 abilitato.

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.perf[host,<path>,<port>]

 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 (come `scheme://host:port/path`, dove solo *host* è obbligatorio). Schemi URL consentiti: *http*, *https*[4]. Uno schema mancante verrà trattato come *http*. Se viene specificato un URL, `path` e `port` devono essere vuoti. La specifica di nome utente/password durante la connessione a server che richiedono 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)

Commenti:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Ma per un indirizzo IPv6, Zabbix agent deve essere compilato con il supporto IPv6 abilitato.

Esempi:

```
web.page.perf[www.example.com,index.php,80]
web.page.perf[https://www.example.com]
```

web.page.regexp[host,<path>,<port>,regexp,<length>,<output>]

 Trova una stringa nella pagina web.
 Valore restituito: la stringa corrispondente, oppure quanto specificato dal parametro opzionale `output`.
 Vedere le 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 non è specificato, verrà considerato *http*. Se viene specificato un URL, `path` e `port` devono essere vuoti. La specifica di nome utente/password durante la connessione a server che richiedono 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)
- **regexp** - un'espressione regolare che descrive il modello richiesto;
- **length** - il numero massimo di caratteri da restituire;
- **output** - un modello opzionale di 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 l'N-esimo gruppo corrispondente (oppure con una stringa vuota se N supera il numero di gruppi acquisiti).

Commenti:

- Questo item diventa non supportato se la risorsa specificata in `host` non esiste o non è disponibile.
- `host` può essere un nome host, un nome di dominio, un indirizzo IPv4 o IPv6. Tuttavia, per un indirizzo IPv6, Zabbix agent deve essere compilato con il supporto IPv6 abilitato.
- L'estrazione del contenuto tramite il parametro `output` avviene sull'agent.

Esempi:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 I metadati del host dell'agent.
 Valore restituito: *String*.
 Vedere piattaforme supportate.

Restituisce il valore dei parametri HostMetadata o HostMetadataItem, oppure una stringa vuota se nessuno di essi è definito.

agent.hostname

 Il nome host dell'agent.
 Valore restituito: *String*.
 Vedere 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'host corrente.

agent.ping

 Il controllo della disponibilità dell'agent.
 Valore restituito: Nothing - 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*.
 Vedere le piattaforme supportate.

Parametri:

- **ip** - l'elenco di IP/DNS/maschere di rete di 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:

- Questo item restituisce un insieme selezionato di metriche interne. Per i dettagli, vedere Monitoraggio remoto delle statistiche di Zabbix.
- Si noti che la richiesta delle statistiche sarà accettata solo dagli indirizzi elencati nel parametro 'StatsAllowedIP' del server/proxy nell'istanza di destinazione.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Restituisce il numero di item monitorati nella coda che sono in ritardo sullo 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*.
 Vedere le piattaforme supportate.

Parametri:

- **ip** - l'elenco di IP/DNS/maschere di rete di 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)

Si noti che la richiesta di statistiche sarà accettata solo dagli indirizzi elencati nel parametro 'StatsAllowedIP' del server/proxy nell'istanza di destinazione.

Note a piè di pagina

[1] Nota specifica per Linux. Zabbix agent deve avere accesso in sola lettura al filesystem */proc*. Le patch del kernel da 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 "obsolete" se i valori degli item non vengono consultati per più di 3 ore. Ciò può accadere se un sistema ha 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 di 3 ore o superiore, 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, incluse tutte le periferiche a blocchi come sda, sdb e le relative partizioni (sda1, sda2, sdb3...) e i dispositivi multipli (MD raid) basati su tali periferiche/partizioni a blocchi e i volumi logici (LVM) basati su tali periferiche/partizioni a blocchi. In tali casi i valori restituiti devono essere considerati solo come valori relativi (dinamici nel tempo) e non come valori assoluti.

[4] SSL (HTTPS) è supportato solo se agent è compilato con il supporto cURL. Altrimenti l'item diventerà non supportato.

[5] I valori `bytes` ed `errors` non sono supportati per le interfacce loopback sui 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 possono essere restituiti poiché net-snmp include codice legacy di cmu-snmp risalente addirittura al 1997 che, quando non riesce a leggere i valori dei byte dalle statistiche dell'interfaccia, restituisce il contatore dei pacchetti (che esiste sulle interfacce loopback) moltiplicato per un valore arbitrario di 308. Questo presuppone che la lunghezza media di un pacchetto sia di 308 ottetti, il che è una stima molto approssimativa, poiché il limite MTU sui sistemi Solaris per le interfacce loopback è di 8892 byte. Non si deve presumere che questi valori siano 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 abbia i permessi di accesso 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 check (ad esempio, 'vfs.file.contents') nel parametro `encoding`. L'elenco delle codifiche supportate (identificatori di code page) è disponibile nella documentazione di libiconv (GNU Project) oppure nella documentazione di Microsoft Windows SDK per "Code Page Identifiers". Si noti che Microsoft talvolta contrassegna alcune code page come "available only to managed applications", cioè esposte solo tramite il runtime .NET e non disponibili tramite le API native Win32. Zabbix agent implementa una propria logica di conversione della codifica; pertanto, tali code page 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 (oppure 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'. Viene effettuato un tentativo di determinare la codifica corretta utilizzando il byte order mark (BOM) all'inizio del file. Se il BOM non è presente, viene invece applicata la risoluzione standard (vedere sopra).

Risoluzione dei problemi degli item agent

Nel caso dei controlli passivi, per evitare che un item non riceva alcun valore perché la richiesta del server all'agent va prima in timeout, è necessario tenere presente quanto segue:

- Se la versione dell'agent è precedente a quella del server, il valore di *Timeout* nella configurazione dell'item (o il timeout globale) potrebbe dover essere maggiore del valore `Timeout` nel file di configurazione dell'agent.
- Se la versione dell'agent è successiva a quella del server, il valore `Timeout` nel file di configurazione del server potrebbe dover essere maggiore del valore `Timeout` nel file di configurazione dell'agent.

1 Agente Zabbix 2

Zabbix agent 2 supporta tutte le chiavi item supportate per Zabbix agent su Unix e Windows. Questa pagina fornisce dettagli sulle chiavi item aggiuntive, che possono essere utilizzate 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.

Chiave item	Descrizione	Plugin
ceph.df.details	L'utilizzo dei dati del cluster e la distribuzione tra i pool.	Ceph
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. Utilizzato per il low-level discovery.	
docker.data.usage	Informazioni sull'utilizzo corrente dei dati.	
docker.images	Restituisce l'elenco delle immagini.	
docker.images.discovery	Restituisce l'elenco delle immagini. Utilizzato per il low-level discovery.	
docker.info	Le informazioni di sistema.	

Chiave item	Descrizione	Plugin
docker.ping	Verifica se il demone Docker è attivo o meno.	
ember.get	Restituisce il risultato del dispositivo richiesto.	Ember+
memcached.ping	Verifica se una connessione è attiva o meno.	Memcached
memcached.stats	Ottiene l'output del comando STATS.	
mongodb.collection.stats	Restituisce una varietà di statistiche di archiviazione per una determinata collection.	MongoDB
mongodb.cfg.discovery	Restituisce un elenco dei server di configurazione rilevati.	MongoDB
mongodb.collections.discovery	Restituisce un elenco delle collection rilevate.	
mongodb.collections.usage	Restituisce le statistiche di utilizzo per le collection.	
mongodb.connpool.stats	Restituisce informazioni relative alle connessioni in uscita aperte dall'istanza di database corrente verso altri membri del cluster shardato o del replica set.	
mongodb.db.stats	Restituisce le statistiche che riflettono un determinato stato del sistema di 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, utilizzando dati raccolti 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 sul quale 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 di 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
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 e a una replica Always On (primaria o secondaria) e che si trovano sul server al quale è stata stabilita la connessione.	
mssql.mirroring.get	Restituisce le informazioni di mirroring.	
mssql.nonlocal.db.get	Restituisce i database che partecipano a un gruppo di disponibilità e a una replica Always On (primaria o secondaria) situati su altri server (il database non è locale all'istanza SQL Server alla quale 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 di 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 di 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
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 ECC della memoria del dispositivo GPU.	
nvml.device.errors.register	Informazioni sugli errori ECC dei registri del dispositivo GPU.	

Chiave item	Descrizione	Plugin
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.	
nvml.device.memory.bar1.get	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 delle prestazioni del dispositivo GPU.	
nvml.device.power.limit	Limite di potenza del dispositivo GPU.	
nvml.device.power.usage	Consumo di potenza del dispositivo GPU.	
nvml.device.serial	Numero di serie del dispositivo GPU.	
nvml.device.sm.frequency	Frequenza di clock dello streaming multiprocessor 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 di 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 file di dati.	
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 di tablespace.	
oracle.user.info	Restituisce le informazioni sugli utenti Oracle.	
oracle.version	Restituisce la versione del server di database.	
pgsql.autovacuum.count	Il numero di worker autovacuum.	PostgreSQL
pgsql.archive	Le informazioni sui file archiviati.	
pgsql.bgwriter	Il numero combinato di checkpoint per il cluster di database, suddiviso per tipo di checkpoint.	
pgsql.cache.hit	Il tasso di hit della buffer cache 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 con bloat 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 riepilogativi 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 flush lag, il write lag e il replay lag per ciascun processo sender.	
pgsql.replication.process.discovery	Il rilevamento dei nomi dei processi di replica.	

Chiave item	Descrizione	Plugin
pgsql.replication.recovery_role	Lo stato di recovery.	
pgsql.replication.status	Lo stato della replica.	
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 dello slow log da quando Redis è stato avviato.	
smart.attribute.discovery	Restituisce un elenco di attributi dei dispositivi S.M.A.R.T.	S.M.A.R.T.
smart.disk.discovery	Restituisce un elenco di 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

Vedere anche:

- Plugin integrati
- Plugin caricabili

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 uno o più topic specifici (con caratteri jolly) del broker fornito e attende le pubblicazioni.
 Valore restituito: dipende dal contenuto del topic. Se vengono utilizzati caratteri jolly, restituisce il contenuto del topic come 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 utilizzerà `tcp://localhost:1883`. Se il protocollo o la porta vengono omessi, verranno utilizzati 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 controllo attivo (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/` in Ubuntu). Per TLS, utilizzare 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 un numero maggiore di tipi di record e flag personalizzabili.
 Valori restituiti: *oggetto JSON*

Parametri:

- **ip** - 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** - 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 utilizzato 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 con combinazione di maiuscole e minuscole non sono supportati.
- Per le ricerche DNS inverse (quando `type` è impostato su *PTR*), è possibile fornire il nome DNS sia in formato invertito che 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 utilizzati insieme a *noedns0*, poiché entrambi richiedono *edns0*. Si noti che i valori devono essere solo in minuscolo; i valori in maiuscolo o con combinazione di maiuscole e minuscole non sono supportati.
- I nomi di dominio internazionalizzati non sono supportati; utilizzare invece nomi codificati IDNA.
- L'output è un oggetto contenente informazioni sui record DNS in base ai parametri forniti (vedere maggiori dettagli).

Esempi:

```
net.dns.get[192.0.2.0,example.com,DNSKEY,3,3,tcp,"cdflag,rdflag,nsid"]

net.dns.get[,198.51.100.1,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,1.100.51.198.in-addr.arpa,PTR,,,,"cdflag,rdflag,nsid"]

net.dns.get[,2a00:1450:400f:800::200e,PTR,,,,"cdflag,rdflag,nsid"]
net.dns.get[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR,,,,"cdflag,rdfla
```

nvml.device.count

 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 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.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` 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;
- **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 direttamente JSON 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 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.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 "(" (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.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 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.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 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.pdb.info[connString,<user>,<password>,<service>,<database>]

 Le informazioni sui Pluggable Databases (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` 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.pdb.discovery[connString,<user>,<password>,<service>]

 Restituisce l'elenco dei PDB. Utilizzato per il 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 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.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 è presente 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 del 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 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.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 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;
- **lockMaxTime** - la durata massima del blocco della sessione, in secondi, per conteggiare la sessione come bloccata in modo prolungato. 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 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;
- **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` 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.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 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;
- **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` vengono omessi, 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 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.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` 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;
- **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 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` 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.

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 utilizzato 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 utilizzato 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 del 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 utilizzato 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 utilizzato il valore di Plugins.PostgreSQL.Default.Database da postgresql.conf).

pgsql.custom.query[uri,<username>,<password>,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;
- **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 FrozenXID più vecchio del database.
 Valore restituito: *Intero*.

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.db.bloating_tables[uri,<username>,<password>,<database name>]

 Il numero di tabelle con bloating 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 utilizzato il valore di Plugins.PostgreSQL.Default.Database da postgresql.conf).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 L'elenco dei database PostgreSQL. Utilizzato per il 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 utilizzato 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. Utilizzato 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 utilizzato 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 utilizzato il valore di Plugins.PostgreSQL.Default.Database da postgresql.conf).

pgsql.locks[uri,<username>,<password>,<database name>]

 Le informazioni sui lock concessi per database. Utilizzato 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 utilizzato il valore di Plugins.PostgreSQL.Default.Database da postgresql.conf).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 L'età del più vecchio XID.
 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.ping[uri,<username>,<password>,<database name>]

 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:

- **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.queries[uri,<username>,<password>,<database name>,time period]

 Metriche delle query per 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 utilizzato 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 di PostgreSQL;
- **database name** - il nome del database (se omesso, viene utilizzato 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 di PostgreSQL;
- **database name** - il nome del database (se omesso, viene utilizzato 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 utilizzato 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 degli 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 aggiuntivo in 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 usare per la scansione dei dischi. Valori possibili: *id*, *name* (predefinito). *id* non è supportato su Windows e, se selezionato, restituirà un errore

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} conterrà un'aggiunta nel 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; come valore può essere utilizzata la macro {#PATH};
- **raid_type** - il tipo di RAID; come valore può essere utilizzata la macro {#RAID}

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 il rilevamento 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[nome unità,<interface>]

 Restituisce tutte le proprietà di un'unità systemd.
 Valore restituito: *oggetto JSON*.

Parametri:

- **nome unità** - il nome dell'unità (potresti voler usare la macro {#UNIT.NAME} nel prototipo di 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 (potresti voler usare la macro {#UNIT.NAME} nel prototipo di item per individuare il nome);
- **property** - proprietà dell'unità (ad es. ActiveState (predefinito), LoadState, Description);
- **interface** - il tipo di interfaccia dell'unità (ad es. Unit (predefinito), Socket, Service).

Commenti:

- Questo item è supportato solo sulla piattaforma Linux;
- Questo item consente di recuperare una proprietà specifica da uno specifico tipo di interfaccia come descritto nella dbus API.

Esempi:

```
systemd.unit.info["{#UNIT.NAME}"] #collect active state (active, reloading, inactive, failed, activating,
systemd.unit.info["{#UNIT.NAME}",LoadState] #collect load state info on discovered systemd units
systemd.unit.info[mysqld.service,Id] #retrieve the service technical name (mysqld.service)
systemd.unit.info[mysqld.service,Description] #retrieve the service description (MySQL Server)
systemd.unit.info[mysqld.service,ActiveEnterTimestamp] #retrieve the last time the service entered the act
systemd.unit.info[dbus.socket,NConnections,Socket] #collect the number of connections from this socket uni
```

systemd.unit.discovery[<type>]

 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>]

 Convalida i certificati e restituisce i dettagli del certificato.
 Valore restituito: *oggetto JSON*.

Parametri:

- **hostname** - può essere un IP o un DNS.
Può contenere lo schema URL (solo *https*), il percorso (verrà ignorato) e la porta.
Se una porta è fornita sia nel primo che nel secondo parametro, i relativi valori devono corrispondere.
Se viene specificato address (il 3° parametro), hostname viene utilizzato solo per SNI e per la verifica del nome host;
- **port** - il numero di porta (il valore predefinito per HTTPS è 443);
- **address** - può essere un IP o un DNS. Se specificato, verrà utilizzato per la connessione e hostname (il 1° parametro) verrà utilizzato 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 utilizzato per la connessione e il 3° parametro verrà utilizzato per SNI e per la verifica dell'host.

Commenti:

- Questo item diventa non supportato se la destinazione specificata nella configurazione del host non esiste, non è disponibile oppure se l'handshake TLS fallisce con qualsiasi errore diverso da un certificato non valido;
- Attualmente, l'estensione X.509 AIA (Authority Information Access), le CRL e OCSP (incluso OCSP stapling), e Certificate Transparency non sono supportati;
- 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 utilizzato per firmare il certificato (ad esempio, `SHA256-RSA`).
    - *issuer*: l'emittente del certificato.
    - *not_before*: la data di inizio della validità del certificato.
    - *not_after*: la data di scadenza del certificato.
    - *subject*: il soggetto del certificato.
    - *public_key_algorithm*: l'algoritmo utilizzato per la chiave pubblica (ad esempio, `RSA`).
    - *alternative_names*: i nomi alternativi del soggetto (se presenti), altrimenti `null`.
  - result: contiene il risultato della convalida.
    - *value*: lo stato della convalida (vedere i possibili valori di seguito).
    - *message*: messaggio dettagliato di convalida (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 convalida del certificato. I valori possibili includono:
  - *valid* – il certificato è valido e attendibile.
  - *valid-but-self-signed* – il certificato è valido ma autofirmato, cioè il suo subject corrisponde al suo issuer.
  - *invalid* – il certificato non è valido a causa di un problema come scadenza, nome host errato 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"
}
```


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 lo Zabbix agent UNIX:

- La chiave item è un collegamento ai dettagli completi dell'item dello Zabbix agent UNIX
- Sono inclusi i commenti relativi a Windows

Item key	Descrizione	Gruppo di item
log	Il monitoraggio di un file di log. Questo item non è supportato per il Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	Monitoraggio log
log.count	Il conteggio delle righe corrispondenti in un file di log monitorato. Questo item non è supportato per il Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
logrt	Il monitoraggio di un file di log soggetto a rotazione. Questo item non è supportato per il Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
logrt.count	Il conteggio delle righe corrispondenti in un file di log monitorato soggetto a rotazione. Questo item non è supportato per il Windows Event Log. Il parametro `persistent_dir` non è supportato su Windows.	
modbus.get	Legge 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.	Rete
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.	

Item key	Descrizione	Gruppo di item
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 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 da contatori a 64 bit, se disponibili. I contatori a 64 bit per le statistiche delle interfacce sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent utilizza contatori a 32 bit. I nomi delle interfacce multibyte su Windows sono supportati. È possibile ottenere le descrizioni delle interfacce di rete su Windows con gli item net.if.discovery o net.if.list.	
net.if.out	Le statistiche del traffico in uscita su un'interfaccia di rete. Su Windows, l'item ottiene i valori da contatori a 64 bit, se disponibili. I contatori a 64 bit per le statistiche delle interfacce sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent utilizza contatori a 32 bit. I nomi delle interfacce multibyte su Windows sono supportati. 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 da contatori a 64 bit, se disponibili. I contatori a 64 bit per le statistiche delle interfacce sono stati introdotti in Windows Vista e Windows Server 2008. Se i contatori a 64 bit non sono disponibili, l'agent utilizza contatori a 32 bit. 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.	Processi
proc.num	Il numero di processi. Su Windows sono supportati solo i parametri `name` e `user`.	
system.cpu.discovery	L'elenco delle CPU/core CPU rilevati.	Sistema
system.cpu.load	Il carico della CPU. Quando viene avviato un processo collector su Zabbix agent, vengono inizializzati i seguenti contatori delle prestazioni e successivamente utilizzati 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 utilizzando il contatore delle prestazioni *Processor Time*. Si noti che a partire da Windows 8 il Task Manager mostra l'utilizzo della CPU in base al contatore delle prestazioni *Processor Utility*, mentre nelle versioni precedenti veniva utilizzato il contatore *Processor Time* (vedere maggiori dettagli). *system* è l'unico parametro `type` supportato su Windows.	

Item key	Descrizione	Gruppo di item
system.hostname	Il nome host del sistema. Il valore viene acquisito su Windows tramite le funzioni GetComputerName() (per **netbios**), GetComputerNameExA() (per **fqdn**) oppure gethostname() (per **host**). Vedere anche una descrizione più dettagliata.	
system.localtime	L'ora del sistema.	
system.run	Esegue il comando specificato sul 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.).	
system.swap.size	La dimensione dello spazio di swap in byte o in percentuale del 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 utilizzare 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 e spazi aggiuntivi.	
system.uptime	Il tempo di attività del sistema in secondi.	
vfs.dir.count	Il conteggio delle voci della directory. Su Windows, i collegamenti simbolici alle directory vengono ignorati e gli hard link vengono conteggiati una sola volta.	File system virtuali
vfs.dir.get	L'elenco delle voci della directory. Su Windows, i collegamenti simbolici alle directory vengono ignorati e gli hard link vengono conteggiati una sola volta.	
vfs.dir.size	La dimensione della directory. Su Windows, qualsiasi collegamento simbolico viene ignorato e gli hard link vengono presi in considerazione una sola volta.	
vfs.file.cksum	Il checksum del file, calcolato con l'algoritmo UNIX cksum.	
vfs.file.contents	Recupera il contenuto di un file.	
vfs.file.exists	Verifica se il file esiste. Su Windows, le virgolette doppie devono essere precedute dal carattere di escape backslash '\' e l'intera chiave item deve essere racchiusa tra virgolette doppie quando si utilizza l'utilità da riga di comando per chiamare zabbix_get.exe o agent2. Si noti che l'item può diventare non supportato su Windows se viene cercata 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 temporali 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 disco disponibile, le statistiche inode e le opzioni di mount. La macro {#FSLABEL} è supportata su Windows.	
vfs.fs.size	Lo spazio disco in byte o in percentuale del totale.	
vm.memory.size	La dimensione della memoria in byte o in percentuale del totale.	Memoria virtuale

Item key	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 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.	
zabbix.stats	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 log eventi.
 Valore restituito: *Log*.

Parametri:

- **name** - il nome del canale del log eventi (*Log Name* nella GUI del Visualizzatore eventi);
- **regexp** - una espressione regolare che descrive il modello richiesto (con distinzione tra maiuscole e minuscole);

- **severity** - un'espressione regolare che descrive la gravità (senza distinzione tra maiuscole e 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 (senza distinzione tra maiuscole e minuscole);
- **eventid** - un'espressione regolare che descrive l'identificatore o gli identificatori dell'evento (con distinzione tra maiuscole e minuscole);
- **maxlines** - il numero massimo di nuove righe al secondo che l'agent invierà al server o al proxy Zabbix. Questo parametro sovrascrive il valore di 'MaxLinesPerSecond' in zabbix_agentd.conf.
- **mode** - valori possibili: *all* (predefinito) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati).

Commenti:

- L'item deve essere configurato come controllo attivo;
- L'agent non è in grado di inviare eventi dal log "Forwarded events";
- Windows Eventing 6.0 è supportato;
- La selezione di un tipo di informazione diverso da Log per questo item comporterà la perdita del timestamp locale, nonché delle informazioni sulla gravità e sulla sorgente del log;
- Vedere anche le informazioni aggiuntive 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] #qui viene fatto riferimento a un'espressione regolare personalizzata denomi
```

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* nella GUI del Visualizzatore eventi);
- **regexp** - una espressione regolare che descrive il modello richiesto (con distinzione tra maiuscole e minuscole);
- **severity** - un'espressione regolare che descrive la gravità (senza distinzione tra maiuscole e 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 (senza distinzione tra maiuscole e minuscole);
- **eventid** - un'espressione regolare che descrive gli identificatori di evento (con distinzione tra maiuscole e 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) oppure *skip* - salta l'elaborazione dei dati meno recenti (influisce solo sugli item appena creati).

Commenti:

- L'item deve essere configurato come controllo attivo;
- L'agent non è in grado di inviare eventi dal registro "Forwarded events";
- Windows Eventing 6.0 è supportato;
- La selezione di un tipo di informazione diverso da Log per questo item comporterà la perdita del timestamp locale, nonché delle informazioni su gravità e sorgente del log;
- Vedere anche le informazioni aggiuntive 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 delle 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 la memorizzazione del valore medio. `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 utilizzato per i contatori che richiedono più di un campione (come l'utilizzo della CPU), quindi il controllo restituisce ogni volta un valore medio relativo agli ultimi "interval" secondi;
- È possibile utilizzare Performance Monitor per ottenere l'elenco dei contatori disponibili.
- Vedi anche: Contatori delle prestazioni di Windows.

perf_counter_en[counter,<interval>]

 Il valore di qualsiasi contatore delle 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 la memorizzazione del valore medio. `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 utilizzato per i contatori che richiedono più di un campione (come l'utilizzo della CPU), quindi il controllo restituisce ogni volta un valore medio relativo agli 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 di registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Mi 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 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 utilizzata 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 lettura e scrittura
io_other_op - numero di operazioni di I/O eseguite dal processo, diverse dalle operazioni di lettura e scrittura
gdiobj - numero di oggetti GDI utilizzati dal processo
userobj - numero di oggetti USER utilizzati 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, è richiesto un agent Zabbix a 64 bit affinché questo item funzioni correttamente.

Esempi:

```
proc_info[iexplore.exe,wkset,sum] #recupera la quantità di memoria fisica occupata da tutti i processi Int
proc_info[iexplore.exe,pf,avg] #recupera il numero medio di page fault per i processi Internet Explorer
```

registry.data[key,<nome valore>]

 Restituisce i dati per il nome valore specificato nella chiave del Registro di sistema di Windows.
 Valore restituito: *Integer*, *string* oppure *text* (a seconda del tipo di valore)

Parametri:

- **key** - la chiave di registro inclusa la chiave radice; sono consentite le abbreviazioni della radice (ad esempio HKLM);
- **nome valore** - il nome del valore di registro nella chiave (stringa vuota "" per impostazione predefinita). Viene restituito il valore predefinito se il nome valore non viene fornito.

Commenti:

- Abbreviazioni della 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 doppi apici.

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 presenti nella chiave specificata.
 Valore restituito: *oggetto JSON*.

Parametri:

- **key** - la chiave di registro, inclusa la chiave radice; sono consentite le abbreviazioni della radice (ad esempio HKLM) (vedere i commenti per registry.data[] per l'elenco completo delle abbreviazioni);
- **mode** - valori possibili:
values (predefinito) oppure *keys*;
- **name regexp** - individua solo i valori con nomi che corrispondono all'espressione regolare (impostazione predefinita: individua tutti i valori). Consentito solo con *values* come mode.

Le chiavi con spazi devono essere racchiuse tra doppi apici.

Esempi:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
The JSON will include details of the key, last subkey, value name, value type and value data.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #restituisce i dati di tutti
The JSON will include details of the key, last subkey, value name, value type and value data.
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #restituisce tutte le sottochi
The JSON will include details of the key and last subkey.
```

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` uguale a *state*, *startup*; *String* - con `param` uguale a *displayname*, *path*, *user*; *Text* - con `param` uguale a *description*
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 tramite trigger, 6 - avvio automatico ritardato tramite trigger, 7 - avvio manuale tramite trigger

Parametri:

- **service** - il nome reale del servizio oppure il suo nome visualizzato come appare nello snap-in Servizi di MMC;
- **param** - *state* (predefinito), *displayname*, *path*, *user*, *startup* oppure *description*.

Commenti:

- Gli item come `service.info[service,state]` e `service.info[service]` restituiscono le stesse informazioni;
- Solo con `param` uguale a *state* questo item restituisce un valore per servizi non esistenti (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] - il 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 il 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.

Passaggio 1

Ottenere il nome del servizio.

È possibile ottenere il nome del servizio aprendo lo snap-in Servizi di MMC e visualizzando le proprietà del servizio. Nella scheda *Generale* dovrebbe essere presente un campo chiamato "Nome servizio". Il valore riportato è il nome che verrà utilizzato durante la configurazione di un item per il monitoraggio. Ad esempio, se si desidera monitorare il servizio "workstation", il nome del 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.

2 SNMP agent

Panoramica

Potresti voler utilizzare il monitoraggio SNMP su dispositivi come stampanti, switch di rete, router o UPS, che di solito supportano SNMP e sui quali sarebbe poco pratico 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, come la visualizzazione dei valori in HEX invece che in UTF-8 o viceversa.

I controlli SNMP vengono eseguiti solo tramite il protocollo UDP.

I demoni server e proxy di Zabbix 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
```

Sebbene non coprano tutti i casi problematici, sono utili per identificare 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 ritentativo non si applica agli errori di risoluzione DNS.

Per i controlli SNMP legacy (un singolo numero OID o una stringa), il server/proxy Zabbix ritenterà almeno una volta dopo un tentativo di query non riuscito: tramite il meccanismo di ritentativo della libreria SNMP oppure tramite il meccanismo interno di elaborazione combinata.

Warning:
Se monitori dispositivi SNMPv3, assicurati 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 persistano il proprio engineBoots. Alcuni dispositivi non lo fanno, con il risultato che i loro messaggi SNMP vengono 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 associazioni SNMPv3 EngineID → IP e riutilizza gli EngineID memorizzati per i controlli successivi invece di inviare ogni volta una probe, riducendo il traffico di rete. Se un EngineID non può essere riutilizzato, verrà eseguito un nuovo tentativo con una probe per individuare il nuovo EngineID.

Configurazione del monitoraggio SNMP

Per iniziare a monitorare un dispositivo tramite SNMP, è necessario eseguire i seguenti passaggi:

Passaggio 1

Individua la stringa SNMP (o OID) dell'item che vuoi monitorare.

Per ottenere un elenco di stringhe SNMP, usa il comando **snmpwalk** (parte del software net-snmp, che dovrebbe essere installato come parte dell'installazione di Zabbix) o uno strumento equivalente:

```
snmpwalk -v 2c -c public <host IP> .
```

Poiché qui '2c' indica la versione SNMP, puoi anche sostituirlo con '1', per indicare SNMP Version 1 sul dispositivo.

Questo dovrebbe fornirti un elenco di stringhe SNMP e del loro ultimo valore.
Se così non fosse, è possibile che la 'community' SNMP sia diversa dalla 'public' standard; in tal caso dovrai scoprire quale sia.

Puoi quindi scorrere l'elenco finché non trovi la stringa che vuoi monitorare, ad esempio,
se volessi monitorare i byte in ingresso al 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 restituire qualcosa di simile al seguente:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

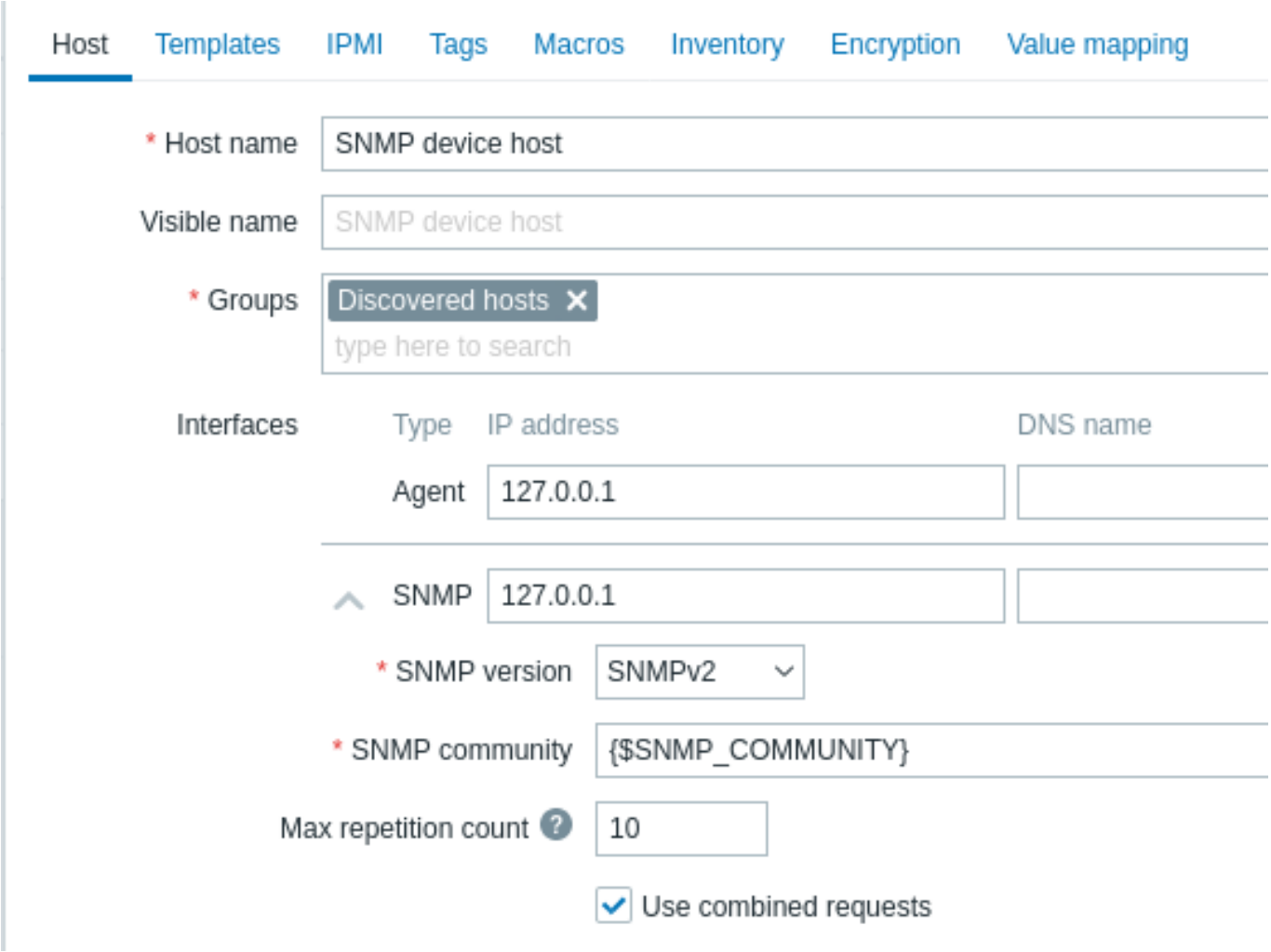
Anche in questo caso, l'ultimo numero nell'OID è il numero della porta.

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 mostrati come "STRING", "Hex-STRING", "OID" e altri, a seconda della presenza di un display hint.

Passaggio 2

Crea un host corrispondente a un dispositivo.



Aggiungi un'interfaccia SNMP per l'host:

- Inserisci l'indirizzo IP/nome DNS e il numero di porta.
- Seleziona la *versione SNMP* dal menu a discesa.
- Aggiungi le credenziali dell'interfaccia in base alla versione SNMP selezionata:
  - SNMPv1, v2 richiedono solo la community (di solito 'public').
  - SNMPv3 richiede opzioni più specifiche (vedi sotto).
- Specifica il valore massimo di ripetizione (predefinito: 10) per le richieste bulk SNMP native (GetBulkRequest-PDUs); solo per gli item `discovery[]` e `walk[]` in SNMPv2 e v3. Nota che impostare questo valore troppo alto può causare il timeout del controllo dell'agent SNMP.
- Seleziona la casella di controllo *Usa richieste combinate* per consentire l'elaborazione combinata delle richieste SNMP (non correlata alle richieste bulk SNMP native "walk" e "get").

Parametro SNMPv3	Descrizione
Nome contesto	Inserisci il nome del contesto per identificare l'item sulla subnet SNMP. Le macro utente vengono risolte in questo campo.

Parametro SNMPv3	Descrizione
Nome di sicurezza	Inserisci il nome di sicurezza. Le macro utente vengono risolte in questo campo.
Livello di sicurezza	Seleziona il livello di sicurezza: **noAuthNoPriv** - non vengono utilizzati protocolli di autenticazione né di privacy **AuthNoPriv** - viene utilizzato il protocollo di autenticazione, il protocollo di privacy no **AuthPriv** - vengono utilizzati sia il protocollo di autenticazione sia quello di privacy
Protocollo di autenticazione	Seleziona il protocollo di autenticazione - *MD5*, *SHA1*; con net-snmp 5.8 e versioni successive *SHA224*, *SHA256*, *SHA384* o *SHA512*.
Passphrase di autenticazione	Inserisci la passphrase di autenticazione. Le macro utente vengono risolte in questo campo.
Protocollo di privacy	Seleziona il protocollo di privacy - *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) o *AES256C* (Cisco). Vedi le note sul supporto del protocollo di privacy
Passphrase di privacy	Inserisci la passphrase di privacy. Le macro utente vengono risolte in questo campo.

In caso di credenziali SNMPv3 errate (nome di sicurezza, protocollo/passphrase di autenticazione, protocollo di privacy):

- Zabbix riceve un ERROR da net-snmp, tranne nel caso di *Passphrase di privacy* 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 *Protocollo di autenticazione*, *Passphrase di autenticazione*, *Protocollo di privacy* o *Passphrase di privacy*, effettuate senza modificare il *Nome di sicurezza*, vengono normalmente applicate automaticamente quando la corrispondente interfaccia SNMPv3 viene aggiornata in Zabbix. Nei casi in cui venga modificato anche il *Nome di sicurezza*, tutti i parametri verranno aggiornati immediatamente.

Puoi utilizzare uno dei template SNMP forniti che aggiungerà automaticamente un insieme di item. Prima di utilizzare un template, verifica che sia compatibile con l'host.

Fai clic su *Aggiungi* per salvare l'host.

Supporto dei protocolli 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 superiori non sono supportati out-of-the-box sui sistemi operativi precedenti a 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+, utilizzare una delle seguenti opzioni:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Se l'output contiene `--enable-blumenthal-aes`, AES192+ è supportato.

Si noti 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 libreria net-snmp non supporta i protocolli AES192 e superiori, ricompilare net-snmp con l'opzione `--enable-blumenthal-aes`, quindi ricompilare Zabbix server specificando l'opzione `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config`.

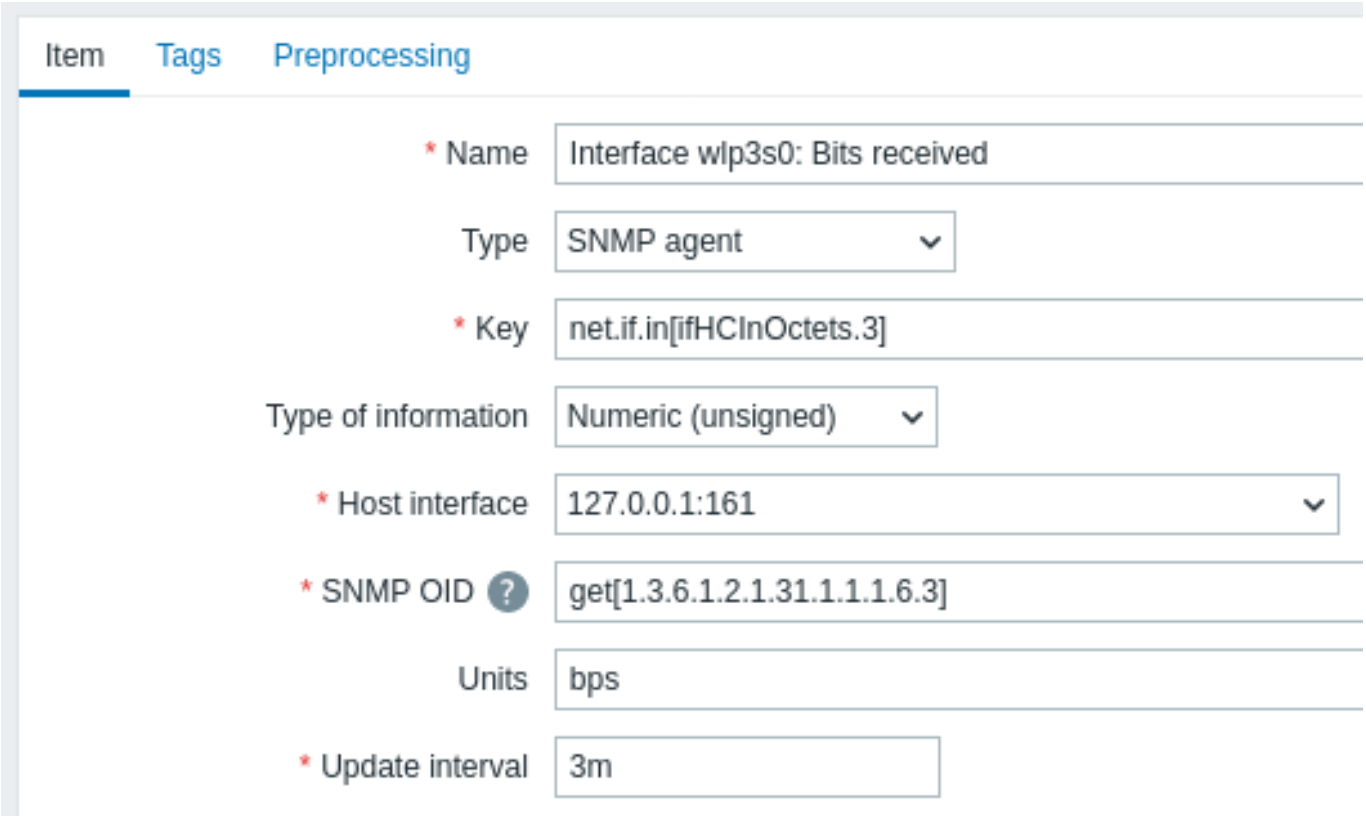
Passaggio 3

Crea un item per il monitoraggio.

Ora torna in Zabbix e fai clic su *Items* per l'host SNMP che hai creato in precedenza. A seconda che tu abbia utilizzato o meno un template durante la creazione del tuo host, vedrai un elenco di item SNMP associati al tuo host oppure semplicemente un elenco

vuoto. Lavoreremo partendo dal presupposto che creerai tu stesso l'item utilizzando le informazioni che hai appena raccolto con snmpwalk e snmpget, quindi fai clic su *Create item*.

Compila i parametri richiesti nel modulo del nuovo item:



Parameter	Description
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.

Parameter	Description
SNMP OID	Usa 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 in modo asincrono le richieste bulk SNMP native (GetBulkRequest-PDUs). Le impostazioni di timeout per questo item possono essere configurate nel modulo di configurazione dell'item. Valuta l'impostazione di un valore di timeout basso per evitare lunghi ritardi se il dispositivo non è raggiungibile, poiché verranno effettuati fino a 5 tentativi se i precedenti vanno in timeout o falliscono (ad esempio, un timeout di 3 secondi può comportare un'attesa di 15 secondi). Puoi usare questo come item master, con item dipendenti che estraggono dati dal master tramite preprocessing. 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 tentato il recupero di un singolo record senza richiesta bulk. I nomi MIB sono supportati come parametri; quindi `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 usate con interfacce SNMPv2 e v3, mentre GetNext con interfacce SNMPv1; il numero massimo di ripetizioni per le richieste bulk è configurato a livello di interfaccia. Il parametro max repetitions 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, non tutti i dispositivi potrebbero supportare valori molto elevati, il che potrebbe causare problemi. Questo item restituisce l'output dell'utilità snmpwalk con i parametri -Oe -Ot -On. Puoi usare questo item come item master in SNMP discovery. **get[OID]** - recupera in modo asincrono un valore *singolo*. 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. Valuta l'impostazione di un valore di timeout basso per evitare lunghi ritardi se il dispositivo non è raggiungibile, poiché verranno effettuati fino a 5 tentativi se i precedenti vanno in timeout o falliscono (ad esempio, un timeout di 3 secondi può comportare un'attesa di 15 secondi). **OID** - (legacy) inserisci un singolo OID testuale o numerico per recuperare in modo sincrono un singolo valore, 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. Si **consiglia** di usare item `walk[OID]` e `get[OID]` per 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 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 SNMP asincroni è definito dal parametro StartSNMPPollers. Tieni presente che per le statistiche del traffico di rete, restituite da uno qualsiasi dei metodi, è necessario aggiungere un passaggio *Change per second* nella scheda *Preprocessing*; in caso contrario otterrai il valore cumulativo dal dispositivo SNMP invece dell'ultima variazione.

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

Ora salva l'item e vai in *Monitoring* > *Latest data* per i tuoi 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 (GetBulkRequest-PDUs), disponibile nelle versioni SNMP 2/3.

Una richiesta GetBulk in SNMP esegue più richieste GetNext e restituisce il risultato in un’unica risposta. Può essere utilizzata per normali item SNMP così come per il discovery SNMP, per ridurre al minimo i round trip di rete.

L’item SNMP **walk[OID1,OID2,...]** può essere utilizzato come item master che raccoglie i dati in un’unica richiesta, con item dipendenti che analizzano la risposta secondo necessità utilizzando il preprocessing.

Si noti che l’utilizzo di richieste bulk SNMP native non è correlato all’opzione di combinare richieste SNMP, che è il modo proprio di Zabbix di combinare più richieste SNMP (vedere la sezione successiva).

Per gli item bulk SNMP 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 di configurazione dell’item) è impostato per un’intera sessione. Il timeout viene applicato indipendentemente dal fatto che i dati vengano recuperati completamente; se i dati vengono ricevuti solo parzialmente (ad esempio, i dati vengono raccolti correttamente solo per uno tra più OID), allora 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 di continuare la sessione dall’ultima richiesta se un singolo pacchetto è andato perso o è arrivato troppo tardi. 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 i precedenti scadono o falliscono (ad esempio, un timeout di 3 secondi può comportare un tempo di attesa di 15 secondi).

Funzionamento interno dell’elaborazione combinata

Zabbix server e proxy 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 low-level discovery SNMP

Tutti gli item SNMP su una singola interfaccia con parametri identici vengono pianificati per essere interrogati nello stesso momento. I primi due tipi di item vengono presi dai poller in batch di al massimo 128 item, mentre le regole di low-level discovery vengono elaborate singolarmente, come in precedenza.

A un livello inferiore, esistono due tipi di operazioni eseguite per interrogare i valori: ottenere più oggetti specificati e percorrere un albero OID.

Per l’”ottenimento”, viene utilizzato un GetRequest-PDU con al massimo 128 associazioni di variabili. Per il ”walking”, viene utilizzato un GetNextRequest-PDU per SNMPv1 e GetBulkRequest con il campo ”max-repetitions” di al massimo 128 per SNMPv2 e SNMPv3.

Pertanto, i vantaggi dell’elaborazione combinata per ciascun tipo di item SNMP sono illustrati di seguito:

- gli item SNMP regolari beneficiano dei miglioramenti nell'"ottenimento";
- gli item SNMP con indici dinamici beneficiano sia dei miglioramenti nell'"ottenimento" sia di quelli nel "walking": l'"ottenimento" viene utilizzato per la verifica dell'indice e il "walking" per costruire la cache;
- le regole SNMP di low-level discovery beneficiano dei miglioramenti nel "walking".

Tuttavia, esiste un problema tecnico: non tutti i dispositivi sono in grado di restituire 128 valori per richiesta. Alcuni restituiscono sempre una risposta corretta, ma altri rispondono con un errore "tooBig(1)" oppure non rispondono affatto quando la risposta potenziale supera un certo limite.

Per trovare un numero ottimale di oggetti da interrogare per un determinato dispositivo, Zabbix utilizza la seguente strategia. Inizia con cautela interrogando 1 valore in una richiesta. Se l'operazione ha successo, interroga 2 valori in una richiesta. Se anche questa ha successo, interroga 3 valori in una richiesta e continua in modo analogo moltiplicando il numero di oggetti interrogati per 1,5, 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 si rifiuta di fornire una risposta corretta (ad esempio, per 42 variabili), Zabbix esegue due operazioni.

Innanzitutto, per il batch corrente di item dimezza il numero di oggetti in una singola richiesta e interroga 21 variabili. Se il dispositivo è attivo, nella stragrande maggioranza dei casi la richiesta dovrebbe funzionare, perché 28 variabili erano note per funzionare e 21 è significativamente inferiore. Tuttavia, se anche questo fallisce, Zabbix torna a interrogare i valori uno per uno. Se anche a questo punto fallisce, allora il dispositivo sicuramente non sta rispondendo e la dimensione della richiesta non è il problema.

La seconda cosa che Zabbix fa per i batch successivi di item è iniziare con l'ultimo numero di variabili che ha avuto successo (28 nel nostro esempio) e continuare ad aumentare la dimensione delle richieste di 1 finché non viene raggiunto 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 fallirà 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 richieste di grandi dimensioni falliscono con questo numero di variabili, ciò può significare una di due cose. I criteri esatti che un dispositivo utilizza per limitare la dimensione della risposta non possono essere conosciuti, ma si cerca di approssimarli usando il numero di variabili. Quindi, 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 lo supera. La seconda possibilità è che un pacchetto UDP in una delle due direzioni sia semplicemente andato perso. Per questi motivi, se Zabbix riceve una richiesta non riuscita, riduce il numero massimo di variabili per cercare di rientrare maggiormente nell'intervallo di funzionamento sicuro del dispositivo, ma solo fino a due volte.

Nell'esempio sopra, se una richiesta con 32 variabili fallisce, Zabbix ridurrà il conteggio a 31. Se anche questa fallisce, Zabbix ridurrà il conteggio a 30. Tuttavia, Zabbix non ridurrà il conteggio al di sotto di 30, perché presumerà che ulteriori errori siano dovuti alla perdita di pacchetti UDP, piuttosto che al limite del dispositivo.

Se invece un dispositivo non è in grado di gestire correttamente le richieste combinate per altri motivi e l'euristica descritta sopra non funziona, esiste un'impostazione "Use combined requests" per ogni interfaccia che consente di disabilitare le richieste combinate per quel dispositivo.

Se le richieste combinate causano risposte parziali o malformate che producono calcoli per secondo (delta) errati (ad esempio, picchi apparenti nei contatori di interfaccia), disabilitare *Use combined requests* per l'interfaccia interessata per forzare richieste separate per singolo item; questo spesso evita falsi picchi. In alternativa, considerare l'uso di item asincroni `get[]` o `walk[]`, che vengono eseguiti in modo asincrono e non sono soggetti al batching per interfaccia di *Use combined requests* — possono essere utilizzati al posto dei controlli OID sincroni legacy per evitare problemi correlati alle richieste combinate. Cercare nel log di server/proxy voci simili a quella mostrata nella sezione Overview per aiutare a identificare i dispositivi interessati.

Inoltre, se l'interfaccia diventa frequentemente non disponibile, potrebbe essere necessario aumentare il parametro `UnavailableDelay` nei file di configurazione di Zabbix server o Zabbix proxy per ridurre la frequenza delle richieste. Gli item possono diventare non supportati se durante la discovery o le operazioni di OID walk vengono ricevuti dati parziali.

1 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, usa 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. Maiuscole e minuscole sono considerate distinte.

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 di ricerca degli indici

Quando viene richiesto un item con indice dinamico, Zabbix recupera e mette in cache l'intera tabella SNMP sotto l'OID di base per l'indice, anche se una corrispondenza potrebbe essere 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.

2 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**, **ifIndex.0** viene tradotto in **1.3.6.1.2.1.2.2.1.1.0**.

La tabella contiene l'elenco degli OID speciali.

OID speciale	Identificatore	Descrizione
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 testuale 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 ai protocolli fisici/di collegamento immediatamente "sotto" il livello di rete nello stack di protocolli.
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.

OID speciale	Identificatore	Descrizione
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 di 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 di sottorete 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 evitare che potessero essere consegnati a un protocollo di livello superiore. Una possibile ragione per scartare un tale pacchetto 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 tali da impedirne 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 evitare che venissero trasmessi. Una possibile ragione per scartare un tale pacchetto 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).

3 file MIB

Introduzione

MIB è l'acronimo di Management Information Base. I file MIB consentono di utilizzare la rappresentazione testuale di un OID (Object Identifier). È possibile usare OID grezzi durante il monitoraggio di dispositivi SNMP con Zabbix, ma se si preferisce utilizzare la rappresentazione testuale, è necessario installare i file MIB.

Per 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.

3 trappole SNMP

Panoramica

La ricezione delle trap SNMP è l'opposto dell'interrogazione dei dispositivi con SNMP abilitato.

In questo caso, le informazioni vengono inviate da un dispositivo con SNMP abilitato a snmptrapd e vengono raccolte o "intercettate" da Zabbix server o Zabbix proxy da file.

Di solito, le trap vengono inviate al verificarsi di un cambiamento di condizione e l'agent si connette al server sulla porta 162 (a differenza della porta 161 sul lato agent utilizzata per le interrogazioni). L'uso delle trap può rilevare alcuni problemi brevi che si verificano nell'intervallo tra le interrogazioni e che potrebbero non essere rilevati dai dati delle interrogazioni.

La ricezione delle trap SNMP in Zabbix è progettata per funzionare con **snmptrapd** e con uno dei meccanismi per passare le trap a Zabbix: uno script Bash o Perl oppure SNMPTT.

Note:
Il modo più semplice per configurare il monitoraggio delle trap dopo aver configurato Zabbix è utilizzare la soluzione con script Bash, perché Perl e SNMPTT spesso mancano nelle distribuzioni moderne e richiedono una configurazione più complessa. Tuttavia, questa soluzione utilizza uno script configurato come `traphandle`. Per prestazioni migliori nei sistemi di produzione, utilizzare la soluzione Perl incorporata (uno script con l'opzione `do perl` oppure SNMPTT).

Il flusso di lavoro per la ricezione di una trap:

1. `snmptrapd` riceve una trap
2. `snmptrapd` passa la trap allo script ricevitore (Bash, Perl) oppure a SNMPTT
3. Il ricevitore analizza, formatta e scrive la trap in un file
4. Zabbix SNMP trapper legge e analizza il file delle trap
5. Per ogni trap, Zabbix trova tutti gli item "SNMP trapper" con interfacce host corrispondenti all'indirizzo della trap ricevuta. Si noti che durante la corrispondenza viene utilizzato solo il valore "IP" o "DNS" selezionato nell'interfaccia host.
6. Per ogni item trovato, la trap viene confrontata con la regexp in `snmptrap[regexp]`. La trap viene impostata come valore di **tutti** gli item corrispondenti. Se non viene trovato alcun item corrispondente ed esiste un item `snmptrap.fallback`, la trap viene impostata come valore di quest'ultimo.
7. Se la trap non è stata impostata come valore di alcun item, Zabbix per impostazione predefinita registra la trap non corrispondente. (Questa opzione è configurata da "Log unmatched SNMP traps" in Administration > General > Other.)

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 delle trap SNMP

Questo tipo di item richiede la seguente configurazione del frontend.

1. Creare un'interfaccia SNMP per il proprio host

   - In *Data collection* > *Hosts*, creare/modificare l'host e, nel campo *Interfaces*, aggiungere il tipo di interfaccia "SNMP", specificando l'indirizzo IP o DNS.

L'indirizzo di ciascuna trap ricevuta verrà confrontato con gli indirizzi IP e DNS di tutte le interfacce SNMP per trovare gli host corrispondenti.

2. Configurare l'item

   - In *Data collection* > *Hosts*, creare/modificare l'item necessario.
   - Nel campo *Key*, utilizzare una delle chiavi trap SNMP:

Key		
Description	Return value	Comments
snmptrap[regexp] Intercetta tutte le trap SNMP che corrispondono alla espressione regolare specificata in **regexp**. Se regexp non è specificato, intercetta qualsiasi trap.	Trap SNMP	Questo item può essere impostato solo per interfacce SNMP. Le macro utente e le espressioni regolari globali sono supportate nel parametro di questa chiave item.
snmptrap.fallback		

Key		
Intercetta tutte le trap SNMP che non sono state intercettate da nessuno degli item snmptrap[] per quell'interfaccia.	Trap SNMP	Questo item può essere impostato solo per interfacce SNMP.

Note:
La corrispondenza di espressioni regolari multilinea non è attualmente supportata.

- Impostare *Type of information* su "Log" affinché i timestamp vengano analizzati. Anche altri formati come "Numeric" sono accettabili, ma potrebbero richiedere un gestore trap personalizzato.

Configurazione del monitoraggio delle trap SNMP

Configurazione di Zabbix server/proxy

Per leggere le trap, Zabbix server o proxy deve essere configurato per avviare il processo SNMP trapper e puntare al file delle trap in cui vengono scritte da SNMPTT o da un ricevitore di trap Bash/Perl. Per farlo, modificare 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 per un ricevitore di trap Bash può essere utilizzato per passare le trap a Zabbix server da snmptrapd usando il file trapper. Per configurarlo, aggiungere l'opzione `traphandle` al file di configurazione di snmptrapd (`snmptrapd.conf`), vedere esempio.

Note:
Potrebbe essere necessario riavviare snmptrapd affinché recepisca le modifiche alla sua configurazione.

Configurazione del ricevitore trap Perl

Requisiti: Perl, Net-SNMP compilato con --enable-embedded-perl (eseguito per impostazione predefinita a partire da Net-SNMP 5.4)

Un ricevitore trap Perl (vedere misc/snmptrap/zabbix_trap_receiver.pl) può essere utilizzato per passare le trap direttamente al server Zabbix 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 racchiuso tra virgolette, snmptrapd si rifiuterà di avviarsi con messaggi simili a questi:


```
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 utilizzare SNMPTT.

> **Note:**
> Per ottenere le migliori prestazioni, SNMPTT deve essere configurato come daemon utilizzando **snmptthandler-embedded** per inoltrargli le trap. Vedere le istruzioni per la configurazione di SNMPTT.

Quando SNMPTT è configurato per ricevere le trap, configurare `snmptt.ini`:

1. abilitare l'uso del modulo Perl del pacchetto NET-SNMP:

```
net_snmp_perl_enable = 1
```

2. registrare le trap nel file trap che verrà letto da Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. impostare il formato di data e 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; reintrodotto in RHEL 8.3. Per maggiori informazioni, vedere i problemi noti.

Ora formattare le trap affinché Zabbix possa riconoscerle (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 in 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 delle trap SNMP di seguito.

> **Attention:**
> Non utilizzare trap sconosciute: Zabbix non sarà in grado di riconoscerle. Le trap sconosciute possono essere gestite definendo un evento generale in snmptt.conf:

> `EVENT general .* "General event" Normal`

Formato della trap SNMP

Tutti i ricevitori di trap Perl personalizzati e la configurazione delle trap SNMPTT devono formattare la trap nel modo seguente:

```
[timestamp] [la trap, parte 1] ZBXTRAP [address] [la trap, parte 2]
```

dove

- [timestamp] - il timestamp nel formato "%Y-%m-%dT%H:%M:%S%z"
- ZBXTRAP - intestazione che indica che una nuova trap inizia in questa riga
- [address] - indirizzo IP utilizzato per trovare l'host per questa trap

Si noti che "ZBXTRAP" e "[address]" verranno rimossi dal messaggio durante l'elaborazione. Se la trap è formattata diversamente, Zabbix potrebbe analizzare le 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 la 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 deve prima rinominare il vecchio file e solo successivamente eliminarlo, in modo che nessuna trap vada persa:

1. Zabbix apre il file delle trap nell'ultima posizione nota e passa al passaggio 3
2. Zabbix verifica se il file attualmente aperto è stato ruotato confrontando il numero di inode con il numero di inode del file delle trap definito. Se non è presente alcun file aperto, Zabbix reimposta l'ultima posizione e torna al passaggio 1.
3. Zabbix legge i dati dal file attualmente aperto e imposta la nuova posizione.
4. I nuovi dati vengono analizzati. Se questo era il file ruotato, il file viene chiuso e si torna al passaggio 2.
5. Se non ci sono nuovi dati, Zabbix resta in attesa per 1 secondo e torna al passaggio 2.

File system

A causa dell'implementazione del file trap, Zabbix ha bisogno che il file system supporti gli inode per differenziare i file (le informazioni vengono acquisite da a chiamata stat()).

Esempi di configurazione con diverse versioni del protocollo SNMP

Questo esempio utilizza snmptrapd e uno script ricevitore Bash per passare le trap al server Zabbix.

Configurazione:

1. Configurare Zabbix per avviare SNMP trapper e impostare il file delle 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/Dock
```

Se necessario, modificare la variabile ZABBIX_TRAPS_FILE nello script. Per utilizzare il valore predefinito, creare prima la directory padre:

```
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 per applicare le modifiche alla sua configurazione.

4. Creare un item SNMP TEST (tenere presenti i requisiti di configurazione iniziali):

   Tipo: SNMP trap
   Tipo di informazione: Log Interfaccia host: SNMP 127.0.0.1
   Chiave: `snmptrap["linkUp"]`
   Formato orario del log: yyyyMMdd.hhmmss

Si noti che viene utilizzato il formato di data e ora ISO 8601.

5. Successivamente configureremo `snmptrapd` per la versione del protocollo SNMP scelta e invieremo trap di test utilizzando l'utilità `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 di trap SNMP.

Si noti che, sebbene SNMPv2 sia ancora ampiamente utilizzato negli ambienti di produzione, non offre alcuna cifratura né una reale autenticazione del mittente. I dati vengono inviati in testo semplice e pertanto queste versioni del protocollo dovrebbero essere utilizzate solo in ambienti sicuri come una rete privata e non dovrebbero mai essere utilizzate su reti pubbliche o di terze parti.

La versione 1 di SNMP oggi non è praticamente più utilizzata, poiché non supporta i contatori a 64 bit ed è considerata un protocollo legacy.

Per abilitare l'accettazione di trap SNMPv1 o SNMPv2 è necessario aggiungere la seguente riga a `snmptrapd.conf`. Sostituire "secret" con la community string SNMP configurata sui mittenti di trap SNMP:

```
authCommunity log,execute,net secret
```

Successivamente possiamo inviare una trap di test utilizzando `snmptrap`. In questo esempio useremo il comune OID "link up":

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 risolve i problemi di sicurezza di SNMPv1/v2 e fornisce autenticazione e cifratura. È possibile utilizzare MD5 o più metodi di autenticazione SHA e DES/più varianti AES come cifrario.

Per abilitare l'accettazione di SNMPv3 aggiungere le seguenti righe a `snmptrapd.conf`:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

> **Attention:**
> Si noti la parola chiave "execute", che consente di eseguire script per questo modello di sicurezza utente.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

> **Warning:**
> Se si desidera utilizzare metodi di cifratura 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 si vedranno 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)

4 controlli IPMI

Panoramica

In Zabbix è possibile monitorare lo stato di salute e la disponibilità dei dispositivi Intelligent Platform Management Interface (IPMI). Per eseguire i controlli IPMI, lo Zabbix server 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 oppure no.

Il monitoraggio IPMI di Zabbix funziona solo per i dispositivi che supportano IPMI (HP iLO, DELL DRAC, IBM RSA, Sun SSP, ecc.).

Un processo manager IPMI pianifica i controlli IPMI eseguiti dagli IPMI poller. Un host viene sempre interrogato da un solo IPMI poller alla volta, riducendo il numero di connessioni aperte ai controller BMC. Pertanto, è possibile aumentare in sicurezza il numero di IPMI poller senza preoccuparsi di sovraccaricare i controller BMC. Il processo manager IPMI viene avviato automaticamente quando viene avviato almeno un IPMI poller.

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, pertanto gli item IPMI aggiunti non funzioneranno. Per modificare questa impostazione, 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 risulti:

```
StartIPMIPollers=3
```

Salvare il file e riavviare successivamente zabbix_server.

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.

5 Controlli semplici

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 tutti i dettagli.

Vedere anche chiavi degli item di monitoraggio VMware.

Chiave item	Descrizione
icmpping	L’accessibilità dell’host tramite ping ICMP.
icmppingloss	La percentuale di pacchetti persi.
icmppingretry	L’accessibilità dell’host tramite ping ICMP con tentativi ripetuti.
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>]

 L’accessibilità dell’host tramite ping ICMP.
 Valore restituito: *0* - ping ICMP non riuscito; *1* - ping ICMP riuscito.

Parametri:

- **target** - l’IP dell’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** - usato per consentire il reindirizzamento: se vuoto (valore predefinito), le risposte reindirizzate vengono considerate come host di destinazione non disponibile; se impostato su *allow_redirect*, le risposte reindirizzate vengono considerate come host di destinazione disponibile.

Vedere anche la tabella dei valori predefiniti.

Esempio:

```
icmpping[,4] #Se viene restituito almeno un pacchetto dei quattro, l'item restituirà 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 semplici LDAP (ad es. `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

Specificare il percorso di fping nel parametro *FpingLocation* del file di configurazione di Zabbix server/proxy (o il parametro *Fping6Location* per l'utilizzo di indirizzi IPv6).

fping deve essere eseguibile dall'utente con cui viene eseguito Zabbix server/proxy e tale utente deve disporre di 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 valori per i parametri dei controlli ICMP:

Parameter	Unit	Description	Fping's flag	Defaults set by		Allowed limits by Zabbix	
				fping	**Zabbix**	**min**	**max**
packets	number	Numero di pacchetti di richiesta inviati a una destinazione	-C		3	1	10000
interval	milliseconds	Tempo di attesa tra pacchetti successivi verso una singola destinazione	-p	1000		20	unlimited
size	bytes	Dimensione del pacchetto in byte 56 byte su x86, 68 byte su x86_64	-b	56 o 68		24	65507
timeout	milliseconds	**fping v3.x** - timeout di attesa dopo l'invio dell'ultimo pacchetto, influenzato dal flag *-C* **fping v4.x** - timeout individuale per ciascun pacchetto	-t	**fping v3.x** - 500 **fping v4.x** e versioni successive - ereditato dal flag *-p*, ma non superiore a 2000		50	unlimited

Parameter	Unit	Description	Fping's flag	Defaults set by		Allowed limits by Zabbix	
retries	number	Numero di volte in cui verrà effettuato un tentativo di ping verso una destinazione, escluso il primo tentativo	-r	3	1	0	unlimited
backoff factor	number	Valore per cui viene moltiplicato il tempo di attesa 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 item *interval* menzionato nella tabella sopra, che corrisponde all'opzione fping *-p*) 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 che fping consente di usare con *-i* provando 3 valori: 0, 1 e 10. Il valore che ha successo per primo viene quindi utilizzato per i successivi controlli ICMP. Questo processo viene eseguito individualmente da ciascun processo ICMP pinger.

Le opzioni di fping rilevate automaticamente vengono invalidate ogni ora e rilevate nuovamente al successivo tentativo 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 gli indirizzi IP da controllare con una qualsiasi delle chiavi *icmpping** in un file temporaneo, che viene poi passato a fping. Se gli item hanno parametri della chiave diversi, solo quelli con parametri della chiave identici vengono scritti in un unico file. Tutti gli indirizzi IP scritti nello stesso file verranno controllati da fping in parallelo, quindi il processo ICMP pinger di Zabbix impiegherà una quantità di tempo fissa indipendentemente dal numero di indirizzi IP nel file.

1 Elementi chiave di monitoraggio VMware

L'elenco del monitoraggio VMware elementi chiave è stato spostato nella sezione monitoraggio VMware.

6 Monitoraggio del file di registro

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 modelli di stringa.

Per monitorare un file di log è necessario disporre di:

- Zabbix agent in esecuzione sul host
- item di monitoraggio dei log configurato

Attention:
Il limite di dimensione di un file di log monitorato dipende dal supporto per file di grandi dimensioni.

Configurazione

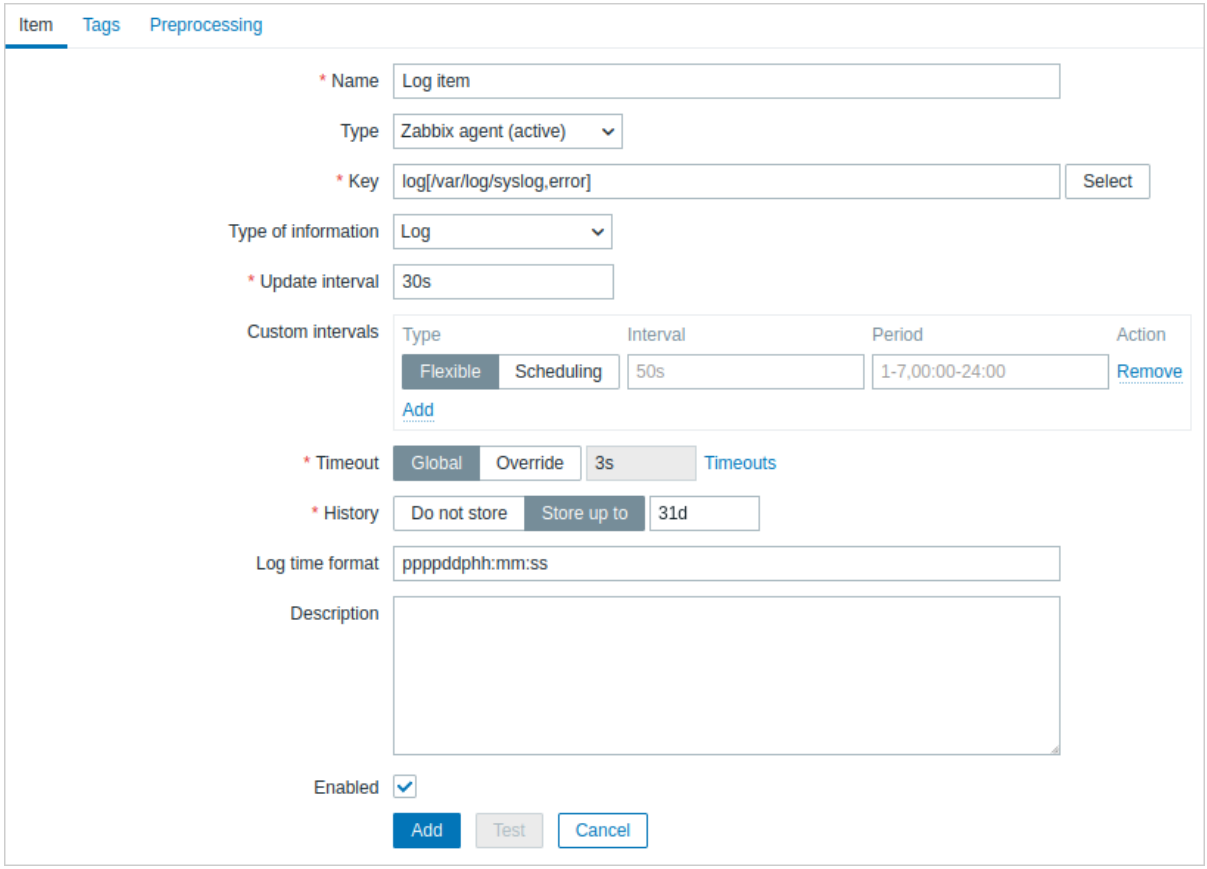
Verificare i parametri dell'agent

Assicurarsi che nel file di configurazione dell'agent:

- il parametro `Hostname` corrisponda al nome del host nel frontend.
- i server nel parametro `ServerActive` siano specificati per l'elaborazione dei controlli attivi.

Configurazione dell'item

Configurare un item di monitoraggio dei log.



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

In particolare, per gli item di monitoraggio dei log è necessario inserire:

Type	Selezionare qui **Zabbix agent (active)**.
Key	Utilizzare una delle seguenti chiavi item: **log[]** oppure **logrt[]**: Queste due chiavi item consentono di monitorare i log e filtrare le voci di log in base al regexp del contenuto, se presente. Ad esempio: `log[/var/log/syslog,error]`. Assicurarsi che il file abbia i permessi di lettura per l'utente 'zabbix', altrimenti lo stato dell'item verrà impostato su 'unsupported'. **log.count[]** oppure **logrt.count[]**: Queste due chiavi item consentono di restituire solo il numero di righe corrispondenti. Per i dettagli sull'utilizzo di queste chiavi item e dei relativi parametri, vedere la sezione delle chiavi Zabbix agent item supportate.
Type of information	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 il tipo di informazione appropriato diverso da Log. Si noti che la scelta di un tipo di informazione diverso da Log comporterà la perdita del timestamp locale.
Update interval (in sec)	Il parametro definisce con quale frequenza Zabbix agent controllerà eventuali modifiche nel file di log. Impostandolo a 1 secondo si avrà la certezza di ricevere i nuovi record il prima possibile.

Log time format	In questo campo è possibile specificare facoltativamente il modello per analizzare il 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 verrà impostato a 0 nel tempo Unix, che rappresenta il 1 gennaio 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 carattere per il PID, seguite da data, ora e dal resto del messaggio. Il formato orario 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".

Note importanti

- Il server e l'agent mantengono traccia della dimensione di un log monitorato e dell'ora dell'ultima modifica (per logrt) in due contatori. Inoltre:
  - L'agent utilizza internamente anche i numeri di inode (su UNIX/GNU/Linux), gli indici di file (su Microsoft Windows) e le somme MD5 dei primi 512 byte del file di log per migliorare le decisioni quando i file di log vengono troncati e ruotati.
  - Sui sistemi UNIX/GNU/Linux si presume che i file system in cui sono memorizzati i file di log riportino i numeri di 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 utilizza:
    * Sui file system NTFS indici di file a 64 bit.
    * Sui file system ReFS (solo da Microsoft Windows Server 2012) ID file a 128 bit.
    * Sui file system in cui gli indici di file cambiano (ad esempio FAT32, exFAT) viene utilizzato un algoritmo di fallback per adottare un approccio ragionevole in condizioni incerte quando la rotazione dei file di log produce più file di log con la stessa ora di ultima modifica.
  - I numeri di inode, gli indici di file e le somme MD5 vengono raccolti internamente da Zabbix agent. Non vengono trasmessi a Zabbix server e vengono persi quando Zabbix agent viene arrestato.
  - Non modificare l'ora di ultima modifica di un file di log (ad esempio con `touch`) e non sostituire un file di log monitorato copiando un file di nuovo nel suo nome originale (questo crea un nuovo inode). In entrambi i casi Zabbix potrebbe trattare il file come un file diverso e rileggerlo dall'inizio, producendo avvisi duplicati.
  - Se ci sono diversi file di log corrispondenti per l'item `logrt[]` e Zabbix agent sta seguendo il più recente tra 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[]` controllato.
- 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'ora dell'ultima modifica (il contatore del tempo) sono memorizzati nel database Zabbix e inviati all'agent per assicurarsi che l'agent inizi a leggere il file di log da questo punto nei casi in cui l'agent sia appena stato avviato o abbia ricevuto item che in precedenza erano disabilitati o non supportati. Tuttavia, se l'agent ha ricevuto dal server un contatore della dimensione diverso da zero, 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 successivamente.
- 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 ci sono diversi file corrispondenti con la stessa ora di ultima modifica, allora 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 riportati nel loro ordine originale. Per altri schemi di rotazione l'ordine originale dei file di log non verrà rispettato, il che può portare alla segnalazione dei record dei file di log corrispondenti in 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 *Intervallo di aggiornamento* 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 sovrascrive il valore predefinito fornito dal parametro **MaxLinesPerSecond** nel file di configurazione dell'agent.
- Per trovare la stringa richiesta Zabbix elaborerà 10 volte più righe nuove di quelle impostate in MaxLinesPerSecond. Pertanto, ad esempio, se un item `log[]` o `logrt[]` ha un *Intervallo di aggiornamento* 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 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 controllo. Se l'*Intervallo di aggiornamento* è impostato su 2 secondi, i limiti per un controllo saranno 2 volte più alti rispetto a un *Intervallo di aggiornamento* 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 ci sono valori non-log al suo interno. Quindi, affinché i valori **maxlines** vengano inviati in una sola connessione (e non in più connessioni), il parametro BufferSize dell'agent deve essere almeno maxlines x 2. Zabbix agent può caricare dati durante la raccolta dei log e quindi liberare il buffer, mentre Zabbix agent 2 interromperà la raccolta dei log finché i dati non saranno caricati e il buffer liberato, operazione eseguita in modo asincrono.
- In assenza di item di log, tutta la dimensione del buffer dell'agent viene utilizzata per valori non-log. Quando arrivano valori di log, essi sostituiscono i valori non-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 potrebbe essere analizzato di nuovo e in modo diverso dopo il riavvio dell'agent.
- 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 "." è preceduto da escape oppure è il primo simbolo del nome del file.
- Le espressioni regolari per `logrt` sono supportate solo nel nome file; la corrispondenza tramite espressione regolare della directory non è supportata.
- Sulle piattaforme UNIX un item `logrt[]` diventa NOTSUPPORTED se una directory in cui ci si aspetta di trovare i file di log non esiste.
- Su Microsoft Windows, se una directory non esiste l'item non diventerà NOTSUPPORTED (ad esempio, se il nome della directory è scritto 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 proprio file di log dell'agent, tranne quando DebugLevel=4 o DebugLevel=5.
- La ricerca di un punto interrogativo usando 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 potremmo voler 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 hanno la capacità di estrarre i valori desiderati dalle righe corrispondenti. Ciò viene ottenuto tramite il parametro aggiuntivo **output** negli item `log` e `logrt`.

L'uso del parametro 'output' consente di indicare il "gruppo di cattura" della corrispondenza che potrebbe interessarci.

Quindi, per esempio

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

dovrebbe consentire di restituire il conteggio delle voci come trovato 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** si riferisce 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 trigger.

Utilizzo del parametro maxdelay

Il parametro `maxdelay` negli item di log consente di ignorare alcune righe meno recenti dei file di log per fare in modo che vengano analizzate le righe più recenti entro i `maxdelay` secondi.

> **Warning:**
> Specificare `maxdelay` > 0 può portare a **ignorare record importanti del file di log e a perdere avvisi**. Usarlo con cautela, a proprio 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 verranno inviate al server come configurato negli item `log` e `logrt`.

La protezione integrata contro il sovraccarico consiste in un parametro `maxlines` configurabile (protegge il server da troppe righe di log corrispondenti in arrivo) e in un limite di 10*`maxlines` (protegge CPU e I/O dell'host dal sovraccarico causato dall'agent in un singolo controllo). Tuttavia, la protezione integrata presenta ancora 2 problemi. Primo, un gran numero di messaggi potenzialmente poco informativi viene segnalato al server e occupa spazio nel database. Secondo, a causa del numero limitato di righe analizzate al secondo, l'agent può rimanere indietro rispetto ai record di log più recenti per ore. Molto probabilmente, si preferisce essere informati prima sulla situazione corrente nei file di log invece di scorrere vecchi record per ore.

La soluzione a entrambi i problemi è usare il 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. Da questi valori l'agent calcola un ritardo stimato, cioè quanti secondi sarebbero necessari per analizzare tutti i record rimanenti in un file di log.

Se il ritardo non supera `maxdelay`, l'agent procede con l'analisi del file di log come di consueto.

Se il ritardo è maggiore di `maxdelay`, allora l'agent **ignora una porzione del file di log "saltandola"** fino a una nuova posizione stimata, in modo che le righe rimanenti possano essere analizzate entro `maxdelay` secondi.

Si noti che l'agent non legge nemmeno le righe ignorate nel buffer, ma calcola una posizione approssimativa a cui saltare nel file.

Il fatto che vengano saltate righe del file di log viene registrato nel file di log dell'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" l'agent regola 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, si possono osservare nessun "salto", "salti" rari o frequenti, "salti" grandi o piccoli, oppure persino un piccolo "salto" a ogni controllo. Anche le fluttuazioni del carico di sistema e della latenza di rete influiscono sul calcolo del ritardo e quindi sul "saltare" in avanti per rispettare il parametro `maxdelay`.

Non è consigliato impostare `maxdelay` < `update interval` (potrebbe causare frequenti piccoli "salti").

Note sulla gestione della rotazione dei file di log con 'copytruncate'

`logrt` con l'opzione `copytruncate` presuppone che file di log diversi contengano record diversi (almeno i loro timestamp siano 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, l'altro è una copia.

`logrt` con l'opzione `copytruncate` cerca di elaborare correttamente le copie dei file di log senza segnalare duplicati. Tuttavia, non è consigliato, ad esempio, produrre più copie del file di log con lo stesso timestamp, eseguire la rotazione del file di log più frequentemente dell'intervallo di aggiornamento dell'item logrt[], o riavviare spesso l'agent. L'agent cerca di gestire tutte queste situazioni in modo ragionevolmente corretto, ma non è possibile garantire buoni risultati in tutte le circostanze.

Note sui file persistenti per gli elementi 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. In base alla dimensione effettiva del file di log e all'ora di modifica riportata 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 tracciare 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 memorizzare in un file questo stato dell'item log[], log.count[], logrt[] o logrt.count[]. 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 entrambe le copie mirrorate. Successivamente, 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 dei log elaborati 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.

Funzionamento 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 separazione 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 sul supporto di memorizzazione (fsync() non viene chiamato).

La sovrascrittura con i dati più recenti viene eseguita dopo la corretta segnalazione a Zabbix server del record del file di log corrispondente o dei metadati (dimensione del log elaborato e ora di modifica). Ciò può avvenire con una frequenza pari a ogni controllo dell'item, se il file di log continua a cambiare.

Non sono previste azioni speciali 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 da quella precedente.

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 potrebbero risultare utili.

Se l'agent viene arrestato prima che siano trascorse 24 ore, i file obsoleti non verranno eliminati, poiché Zabbix agent non riceve più dal Zabbix server informazioni sulla loro posizione.

Warning:
Riconfigurare **persistent_dir** di un item di log riportandolo alla vecchia posizione **persistent_dir** mentre l'agent è arrestato, senza che l'utente elimini il vecchio file persistente, causerà il ripristino dello stato dell'agent dal vecchio file persistente, con conseguenti messaggi mancati o falsi avvisi.

Denominazione e posizione dei file persistenti

Zabbix agent distingue i controlli attivi in base alle loro 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, non nell'agent).

Il nome del file è composto dalla somma MD5 della chiave dell'item con la lunghezza della chiave dell'item aggiunta per ridurre la possibilità di collisioni. Ad esempio, lo stato dell'item logrt[/home/zabbix50/test.log„„„„/home/zabbix50/agent_private] verrà mantenuto nel file persistente c963ade4008054813bbc0a650bb8e09266.

Più item di log possono utilizzare lo stesso valore di **persistent_dir**.

persistent_dir viene specificato tenendo conto di layout specifici del file system, punti di mount, opzioni di mount e configurazione del mirroring dello storage: il file persistente deve trovarsi nello stesso file system con mirroring del file di log monitorato.

Se la directory **persistent_dir** non può essere creata o non esiste, oppure se i diritti di accesso per Zabbix agent non consentono di creare/scrivere/leggere/eliminare file, l'item di 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, disco pieno), gli errori vengono registrati nel file di log dell'agent, ma il log item 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 trova 70 record corrispondenti, i primi 50 record verranno inviati in un batch, il file persistente verrà aggiornato, quindi i restanti 20 record verranno inviati (eventualmente con un certo ritardo, quando verranno accumulati più dati) nel secondo batch, e il file persistente verrà aggiornato di nuovo.

Azioni in caso di errore di comunicazione tra agent e server

Ogni riga corrispondente degli item `log[]` e `logrt[]` e ogni risultato di controllo degli item `log.count[]` e `logrt.count[]` richiede uno slot libero nell'area dedicata del 50% nel buffer di invio dell'agent. Gli elementi del buffer vengono inviati regolarmente al server (o proxy) e gli slot del buffer tornano nuovamente liberi.

Finché ci sono slot liberi nell'area log designata del buffer di invio dell'agent e la comunicazione tra agent e server (o proxy) non riesce, i risultati del monitoraggio dei log vengono accumulati nel buffer di invio. Questo aiuta a mitigare brevi interruzioni della comunicazione.

Durante interruzioni della comunicazione più lunghe, tutti gli slot log vengono occupati e vengono intraprese le seguenti azioni:

- I controlli degli item `log[]` e `logrt[]` vengono arrestati. Quando la comunicazione viene ripristinata e sono disponibili slot liberi nel buffer, i controlli riprendono dalla posizione precedente. Nessuna riga corrispondente viene persa, viene solo segnalata più tardi.
- I controlli `log.count[]` e `logrt.count[]` vengono arrestati se `maxdelay = 0` (predefinito). Il comportamento è simile a quello degli item `log[]` e `logrt[]` come 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 arrestato. 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.
- 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" sulle righe del file di log, allora viene mantenuta la posizione successiva al "salto" e il risultato conteggiato viene scartato. In questo modo, l'agent cerca di stare al passo con un file di log in crescita anche in caso di errore di 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 NOTSUPPORTED con un messaggio di errore. Per continuare il monitoraggio dell'item di log, l'espressione regolare deve essere corretta.

Se l'espressione regolare viene compilata correttamente, ma fallisce in fase di esecuzione (su alcuni o su tutti i record di log), allora l'item di log rimane supportato e il monitoraggio continua. L'errore di esecuzione viene registrato nel file di log dello Zabbix agent (senza il record del file di log).

La frequenza di registrazione è limitata a un errore di esecuzione per controllo, per consentire allo Zabbix agent di monitorare il proprio file di log. Ad esempio, se vengono analizzati 10 record e 3 record falliscono con un errore di esecuzione della regexp, nel log dell'agent viene prodotto un record.

Eccezione: se `MaxLinesPerSecond=1` e l'intervallo di aggiornamento=1 (è consentito analizzare un solo record per controllo), gli errori di esecuzione della regexp non vengono registrati.

zabbix_agentd registra la chiave dell'item in caso di errore di esecuzione, zabbix_agent2 registra l'ID dell'item per aiutare a identificare quale item di log presenta errori di esecuzione. In caso di errori di esecuzione, si consiglia di riprogettare l'espressione regolare.

7 Elementi calcolati

Panoramica

Un item calcolato consente di creare un calcolo basato sui valori di alcuni item esistenti. Ad esempio, potresti voler calcolare la media oraria del valore di un item oppure calcolare il valore totale per un gruppo di item. È proprio a questo che serve un item calcolato.

I calcoli possono usare entrambi:

- valori singoli di item individuali
- un filtro complesso per selezionare più item per l'aggregazione (vedi calcoli aggregati per i dettagli)

Gli item calcolati sono un modo per creare origini dati virtuali. Tutti i calcoli vengono eseguiti solo da Zabbix server. I valori vengono calcolati periodicamente in base all'espressione aritmetica utilizzata.

I dati risultanti vengono memorizzati nel database di Zabbix come per qualsiasi altro item; vengono salvati sia i valori di storico sia quelli di trend ed è possibile generare grafici.

> **Note:**
> Se il risultato del calcolo è un valore float, verrà troncato a un intero se il tipo di informazione dell'item calcolato è *Numeric (unsigned)*.
> Inoltre, 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.

Gli item calcolati condividono la loro sintassi con le espressioni dei trigger. Se hai familiarità con le espressioni dei trigger, puoi considerare gli item calcolati come funzioni che analizzano alcuni valori, ma senza il confronto con una soglia.

Negli item calcolati è consentito il confronto con stringhe. Gli item calcolati possono essere referenziati da macro o da altre entità come qualsiasi altro tipo di item.

Per utilizzare gli item calcolati, scegli il tipo di item **Calculated**.

Campi configurabili

La **key** è un identificatore univoco dell'item (per host). Puoi creare qualsiasi nome di key utilizzando i simboli supportati.

La definizione del calcolo deve essere inserita nel campo **Formula**. Non esiste alcuna connessione tra la formula e la key. I parametri della key non vengono utilizzati nella formula in alcun modo.

La sintassi di una formula semplice è:

`function(/host/key,<parameter1>,<parameter2>,...)`

dove:

Element	Description
`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`	Key dell'item utilizzato per il calcolo. Gli item che restituiscono valori con tipo di dato binario o JSON data type non sono supportati.
`parameter(s)`	Parametri della funzione, se richiesti. Sono supportati i suffissi temporali e i suffissi di dimensione della memoria.

Attention:
Le macro utente nella formula verranno espanse se utilizzate per fare riferimento a un parametro di funzione, a un parametro di filtro item o a una costante. Le macro utente NON verranno espanse se fanno riferimento a una funzione, al nome host, alla key dell'item, a un parametro della key dell'item o a un operatore.

Una formula più complessa può utilizzare una combinazione di funzioni, operatori e parentesi. Puoi usare tutte le funzioni e gli operatori supportati nelle espressioni dei trigger. La logica e la precedenza degli operatori sono esattamente le stesse.

A differenza delle espressioni dei trigger, Zabbix elabora gli item calcolati in base all'intervallo di aggiornamento dell'item, non alla ricezione di un nuovo valore.

Tutti gli item a cui fanno riferimento le funzioni di storico nella formula dell'item calcolato devono esistere e raccogliere dati. Inoltre, se modifichi la key dell'item di un item referenziato, devi aggiornare manualmente tutte le formule che utilizzano quella key.

Un item calcolato può diventare non supportato in diversi casi:

- item referenziato/i
  - non trovato/i
  - disabilitato/i
  - appartenente/i a un host disabilitato
  - non supportato/i (tranne con la funzione nodata() e gli operatori con valori sconosciuti)
- nessun dato per calcolare una funzione
- divisione per zero
- sintassi non corretta utilizzata

Esempi di utilizzo

Esempio 1

Calcolo della percentuale di spazio libero su "/".

Utilizzo della funzione **ultimo**:

`100*ultimo(//vfs.fs.size[/,gratuito])/ultimo(//vfs.fs.size[/,totale])`

Zabbix prenderà i valori più recenti per gli spazi su disco liberi e totali e calcolare la percentuale in base alla formula fornita.

Esempio 2

Calcolo di una media di 10 minuti del numero di valori elaborati da Zabbix.

Uso della funzione **avg**:

`avg(/Zabbix Server/zabbix[wcache,values],10m)`

Si noti che un uso estensivo di item calcolati con lunghi periodi di tempo può influire sulle prestazioni del server Zabbix.

Esempio 3

Calcolo della larghezza di banda totale su eth0.

Somma di due funzioni:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Esempio 4

Calcolo della percentuale del traffico in ingresso.

Espressione più complessa:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Vedere anche: Esempi di calcoli aggregati

1 Calcoli aggregati

Panoramica

I calcoli aggregati sono un tipo di elemento calcolato che consente di raccogliere informazioni da diversi elementi tramite il server Zabbix e quindi calcolare un aggregato, a seconda della funzione di aggregazione utilizzata.

I calcoli aggregati non richiedono alcun agent in esecuzione sull'host da monitorare.

Sintassi

Per recuperare gli aggregati utilizzare una delle funzioni aggregate 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.

> **Note:**
> Se l'aggregato produce un valore float, questo verrà troncato a un intero se il tipo di informazione dell'item aggregato è *Numeric (unsigned)*.

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 utilizzo

Esempi di chiavi per calcoli aggregati.

Esempio 1

Spazio totale su disco del gruppo host 'MySQL Servers'.

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Esempio 2

Somma degli ultimi valori di tutti gli item che corrispondono a net.if.in[*] sull'host.

```
sum(last_foreach(/host/net.if.in[*]))
```

Esempio 3

Carico medio del processore del gruppo di host 'MySQL Servers'.

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Esempio 4

Media a 5 minuti del numero di query al secondo per il gruppo di host 'MySQL Servers'.

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Esempio 5

Carico medio della CPU su tutti gli host in più gruppi di host che hanno i tag specifici.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Esempio 6

Calcolo utilizzato sulle somme degli ultimi valori degli item di un intero gruppo di host.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Esempio 7

Il numero totale di item non supportati nel gruppo host 'Zabbix servers'.

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Esempio 8

Somma degli ultimi risultati numerici dei controlli DNS su tutti gli host. La forma dell'item mostrata è `net.dns[192.0.2.0,example.com,A]` come esempio di una possibile chiave.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Si noti che i caratteri jolly devono corrispondere al numero di parametri nella chiave (qui `net.dns` ha tre parametri: ip, name, type).

Esempi di sintassi corretta/non corretta

Le espressioni (incluse le chiamate di funzione) non possono essere utilizzate come parametri delle funzioni history, trend o foreach. Tuttavia, queste funzioni stesse possono essere utilizzate in altri parametri di funzione (non storici).

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))
```

non è possibile garantire che entrambe le parti dell'equazione abbiano sempre lo stesso insieme di valori. Mentre una parte dell'espressione viene valutata, può arrivare un nuovo valore per il periodo richiesto e quindi l'altra parte dell'espressione avrà un insieme di valori diverso.

8 Controlli interni

Panoramica

I controlli interni consentono di monitorare i processi interni di Zabbix. In altre parole, è possibile monitorare ciò che accade in 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 da server o proxy indipendentemente dallo stato di manutenzione del host.

Per utilizzare questo item, scegliere 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 item	Descrizione
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 connettore.
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) sull'host.
zabbix[host,,items_unsupported]	Il numero di item non supportati abilitati sull'host.
zabbix[host,,maintenance]	Lo stato di manutenzione corrente dell'host.
zabbix[host,active_agent,available]	La disponibilità dei controlli agent attivi sull'host.
zabbix[host,discovery,interfaces]	I dettagli di tutte le interfacce configurate dell'host nel frontend di Zabbix.
zabbix[host,,available]	La disponibilità dell'interfaccia principale di un determinato tipo di controlli sull'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 su Zabbix Java gateway.
zabbix[lld_queue]	Il numero di valori accodati nella coda di elaborazione del low-level discovery.
zabbix[preprocessing]	Statistiche dei valori ricevuti dal gestore di preprocessing.
zabbix[preprocessing_queue]	Il numero di valori accodati nella coda di preprocessing.
zabbix[process,,,]	La percentuale di tempo che un determinato processo Zabbix o un gruppo di processi (identificato da <type> e <mode>) ha trascorso in <state>.
zabbix[proxy,,]	Le informazioni su Zabbix proxy.
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.
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 di utilizzo del buffer di memoria del proxy.

Chiave item	Descrizione
zabbix[proxy_buffer,state,changes]	Restituisce il numero di cambi di stato tra le modalità buffer su disco/in memoria dall'avvio.
zabbix[proxy_buffer,state,current]	Restituisce lo stato operativo corrente in cui vengono memorizzati 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 in ritardo di almeno <from> secondi, ma di meno di <to> secondi.
zabbix[rcache,,]	Le statistiche di disponibilità della cache di configurazione di Zabbix.
zabbix[requiredperformance,]	Le prestazioni richieste di Zabbix server o Zabbix proxy, in termini di nuovi valori al secondo previsti.
zabbix[stats,,]	Restituisce le metriche interne di un server o proxy Zabbix. Se vengono specificati <ip> e <port>, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
zabbix[stats,,,queue,,]	Restituisce le metriche interne della coda di un server o proxy Zabbix. Se vengono specificati <ip> e <port>, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
zabbix[tcache,,]	Le statistiche di efficacia della cache delle funzioni di trend di Zabbix.
zabbix[triggers]	Il numero di trigger abilitati nel database di Zabbix, con tutti gli item abilitati su host abilitati.
zabbix[uptime]	Il tempo di attività del processo Zabbix server o proxy, in secondi.
zabbix[vcache,buffer,]	Le statistiche di disponibilità della cache dei valori di Zabbix.
zabbix[vcache,cache,]	Le statistiche di efficacia della cache dei valori di Zabbix.
zabbix[version]	La versione di Zabbix server o proxy.
zabbix[vmware,buffer,]	Le statistiche di 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 di 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[boottime]

 L'ora di avvio del processo di Zabbix server o Zabbix proxy in secondi.
 Valore restituito: *Integer*.

zabbix[cluster,discovery,nodes]

 Rileva i nodi del cluster ad alta disponibilità.
 Valore restituito: *oggetto JSON*.

Commenti:

- Questo item può essere utilizzato nel low-level discovery.
- I campi restituiti corrispondono per lo più alle proprietà dell'oggetto nodo di 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* - sconosciuto; *1* - disponibile; *2* - non disponibile.

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* (per esempio: "8.0.0").

Parametri:

- **param** - *ping* oppure *version*.

Commenti:

- Questo item può essere utilizzato per verificare la disponibilità di Java gateway usando la funzione trigger `nodata()`.
- Il secondo parametro deve essere vuoto ed è riservato per utilizzi futuri.

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.

zabbix[preprocessing]

 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 particolare processo Zabbix o un gruppo di processi (identificati da <type> e <mode>) ha trascorso nello stato <state>. Viene calcolata solo per l'ultimo minuto.
 Valore restituito: *Float*.

Parametri:

- **type** - per i processi 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 proxy: *agent poller*, *availability manager*, *browser poller*, *configuration syncer*, *data sender*, *discovery manager*, *discovery worker*, *history syncer*, *housekeeper*, *http agent poller*, *http poller*, *icmp pinger*, *internal poller ipmi manager*, *ipmi poller*, *java poller*, *odbc poller*, *poller*, *preprocessing manager*, *preprocessing worker*, *self-monitoring*, *snmp poller*, *snmp trapper*, *task manager*, *trapper*, *unreachable poller*, *vmware collector*;
- **mode** - *avg* - 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 (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 sta facendo nulla.

Commenti:

- Se <mode> è un numero di processo Zabbix che non è in esecuzione (ad esempio, con 5 poller in esecuzione viene specificato <mode> uguale a 6), tale item diventerà non supportato.
- Il minimo e il massimo si riferiscono alla percentuale di utilizzo di un singolo processo. Quindi, 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 riportano ciò che stanno facendo nella memoria condivisa e il processo di self-monitoring riepiloga tali dati ogni secondo. Le modifiche di stato (busy/idle) vengono registrate al momento del cambiamento; pertanto un processo che diventa busy viene registrato come tale e non cambia né aggiorna lo stato finché non diventa idle. Questo garantisce che anche i processi completamente bloccati vengano registrati correttamente come busy al 100%.
- Attualmente, "busy" significa "non in sleep", ma in futuro potrebbero essere introdotti stati aggiuntivi: attesa di lock, esecuzione di query al database, ecc. Si noti che i poller asincroni sono considerati busy se hanno raggiunto il limite impostato dal parametro di configurazione MaxConcurrentChecksPerPoller 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] #il tempo medio dei processi poller trascorso a fare qualcosa durante l'ult
zabbix[process,"icmp pinger",max,busy] #il tempo massimo trascorso a fare qualcosa da un qualsiasi process
zabbix[process,"history syncer",2,busy] #il tempo trascorso a fare qualcosa da history syncer numero 2 dur
zabbix[process,trapper,count] #il numero di processi trapper attualmente in esecuzione
```

zabbix[proxy,<name>,<param>]

 Le informazioni sul proxy Zabbix.
 Valore restituito: *Intero*.

Parametri:

- **name** - il nome del proxy;
- **param** - *lastaccess* - il timestamp dell'ultimo messaggio heartbeat ricevuto dal proxy;
delay - per quanto tempo i valori raccolti non sono stati inviati; calcolato come "proxy delay" + ("ora corrente del server" - "proxy lastaccess"), dove "proxy delay" è la differenza tra l'ora corrente del proxy e il timestamp del valore non inviato più vecchio sul proxy.

Commenti:

- Questo item viene sempre elaborato dal server Zabbix indipendentemente dalla posizione dell'host (sul server o sul 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, ultimo accesso, 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*.

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.

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.

zabbix[proxy group,<name>,proxies]

 L'elenco dei proxy Zabbix in un gruppo di proxy con nome, modalità, crittografia, compressione, versione, ultimo accesso, 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/offline/online).
 Valore restituito: *JSON*.

Parametri:

- **name** - il nome del gruppo di proxy.

zabbix[proxy group,<name>,state]

 Lo stato di un gruppo di proxy.
 Valore restituito: *0* - sconosciuto; *1* - offline; *2* - in ripristino; *3* - online; *4* - in degrado.

Parametri:

- **name** - il nome del gruppo di proxy.

zabbix[proxy group,discovery]

 Restituisce un elenco di gruppi di proxy con dati di configurazione e dati in tempo reale. I dati di configurazione includono il nome del gruppo di proxy, il ritardo di failover e il numero minimo richiesto di proxy online. I dati in tempo reale includono lo stato del gruppo di proxy (vedere i commenti per i dettagli), il numero di proxy online e la percentuale di proxy online.
 Valore restituito: *JSON*.

Commenti:

- Questo item non restituisce i proxy senza gruppo.
- Se è presente un valore non valido per "failover_delay" o "min_online", viene riportato un valore speciale *-1* per indicarlo. Possono verificarsi valori non validi se per la configurazione vengono utilizzate macro e queste non possono essere espanse in un valore valido.
- Lo stato del gruppo di proxy viene riportato come intero: *0* - sconosciuto; *1* - offline; *2* - in ripristino; *3* - online; *4* - in degradazione.

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>]

 Le statistiche di utilizzo del buffer di memoria del proxy.
 Valori restituiti: *Integer* (per la dimensione); *Float* (per la percentuale).

Parametri:

- **mode**: *total* - la dimensione totale del buffer (può essere usata per verificare se il buffer di memoria è abilitato);
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:

- Restituisce un errore 'Proxy memory buffer is disabled' quando il buffer di memoria è disabilitato;
- Questo item non è supportato su Zabbix server.

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 di funzionamento corrente in cui vengono memorizzati i nuovi dati.
 Valori restituiti: *0* - disco; *1* - memoria.

Commenti:

- "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: *Intero*.

Commenti:

- Questo item non è supportato su Zabbix server.

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: *Intero*.

Parametri:

- **from** - in ritardo di almeno (il valore predefinito è 6 secondi);
- **to** - in ritardo al massimo (il valore predefinito è infinito).

Commenti:

- I suffissi temporali (s,m,h,d,w) sono supportati nei parametri.

> **Attention:**
> Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calcolo delle code durante la manutenzione per i dettagli.

zabbix[rcache,<cache>,<mode>]

 The availability statistics of the Zabbix configuration cache.
 Return values: *Integer* (for size); *Float* (for percentage).

Parameters:

- **cache** - *buffer*;
- **mode** - *total* - the total size of buffer
free - the size of free buffer
pfree - the percentage of free buffer
used - the size of used buffer
pused - the percentage of used buffer

zabbix[requiredperformance]

 Le prestazioni richieste di Zabbix server o Zabbix proxy, espresse come nuovi valori al secondo previsti.
 Valore restituito: *Float*.

Commenti:

- Corrisponde approssimativamente a "Prestazioni richieste del server, nuovi valori al secondo" in *Reports* > *System information*.

zabbix[stats,<ip>,<port>]

 Restituisce le metriche interne di un server o proxy Zabbix. Se vengono specificati <ip> e <port>, le metriche vengono recuperate dall'istanza remota; altrimenti, dall'istanza locale.
 Valori restituiti: *oggetto JSON*.

Parametri:

- **ip** - l'elenco di IP/DNS/maschere di rete di 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 stats sarà accettata solo dagli indirizzi elencati nel parametro 'StatsAllowedIP' del server/proxy nell'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 della coda interna (vedi `zabbix[queue,<from>,<to>]`) 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/maschere di rete di 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** - in ritardo di almeno (il valore predefinito è 6 secondi);
- **to** - in ritardo di al massimo (il valore predefinito è infinito).

Commenti:

- La richiesta stats sarà accettata solo dagli indirizzi elencati nel parametro 'StatsAllowedIP' server/proxy sull'istanza di destinazione.
- Un insieme selezionato di metriche interne viene restituito da questo item. Per i dettagli, vedere Monitoraggio remoto delle statistiche di Zabbix.

Attention:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; per i dettagli, vedere Calcolo delle code durante la manutenzione.

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 - mancate corrispondenze della cache;
pmisses - percentuale di mancate corrispondenze 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 molto probabilmente indica che la dimensione della cache può essere ridotta.

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 memoria insufficiente.

Parametri:

- **parameter** - *requests* - il numero totale di richieste;
hits - il numero di hit della cache (valori della cronologia presi dalla cache);
misses - il numero di miss della cache (valori della cronologia presi dal database);
mode - la modalità operativa della cache dei valori.

Commenti:

- Una volta attivata la modalità memoria insufficiente, 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 utilizzare questa chiave con il passaggio di preprocessing *Change per second* per ottenere statistiche dei valori al secondo.
- Questo item non è supportato su Zabbix proxy.

zabbixversion

 La versione del server o del proxy Zabbix.
 Valore restituito: *String*. Ad esempio: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 The availability statistics of the Zabbix vmware cache.
 Return values: *Integer* (for size); *Float* (for percentage).

Parameters:

- **mode** - *total* - the total size of buffer
free - the size of free buffer
pfree - the percentage of free buffer
used - the size of used buffer
pused - the percentage of used buffer

zabbix[vps,written]

 Il numero totale di valori dello storico scritti nel database.
 Valore restituito: *Integer*.

zabbix[wcache,<cache>,<mode>]

 Le statistiche e la disponibilità della cache di scrittura di Zabbix.
 Valori restituiti: *Integer* (per numero/dimensione); *Float* (per percentuale).

Parametri:

- **cache** - *values*, *history*, *index* o *trend*;
- **mode** - (con *values*) *all* (predefinito) - il numero totale di valori elaborati da Zabbix server/proxy, eccetto gli item non supportati (contatore);
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 testo 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 dell'item ha comportato che l'item diventasse non supportato o mantenesse tale stato (contatore);
(con cache *history*, *index*, *trend*) *pfree* (predefinito) - la percentuale di buffer libero;
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 della cache `trend` non è supportato con Zabbix proxy.
- La cache history viene utilizzata per memorizzare i valori degli item. Un numero basso indica problemi di prestazioni lato database.
- La cache history index viene utilizzata per indicizzare i valori memorizzati nella cache history.
- Dopo che la cache history è stata riempita e poi svuotata, la cache history index manterrà comunque alcuni dati. Questo comportamento è previsto e aiuta il sistema a funzionare in modo più efficiente evitando l'elaborazione aggiuntiva necessaria per ridimensionare costantemente la memoria.
- La cache trend memorizza l'aggregato dell'ora corrente per tutti gli item che ricevono dati.
- È possibile utilizzare la chiave zabbix[wcache,values] con il passaggio di preprocessing *Change per second* per ottenere statistiche dei valori al secondo.

9 controlli SSH

Panoramica

I controlli SSH vengono eseguiti come monitoraggio senza agent. Zabbix agent non è necessario per i controlli SSH.

Per eseguire i controlli SSH, Zabbix server deve essere inizialmente configurato con il supporto SSH2 (libssh o libssh2). Vedere anche: Requisiti.

Attention:
A partire da RHEL 8, è supportato solo libssh. Per le 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 tramite file 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=
```

Rimuoverne il commento e impostare il percorso completo della cartella in cui saranno collocate le chiavi pubblica e privata:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Salvare il file e successivamente riavviare 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 pubblica e privata verranno generate da un comando ssh-keygen all'interno della directory home.

Di solito i pacchetti di installazione del server Zabbix provenienti da 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 del server Zabbix `SSHKeyLocation` menzionato sopra.

Note:
I passaggi seguenti 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 cambiare la directory home dell'account utente *zabbix*, tutti i processi in esecuzione che la stanno utilizzando devono essere arrestati:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Per cambiare la posizione della directory home tentando anche di spostarla (se esiste), deve essere eseguito il seguente comando:

```
usermod -m -d /home/zabbix zabbix
```

È anche possibile che una directory home non esistesse nella vecchia posizione, quindi dovrà essere creata nella nuova posizione. Un tentativo sicuro per farlo è:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Per assicurarsi che tutto sia sicuro, 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 riavviati:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Ora è possibile eseguire i passaggi per generare le chiavi pubblica e privata 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 pubblica e privata (*id_rsa.pub* e *id_rsa*) sono state generate per impostazione predefinita nella directory */home/zabbix/.ssh*, che corrisponde al parametro di configurazione del server Zabbix `SSHKeyLocation`.

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 utilizzata 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.
```

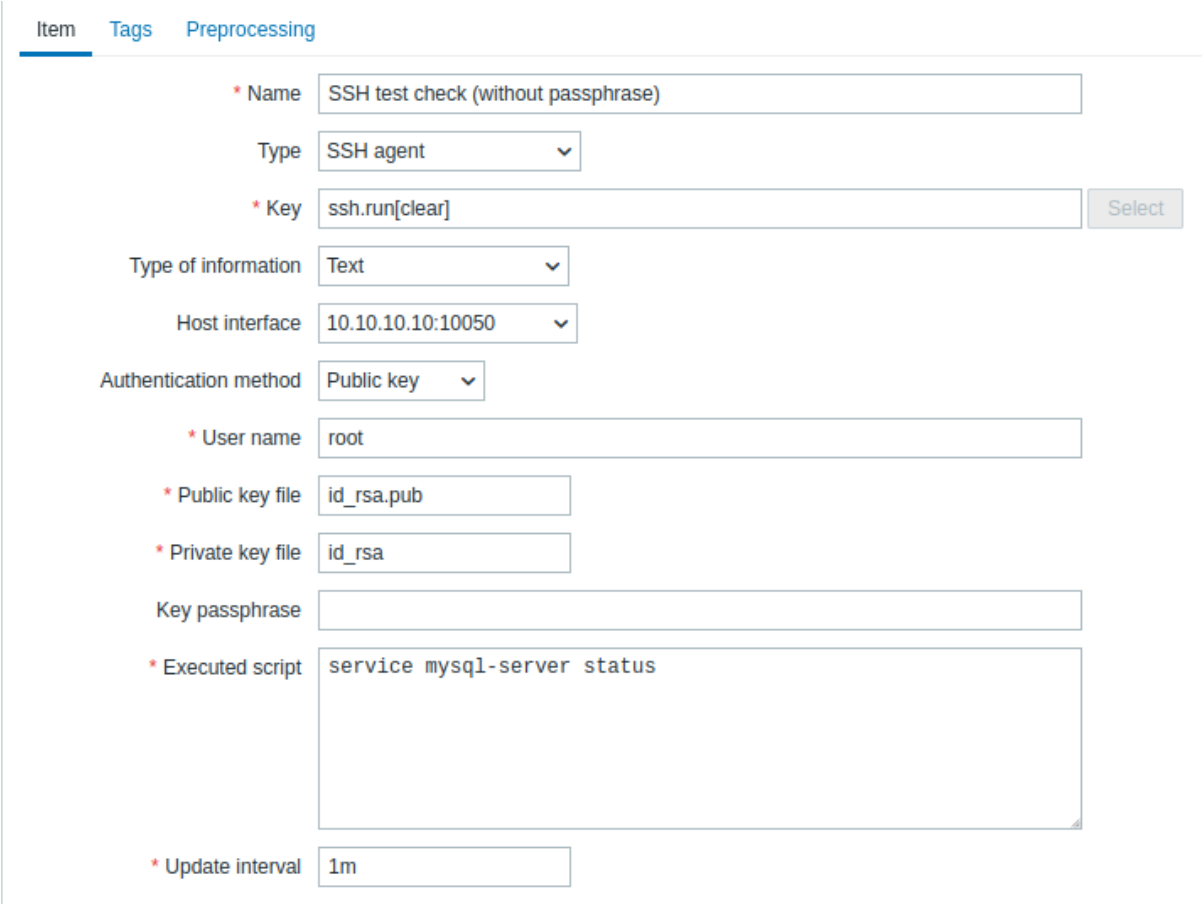
Ora è possibile verificare l'accesso SSH utilizzando la chiave privata predefinita (*/home/zabbix/.ssh/id_rsa*) per l'account utente *zabbix*:

```
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 degli item

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 i valori restituiti saranno formattati su più righe.



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

I campi che richiedono informazioni specifiche per gli item SSH sono:

Parameter	Description	Comments
Type	Selezionare qui **SSH agent**.	

Parameter	Description	Comments

Key	Chiave item univoca (per host) nel formato **ssh.run[breve descrizione univoca,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]**	**breve descrizione univoca** è obbligatoria e deve essere univoca per ciascun item SSH per host.
		
		La porta predefinita è 22, non la porta specificata nell'interfaccia a cui questo item è assegnato.
		
		ssh options consentono 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 quoted); è possibile passare più chiavi di opzione separate da punto e virgola.
		
		Sono supportate le seguenti chiavi di opzione: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Il supporto per chiave e valore dell'opzione dipende dalla libreria SSH (ad esempio, `PubkeyAcceptedKeyTypes` è supportato solo con libssh); se un'opzione non è supportata, verrà restituito un errore e l'item diventerà non supportato.
		
		Si noti che il segno "+" per aggiungere impostazioni dei cifrari e "!" per disabilitare impostazioni specifiche dei cifrari (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). Si noti che l'uso di un sottosistema può richiedere anche l'uso di una sintassi di script specifica nel parametro *Executed script*.
		
		Esempi:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	Uno tra "Password" o "Public key".	
User name	Nome utente (fino a 255 caratteri) per l'autenticazione sull'host remoto. Obbligatorio.	
Public key file	Nome del file della chiave pubblica se *Authentication method* è "Public key". Obbligatorio.	Esempio: *id_rsa.pub* - nome file predefinito della chiave pubblica generata dal comando ssh-keygen.
Private key file	Nome del file della chiave privata se *Authentication method* è "Public key". Obbligatorio.	Esempio: *id_rsa* - nome file predefinito della chiave privata.

Parameter	Description	Comments
Password or *Key passphrase*	Password (fino a 255 caratteri) per l'autenticazione oppure Passphrase **se** è stata usata per la chiave privata.	Lasciare vuoto il campo *Key passphrase* se la passphrase non è stata usata. Vedere anche i problemi noti relativi all'uso della passphrase.

Executed script	Comando/i shell eseguito/i usando una sessione remota SSH.	Il valore restituito dal/i comando/i shell eseguito/i è limitato a 16MB (inclusi gli spazi finali che vengono troncati); si applicano anche i limiti del database.
		
		Si noti che la libreria libssh2 può troncare gli script eseguibili a ~32kB.
		
		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>`
		`]]>]]>`

10 controlli Telnet

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.

11 Controlli esterni

Panoramica

Il controllo esterno è un controllo eseguito da Zabbix server avviando uno script shell o un file binario. Tuttavia, quando gli host sono monitorati da uno Zabbix proxy, i controlli esterni vengono eseguiti dal proxy.

I controlli esterni non richiedono alcun agent in esecuzione su un host monitorato.

La sintassi della chiave item è:

```
script[<parameter1>,<parameter2>,...]
```

Dove:

ARGUMENT	DEFINITION
script	Nome di uno script shell o di un file 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 nella 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, pertanto eventuali permessi di accesso o variabili d'ambiente dovranno essere gestiti in uno script wrapper, se necessario. Anche i permessi del comando devono consentire a tale utente di eseguirlo. Solo i comandi presenti nella directory specificata sono disponibili per l'esecuzione.

Warning:
Non abusare dei controlli esterni, 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 del host.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

Supponendo che il host sia configurato per utilizzare un 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.

12 oggetti Trapper

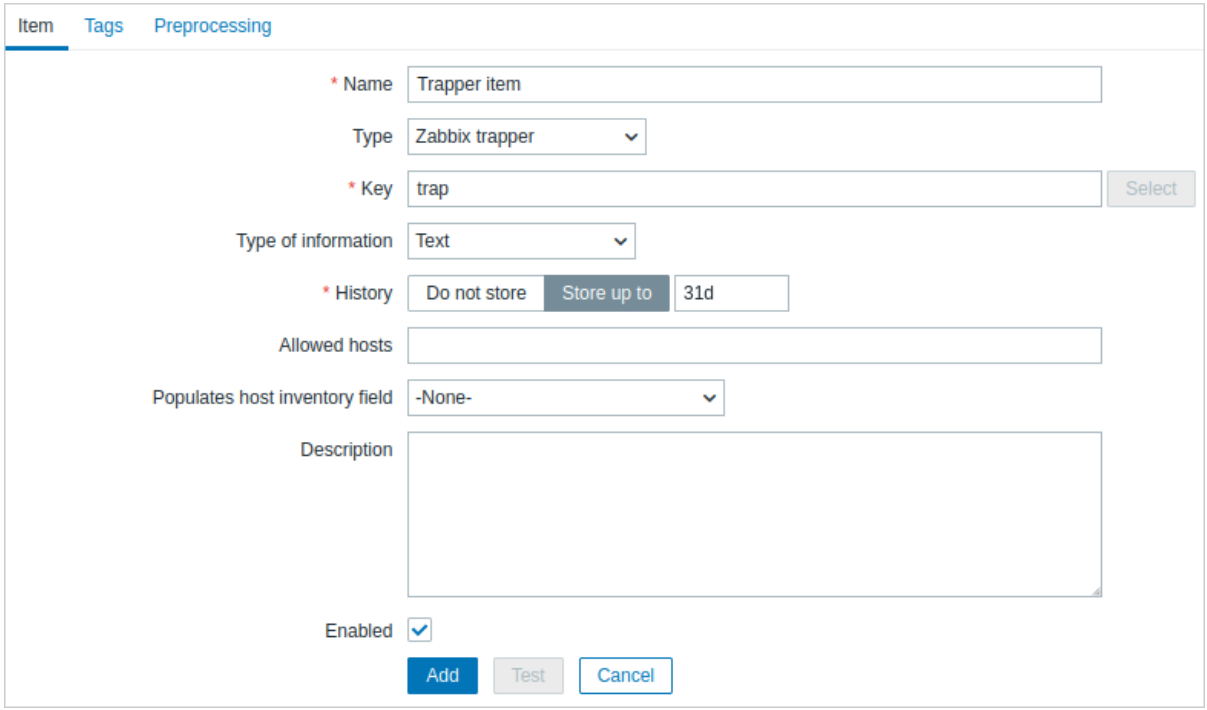
Panoramica

Gli item trapper accettano dati in ingresso invece di interrogarli. Questo è utile per qualsiasi dato che si desidera inviare a Zabbix.

Configurazione

Per configurare un item trapper:

1. Andare in *Raccolta dati* → *Host*.
2. Fare clic su *Item* nella riga dell'host.
3. Fare clic su *Crea item*.
4. Inserire i parametri dell'item nel modulo di configurazione.



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

I campi che richiedono informazioni specifiche per gli item trapper sono:

Type	Selezionare "Zabbix trapper".
Key	Inserire una chiave che verrà utilizzata per riconoscere l'item durante l'invio dei dati a Zabbix server.
Type of information	Selezionare il tipo di informazione 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. 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 utilizzato per consentire qualsiasi indirizzo IPv4. Si noti 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, macro utente e macro host {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 il salvataggio dell'item affinché Zabbix server recepisca le modifiche da un aggiornamento della cache di configurazione.

Invio dei dati

L'invio di dati al server o al proxy Zabbix è possibile utilizzando l'utilità Zabbix sender oppure il protocol di Zabbix sender. L'invio di dati al server Zabbix è possibile anche utilizzando il metodo API `history.push`.

Zabbix sender

Per inviare dati a Zabbix server o proxy utilizzando l'utilità Zabbix sender, è possibile eseguire il seguente comando per inviare il "valore di test":

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Per inviare il "valore di test", vengono utilizzate 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 (assicurarsi di utilizzare il nome host tecnico invece di quello 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 utilizzate nella chiave dell'item per verificare l'esistenza della chiave dell'item corrispondente per l'host di destinazione.

Per ulteriori informazioni sulla comunicazione tra Zabbix sender e Zabbix server o proxy, vedere *protocollo Zabbix sender*.

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:

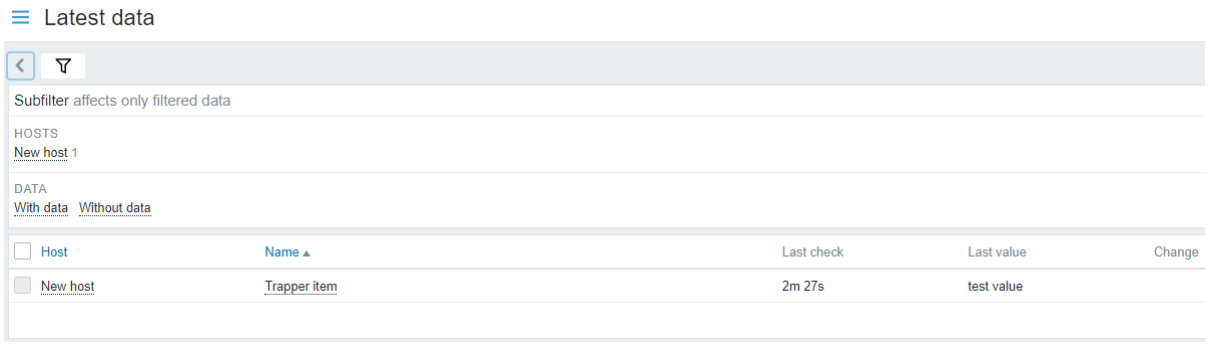
- l'utente che invia i dati non dispone del permesso di *lettura* per l'host dell'item;
- l'host è disabilitato o in manutenzione senza raccolta dati;
- l'item non esiste o non è ancora incluso nella cache di configurazione del server;
- l'item è disabilitato oppure il suo tipo è diverso da Zabbix trapper 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 in *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.

13 Monitoraggio 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.

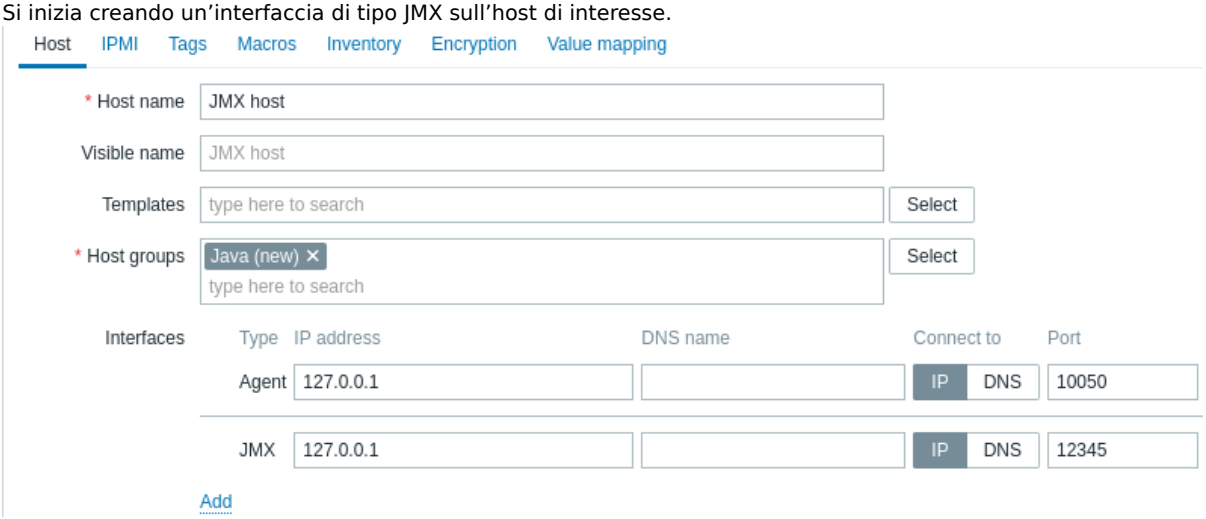
Per una descrizione dettagliata, vedi Monitoring and Management Using JMX.

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 sull'host di interesse.

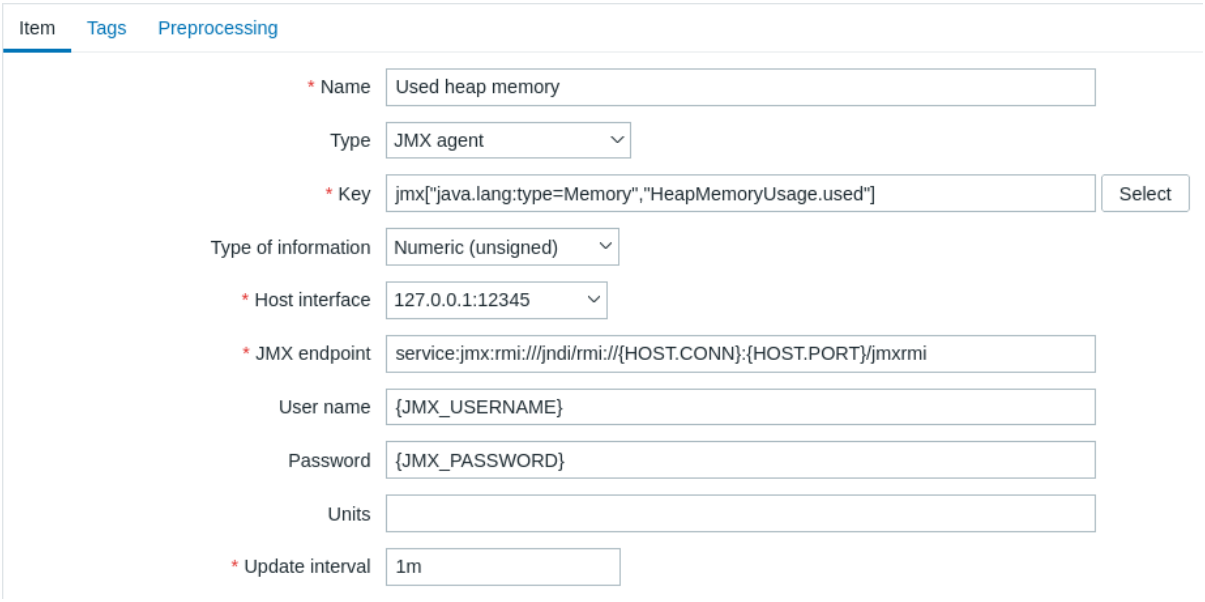


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Aggiunta di un item agent JMX

Per ogni contatore JMX di interesse, aggiungere un item **JMX agent** associato a tale interfaccia.

La chiave nella schermata seguente è `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



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

I campi che richiedono informazioni specifiche per gli item JMX sono:

Type	Impostare qui **JMX agent**.
Key	La chiave item `jmx[]` contiene tre parametri: **object name** - il nome oggetto di un MBean **attribute name** - il nome di un attributo MBean con nomi di campi dati compositi opzionali separati da punti **unique short description** - una descrizione univoca che consente più item JMX con lo stesso nome oggetto e nome attributo sullo host (opzionale) Vedere sotto per maggiori dettagli sulle chiavi item JMX. È possibile individuare MBean e attributi MBean utilizzando 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 utilizzando le macro {HOST.*} come fatto nell'endpoint JMX predefinito. Sono supportate le macro {HOST.*} e le macro utente.
User name	Specificare il nome utente (fino a 255 caratteri), se è stata configurata l'autenticazione nell'applicazione Java. Sono supportate le macro utente.
Password	Specificare la password (fino a 255 caratteri), se è stata configurata l'autenticazione nell'applicazione Java. Sono supportate le macro utente.

Se si desidera monitorare un contatore Boolean che può essere "true" oppure "false", specificare il tipo di informazione come "Numeric (unsigned)" e selezionare il passaggio di preprocessing "Boolean to decimal" nella scheda Preprocessing. Il server memorizzerà i valori Boolean rispettivamente come 1 o 0.

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

Il 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 ciò che c'è da sapere. 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 un endpoint personalizzato con JBoss EAP 6.4

Gli endpoint personalizzati consentono di lavorare con protocolli di trasporto diversi da 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 in conformità con la documentazione.
- Zabbix server 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 alcuni semplici parametri in zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

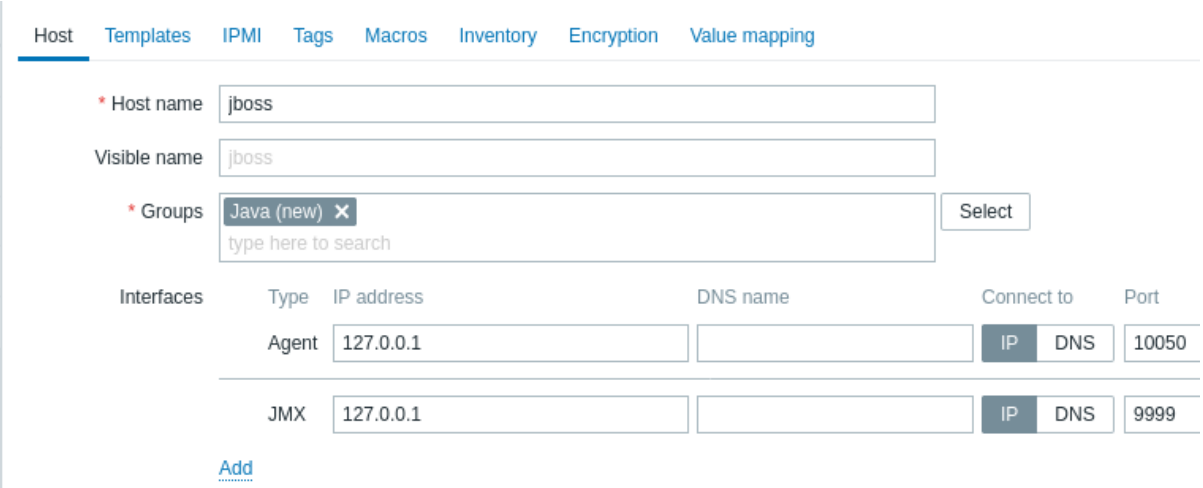
E nel file di configurazione `zabbix_java/settings.sh` (oppure `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Verifica che JBoss sia in ascolto sulla sua porta di gestione standard:

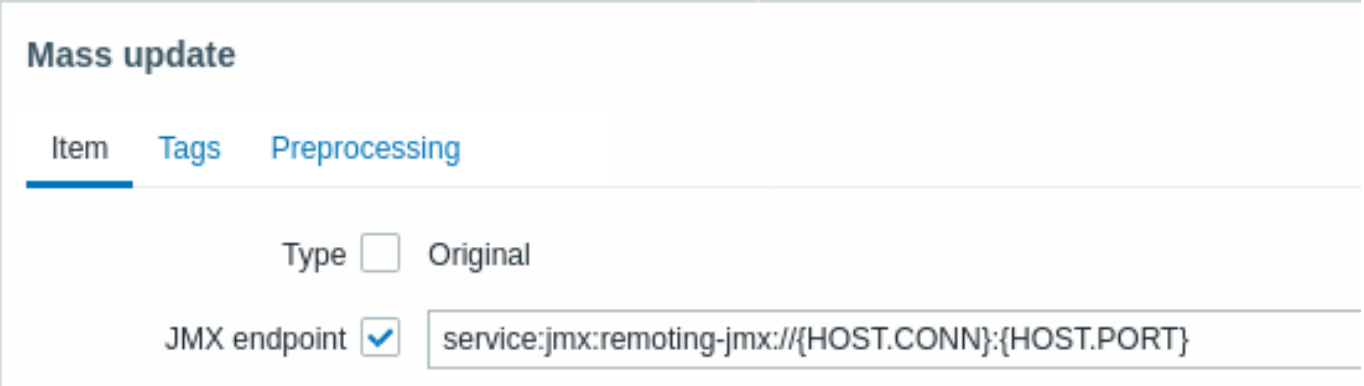
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

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 invece di RMI, possiamo aggiornare in massa il parametro endpoint JMX per gli item nel nostro template JMX di conseguenza:

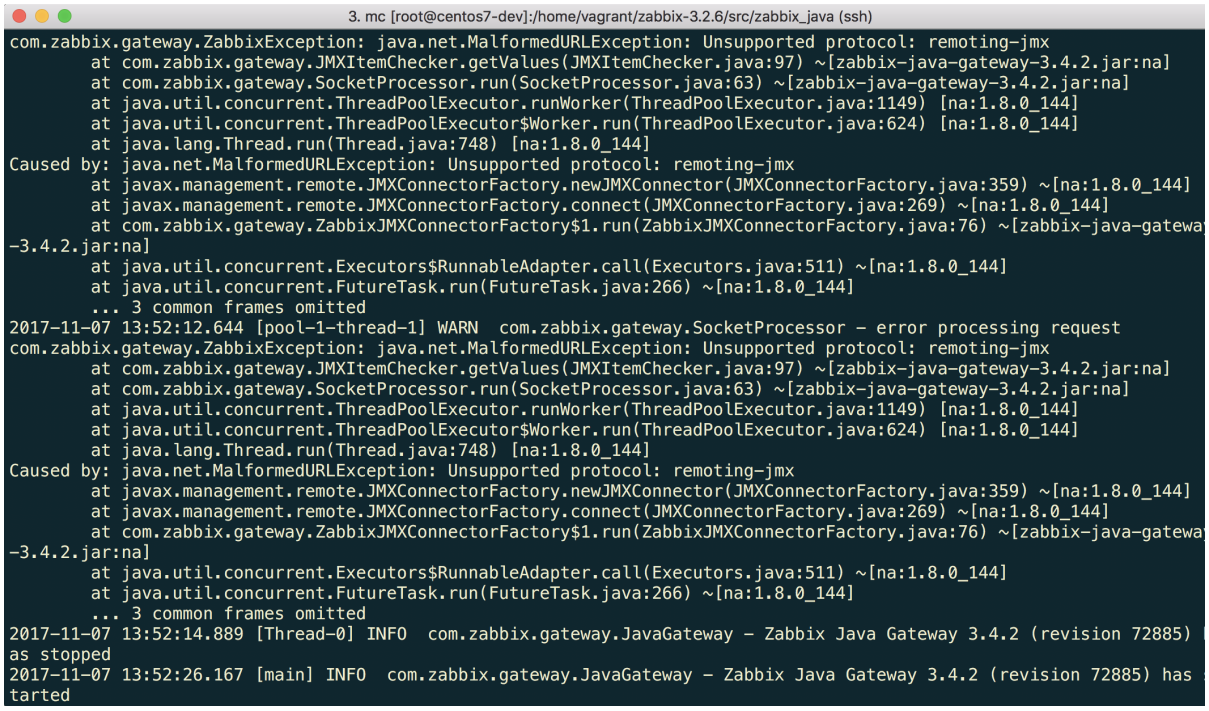
```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Aggiorniamo la cache di configurazione:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Tieni presente che inizialmente potresti riscontrare 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, vedere che i dati di monitoraggio JMX iniziano ad arrivare in Zabbix (vedi anche: Ultimi dati).

14 Monitoraggio ODBC

Panoramica

Il monitoraggio ODBC corrisponde al tipo di item **Database monitor** nel frontend di Zabbix.

ODBC è un’API middleware del linguaggio di programmazione C per l’accesso ai sistemi di gestione di database (DBMS). Il concetto di 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. Ciò 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 nel repository dei pacchetti per impostazione predefinita. 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, utilizzare il gestore di pacchetti del sistema scelto:

```
##### 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 con 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`).

Parameter	Description
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

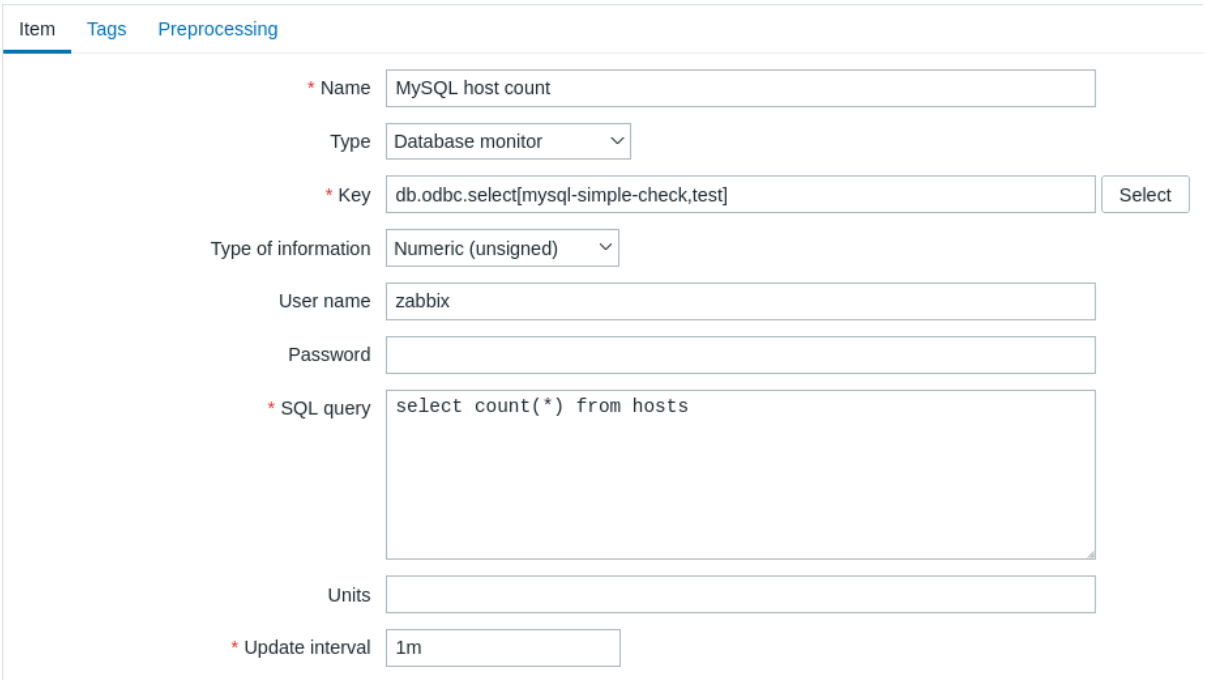
Per verificare se la connessione ODBC funziona correttamente, puoi utilizzare l'utilità `isql` (inclusa nel pacchetto `unixODBC`):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
```

```
| quit                                   |
|                                        |
+----------------------------------------+
```

Configurazione degli 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, è necessario specificare:

Type	Selezionare qui "Database monitor".
Key	Inserire una delle chiavi item supportate: **db.odbc.select[]** - questo item restituisce un 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	Inserire 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 utilizzata una stringa di connessione e il campo *User name* non è vuoto, allora viene aggiunto alla stringa di connessione come `UID=<user>`.
Password	Inserire 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 utilizzata una stringa di connessione e il campo *Password* non è vuoto, allora viene aggiunto alla stringa di connessione come `PWD=<password>`. I caratteri speciali sono supportati in questo campo. La password verrà aggiunta alla stringa di connessione dopo il nome utente, ad esempio come `UID=<username>;PWD=P?;)*word`. Per testare la stringa risultante, è possibile eseguire il seguente comando: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Inserire la query SQL. Si noti che con `db.odbc.select[]` la query deve restituire un solo valore.
Type of information	Selezionare 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 viene avviato alcun processo *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 utilizzato 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 set di risultati 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 memorizzata 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[<breve descrizione univoca>,<dsn>,<stringa di connessione>]

 Trasforma il risultato della query SQL in un array JSON.
 Valore restituito: *oggetto JSON*.

Parametri:

- **breve descrizione univoca** - una breve descrizione univoca per identificare l'item (da usare 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.
- Possono essere restituite più righe/colonne in formato JSON. Questo item può essere usato come item master che raccoglie tutti i dati in una singola chiamata di sistema, mentre il preprocessing JSONPath può essere usato negli item dipendenti per estrarre i singoli valori. Per ulteriori informazioni, vedere un esempio del formato restituito, usato nel low-level discovery.

Esempio:

```
# Connessione per il driver MySQL ODBC 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.

15 Elementi dipendenti

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.

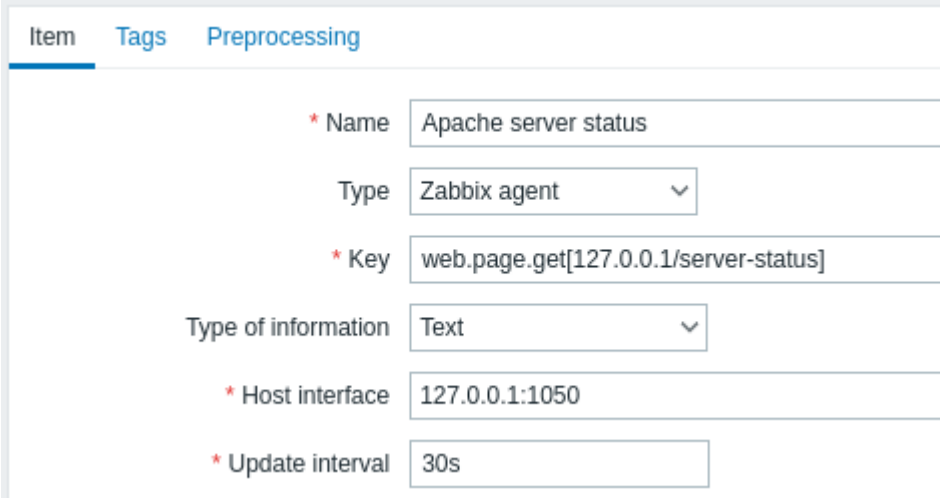
Limitazioni

- 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 degli item

Un item dipendente dipende dal suo item master per i dati. Per questo motivo, l'**item master** deve essere configurato (o esistere) per primo:

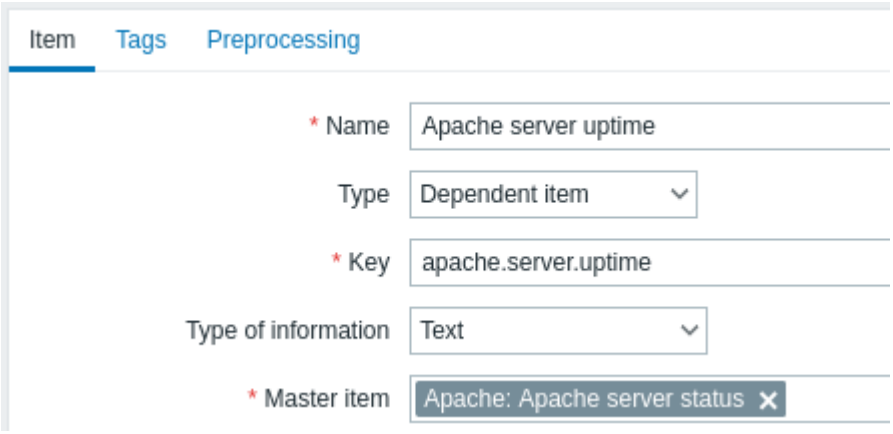
- Vai a: *Raccolta dati* → *Host*
- Fai clic su *Item* nella riga dell'host
- Fai clic su *Crea item*
- Inserisci i parametri dell'item nel modulo



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

Fai clic su *Aggiungi* per salvare l'item master.

Quindi puoi configurare un **item dipendente**.

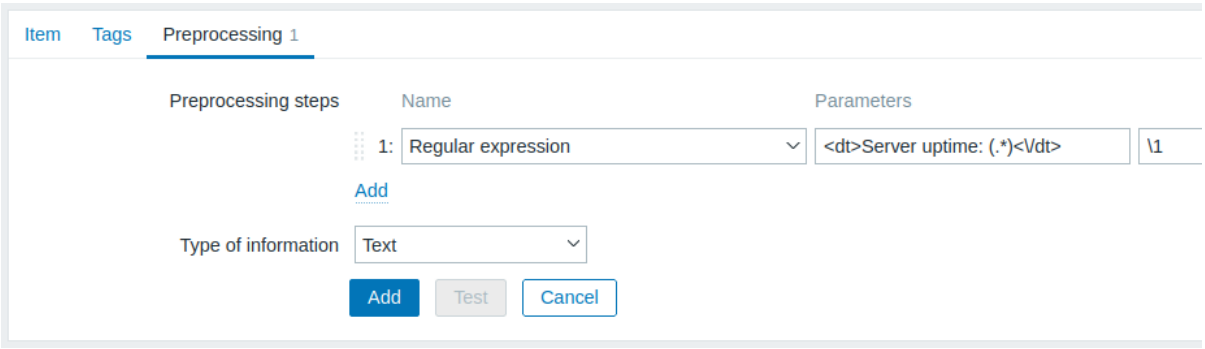


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

I campi che richiedono informazioni specifiche per gli item dipendenti sono:

Tipo	Seleziona qui **Item dipendente**.
Chiave	Inserisci una chiave che verrà utilizzata per riconoscere l'item.
Item master	Seleziona l'item master. Il valore dell'item master verrà utilizzato per popolare il valore dell'item dipendente.
Tipo di informazione	Seleziona il tipo di informazione che corrisponderà al formato dei dati che verranno memorizzati.

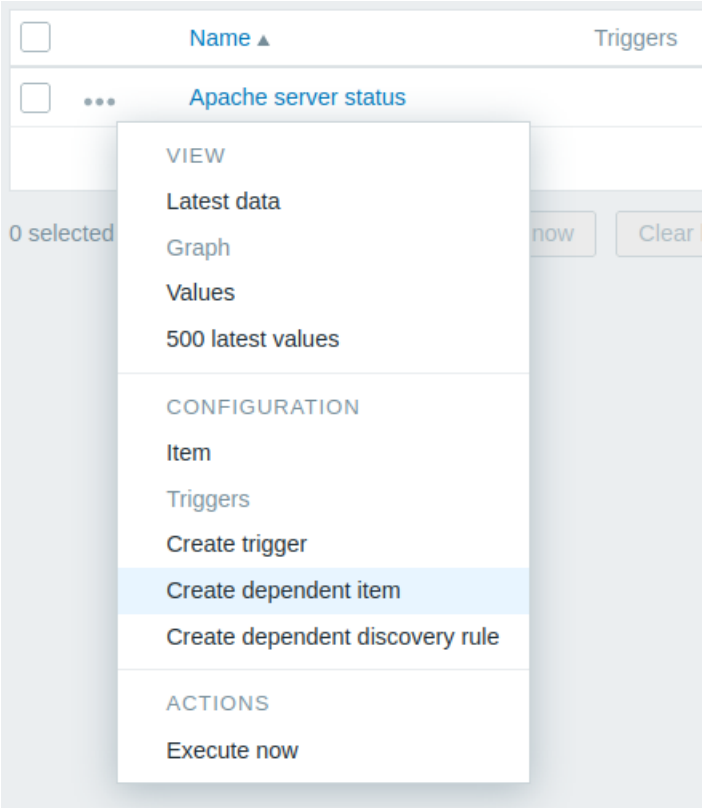
Puoi utilizzare la pre-elaborazione del valore dell'item per estrarre la parte richiesta del valore dell'item master.



Senza pre-elaborazione, il valore dell'item dipendente sarà esattamente lo stesso del valore dell'item master.

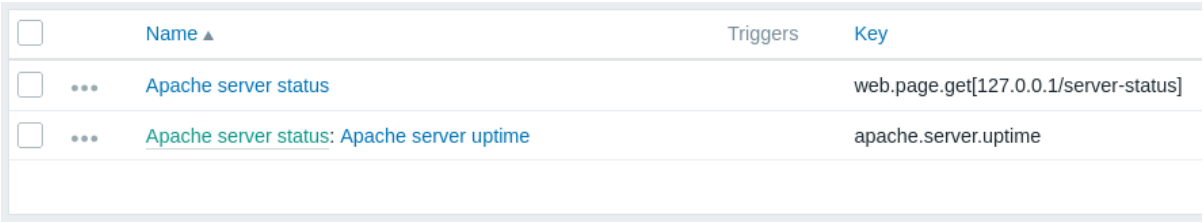
Fai clic su *Aggiungi* per salvare l'item dipendente.

È possibile accedere a una scorciatoia per creare più rapidamente un item dipendente facendo clic sul pulsante ••• nell'elenco degli item e selezionando *Crea item dipendente*.



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.

16 Agente HTTP

Panoramica

Questo tipo di item consente il polling dei dati tramite il protocollo HTTP/HTTPS. È inoltre possibile il trapping utilizzando l'utility Zabbix sender o il protocol di Zabbix sender (per inviare dati a Zabbix server o proxy), oppure utilizzando 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 su un host monitorato.

HTTP agent supporta sia HTTP che HTTPS. Zabbix può facoltativamente seguire i reindirizzamenti (vedere l'opzione *Follow redirects* di seguito). Il numero massimo di reindirizzamenti è fissato nel codice a 10 (utilizzando 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 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 asincroni 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:

- Andare in: *Raccolta dati* → *Host*
- Fare clic su *Item* nella riga dell'host
- Fare clic su *Crea item*
- Inserire i parametri dell'item nel modulo

Item | Tags | Preprocessing

* Name: HTTP agent item

Type: HTTP agent

* Key: http_value_search [Select]

Type of information: Numeric (unsigned)

* URL: http://localhost:9200/str/values/_search [Parse]

Query fields:

Name		Value	
scroll	⇒	10s	Remove

Add

Request type: POST

Request body type: Raw data | JSON data | XML data

Request body:

```
{
  "query":{
    "bool":{
     "must":{
      "match":{
        "itemid":28275
}
```

Headers:

Name		Value	
name	⇒	value	Remove

Add

Required status codes: 200

Follow redirects: ☑

Retrieve mode: Body | Headers | Body and headers

Convert to JSON: ☐

HTTP proxy: [protocol://][user[:password]@]proxy.example.com[:port]

HTTP authentication: None

SSL verify peer: ☐

SSL verify host: ☐

SSL certificate file:

SSL key file:

SSL key password:

Host interface: 127.0.0.1:10050

Units:

* Update interval: 1m

Custom intervals:

Type	Interval	Period	Action
Flexible | Scheduling	50s	1-7,00:00-24:00	Remove

Add

* Timeout: Global | Override 3 Timeouts

* History: Do not store | Store up to 31d

* Trends: Do not store | Store up to 365d

Value mapping: type here to search [Select]

Enable trapping: ☐

Populates host inventory field: -None-

Description:

Enabled: ☑

Add | Test | Cancel

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

I campi che richiedono informazioni specifiche per gli item HTTP sono:

Parametro	Descrizione
Type	Selezionare qui **HTTP agent**.
Key	Inserire una chiave item univoca.
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 con caratteri Unicode. Verranno convertiti automaticamente in punycode ASCII durante l'esecuzione del controllo HTTP. Il pulsante *Parse* può essere utilizzato per separare i campi di query opzionali (come ?name=Admin&password=mypassword) dall'URL, spostando attributi e valori in *Query fields* per la codifica URL automatica. 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_URL.
Query fields	Variabili per l'URL (vedere sopra). Specificate come coppie attributo-valore. I valori vengono codificati automaticamente come URL. I valori provenienti dalle macro vengono risolti e quindi codificati automaticamente come 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_URL.
Request type	Selezionare il tipo di metodo della richiesta: *GET*, *POST*, *PUT* o *HEAD*
Request body type	Selezionare il tipo di corpo della richiesta: **Raw data** - corpo della richiesta HTTP personalizzato; le macro vengono sostituite ma non viene eseguita alcuna codifica **JSON data** - corpo della richiesta HTTP in formato JSON. Le macro possono essere usate come stringa, numero, true e false; le macro usate come stringhe devono essere racchiuse tra virgolette doppie. I valori provenienti dalle macro vengono risolti e quindi sottoposti automaticamente a escape. Se "Content-Type" non è specificato negli header, verrà usato per impostazione predefinita "Content-Type: application/json" **XML data** - corpo della richiesta HTTP in formato XML. Le macro possono essere usate come nodo di testo, attributo o sezione CDATA. I valori provenienti dalle macro vengono risolti e quindi sottoposti automaticamente a escape in un nodo di testo e in un attributo. Se "Content-Type" non è specificato negli header, verrà usato per impostazione predefinita "Content-Type: application/xml"
Request body	Inserire 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 low-level discovery.
Headers	Header HTTP personalizzati che verranno inviati 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_HTTPHEADER.
Required status codes	Elenco dei codici di stato HTTP previsti. Se Zabbix riceve un codice non presente nell'elenco, l'item diventerà unsupported. Se vuoto, non viene eseguito alcun controllo. Ad esempio: 200,201,210-299 Macro supportate nell'elenco: macro utente, macro di low-level discovery. Questo usa l'opzione cURL CURLINFO_RESPONSE_CODE.
Follow redirects	Selezionare la casella per seguire i reindirizzamenti HTTP. Questo imposta l'opzione cURL CURLOPT_FOLLOWLOCATION.
Retrieve mode	Selezionare la parte della risposta che deve essere recuperata: **Body** - solo corpo **Headers** - solo header **Body and headers** - corpo e header

Parametro	Descrizione
Convert to JSON	Gli header vengono salvati come coppie attributo-valore sotto la chiave "header" se *Retrieve mode* non è impostato su *Body*. Se viene rilevato 'Content-Type: application/json', il corpo viene salvato come oggetto; altrimenti viene memorizzato come stringa, ad esempio: `{ "header": { "<key>": "<value>", "<key2>": "<value>" }, "body": <body> }`
HTTP proxy	È possibile specificare un proxy HTTP da usare, nel formato `[protocol://][username[:password]@]proxy.example.com[:port]`. Il prefisso facoltativo `protocol://` può essere usato per specificare protocolli proxy alternativi (ad es. https, socks4, socks5; vedere la documentazione; il supporto del 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 si specifica il protocollo errato, la connessione fallirà e l'item diventerà unsupported. Per impostazione predefinita verrà usata la porta 1080. Se specificato, il proxy sovrascriverà le variabili di ambiente relative al proxy come http_proxy, HTTPS_PROXY. Se non specificato, il proxy non sovrascriverà le variabili di ambiente relative al proxy. Il valore inserito viene passato "così com'è", senza alcun controllo di validità. *Nota*: 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_PROXY.
HTTP authentication	Selezionare l'opzione di autenticazione: **None** - non viene usata alcuna autenticazione; **Basic** - viene usata l'autenticazione di base; **NTLM** - viene usata l'autenticazione NTLM (Windows NT LAN Manager); **Kerberos** - viene usata l'autenticazione Kerberos (vedere anche: Configurazione di Kerberos con Zabbix); **Digest** - viene usata l'autenticazione Digest. Questo imposta l'opzione cURL CURLOPT_HTTPAUTH.
User name	Inserire il nome utente (fino a 255 caratteri). Questo campo è disponibile se *HTTP authentication* è impostato su Basic, NTLM, Kerberos o Digest. Sono supportate le macro utente e le macro di low-level discovery.
Password	Inserire la password utente (fino a 255 caratteri). Questo campo è disponibile se *HTTP authentication* è impostato su Basic, NTLM, Kerberos o Digest. Sono supportate le macro utente e le macro di low-level discovery.
SSL verify peer	Selezionare la casella per verificare il certificato SSL del server web. Il certificato del server verrà prelevato automaticamente dalla posizione dell'autorità di certificazione (CA) a livello di sistema. È possibile sovrascrivere la posizione dei file CA usando il parametro di configurazione del server o proxy Zabbix SSLCALocation. Questo imposta l'opzione cURL CURLOPT_SSL_VERIFYPEER.
SSL verify host	Selezionare la casella per verificare che il campo Common Name o il campo Subject Alternate Name del certificato del server web corrispondano. Questo imposta l'opzione cURL CURLOPT_SSL_VERIFYHOST.
SSL certificate file	Nome del file del certificato SSL usato 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 del file della chiave SSL. Se la chiave è cifrata, specificare la password nel campo SSL key password. La directory che contiene questo file è specificata dal parametro di configurazione del server o proxy Zabbix SSLCertLocation. 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_SSLCERT.

Parametro	Descrizione
SSL key file	Nome del file della chiave privata SSL usato per l'autenticazione del client. Il file della chiave privata deve essere in formato PEM[1]. La directory che contiene questo file è specificata dal parametro di configurazione del server o proxy Zabbix SSLKeyLocation. 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 low-level discovery. Questo imposta l'opzione cURL CURLOPT_SSLKEY.
SSL key password	Password del file della chiave privata SSL. Macro supportate: macro utente, macro di low-level discovery. Questo imposta l'opzione cURL CURLOPT_KEYPASSWD.
Timeout	Zabbix non impiegherà più del tempo impostato per elaborare l'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 impiegherà più di 2 x *Timeout* secondi per un controllo. Questo imposta l'opzione cURL CURLOPT_TIMEOUT. Per ulteriori informazioni sul parametro *Timeout*, vedere attributi generali dell'item.
Enable trapping	Con questa casella selezionata, l'item funzionerà anche come item trapper e accetterà i dati inviati al server o proxy Zabbix usando l'utilità Zabbix sender o il protocol di Zabbix sender, oppure accetterà i dati inviati al server Zabbix usando il metodo API `history.push`. Per ulteriori informazioni sull'invio dei dati, vedere: Item trapper.
Allowed hosts	Visibile solo se la casella *Enable trapping* è selezionata. Elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS. 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' 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. Si noti 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. In questo campo sono consentite le macro host: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Se il campo *HTTP proxy* viene lasciato vuoto, un altro modo per usare un proxy HTTP è impostare le variabili di ambiente relative al proxy.
Per HTTP - impostare la variabile di ambiente `http_proxy` per l'utente del server Zabbix. Ad esempio: `http_proxy=http://proxy_ip:proxy_port`.
Per HTTPS - impostare la variabile di ambiente `HTTPS_PROXY`. Ad esempio: `HTTPS_PROXY=http://proxy_ip:proxy_port`. Maggiori dettagli sono disponibili eseguendo il comando shell: *# man curl*.

Attention:
[1] Zabbix supporta solo file di certificato e chiave privata in formato PEM. Nel caso in cui i dati del certificato e della chiave privata siano in un file in formato PKCS #12 (di solito con estensione *.p12 o *.pfx), è possibile generare il file PEM da esso usando 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 utilizzando 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 minuto 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",
```

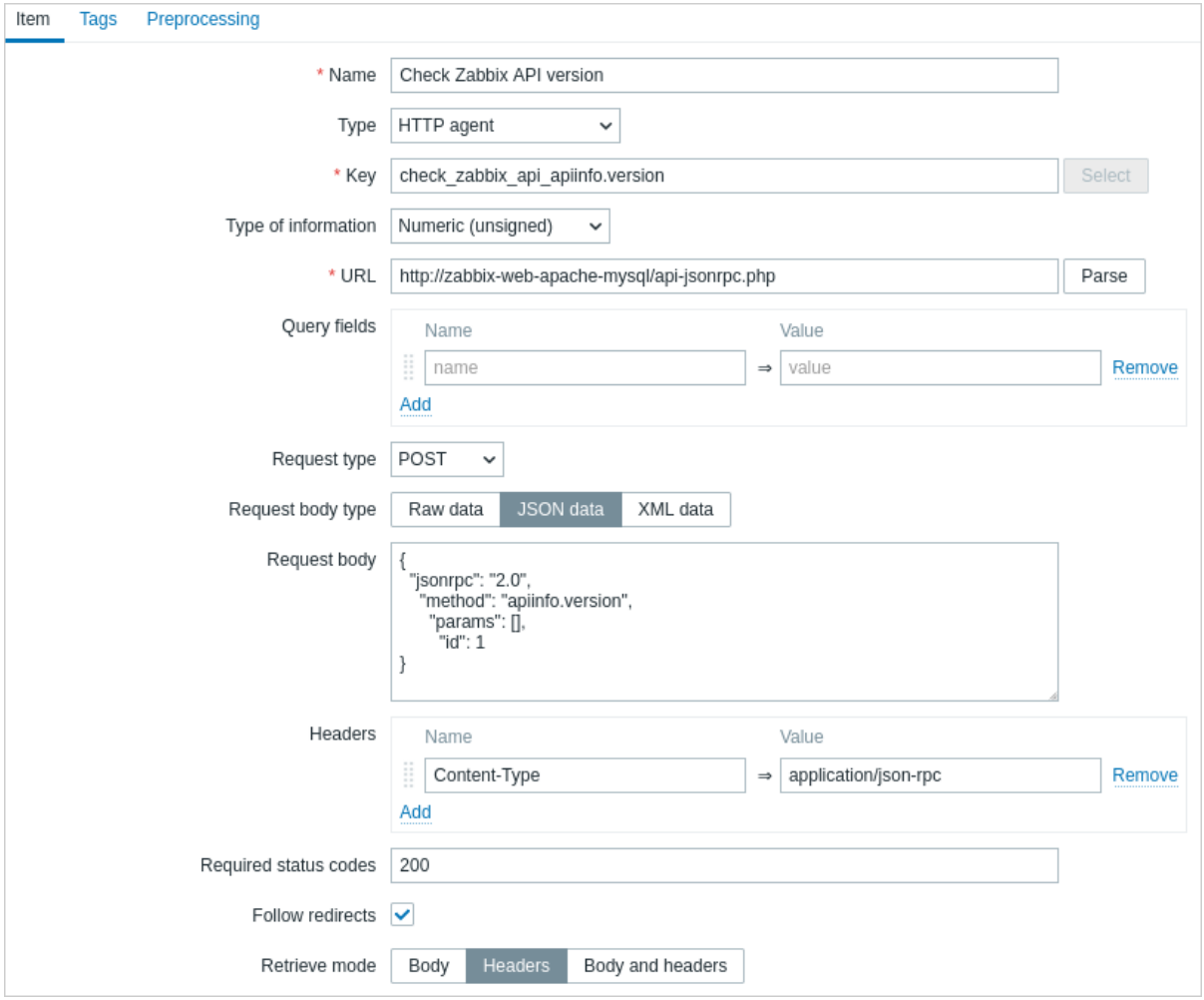
```
                "clock": 1517566136,
                "ns": 25388713,
                "ttl": 604800
            }
        }]
    }
}
```

- Ora utilizza un passaggio di preprocessing JSONPath per ottenere il valore dell'item: `$.hits.hits[0]._source.value`

Esempio 3

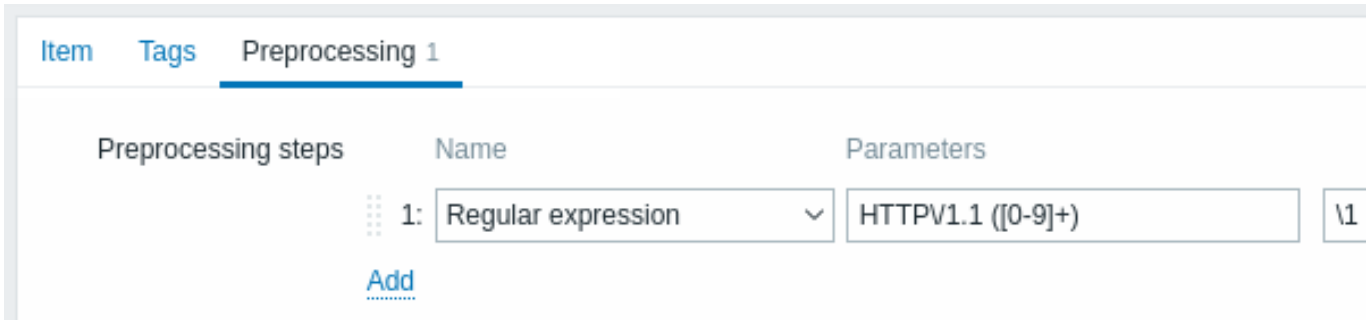
Verifica che la API di Zabbix sia attiva, utilizzando apiinfo.version.

- Configurazione dell'item:

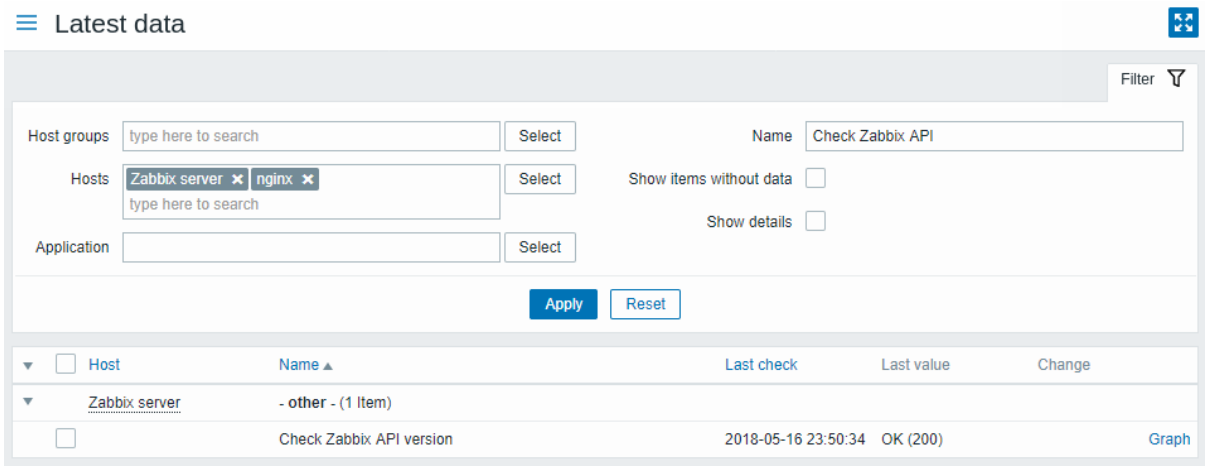


Notare l'uso del metodo POST con dati JSON, l'impostazione delle intestazioni della richiesta e la richiesta di restituire solo le intestazioni:

- Pre-elaborazione del valore dell'item con espressione regolare per ottenere il codice HTTP:



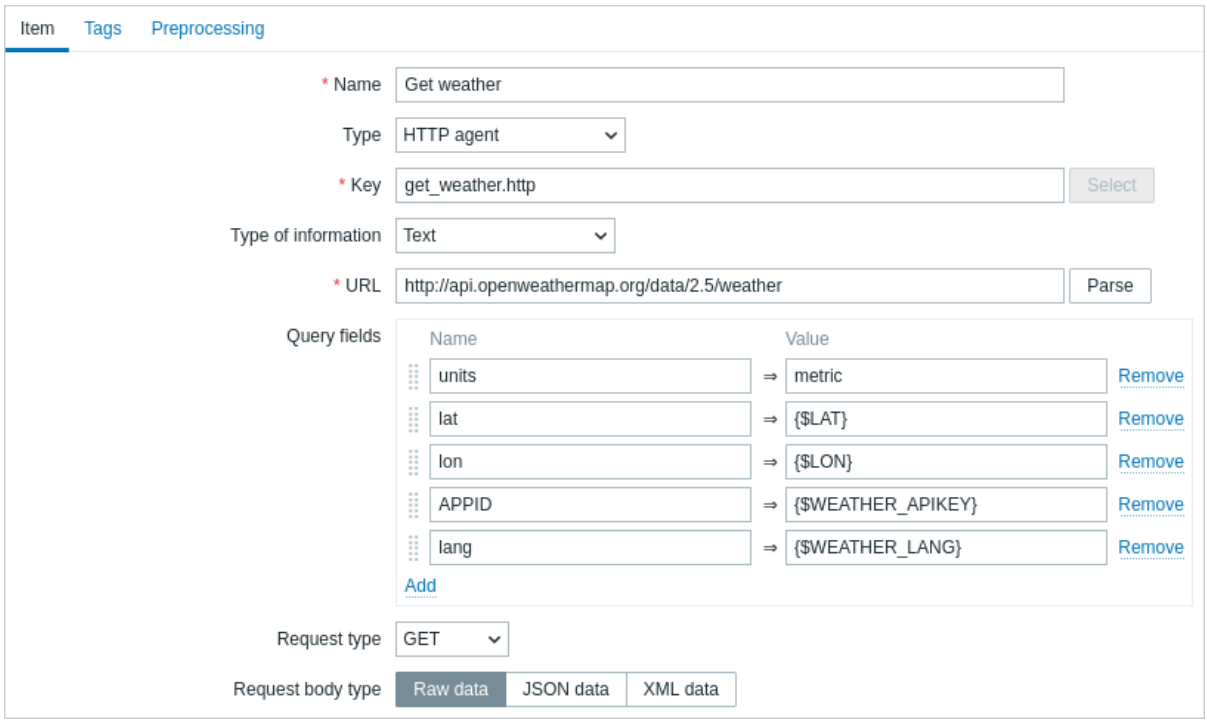
- Verifica del risultato in *Ultimi dati*:



Esempio 4

Recupero delle informazioni meteo collegandosi al servizio pubblico Openweathermap.

- Configurare un item principale per la raccolta massiva dei dati in un unico JSON:



Notare l'uso delle macro nei campi della query. Fare riferimento alla API di Openweathermap per sapere come compilarli.

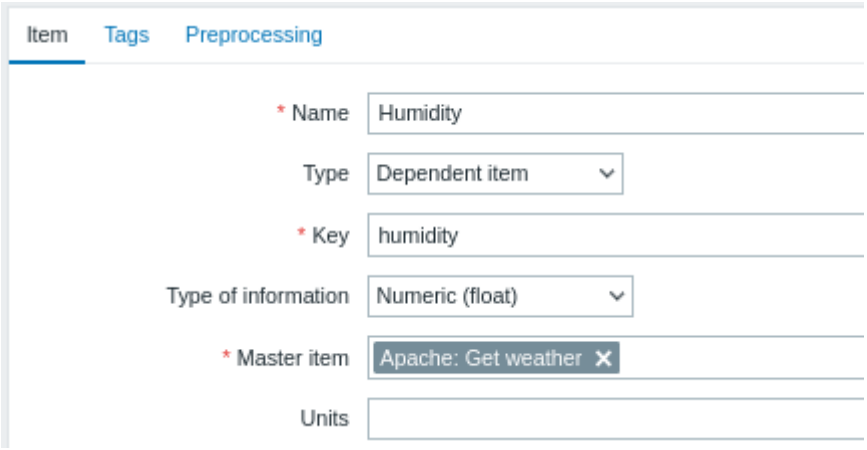
JSON di esempio restituito in risposta da HTTP agent:

```
{
    "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
    }
}
```

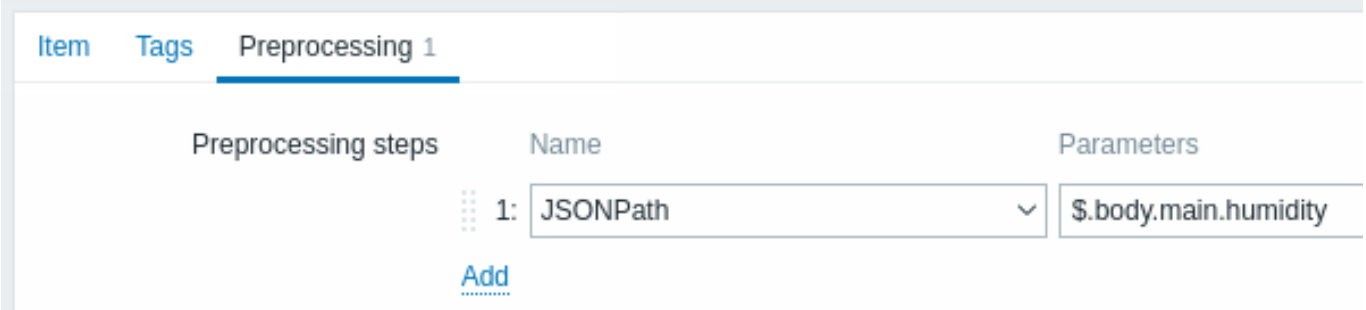

L'attività successiva consiste nel configurare item dipendenti che estraggono i dati dal JSON.

- Configurare un item dipendente di esempio per l'umidità:

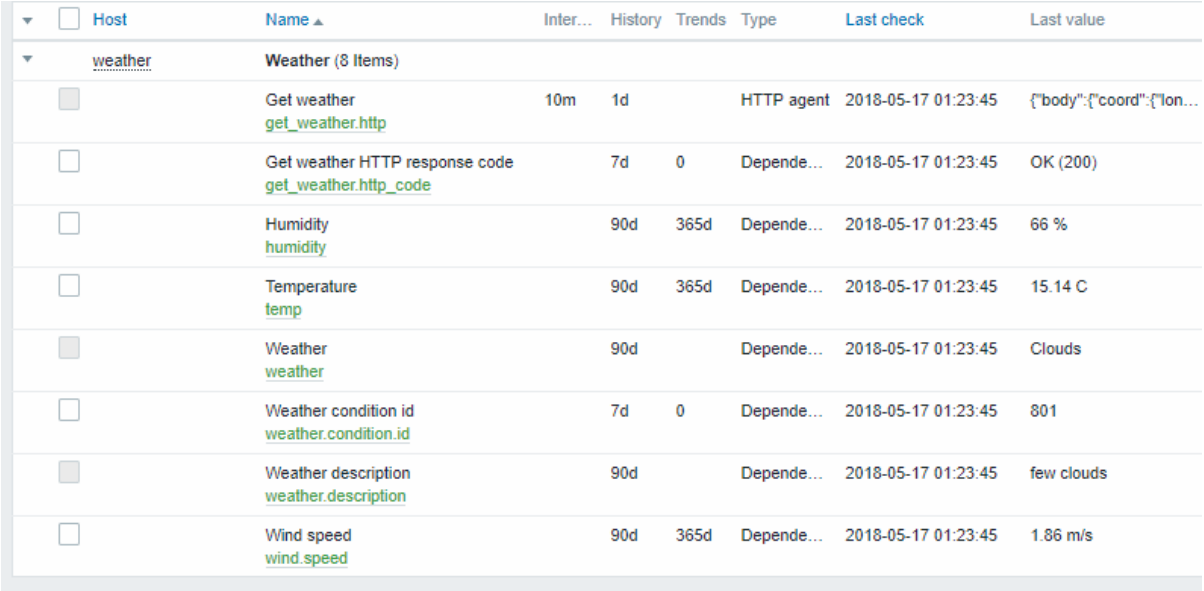


Altre metriche meteo come 'Temperature' vengono aggiunte nello stesso modo.

- Esempio di preprocessing del valore dell'item dipendente con JSONPath:



- Verificare il risultato dei dati meteo in *Ultimi dati*:

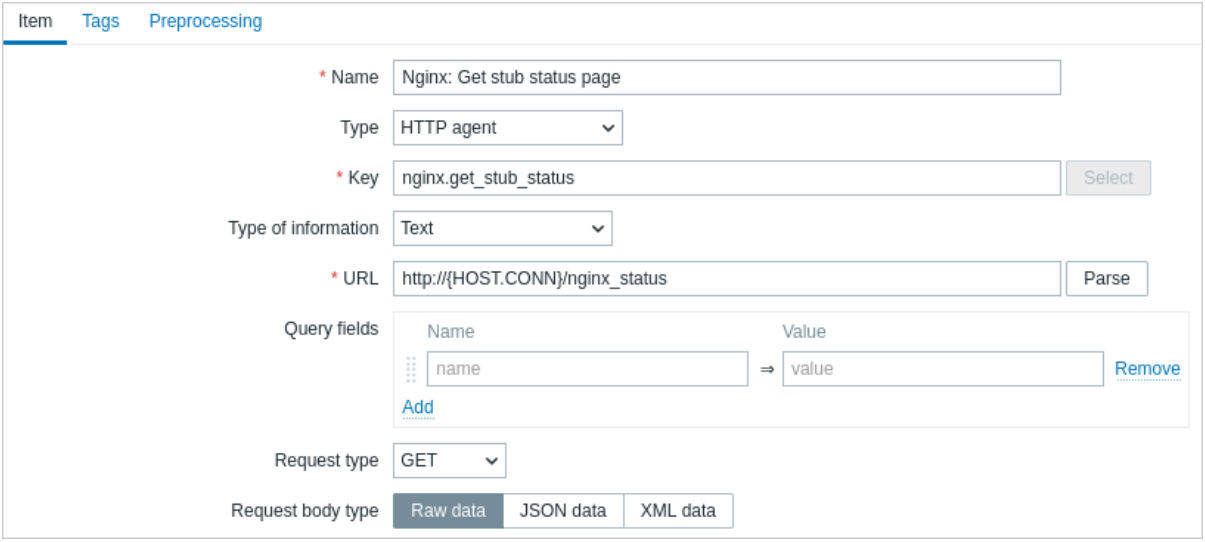


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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 acquisizione in blocco delle sue metriche.

- Configurare Nginx seguendo la guida ufficiale.
- Configurare un item master per la raccolta dei dati in blocco:

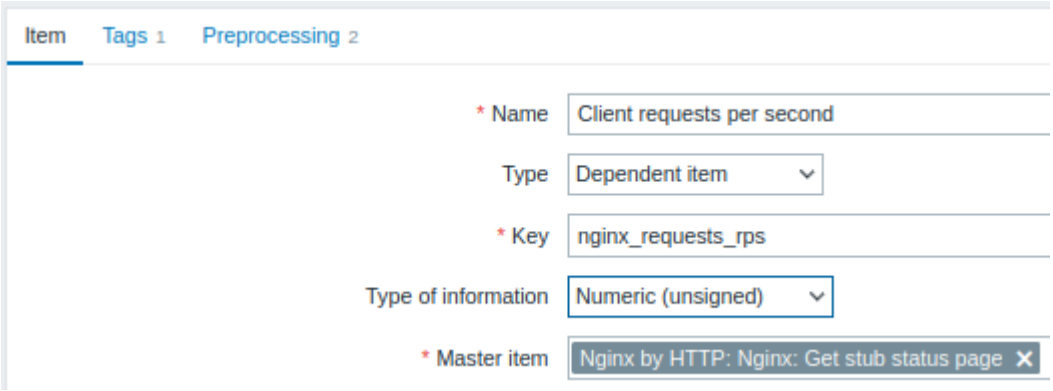


Esempio di output dello stato stub di Nginx:

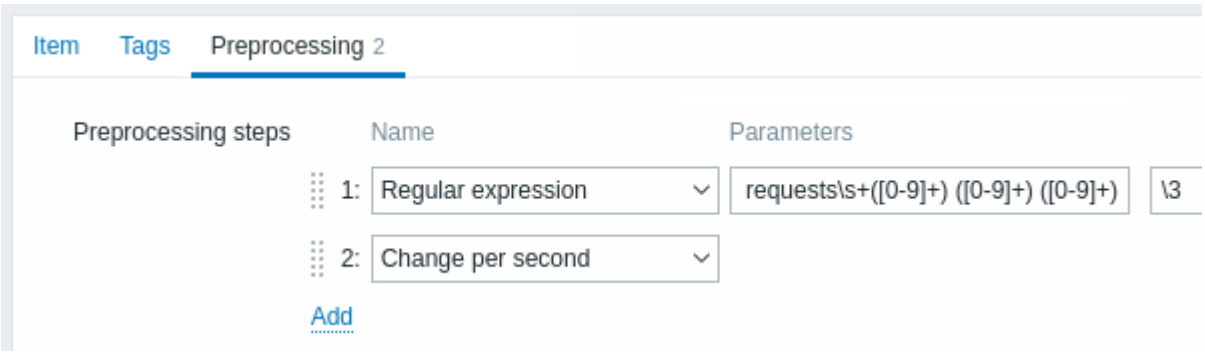
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

L'attività successiva consiste nel configurare item dipendenti che estraggano i dati.

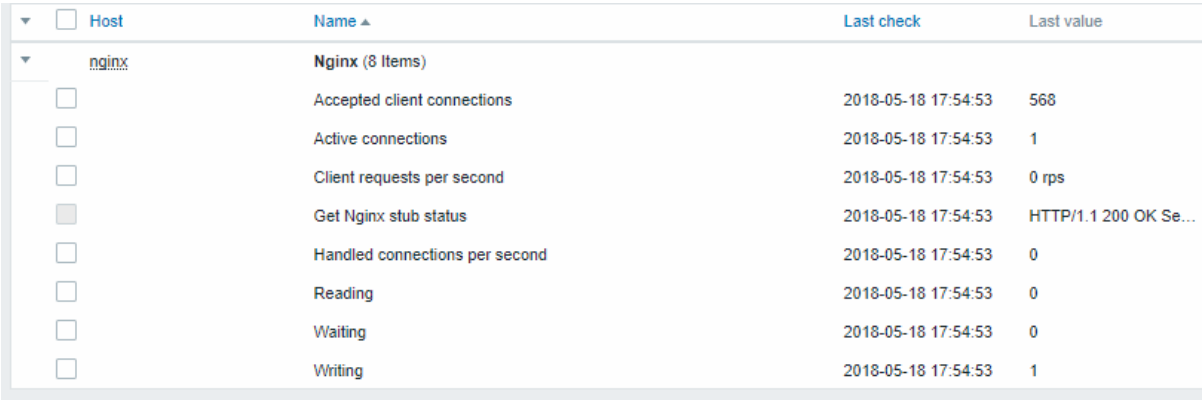
- Configurare un item dipendente di esempio per le richieste al secondo:



- Esempio di preprocessing del valore dell'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 *Ultimi dati*:



Host	Name ▲	Last check	Last value
nginx	**Nginx** (8 Items)		
	Accepted client connections	2018-05-18 17:54:53	568
	Active connections	2018-05-18 17:54:53	1
	Client requests per second	2018-05-18 17:54:53	0 rps
	Get Nginx stub status	2018-05-18 17:54:53	HTTP/1.1 200 OK Se...
	Handled connections per second	2018-05-18 17:54:53	0
	Reading	2018-05-18 17:54:53	0
	Waiting	2018-05-18 17:54:53	0
	Writing	2018-05-18 17:54:53	1

17 Controlli Prometheus

Panoramica

Zabbix può interrogare le metriche esposte nel formato di riga Prometheus.

Per iniziare a raccogliere dati 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 Prometheus per interrogare i dati richiesti dalle metriche raccolte dal master item

Esistono due opzioni di preprocessing dei dati Prometheus:

- *Prometheus pattern* - utilizzato negli item normali per interrogare i dati Prometheus
- *Prometheus to JSON* - utilizzato negli item normali e per il low-level discovery. In questo caso i dati Prometheus interrogati vengono restituiti in formato JSON.

Elaborazione in blocco

L'elaborazione in blocco è supportata per gli item dipendenti. Per abilitare la cache e l'indicizzazione, il preprocessing *Prometheus pattern* deve essere il **primo** passaggio di preprocessing. Quando *Prometheus pattern* è il primo passaggio di preprocessing, i dati Prometheus analizzati vengono memorizzati nella cache e indicizzati in base alla prima condizione `<label>==<value>` nel

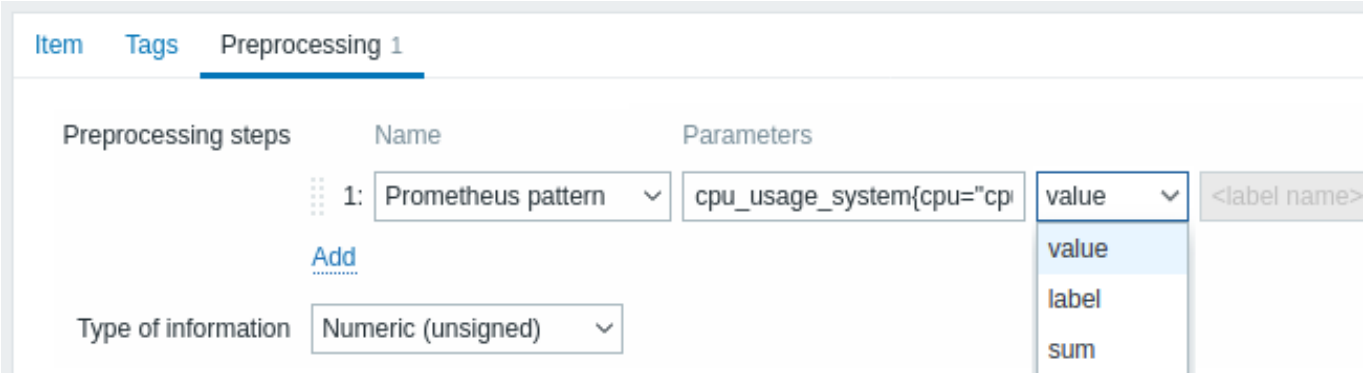
passaggio di preprocessing *Prometheus pattern*. Questa cache viene riutilizzata durante l'elaborazione degli altri item dipendenti in questo batch. Per prestazioni ottimali, la prima label dovrebbe essere quella con il maggior numero di valori diversi.

Se è necessario eseguire altro preprocessing prima del primo passaggio, questo dovrebbe essere spostato nell'item master oppure in un nuovo item dipendente che verrebbe utilizzato come item master per gli item dipendenti.

Configurazione

A condizione che l'HTTP master item sia configurato, è necessario creare un item dipendente che utilizzi un passaggio di preprocessing Prometheus:

- Inserire i parametri generali dell'item dipendente nel modulo di configurazione
- Andare alla scheda Preprocessing
- Selezionare un'opzione di preprocessing Prometheus (*Prometheus pattern* oppure *Prometheus to JSON*)



I seguenti parametri sono specifici dell'opzione di preprocessing *Prometheus pattern*:

Parameter	Description	Examples
Pattern	Per definire il pattern di dati richiesto è possibile utilizzare un linguaggio di query simile al linguaggio di query di Prometheus (vedere la tabella di confronto), ad esempio: <metric name> - seleziona per nome della metrica {__name__="<metric name>"} - seleziona per nome della metrica {__name__=~"<regex>"} - seleziona per nome della metrica corrispondente a un'espressione regolare {<label name>="<label value>",...} - seleziona per nome dell'etichetta {<label name>=~"<regex>",...} - seleziona per nome dell'etichetta corrispondente a un'espressione regolare {__name__=~".*"}==<value> - seleziona per valore della metrica Oppure una combinazione di quanto sopra: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Il valore dell'etichetta può essere qualsiasi sequenza di caratteri UTF-8, ma i caratteri backslash, doppia virgoletta e avanzamento riga devono essere preceduti da escape rispettivamente come \\, \" e \n; gli altri caratteri non devono essere preceduti da escape.	*wmi_os_physical_memory_free_bytes* *cpu_usage_system{cpu="cpu-total"}* *cpu_usage_system{cpu=~".*"}* *cpu_usage_system{cpu="cpu-total",host=~".*"}* *wmi_service_state{name="dhcp"}==1* *wmi_os_timezone{timezone=~".*"}==1*

Parameter	Description	Examples
Result processing	Specificare se restituire il valore, l'etichetta oppure 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 conteggio dei 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 è 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 anche interessare 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, è comune utilizzare una soluzione alternativa che consente di restituire anche la descrizione testuale pertinente. Questo può essere ottenuto con un filtro e specificando l'etichetta. Quindi, 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').

Da Prometheus a JSON

I dati di Prometheus possono essere utilizzati per il low-level discovery. In questo caso sono necessari dati in formato JSON e l'opzione di preprocessing *Prometheus to JSON* restituirà esattamente questo.

Per maggiori dettagli, vedere Discovery using Prometheus data.

Confronto del linguaggio di query

La tabella seguente elenca differenze e somiglianze tra PromQL e il linguaggio di query del preprocessing Prometheus di Zabbix.

Selettore di vettore istantaneo PromQL		Preprocessing Prometheus di Zabbix
Differenze		
Destinazione della query	server Prometheus	Testo semplice nel formato di esposizione Prometheus
Restituisce	Vettore istantaneo	Valore della metrica o dell'etichetta (pattern Prometheus) Array di metriche per valore singolo in JSON (Prometheus in JSON)
Operatori di corrispondenza delle etichette	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Espressione regolare usata nella corrispondenza del nome dell'etichetta o della metrica	RE2	PCRE
Operatori di confronto	Vedi elenco	Per il filtraggio dei valori è supportato solo **==** (uguale)
Somiglianze		
Selezione per nome della metrica uguale a una stringa	<metric name> o {__name__="<metric name>"}	<metric name> o {__name__="<metric name>"}
Selezione per nome della metrica corrispondente a un'espressione regolare	{__name__=~"<regex>"}	{__name__=~"<regex>"}
Selezione per valore di <label name> uguale a una stringa	{<label name>="<label value>",...}	{<label name>="<label value>",...}
Selezione per valore di <label name> corrispondente a un'espressione regolare	{<label name>=~"<regex>",...}	{<label name>=~"<regex>",...}
Selezione per valore uguale a una stringa	{__name__=~".*"} == <value>	{__name__=~".*"} == <value>

18 Elementi di script

Panoramica

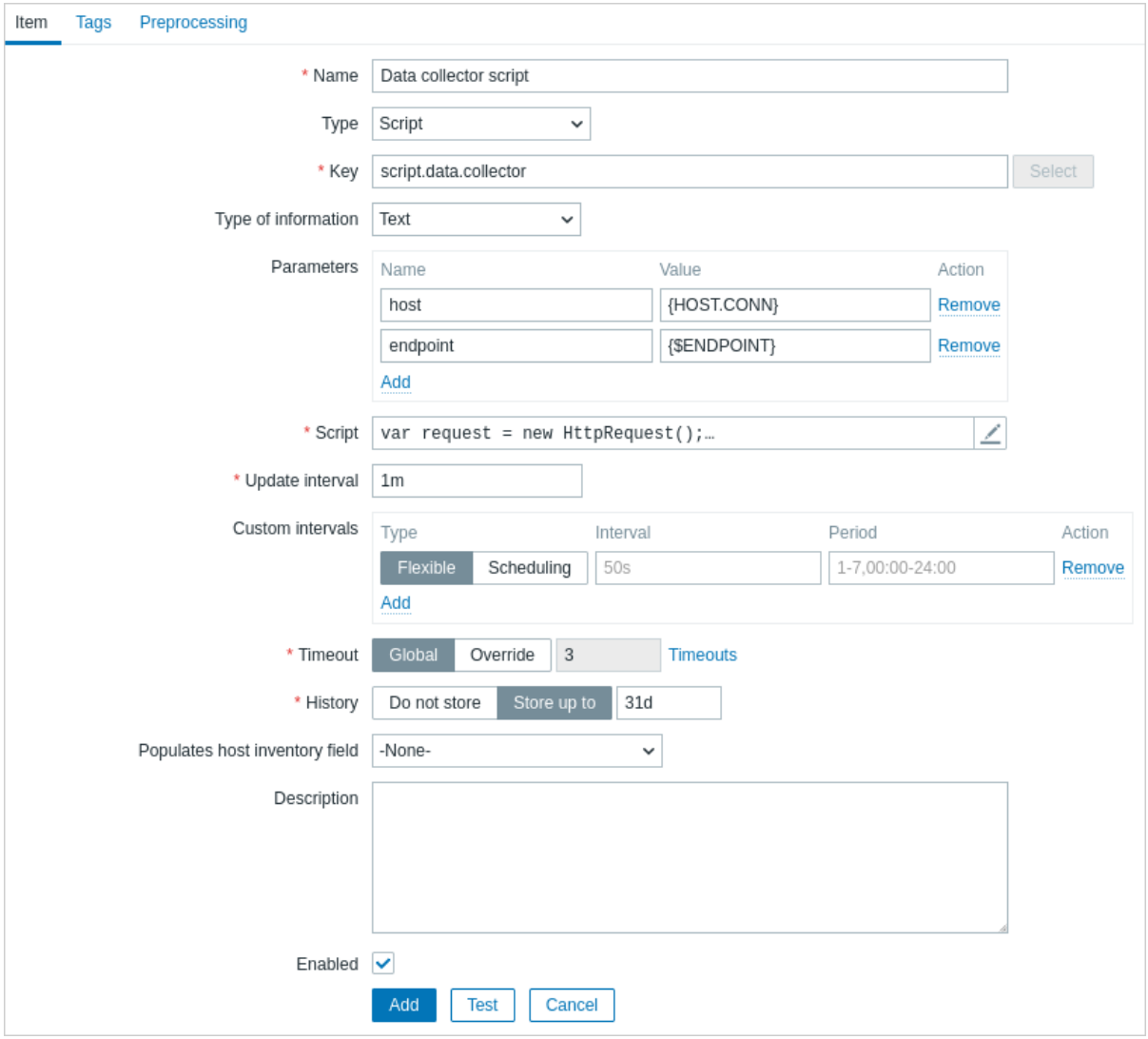
Gli item script possono essere utilizzati per raccogliere dati eseguendo codice JavaScript definito dall'utente, con la possibilità di recuperare dati tramite HTTP/HTTPS. Oltre allo script, è possibile specificare un elenco facoltativo di parametri (coppie di nome e valore) e un timeout.

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.
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*:

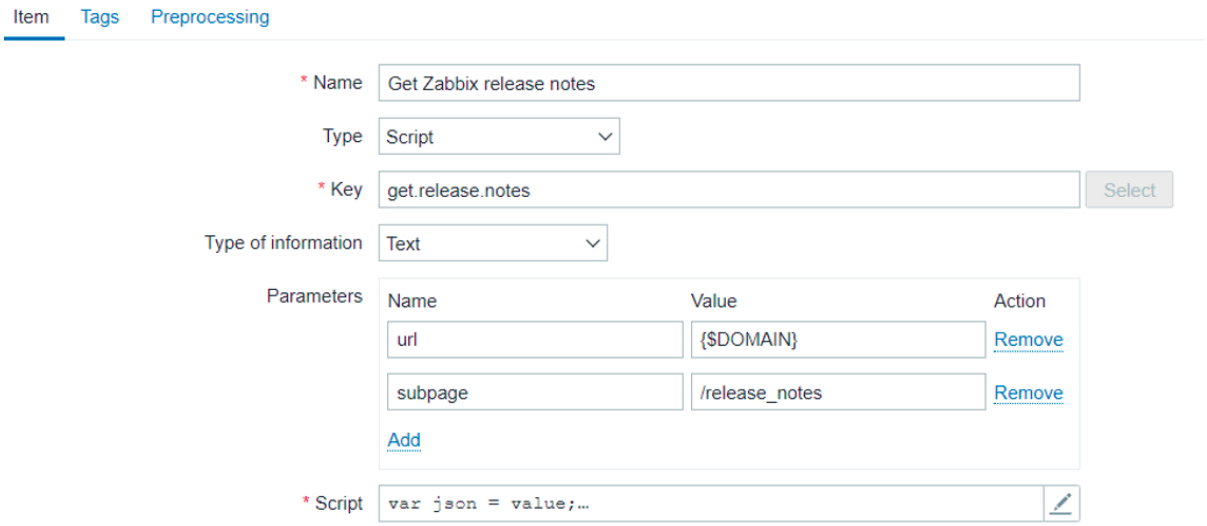
- Creare un item di tipo "Script".
- Nel campo *Script*, inserire:

```
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;
```

19 Item del browser

Panoramica

I Browser item consentono di monitorare siti web complessi e applicazioni web utilizzando un browser.

> **Attention:**
> Il supporto dei Browser item è attualmente sperimentale.

I Browser item raccolgono dati eseguendo codice JavaScript definito dall'utente e recuperando dati tramite HTTP/HTTPS. Questo item può simulare azioni correlate al browser come fare clic, inserire testo, navigare tra pagine web e altre interazioni dell'utente con siti web o applicazioni web.

Oltre allo script, è possibile specificare un elenco facoltativo di parametri (coppie di nome e valore) e un timeout.

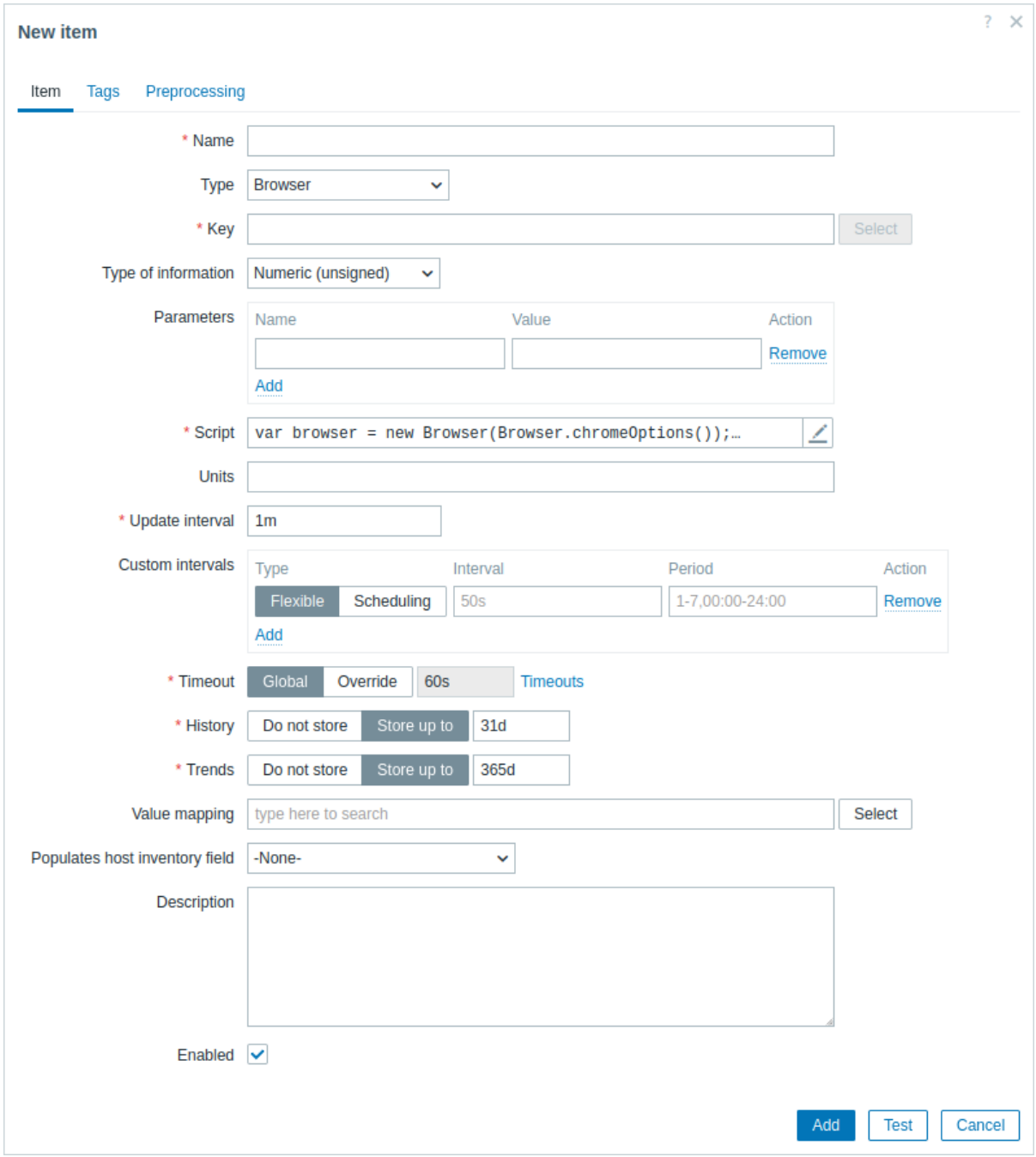
Attention:
L'item implementa parzialmente lo standard W3C WebDriver utilizzando come endpoint di test web Selenium Server oppure un WebDriver semplice (ad esempio, ChromeDriver). Affinché l'item funzioni, impostare l'endpoint nel parametro `WebDriverURL` del file di configurazione di Zabbix server/proxy (se si utilizza ChromeDriver, vedere Security Considerations). Per prestazioni migliori, valutare l'utilizzo di un server dedicato per l'ambiente di test web.

I controlli dei Browser item vengono eseguiti ed elaborati dai browser poller di Zabbix server o proxy. Se necessario, è possibile regolare il numero di istanze pre-fork dei browser poller nel parametro `StartBrowserPollers` del file di configurazione di Zabbix server/proxy.

Per il monitoraggio di siti web complessi e applicazioni web, è disponibile il template Website by Browser come template predefinito.

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 utilizzando il template *Website by Browser*, vedere Monitor websites with Browser items.

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);
}
```

4 Storia 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 form delle proprietà dell'item
- durante l'aggiornamento massivo degli item
- durante la configurazione delle attività di housekeeper

Qualsiasi dato più vecchio verrà rimosso 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 mantenere una cronologia lunga, 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 una buona idea 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 esaminare 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 solo sui dati della cronologia.

Note:
Come modo 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.

5 Parametri utente

Panoramica

A volte potresti voler eseguire un controllo agent che non è predefinito in Zabbix. In questo caso entrano in aiuto i parametri utente.

Puoi scrivere un comando che recuperi i dati di cui hai bisogno e includerlo nel parametro utente nel file di configurazione dell'agent (parametro di configurazione 'UserParameter').

Un parametro utente ha la seguente sintassi:

```
UserParameter=<key>,<command>
```

Come puoi vedere, un parametro utente contiene anche una chiave. La chiave sarà necessaria durante la configurazione di un item. Inserisci una chiave a tua scelta che sia facile da richiamare (deve essere univoca all'interno di un host).

Riavvia l'agent oppure usa l'opzione di controllo runtime dell'agent per applicare il nuovo parametro, ad esempio:

```
zabbix_agentd -R userparameter_reload
```

Quindi, durante la configurazione di un item, inserisci la chiave per fare riferimento al comando del parametro utente che vuoi eseguire.

I parametri utente sono comandi eseguiti da Zabbix agent. Tieni presente che possono essere restituiti fino a 16 MB di dati prima delle fasi di pre-elaborazione del valore dell'item.

/bin/sh viene utilizzato come interprete della riga di comando nei sistemi operativi UNIX. I parametri utente rispettano il timeout del controllo agent; se il timeout viene raggiunto, il processo del parametro utente generato tramite fork viene terminato.

Vedi anche:

- Tutorial passo passo sull'utilizzo dei parametri utente
- Esecuzione dei comandi

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'.

Parametri utente flessibili

I parametri utente flessibili accettano parametri con la chiave. In questo modo, un parametro utente flessibile può costituire la base per creare diversi item.

I parametri utente flessibili hanno la seguente sintassi:

```
UserParameter=key[*],command
```

Parameter	Description
Key	Chiave item univoca. [*] definisce che questa chiave accetta parametri tra parentesi quadre. I parametri vengono forniti durante la configurazione dell'item.
Command	Comando da eseguire per valutare il valore della chiave. *Solo per i parametri utente flessibili*: È possibile usare riferimenti posizionali $1…$9 nel comando per fare riferimento al rispettivo parametro nella chiave item. Zabbix analizza i parametri racchiusi tra [ ] della chiave item e sostituisce di conseguenza $1,...,$9 nel comando. $0 verrà sostituito dal comando originale (prima dell'espansione di $0,...,$9) da eseguire. I riferimenti posizionali vengono interpretati indipendentemente dal fatto che siano racchiusi tra virgolette doppie (") o singole (''). Per usare i riferimenti posizionali senza modificarli, specificare un doppio simbolo del dollaro: ad esempio, awk '{print $$2}'. In questo caso $$2 diventerà effettivamente $2 durante l'esecuzione del comando.

Attention:
I riferimenti posizionali con il simbolo $ vengono cercati e sostituiti da Zabbix agent solo per i parametri utente flessibili. Per i parametri utente semplici, tale elaborazione dei riferimenti viene saltata e, pertanto, non è necessario alcun escaping del simbolo $.

Attention:
Alcuni caratteri non sono consentiti nei parametri utente per impostazione predefinita. Per l'elenco completo dei caratteri, vedere UnsafeUserParameters.

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]
```

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.

1 Estendere gli agent Zabbix

Questo tutorial fornisce istruzioni passo dopo passo su come estendere la funzionalità dell'agent Zabbix con l'uso di un parametro utente.

Passo 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 a 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 Zabbix agent con il flag "-t" (se in esecuzione come root, tuttavia, tenere presente che l'agent potrebbe avere permessi diversi quando viene avviato come demone):

```
zabbix_agentd -t mysql.questions
```

Passaggio 3

Ricaricare i parametri utente dal file di configurazione eseguendo:

```
zabbix_agentd -R userparameter_reload
```

In alternativa, è possibile riavviare l'agent invece di utilizzare il comando di controllo a runtime.

Testare il parametro utilizzando l'utilità zabbix_get.

Passaggio 4

Aggiungere un nuovo item con Key=mysql.questions all'host monitorato. Il tipo dell'item deve essere Zabbix Agent oppure Zabbix Agent (active).

Tenere presente che il tipo dei valori restituiti deve essere impostato correttamente sul server Zabbix. In caso contrario, Zabbix non li accetterà.

6 Contatori delle prestazioni di Windows (Windows performance counters)

Panoramica

È possibile monitorare efficacemente i contatori delle prestazioni di Windows utilizzando la chiave *perf_counter[]*.

Per 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 solo in inglese `perf_counter_en`, vedere chiavi item specifiche per 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 oggetti con più istanze.

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 di item `perf_counter`, `perf_counter_en` e nei parametri di configurazione `PerfCounter`, `PerfCounterEn`.

Tuttavia, non è consigliato utilizzarle a meno che non si possa garantire che i propri indici numerici siano associati 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 usare la chiave `perf_counter_en` o il parametro di configurazione `PerfCounterEn`, che consentono di usare nomi in inglese indipendentemente dalle impostazioni locali del sistema.

Per trovare gli equivalenti numerici, eseguire **regedit**, quindi individuare *Counter* in *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows 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 per ogni parte testuale del contatore delle prestazioni, come in '\System\% Processor Time':

```
System → 2
% Processor Time → 6
```

È quindi possibile usare questi numeri per rappresentare il percorso in forma numerica:

```
\2\6
```

Parametri del contatore 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
or
PerfCounter=UserPerfCounter2,"\4\24",30
```

Con tali parametri configurati, è quindi possibile utilizzare semplicemente *UserPerfCounter1* o *UserPerfCounter2* come chiavi per creare i rispettivi item.

Ricordarsi di riavviare Zabbix agent dopo aver apportato modifiche al file di configurazione.

7 Aggiornamento di massa

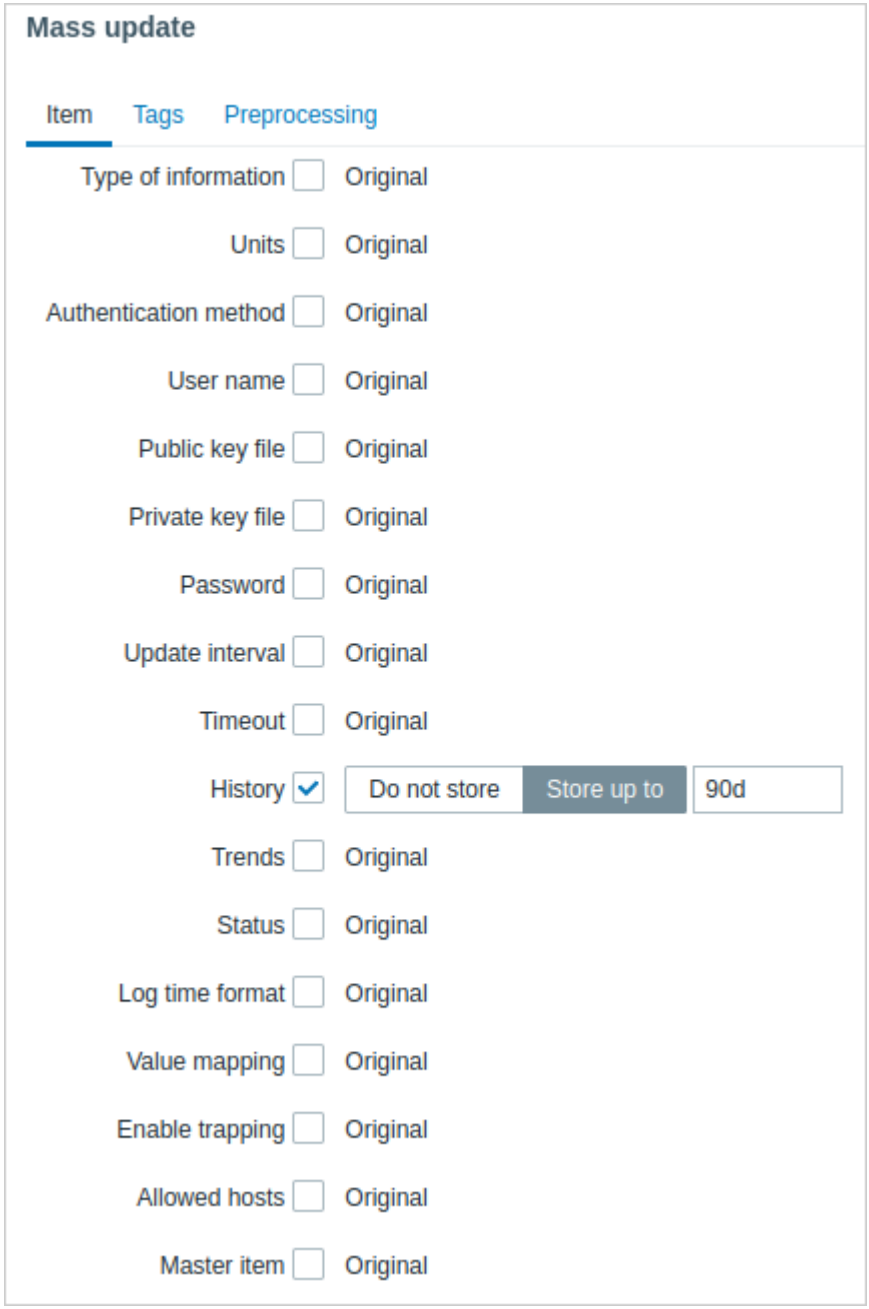
Panoramica

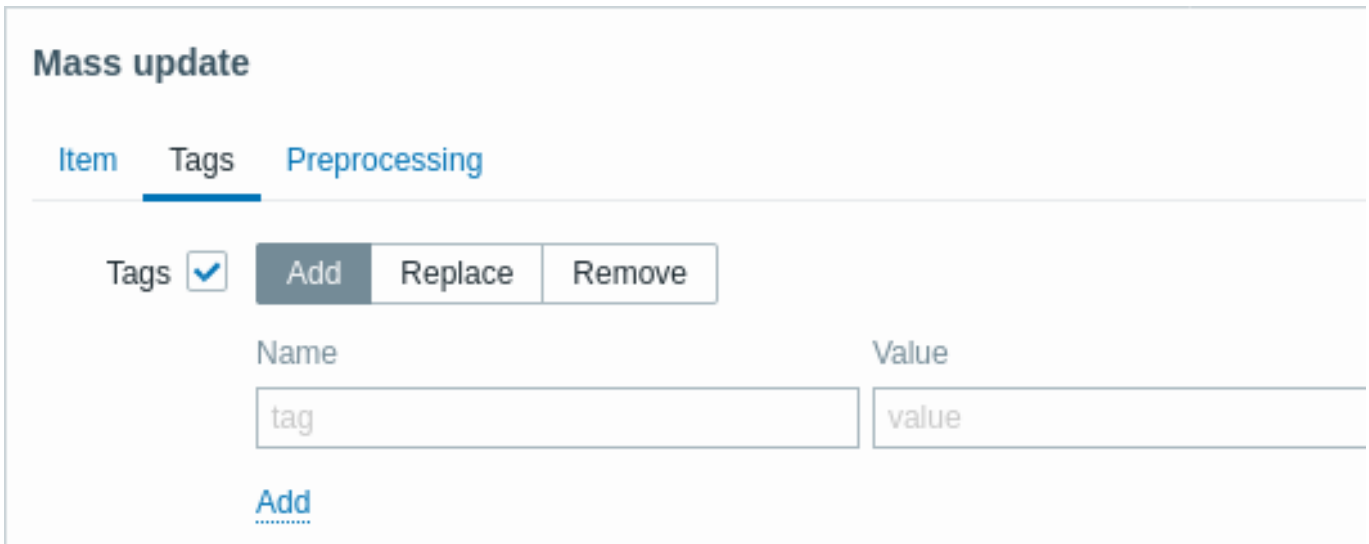
A volte potresti voler modificare alcuni attributi per un numero di oggetti in una singola volta. Invece di aprire ogni singolo oggetto per la modifica, puoi utilizzare la funzione di aggiornamento di massa.

Utilizzo dell'aggiornamento di massa

Per aggiornare in massa alcuni item, 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* oppure *Preprocessing*)
- Selezionare le caselle di controllo degli attributi da aggiornare
- Inserire nuovi valori per gli attributi



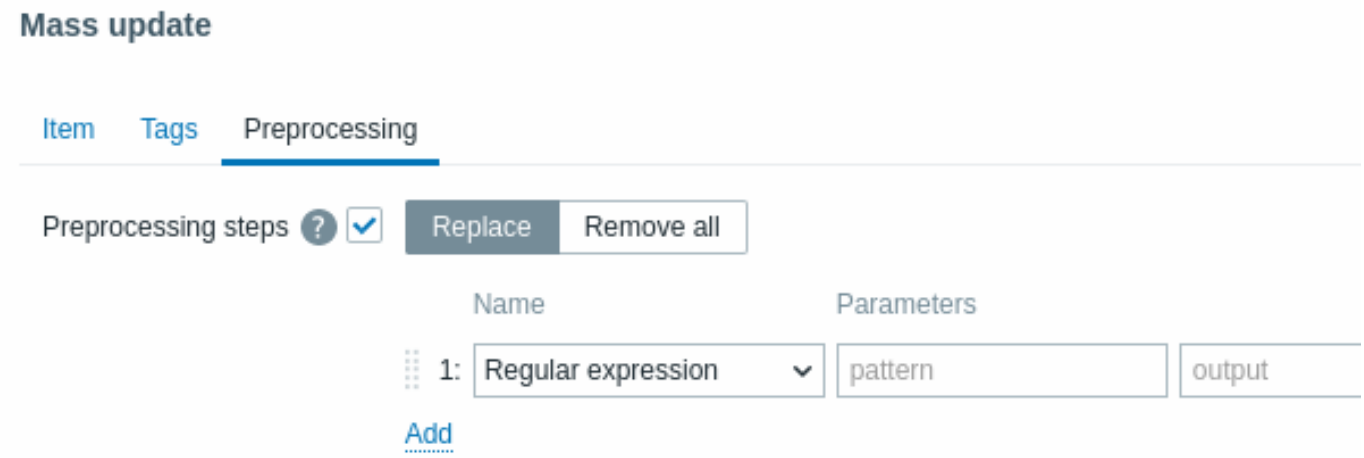


L'opzione *Tag* consente di:

- *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 *Passi di preprocessing* per attivare l'aggiornamento di massa per i passi di preprocessing.

L'aggiornamento di massa *Preprocessing* consente di:

- *Sostituire* - sostituire i passi di preprocessing esistenti sugli item con i passi di preprocessing specificati di seguito
- *Rimuovere tutto* - rimuovere tutti i passi di preprocessing dagli item

Al termine, fare clic su *Aggiorna*.

8 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*.

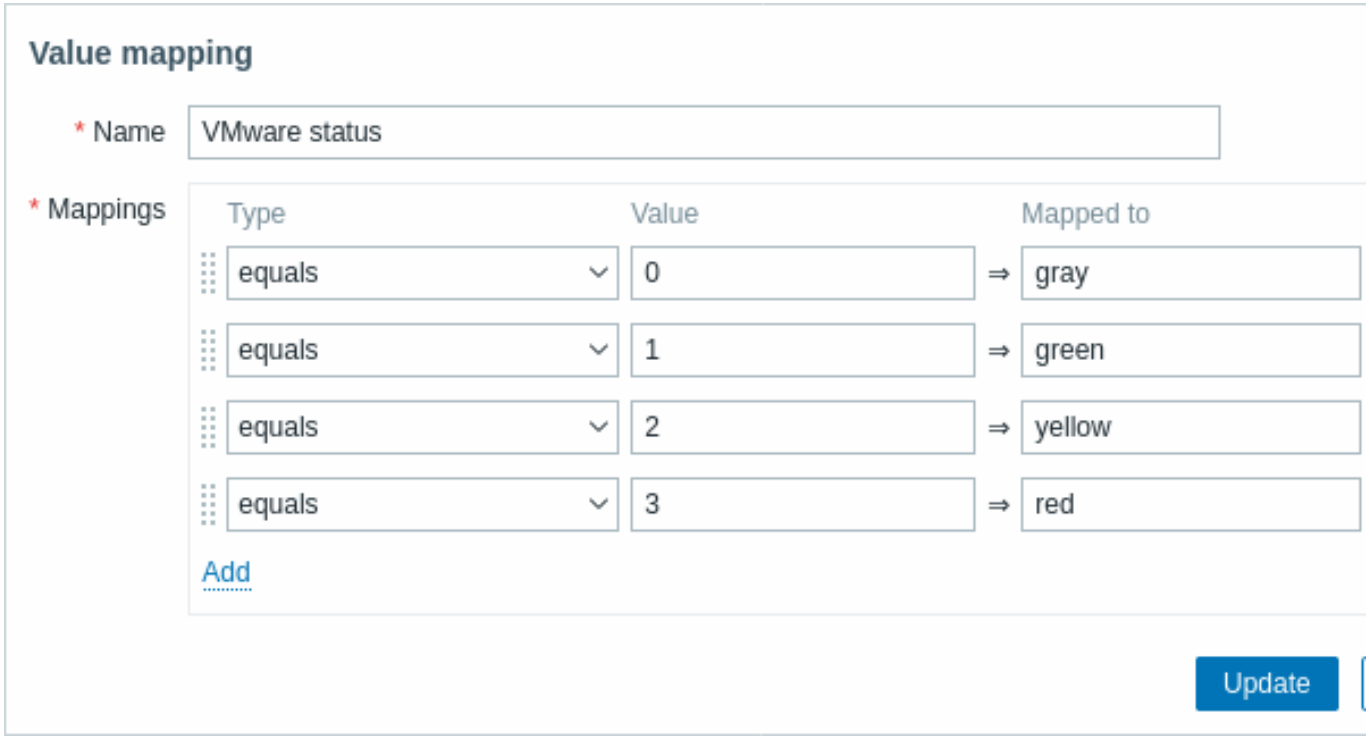
Le mappature dei valori vengono esportate/importate con il rispettivo template o host. Possono anche essere aggiornate in massa utilizzando i moduli di aggiornamento di massa di host e template.

Configurazione

Per configurare una mappatura dei valori, segui questi passaggi:

1. Apri il modulo di configurazione del host o del template.

2. Nella scheda *Mappatura dei valori*, fai clic su *Aggiungi* per aggiungere una nuova mappatura dei valori, oppure fai 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 mappare valori numerici/stringa in rappresentazioni testuali. La mappatura viene applicata nell'ordine delle regole, che può essere modificato trascinandole.

Parametro	Descrizione
Tipo	Tipo di mappatura: **uguale a** - i valori uguali verranno mappati; **maggiore o uguale a** - i valori uguali o maggiori verranno mappati; **minore o uguale a** - i valori uguali o minori verranno mappati; **nell'intervallo** - i valori nell'intervallo verranno mappati; 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 alla espressione regolare (le espressioni regolari globali non sono supportate); **predefinito** - verranno mappati tutti i valori rimanenti, diversi da 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 testuale (fino a 64 caratteri) del valore in ingresso.

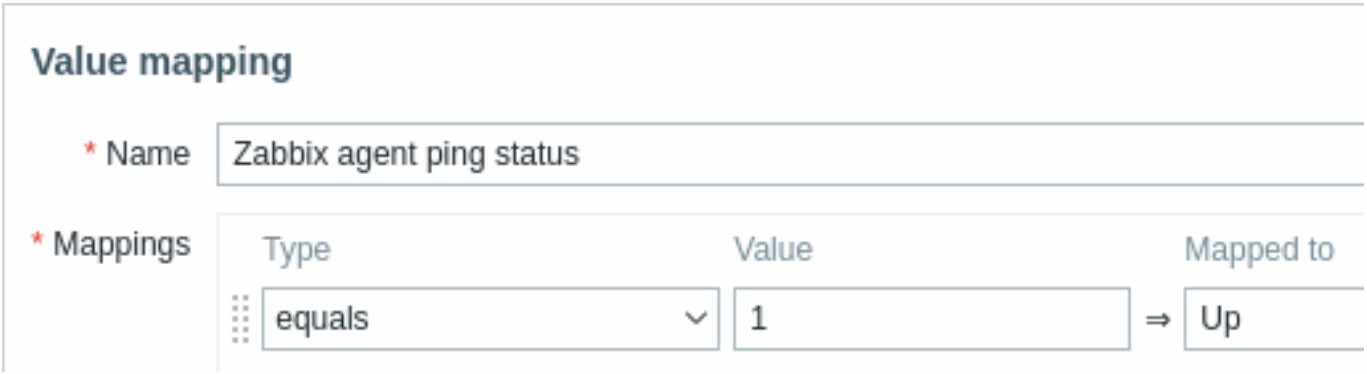
Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Quando si visualizza la mappatura dei valori nell'elenco, sono visibili solo le prime tre mappature; tre puntini indicano che esistono altre mappature.



Esempio di mappatura dei valori

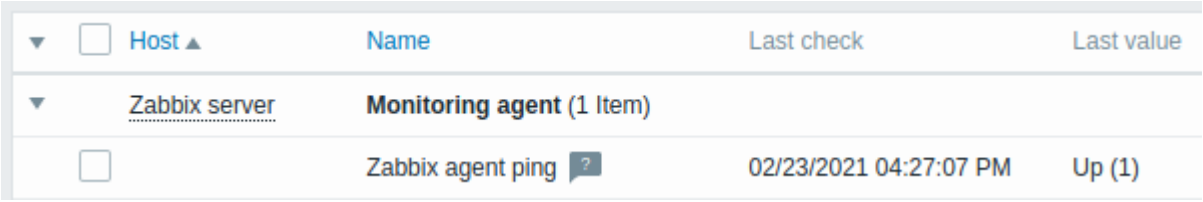
Uno degli item predefiniti dell'agent, *Zabbix agent ping*, utilizza una mappatura dei valori a livello di template "Zabbix agent ping status" per visualizzare i propri valori.



Nel modulo di configurazione dell'item, puoi trovare un riferimento a questa mappatura dei valori nel campo *Value mapping*:



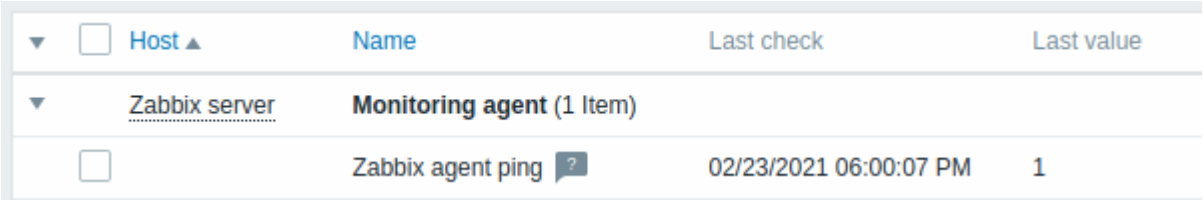
Questa mappatura viene utilizzata nella sezione *Monitoring* → *Latest data* per visualizzare "Up" (con il valore grezzo tra parentesi).



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 la mappatura dei valori, questa abbreviazione non viene applicata al valore mappato ma solo al valore grezzo (visualizzato tra parentesi).

Senza una mappatura dei valori predefinita, vedresti solo ”1”, che 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

9 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

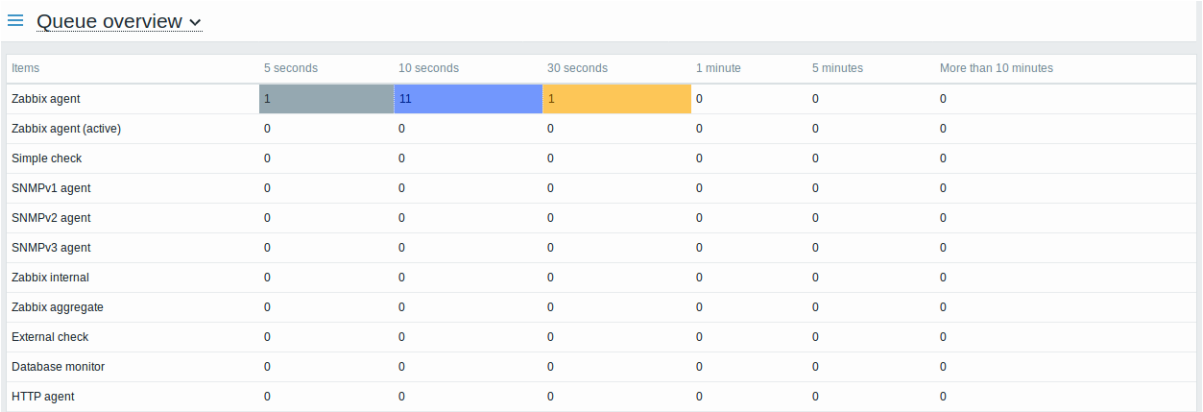
Le statistiche mostrate dalla coda sono un buon indicatore delle prestazioni di Zabbix server.

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, vai su *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 è generalmente "ok", quindi possiamo presumere che il server stia funzionando correttamente.

La coda mostra alcuni item in attesa fino a 30 secondi. Sarebbe utile sapere quali sono questi item.

Per farlo, seleziona *Queue details* nel menu a discesa del titolo. Ora puoi vedere un elenco di quegli 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 capire perché questi item potrebbero essere in ritardo.

Con uno o due item in ritardo probabilmente non c'è motivo di allarmarsi. Potrebbero aggiornarsi in un secondo. Tuttavia, se vedi un gruppo di item in ritardo per troppo tempo, potrebbe esserci un problema più serio.

Vedi anche: Allineamento dei fusi orari quando si utilizzano gli 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

Uno speciale item interno **zabbix[queue,<from>,<to>]** può essere utilizzato 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.

10 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, il file di configurazione del server Zabbix supporta il parametro opzionale **ValueCacheSize**.

Sono supportati due item interni per monitorare la cache dei valori: **zabbix[vcache,buffer,<mode>]** e **zabbix[vcache,cache,<parameter**
Per maggiori dettagli, vedere item interni.

11 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

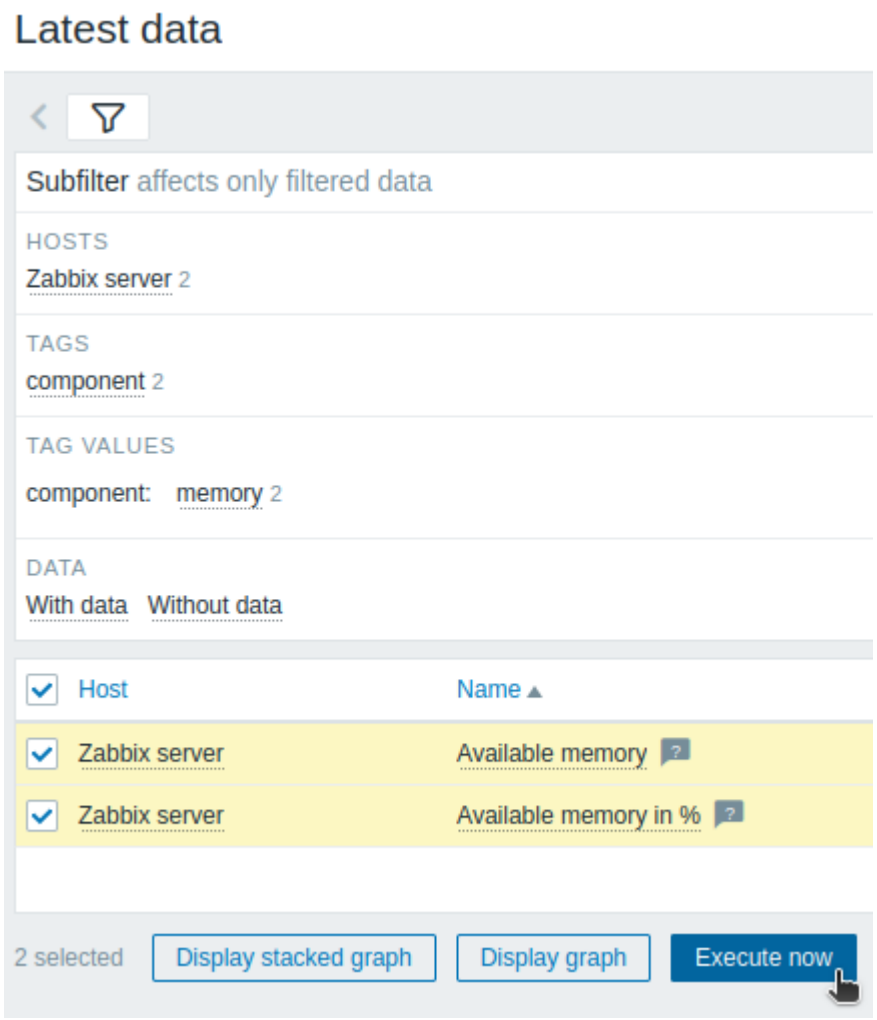
Puoi eseguire controlli passivi da:

- *Monitoring* > *Latest data*
- *Data collection* > *Hosts* > *Items* o *Discovery*
- Il menu item
- Durante la modifica di item o regole di low-level discovery (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 regole di low-level discovery non verranno considerate. Per testare item o regole LLD 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 usare *Execute now* sugli item dei host per i quali dispongono del permesso di *lettura-scrittura*, oppure del permesso di *sola lettura* e dell'azione *Invoke "Execute now" on read-only hosts* abilitata per il loro ruolo.

12 Limitazione dei controlli degli agenti

Panoramica

È possibile controllare quali chiavi item Zabbix agent o agent 2 è autorizzato o negato a utilizzare durante l'esecuzione di controlli item, comandi remoti o script.

Per farlo, utilizzare questi parametri di configurazione dell'agent per definire regole di autorizzazione/negazione:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

Il `<pattern>` deve contenere una singola chiave item e supporta i caratteri jolly (`*`). Il carattere jolly corrisponde a un numero qualsiasi di qualsiasi carattere nella sua posizione e può essere utilizzato per far corrispondere chiavi item o parametri (ad esempio, `vfs.file.*[*]`).

Attention:
Per migliorare la sicurezza, si consiglia di utilizzare chiavi item esatte invece dei caratteri jolly. Per i dettagli, vedere Protezione delle regole di autorizzazione/negazione.

A differenza di altri parametri di configurazione dell'agent, è possibile specificare un numero illimitato di parametri `AllowKey` o `DenyKey`.

Note importanti

- Tutti gli item `system.run` sono disabilitati per impostazione predefinita (anche quando `DenyKey` è vuoto), come se `DenyKey=system.run[*]` fosse impostato come ultima regola. Per questo motivo, puoi consentire specifici item

system.run senza negare esplicitamente altri item system.run.

- Un item specificato in AllowKey deve essere specificato anche in DenyKey (eccetto gli item system.run); in caso contrario, Zabbix agent **non si avvierà**.
- Se possibile, usa AllowKey per consentire solo gli item richiesti e negare tutto il resto. Alcune chiavi possono essere sfruttate impropriamente per leggere file non previsti tramite path traversal (ad esempio, vfs.file.contents["../../../../etc/passwd" e le nuove versioni di Zabbix agent possono introdurre chiavi non coperte dalle tue regole DenyKey.
- La configurazione di AllowKey e DenyKey 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 della riga di comando --print (-p) di Zabbix agent non mostrerà le chiavi degli item negati.
  - Il parametro della riga di comando --test (-t) di Zabbix agent 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 di autorizzazione/negazione

Puoi specificare un numero illimitato di regole AllowKey o DenyKey, anche se il loro ordine è importante.

- Le regole vengono valutate una per una, dall'alto verso il basso.
- Quando una chiave item corrisponde a una regola, viene autorizzata 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 item non corrisponde; l'agent passa
AllowKey=vfs.file.contents[/etc/passwd]     # Il pattern della chiave item corrisponde; l'agent autorizza
DenyKey=vfs.file.*[*]                       # L'agent ignora la regola, poiché la valutazione è già stata
```

Il seguente ordine delle regole negherà il controllo dell'item:

```
DenyKey=vfs.file.*[*]                       # Il pattern della chiave item corrisponde; l'agent nega il co
AllowKey=vfs.file.contents[/etc/passwd]     # L'agent ignora la regola, poiché la valutazione è già stata
AllowKey=vfs.file.contents[/tmp/app.log]    # L'agent ignora la regola, poiché la valutazione è già stata
```

Esempi

I seguenti esempi mostrano schemi di configurazione comuni per AllowKey e DenyKey.

Consentire controlli e comandi specifici

Consentire solo due controlli item vfs.file e due comandi system.run:

```
AllowKey=vfs.file.contents[/tmp/app.log]
AllowKey=vfs.file.size[/tmp/app.log]
AllowKey=system.run[/usr/bin/uptime]
AllowKey=system.run[/usr/bin/df -h /]
DenyKey=vfs.file.*[*]
```

Note:
L'impostazione DenyKey=system.run[*] non è necessaria, perché tutti gli altri comandi system.run sono negati per impostazione predefinita.

Consentire gli script

Consente a Zabbix agent di eseguire script sugli host tramite tutti i metodi disponibili:

- Script globali che possono essere eseguiti nel frontend o tramite API (questo metodo utilizza sempre la chiave system.run[myscript.sh])
- Comandi remoti dalle operazioni delle azioni (questo metodo utilizza sempre la chiave system.run[myscript.sh,nowait])
- item Zabbix agent system.run con lo script, ad esempio:
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

Per controllare il parametro `wait/nowait`, è necessario impostare una regola diversa. Ad esempio, è possibile consentire solo gli item `system.run[myscript.sh,wait]`, escludendo così gli altri metodi:

```
AllowKey=system.run[myscript.sh,wait]
```

Protezione delle regole allow/deny

Questo esempio mostra come proteggere regole `AllowKey` o `DenyKey` eccessivamente permissive.

Considerare le seguenti regole:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

Note:
In Windows, è necessario eseguire l'escape degli spazi nel percorso usando un accento circonflesso (^).

Queste regole contengono un carattere jolly (*), che può essere usato in modo improprio:

- Lo script `test.bat` può essere eseguito con qualsiasi argomento, inclusi quelli non previsti.
- Il pattern `vfs.file.*` corrisponde solo alle chiavi item senza parametri; tuttavia, tutti gli item `vfs.file` richiedono parametri.
- Il pattern `system.cpu.load[*]` corrisponde solo alle chiavi item con parametri; tuttavia gli item `system.cpu.load` non richiedono parametri.

Per proteggere queste regole, consentire esplicitamente l'esecuzione di `test.bat` solo con argomenti specifici e negare i pattern corretti delle chiavi item; per esempio:

```
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[*]
```

È possibile 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

La tabella seguente mostra come vengono confrontati i pattern delle chiavi degli item:

- Una chiave corrisponde al pattern solo se soddisfa **tutte** le condizioni nella colonna *Corrisponde*.
- I parametri devono essere completamente racchiusi tra parentesi quadre (ad esempio, `vfs.file.contents[*` e `vfs.file.contents*utf8]` sono pattern non validi).

Pattern	Corrisponde	Esempi
`*`	Qualsiasi chiave con o senza parametri	

`vfs.file.*`	La chiave inizia con `vfs.file.`
	Nessun parametro
		Corrisponde:
		`vfs.file.size`
		`vfs.file.contents`
		
		Non corrisponde:
		`vfs.file.contents[]`
		`vfs.file.size[/var/log/app.log]`

`vfs.*.contents`	La chiave inizia con `vfs.`
	La chiave termina con `.contents`
	Nessun parametro
		Corrisponde:
		`vfs..contents`
		`vfs.mount.point.file.contents`
		
		Non corrisponde:
		`vfs.contents`
		`vfs.file.contents[]`

Pattern	Corrisponde	Esempi

`vfs.file.*[*]`	La chiave inizia con `vfs.file.`
	Qualsiasi parametro o parametri vuoti
		Corrisponde
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Non corrisponde:
		`vfs.file.get.custom`

`vfs.file.contents`	La chiave è `vfs.file.contents`
	Nessun parametro
		Corrisponde:
		`vfs.file.contents`
		
		Non corrisponde:
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	La chiave è `vfs.file.contents[]`
	Parametri vuoti
		Corrisponde:
		`vfs.file.contents[]`
		
		Non corrisponde:
		`vfs.file.contents`

`vfs.file.contents[*]`	La chiave è `vfs.file.contents`
	Qualsiasi parametro o parametri vuoti
		Corrisponde:
		`vfs.file.contents[/path/to/file]`
		
		Non corrisponde:
		`vfs.file.contents`

`vfs.file.contents[/etc/passwd,*]`	La chiave è `vfs.file.contents`
	Il primo parametro è `/etc/passwd`
	Qualsiasi secondo parametro o secondo parametro vuoto
		Corrisponde:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Non corrisponde:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[*passwd*]`	La chiave è `vfs.file.contents`
	Il primo parametro include `passwd`
	Nessun secondo parametro
		Corrisponde:
		`vfs.file.contents[/etc/passwd]`
		
		Non corrisponde:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`vfs.file.contents[*passwd*,*]`	La chiave è `vfs.file.contents`
	Il primo parametro include `passwd`
	Qualsiasi secondo parametro o secondo parametro vuoto
		Corrisponde:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Non corrisponde:
		`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`
		Corrisponde:
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Non corrisponde:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

3 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 se il problema si verifica.

Il flusso completo del rilevamento dei problemi è il seguente:

1. Assicurati di avere un item che raccolga dati.
2. Definisci il trigger per questo item. Senza un trigger, Zabbix si limita a raccogliere dati, ma non reagisce mai ad essi.

3. Osserva i problemi rilevati:
    - In *Monitoring* > *Problems*
    - Nel widget *Problems*
4. Ricevi notifiche, esegui comandi remoti (se definiti).

Tieni presente che i template ufficiali di Zabbix contengono trigger predefiniti. Consulta la guida sull'applicazione di un template per monitorare 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.

1 Esempi di trigger

Questa pagina è una raccolta di 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 nei 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`

Rilevare modifiche nella query DNS

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Notare le virgolette attorno al secondo operando.

Questo trigger si attiva se il risultato della query non è uguale a quello che normalmente restituisce:

```
example.com           MX       0 mail.example.com
```

Funzione: last
 Item di dati: net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], con le 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

Rilevare 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
 Item di dati: system.localtime

Si noti che system.localtime deve essere configurato come controllo passivo per Zabbix agent; in 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`

2 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 un template a monitorare Linux. È anche 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 del trigger (dettagli della sintassi)
- Best practice

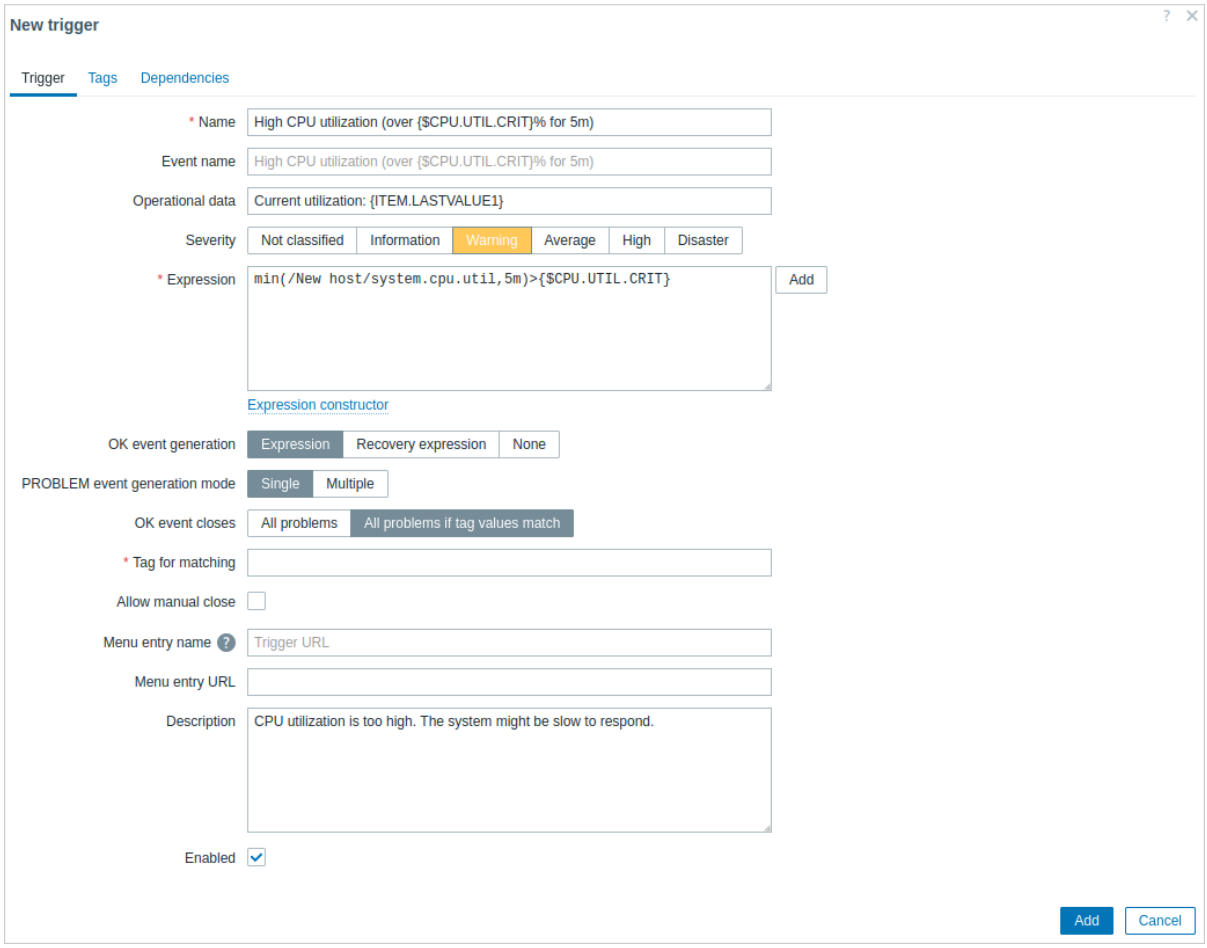
Configurazione

Per configurare un trigger, procedere come segue:

- Andare in: *Raccolta dati* > *Host*
- Fare clic su *Trigger* nella riga dell'host

- Fare clic su *Crea trigger* a destra (oppure sul nome del trigger per modificare un trigger esistente)
- Inserire i parametri del trigger nel modulo

La scheda **Trigger** contiene tutti gli attributi essenziali del trigger.

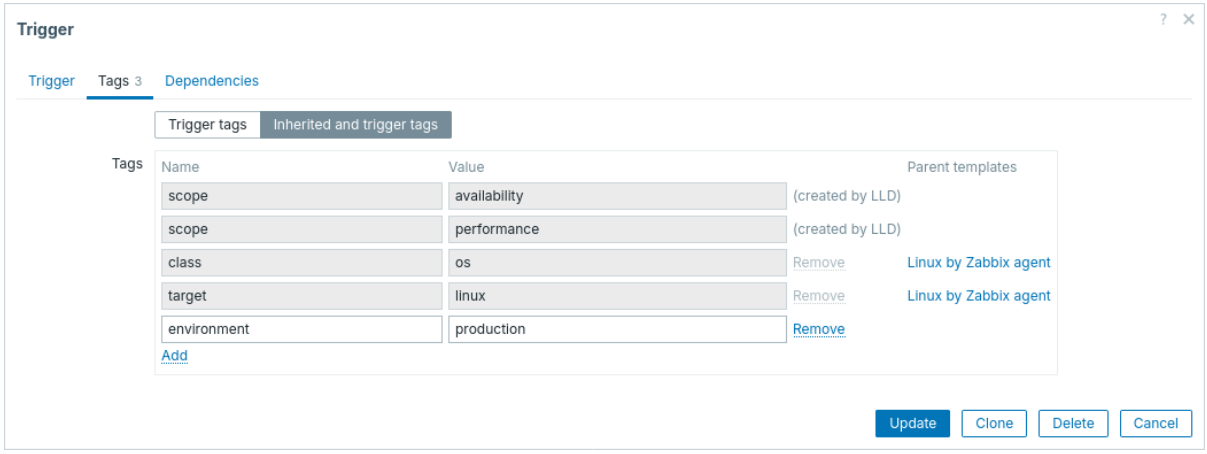


Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

Parametro	Descrizione
Nome	Nome del trigger. Le macro supportate 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 le macro utente {$MACRO}. Le macro **$1, $2...$9** possono essere utilizzate per fare riferimento alla prima, seconda...nona costante dell'espressione. *Nota*: le macro $1-$9 verranno risolte correttamente se fanno riferimento a costanti in espressioni relativamente semplici e lineari. Ad esempio, il nome "Carico del processore sopra $1 su {HOST.NAME}" cambierà automaticamente in "Carico del processore sopra 5 su New host" se l'espressione è last(/New host/system.cpu.load[percpu,avg1])>5.
Nome evento	Se definito, questo nome verrà utilizzato per creare il nome dell'evento problema. Per impostazione predefinita, il nome dell'evento è uguale al nome del trigger. Il nome dell'evento può essere utilizzato per creare avvisi significativi contenenti dati del problema (vedere esempio). È supportato lo stesso insieme di macro del nome del trigger, più le macro di espressione {TIME}, {TIMESTAMP} e {?EXPRESSION}.

Parametro	Descrizione
Dati operativi	I dati operativi consentono di definire stringhe arbitrarie insieme alle macro. Le macro verranno risolte dinamicamente in dati in tempo reale in *Monitoraggio* > *Problemi*. Mentre le macro nel nome del trigger (vedere sopra) verranno risolte nei loro valori nel momento in cui si verifica un problema e diventeranno la base di un nome problema statico, le macro nei dati operativi mantengono la capacità di visualizzare dinamicamente le informazioni più recenti. Se non sono configurati dati operativi a livello di trigger, verranno visualizzati gli ultimi valori di tutti gli item dell'espressione. È supportato lo stesso insieme di macro del nome del trigger.
Gravità	Impostare la gravità richiesta del trigger facendo clic sui pulsanti.
Espressione	Espressione logica utilizzata per definire le condizioni di un problema. Sono supportati i suffissi temporali e i suffissi di dimensione della memoria. Un problema viene creato dopo che tutte le condizioni incluse nell'espressione sono soddisfatte, cioè quando l'espressione restituisce TRUE. Il problema verrà risolto non appena l'espressione restituisce FALSE, a meno che non siano specificate condizioni di ripristino aggiuntive in *Espressione di ripristino*.
Generazione evento OK	Opzioni di generazione dell'evento OK: **Espressione** - gli eventi OK vengono generati in base alla stessa espressione degli eventi problema; **Espressione di ripristino** - gli eventi OK vengono generati se l'espressione del problema restituisce FALSE e l'espressione di ripristino restituisce TRUE; **Nessuno** - in questo caso il trigger non tornerà mai autonomamente allo stato OK.
Espressione di ripristino	Espressione logica (facoltativa) che definisce condizioni aggiuntive che devono essere soddisfatte prima che il problema venga risolto, dopo che l'espressione originale del problema è già stata valutata come FALSE. L'espressione di ripristino è utile per l'isteresi del trigger. **Non** è possibile risolvere un problema solo tramite l'espressione di ripristino se l'espressione del problema è ancora TRUE. Questo campo è disponibile solo se per *Generazione evento OK* è selezionata l'opzione 'Espressione di ripristino'.
Modalità di generazione evento PROBLEM	Modalità per la generazione degli eventi problema: **Singolo** - viene generato un singolo evento quando un trigger entra nello stato 'Problema' per la prima volta; **Multiplo** - viene generato un evento a *ogni* valutazione 'Problema' del trigger.
L'evento OK chiude	Selezionare se l'evento OK chiude: **Tutti i problemi** - tutti i problemi di questo trigger; **Tutti i problemi se i valori dei tag corrispondono** - solo i problemi del trigger con valori dei tag evento corrispondenti.
Tag per la corrispondenza	Inserire il nome del tag evento da utilizzare per la correlazione degli eventi. Questo campo viene visualizzato se per la proprietà *L'evento OK chiude* è selezionato 'Tutti i problemi se i valori dei tag corrispondono' ed è obbligatorio in questo caso.
Consenti chiusura manuale	Selezionare questa opzione per consentire la chiusura manuale degli eventi problema generati da questo trigger. La chiusura manuale è possibile quando si confermano gli eventi problema.
Nome voce di menu	Se non è vuoto, il nome inserito qui (fino a 64 caratteri) viene utilizzato in varie posizioni del frontend come etichetta per l'URL del trigger specificato nel parametro *URL voce di menu*. Se vuoto, viene utilizzato il nome predefinito *URL trigger*. È supportato lo stesso insieme di macro del nome del trigger, più {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}.
URL voce di menu	Se non è vuoto, l'URL inserito qui (fino a 2048 caratteri) è disponibile come collegamento nel menu evento in varie posizioni del frontend, ad esempio facendo clic sul nome del problema in *Monitoraggio* > *Problemi* o nel widget dashboard *Problemi*. È supportato lo stesso insieme di macro del nome del trigger, più {EVENT.ID}, {HOST.ID} e {TRIGGER.ID}. Nota: le macro utente con valori segreti non verranno risolte nell'URL.
Descrizione	Campo di testo utilizzato per fornire maggiori informazioni su questo trigger. Può contenere istruzioni per risolvere uno specifico problema, dettagli di contatto del personale responsabile, ecc. È supportato lo stesso insieme di macro del nome del trigger.
Abilitato	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 **Tag** consente di definire i tag a livello di trigger. Tutti i problemi di questo trigger verranno contrassegnati con i valori inseriti qui.



Inoltre, l'opzione *Tag ereditati e del trigger* 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.

Parametro	Descrizione
Nome/Valore	Impostare tag personalizzati per contrassegnare gli eventi del trigger. I tag sono una coppia composta da nome e valore del tag. È possibile utilizzare solo il nome oppure associarlo a un valore. Un trigger può avere più tag con lo stesso nome, ma valori diversi. Nei tag evento sono supportate le macro utente, le macro utente con contesto, le macro di low-level discovery e le funzioni di macro con `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}`. Le macro di low-level discovery possono essere utilizzate all'interno del contesto 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, sopprimere i problemi di questi trigger durante la manutenzione. Se la lunghezza totale del valore espanso supera 255, verrà troncata a 255 caratteri. Vedere tutte le macro supportate per i tag evento. I tag evento possono essere utilizzati per la correlazione degli eventi, nelle condizioni delle azioni e saranno visibili anche in *Monitoraggio* > *Problemi* o nel widget *Problemi*.

La scheda **Dipendenze** contiene tutte le dipendenze del trigger.

Fare clic su *Aggiungi* per aggiungere una nuova dipendenza.

> **Note:**
> È anche possibile configurare un trigger aprendone uno esistente, premendo il pulsante *Clona* e quindi salvandolo con un nome diverso.

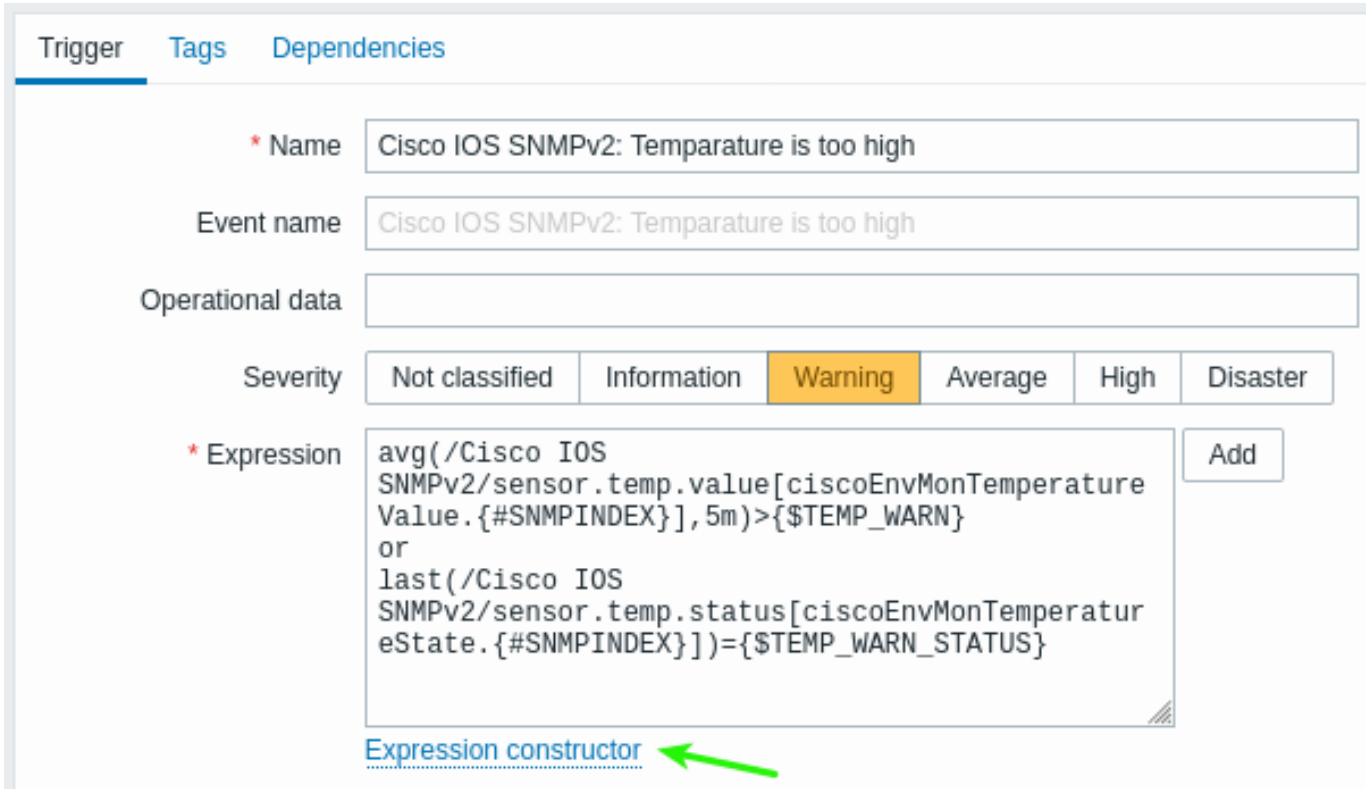
Test delle espressioni dei trigger

È possibile verificare quale sarebbe il risultato dell'espressione in base al valore ricevuto.

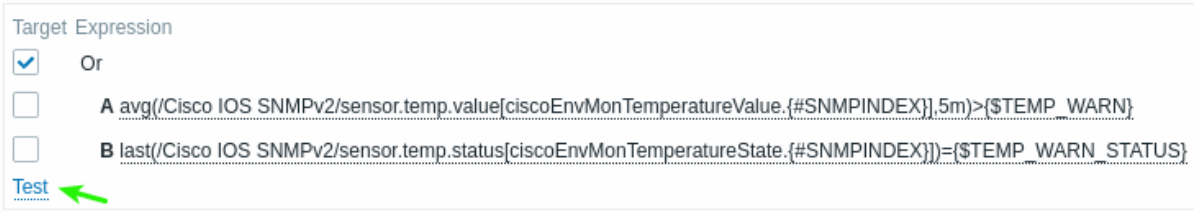
Come esempio viene utilizzata la seguente espressione tratta da un template ufficiale:

```
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}
```

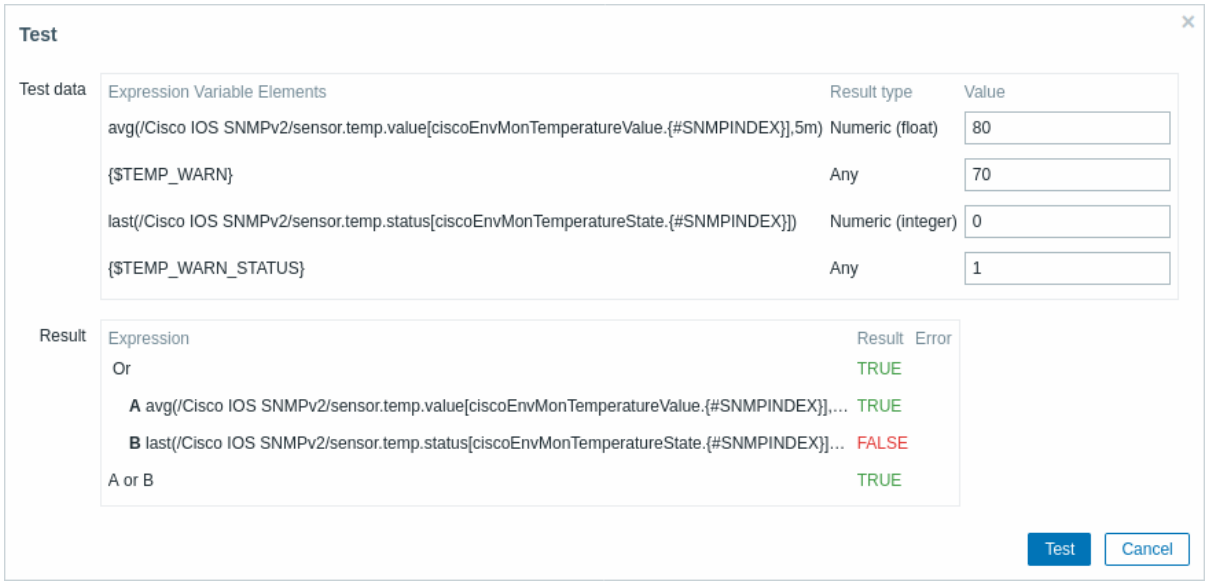
Per testare l'espressione, fare clic su *Expression constructor* sotto il campo dell'espressione.



In Expression constructor sono elencate tutte le espressioni individuali. 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 visualizzare 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 particolare A, "80" è maggiore del valore specificato {$TEMP_WARN}, che in questo esempio è "70". Come previsto, viene visualizzato un risultato "TRUE".

"FALSE" significa che l'espressione specificata non è corretta. In questo caso particolare B, {$TEMP_WARN_STATUS} "1" deve essere uguale al valore specificato, che in questo esempio è "0". 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.

3 Espressione del trigger

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 calcolare la media o trovare una stringa specifica.

Fare clic sul rispettivo gruppo di funzioni per visualizzare maggiori 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 bit a bit	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Funzioni di data e ora	date, dayofmonth, dayofweek, now, time
Funzioni dello storico	change, changecount, count, countunique, find, first, firstclock, fuzzytime, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate
Funzioni dei trend	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
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

Gruppo di funzioni	Funzioni
Funzioni operatore	between, in
Funzioni predittive	forecast, timeleft
Funzioni stringa	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, 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 della funzione

I parametri della funzione consentono di specificare:

- chiave item (come `/host/key`) per le funzioni che fanno riferimento alla cronologia degli item del host
- 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. Questo consente di specificare l'intervallo che ci interessa. Può essere specificato come periodo di tempo (30s, 10m, 1h) oppure come intervallo di valori (#5 - per gli ultimi cinque valori).

È possibile usare i secondi o i suffissi temporali per indicare il periodo di tempo. Se preceduto da un cancelletto, il parametro ha un significato diverso:

Expression	Description
sum(/host/key,**10m)**	Somma dei valori negli ultimi 10 minuti.
sum(/host/key,**#10)**	Somma degli ultimi dieci valori.

I parametri con cancelletto hanno un significato diverso con la funzione **last**: indicano l'N-esimo valore precedente, quindi 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 "ora", dove "ora" è l'ultimo momento di ricalcolo del trigger (vedere Frequenza di calcolo); "ora" non è l'ora corrente del server.

Il periodo di tempo specifica uno dei seguenti casi:

- Considerare tutti i valori tra "ora-periodo di tempo" e "ora" (oppure, con spostamento temporale, tra "ora-spostamento temporale-periodo di tempo" e "ora-time_shift")
- Considerare non più di num valori del passato, fino a "ora"
  - Se sono disponibili 0 valori per il periodo di tempo o il conteggio num specificato, allora il trigger o l'item calcolato che utilizza questa funzione diventa non supportato

Si noti che:

- Se nel trigger viene utilizzata una sola funzione (che fa riferimento allo storico dei dati), "ora" è sempre l'ultimo valore ricevuto. Ad esempio, se l'ultimo valore è stato ricevuto un'ora fa, il periodo di tempo sarà considerato fino all'ultimo valore ricevuto un'ora fa.
- Un nuovo trigger viene calcolato non appena viene ricevuto il primo valore (funzioni dello storico); verrà calcolato entro 30 secondi per le funzioni data e ora e nodata(). Pertanto il trigger verrà calcolato anche se forse 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 temporale era impostato, ad esempio, sugli ultimi dieci valori.

Scostamento temporale

È supportato uno scostamento temporale opzionale con il tempo o il conteggio dei valori come parametro della funzione. Questo parametro consente di fare riferimento ai dati di un periodo di tempo nel passato.

Lo scostamento temporale inizia con `now` - che specifica l'ora corrente, ed è seguito da `+N<time unit>` oppure `-N<time unit>` - per aggiungere o sottrarre N unità di tempo.

Ad esempio, `avg(/host/key,1h:now-1d)` restituirà il valore medio per un'ora di un giorno fa.

Attention:
Lo scostamento temporale specificato in mesi (M) e anni (y) è supportato solo per le funzioni di trend. Le altre funzioni supportano secondi (s), minuti (m), ore (h), giorni (d) e settimane (w).

Scostamento temporale con periodi di tempo assoluti

I periodi di tempo assoluti sono supportati nel parametro di scostamento temporale, 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: `/<time unit>` - definisce l'inizio e la fine dell'unità di tempo, ad esempio, da mezzanotte a mezzanotte per un giorno, `+N<time unit>` oppure `-N<time unit>` - per aggiungere o sottrarre N unità di tempo.

Si noti che il valore dello scostamento temporale può essere maggiore o uguale a 0, mentre il valore minimo del periodo di tempo è 1.

Parameter	Description
1d:now/d	Ieri
1d:now/d+1d	Oggi
2d:now/d+1d	Ultimi 2 giorni
1w:now/w	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
6	**=**	Uguale a. L'operatore è definito come: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		No [1]

Priorità	Operatore	Definizione	**Note per i valori sconosciuti**	Forza il cast dell'operando a float [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 sconosciuto dell'espressione

È possibile che in un'espressione di trigger compaia un operando sconosciuto quando:

- viene utilizzato un item non supportato
- la valutazione della funzione, per un item supportato, restituisce un errore

In questo caso, l'espressione di trigger viene generalmente valutata come `Unknown` (poiché non può essere valutata)

È possibile ricevere una notifica sui trigger sconosciuti.

Eccezioni

Nonostante la presenza di un operando sconosciuto, in alcuni casi le espressioni di trigger possono essere valutate con un risultato noto (Problem/OK):

- La funzione `nodata()` viene valutata indipendentemente dal fatto che l'item di riferimento 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' oppure "Problem"),
  - **Caso 2**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" viene valutata con un risultato noto ('0' oppure "OK").
- Se la valutazione della funzione per un item supportato restituisce un errore, il valore della funzione diventa `Unknown` e partecipa come operando sconosciuto alla successiva valutazione 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 nel caso della divisione per 0).

Attention:
Lo stato sconosciuto dell'espressione non modifica lo stato del trigger (Problem/OK). Quindi, se lo stato del trigger era ”Problem” (vedere Caso 1), rimane nello stato di problema anche se la parte nota viene risolta ('1' diventa '0'), perché l'espressione ora viene valutata come `Unknown` e ciò non modifica lo stato del trigger.

Se un'espressione di 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.

1 Funzioni aggregate

Salvo diversa indicazione, tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni aggregate possono operare con:

- la cronologia degli item, ad esempio `min(/host/key,1h)`
- le funzioni foreach come unico parametro, ad esempio `min(last_foreach(/*/key))` (solo negli item calcolati; non possono essere usate nei trigger)

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
avg	Il valore medio di un item nel periodo 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 quantile φ dai bucket di un istogramma.
item_count	Il numero di item esistenti nella configurazione che corrispondono ai criteri di filtro.
kurtosis	La ”pesantezza delle code” della distribuzione di probabilità nei valori raccolti nel periodo di valutazione definito.
mad	La deviazione assoluta mediana nei valori raccolti nel periodo di valutazione definito.
max	Il valore più alto di un item nel periodo di valutazione definito.
min	Il valore più basso di un item nel periodo di valutazione definito.
skewness	L'asimmetria della distribuzione di probabilità nei valori raccolti nel periodo di valutazione definito.
stddevpop	La deviazione standard della popolazione nei valori raccolti nel periodo di valutazione definito.
stddevsamp	La deviazione standard campionaria nei valori raccolti nel periodo di valutazione definito.
sum	La somma dei valori raccolti nel periodo di valutazione definito.
sumofsquares	La somma dei quadrati nei valori raccolti nel periodo di valutazione definito.
varpop	La varianza della popolazione dei valori raccolti nel periodo di valutazione definito.
varsamp	La varianza campionaria dei valori raccolti nel periodo di valutazione definito.

Parametri comuni

- `/host/key` è un primo parametro comune obbligatorio per le funzioni che fanno riferimento alla cronologia degli item del host
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia degli item del host, dove:
  - **sec** - massimo periodo di valutazione in secondi (è possibile usare i suffissi di tempo), oppure
  - **#num** - massimo intervallo di valutazione negli ultimi valori raccolti (se preceduto da un cancelletto)
  - **time shift** (opzionale) consente di spostare indietro nel tempo il punto di valutazione. Vedi maggiori 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) #il valore medio dell'ultima ora fino a questo momento
avg(/host/key,1h:now-1d) #il valore medio di un'ora compresa tra 25 ore fa e 24 ore fa rispetto a ora
avg(/host/key,#5) #il valore medio degli ultimi cinque valori
avg(/host/key,#5:now-1d) #il valore medio degli ultimi cinque valori, escludendo i valori ricevuti nelle u
```

bucket_percentile(item filter,time period,percentage)

Calcola il percentile dai bucket di un istogramma.

Parametri:

- **item filter** - vedi item filter;
- **time period** - vedi 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))`.

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. Vedere funzioni foreach per i dettagli. Si noti che count_foreach e bucket_rate_foreach supportano parametri aggiuntivi.
- **item filter** - vedere item filter;
- **time period** - vedere time period;
- **operator** (deve essere tra doppi apici). `operator` supportati:
eq - uguale
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like - corrisponde se contiene il pattern (con distinzione tra maiuscole e minuscole)
bitand - AND bit a bit
regexp - corrispondenza con distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza senza distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra doppi apici); supportato se *operator* è specificato nel terzo parametro.

Commenti:

- L'uso di **count()** con una funzione foreach correlata alla cronologia (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 tale effetto.
- I parametri facoltativi *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" 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 terzo parametro, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). Nel caso delle espressioni regolari globali, la distinzione tra maiuscole e 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 i numeri grandi, la differenza tra la rappresentazione decimale (memorizzata nel database) e quella binaria (utilizzata da Zabbix server) può influire sulla quarta cifra decimale.

Esempi:

```
count(max_foreach(/*/net.if.in[*],1h)) #il numero di item net.if.in che hanno ricevuto dati nell'ultima or
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #il numero di file system con oltre il 95% dello spaz
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Calcola il φ-quantile 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. Vedere item filter per maggiori dettagli.

Commenti:

- Supportato solo negli item calcolati;
- Funziona come alias per la funzione *count(exists_foreach(item_filter))*.

Esempi:

```
item_count(/*/agent.ping?[group="Host group 1"]) #il numero di host con l'item *agent.ping* nel "Host grou
```

kurtosis(/host/key,(sec|#num)<:time shift>)

La "pesantezza delle code" 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.

Esempio:

```
kurtosis(/host/key,1h) #kurtosis per l'ultima ora fino ad ora
```

mad(/host/key,(sec|#num)<:time shift>)

La deviazione assoluta mediana dei valori raccolti nel periodo di valutazione definito. Vedi anche: Median absolute deviation.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempio:

```
mad(/host/key,1h) #deviazione assoluta mediana per l'ultima ora fino ad ora
```

max(/host/key,(sec|#num)<:time shift>)

Il valore più alto 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.

Esempio:

```
max(/host/key,1h) - min(/host/key,1h) #calcola la differenza tra i valori massimo e minimo nell'ultima ora
```

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.

Esempio:

```
max(/host/key,1h) - min(/host/key,1h) #calcola la differenza tra i valori massimo e minimo nell'ultima ora
```

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.

Esempio:

```
skewness(/host/key,1h) #l'asimmetria per l'ultima ora fino ad ora
```

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 i parametri comuni.

Esempio:

```
stddevpop(/host/key,1h) #la deviazione standard della popolazione per l'ultima ora fino ad ora
```

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 i parametri comuni.

Per il funzionamento di questa funzione sono richiesti almeno due valori di dati.

Esempio:

```
stddevsamp(/host/key,1h) #la deviazione standard campionaria per l'ultima ora fino ad ora
```

sum(/host/key,(sec|#num)<:time shift>)

La somma dei valori raccolti 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.

Esempio:

```
sum(/host/key,1h) #la somma dei valori dell'ultima ora fino ad ora
```

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.

Esempio:

```
sumofsquares(/host/key,1h) #la somma dei quadrati per l'ultima ora fino ad ora
```

varpop(/host/key,(sec|#num)<:time shift>)

La varianza della popolazione dei valori raccolti entro il periodo di valutazione definito. Vedi anche: Varianza.
 Tipi di valore supportati: *Float*, *Integer*.
 Funzione foreach supportata: *last_foreach*.

Parametri: vedere i parametri comuni.

Esempio:

```
varpop(/host/key,1h) #la varianza della popolazione per l'ultima ora fino ad ora
```

varsamp(/host/key,(sec|#num)<:time shift>)

La varianza campionaria dei valori raccolti entro il periodo di valutazione definito. Vedi anche: Variance.
 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.

Esempio:

```
varsamp(/host/key,1h) #la varianza campionaria per l'ultima ora fino ad ora
```

Vedere tutte le funzioni supportate.

Funzioni foreach

Panoramica

Le funzioni foreach sono utilizzate nei calcoli aggregati per restituire un valore aggregato per ciascun item selezionato dal **filtro item** utilizzato. Viene restituito un array di valori.

> **Attention:**
> Le funzioni foreach sono supportate solo all'interno dei Calculated item come parte delle formule di calcolo aggregato. Non possono essere richiamate in semplici chiavi item o in espressioni trigger al di fuori di questo contesto.

Ad esempio, la funzione *avg_foreach* restituirà un array di valori, in cui ciascun valore è il valore storico *medio* dell'item selezionato, durante l'intervallo di tempo specificato.

Il filtro item fa parte della sintassi utilizzata dalle funzioni foreach. Nel filtro item è supportato l'uso dei caratteri jolly, pertanto gli item richiesti possono essere selezionati in modo piuttosto 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 il "limite superiore del bucket" è il valore del parametro della chiave item definito dal parametro <parameter number>.
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 secondo parametro consente di specificare il periodo di tempo per l'aggregazione. Il periodo di tempo può essere espresso solo come tempo; la quantità di valori (preceduta da **#**) non è supportata.

In questo parametro, per comodità, è possibile utilizzare i simboli di unità supportati, 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 da Zabbix 7.0), che 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

Un terzo parametro opzionale è supportato dalla funzione *bucket_rate_foreach*:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

dove <parameter number> è la posizione del valore "bucket" nella chiave item. Ad esempio, se il valore "bucket" in myItem[aaa,0.2] è '0.2', allora la sua posizione è 2.

Il valore predefinito di <parameter number> è '1'.

count_foreach

Il terzo e il quarto parametro opzionale sono supportati dalla funzione *count_foreach*:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Dove:

- **operator** è l'operatore condizionale per i valori dell'item (deve essere tra doppi apici). `operatori` supportati:
eq - uguale
ne - diverso
gt - maggiore
ge - maggiore o uguale
lt - minore
le - minore o uguale
like - corrisponde se contiene il pattern (con distinzione tra maiuscole e minuscole)
bitand - AND bit a bit
regexp - corrispondenza con distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza senza distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
- **pattern** è il pattern richiesto (gli argomenti stringa devono essere tra doppi apici); supportato se *operator* è specificato nel terzo parametro.

Commenti:

- I parametri opzionali *operator* o *pattern* non possono essere lasciati vuoti dopo una virgola, possono solo essere 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" 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 terzo parametro, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). Nel caso di espressioni regolari globali, la distinzione tra maiuscole e minuscole viene ereditata dalle impostazioni delle espressioni regolari globali. 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 (utilizzata da Zabbix server) può influire sulla quarta cifra decimale.

Vedere 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

Function	Host disabilitato	Host non disponibile con dati	Host non disponibile senza dati	Item disabilitato	Item non supportato	Errore di recupero dati (SQL)
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.

2 Funzioni bit a bit

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
bitand	Il valore dell'"AND bit a bit" tra il valore di un item e la maschera.
bitlshift	Lo spostamento a sinistra bit a bit del valore di un item.
bitnot	Il valore del "NOT bit a bit" del valore di un item.
bitor	Il valore dell'"OR bit a bit" tra il valore di un item e la maschera.
bitrshift	Lo spostamento a destra bit a bit del valore di un item.
bitxor	Il valore dell'"OR esclusivo bit a bit" tra il valore di un item e la 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.

3 Funzioni di data e ora

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Attention:
Le funzioni di data e ora non possono essere utilizzate da sole nell'espressione; nell'espressione deve essere inclusa almeno una funzione di un altro gruppo che faccia riferimento all'item del host (ad eccezione della funzione nodata()). Per informazioni dettagliate su come le funzioni di data e ora operano all'interno delle espressioni, vedere Tempo di calcolo.

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

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.

4 Funzioni della cronologia

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
change	L'entità della differenza tra il valore precedente e l'ultimo valore.
changecount	Il numero di modifiche 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.
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 dell'N-esimo valore item di log 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 l'assenza di dati ricevuti.
percentile	Il percentile P-esimo di un periodo, dove P (percentuale) è specificato dal terzo parametro.
rate	Il tasso medio al secondo dell'aumento di un contatore monotonicamente crescente nel periodo di tempo definito.

Parametri comuni

- `/host/key` è un primo parametro comune obbligatorio per le funzioni che fanno riferimento alla cronologia degli item del host
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia degli item del host, dove:
  - **sec** - periodo di valutazione massimo in secondi (è possibile usare i suffissi di tempo), oppure
  - **#num** - intervallo di valutazione massimo negli ultimi valori raccolti (se preceduto da un cancelletto)
  - **time shift** (opzionale) consente di spostare indietro nel tempo il punto di valutazione. Vedi maggiori 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 entro il periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere i 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 (con distinzione tra maiuscole e minuscole)
bitand - AND

bit a bit
regexp - corrispondenza con distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza senza distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`

- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra virgolette doppie)

Commenti:

- I valori 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 i valori string, text e log sono supportati solo gli operator *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**. 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 conteggiato.
Se *number_to_compare_with* e *mask* sono uguali, è necessario specificare solo *mask* (senza '/').
- Con *regexp* o *iregexp* come operator, il quarto parametro `pattern` può essere un'espressione regolare ordinaria o globale (che inizia con '@'). Nel caso di espressioni regolari globali, la distinzione tra maiuscole e 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 (utilizzata da Zabbix server) può influire sulla 4ª cifra decimale.

Esempi:

```
count(/host/icmpping,30m,,"0")>5 #il ping è fallito più di 5 volte in 30 minuti
count(/host/key,10m,"like","error") #il numero di valori negli ultimi 10 minuti fino ad ora che contengono
count(/host/key,10m,,12) #il numero di valori negli ultimi 10 minuti fino ad ora uguali a '12'
count(/host/key,10m,"gt",12) #il numero di valori negli ultimi 10 minuti fino ad ora maggiori di '12'
count(/host/key,#10,"gt",12) #il numero di valori entro gli ultimi 10 valori fino ad ora maggiori di '12'
count(/host/key,10m:now-1d,"gt",12) #il numero di valori tra 24 ore e 10 minuti fa e 24 ore fa da adesso
count(/host/key,10m,"bitand","6/7") #il numero di valori negli ultimi 10 minuti fino ad ora che hanno '110
count(/host/key,10m:now-1d) #il numero di valori tra 24 ore e 10 minuti fa e 24 ore fa da adesso
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Il numero di valori univoci nel periodo di valutazione definito.
 Tipi di valore supportati: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Vedere i parametri comuni
- **operator** (deve essere tra doppi apici). `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 (con distinzione tra maiuscole e minuscole)
bitand - AND bit a bit
regexp - corrispondenza con distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza senza distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere tra doppi apici)

Commenti:

- I 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 operator *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 un'espressione regolare normale o globale (che inizia con '@'). Nel caso di espressioni regolari globali, la distinzione tra maiuscole e minuscole viene ereditata dalle impostazioni globali delle espressioni regolari. 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 (utilizzata da Zabbix server) può influire sulla quarta cifra decimale.

Esempi:

```
countunique(/host/key,10m) #il numero di valori univoci negli ultimi 10 minuti fino ad ora
countunique(/host/key,10m,"like","error") #il numero di valori univoci negli ultimi 10 minuti fino ad ora
countunique(/host/key,10m,,12) #il numero di valori univoci negli ultimi 10 minuti fino ad ora uguali a '1
countunique(/host/key,10m,"gt",12) #il numero di valori univoci negli ultimi 10 minuti fino ad ora maggior
```

```
countunique(/host/key,#10,"gt",12) #il numero di valori univoci negli ultimi 10 valori fino ad ora maggior
countunique(/host/key,10m:now-1d,"gt",12) #il numero di valori univoci tra 24 ore e 10 minuti fa e 24 ore
countunique(/host/key,10m,"bitand","6/7") #il numero di valori univoci negli ultimi 10 minuti fino ad ora
countunique(/host/key,10m:now-1d) #il numero di valori univoci tra 24 ore e 10 minuti fa e 24 ore fa rispe
```

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:

- Vedere i parametri comuni
- **sec** o **#num** (opzionale) - se non specificato, per impostazione predefinita viene usato il valore più recente
- **operator** (deve essere racchiuso tra doppi apici) `operator` supportati:
eq - uguale (predefinito per integer, float)
ne - diverso
gt - maggiore di
ge - maggiore o uguale
lt - minore di
le - minore o uguale
like (predefinito per string, text, log) - corrisponde se contiene la stringa fornita in `pattern` (con distinzione tra maiuscole e minuscole)
bitand - AND bit a bit
regexp - corrispondenza con distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
iregexp - corrispondenza senza distinzione tra maiuscole e minuscole dell'espressione regolare fornita in `pattern`
- **pattern** - il pattern richiesto (gli argomenti stringa devono essere racchiusi tra doppi apici); espressione regolare Perl Compatible Regular Expression (PCRE) se `operator` è *regexp*, *iregexp*

Commenti:

- Se viene elaborato più di un valore, viene restituito '1' se è presente 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 i 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 '@'). Nel caso di espressioni regolari globali, la distinzione tra maiuscole e 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 ad or
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 rilevato un errore interno del ser
```

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)

Controlla 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 del passive item (come timestamp) e il timestamp del server/proxy Zabbix (l'orologio della raccolta del valore) è minore o uguale a *sec* secondi; 0 - altrimenti.

Parametri:

- Vedi parametri comuni.

Commenti:

- Solitamente utilizzata 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; in Zabbix agent 2 può essere configurato come controllo attivo.
- Può essere utilizzata anche con la chiave `vfs.file.time[/path/file,modify]` per verificare che il file non sia stato aggiornato per molto tempo.
- Questa funzione non è consigliata nelle espressioni di 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.localtime,6 or last(/Host/trap)<>0`.

Esempi:

```
fuzzytime(/host/system.localtime,5m)=0 #l'ora locale del client differisce dall'ora del server/proxy Zabbi
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #l'ora locale del client d
```

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>)

Controlla 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, nello stile Perl Compatible Regular Expression (PCRE) (gli argomenti stringa devono essere racchiusi tra doppi apici)

Esempi:

```
logeventid(/host/eventlog[Security],,"4625")=1 #a log entry with ID matching "4625" (failed authentication
logeventid(/host/eventlog[System],,"6008|41")=1 #a log entry with ID matching "6008" or "41" found
```

logseverity(/host/key,<#num<:time shift»)

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 l'origine del log dell'ultima voce di log corrisponde a un'espressione regolare.
 Tipi di valore supportati: *Log*.
 Restituisce: 0 - non corrisponde; 1 - corrisponde.

Parametri:

- Vedere parametri comuni
- **#num** (opzionale) - l'N-esimo valore più recente
- **pattern** (opzionale) - l'espressione regolare che descrive il modello richiesto, nello stile Perl Compatible Regular Expression (PCRE) (gli argomenti stringa devono essere racchiusi tra doppi apici)

Normalmente utilizzato per i log eventi di Windows.

Esempi:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #trovata una voce di log con origine corrispondent
logsource(/host/eventlog[System],,"Service Control Manager")=1 #trovata una voce di log con origine corris
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

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 che non siano stati ricevuti dati.
 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 perché il processo history syncer calcola questa funzione solo ogni 30 secondi; `nodata(/host/key,0)` non è consentito
- **mode** - se impostato su *strict* (tra doppi apici), questa funzione sarà insensibile 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 salteranno i dati per il periodo di ritardo. Si noti che per i proxy passivi la soppressione viene attivata se la connessione viene ripristinata più di 15 secondi e non meno di 2 secondi dopo. Per i proxy attivi la soppressione viene attivata se la connessione viene ripristinata più di 15 secondi dopo. 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 mostrerà un errore se, entro il periodo del 1° parametro:
- non ci sono dati e Zabbix server 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 sono visualizzati nella colonna *Info* nella configurazione dei trigger.
- Questa funzione potrebbe non funzionare correttamente se ci sono differenze di orario tra Zabbix server, proxy e agent. Vedere anche: Requisito di sincronizzazione dell'ora.

Esempio:

```
nodata(/host/agent.ping,5m)=1 #trigger if no data has been received from Zabbix agent for 5 minutes
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

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
```

Vedere tutte le funzioni supportate.

5 Funzioni di trend

Le funzioni di trend, a differenza delle funzioni di history, utilizzano i dati di trend per i calcoli.

I trend memorizzano valori aggregati orari. Le funzioni di trend utilizzano queste medie orarie e sono quindi utili per analisi a lungo termine.

I risultati delle funzioni di trend vengono memorizzati nella cache, quindi chiamate multiple alla stessa funzione con gli stessi parametri recuperano le informazioni dal database una sola volta. La cache delle funzioni di trend è controllata dal parametro del server TrendFunctionCacheSize.

I trigger che fanno riferimento **solo** a funzioni di trend vengono valutati una volta per il periodo di tempo più piccolo nell'espressione. Per 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 di trend sia di history (o funzioni di data e ora e/o nodata()), viene calcolato in conformità ai principi usuali.

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
baselinedev	Restituisce il numero di deviazioni (secondo l'algoritmo stddevpop) tra l'ultimo periodo di dati e gli stessi periodi di dati nelle stagioni precedenti.
baselinewma	Calcola la baseline facendo la media dei dati dello stesso intervallo temporale in più periodi di tempo uguali ("stagioni") utilizzando l'algoritmo della media mobile pesata.
trendavg	La media dei valori di trend nel periodo di tempo definito.
trendcount	Il numero di valori di history recuperati con successo utilizzati per calcolare il valore di trend nel periodo di tempo definito.
trendmax	Il massimo dei valori di trend nel periodo di tempo definito.
trendmin	Il minimo dei valori di trend nel periodo di tempo definito.
trendstl	Restituisce il tasso di anomalie durante il periodo di rilevamento: un valore decimale compreso tra 0 e 1 pari a `((numero di valori anomali)/(numero totale di valori))`.
trendsum	La somma dei valori di trend nel 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.

6 Funzioni matematiche

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni matematiche sono supportate con i tipi di valore float e integer, salvo diversa indicazione.

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
abs	Il valore assoluto di un valore.
acos	L'arcocoseno 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 ascisse specificate come angolo, espressa in radianti.
avg	Il valore medio dei valori dell'item di riferimento.
cbrt	La radice cubica di un valore.
ceil	Arrotonda il valore per eccesso all'intero più vicino maggiore o uguale.
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).
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 più vicino minore o uguale.
log	Il logaritmo naturale.
log10	Il logaritmo decimale.
max	Il valore più alto dei valori dell'item di riferimento.
min	Il valore più basso dei valori dell'item di riferimento.
mod	Il resto della divisione.
pi	La costante Pi greco (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 alle posizioni 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 di riferimento.
tan	La tangente di un valore.

Function	Description
truncate	Tronca il valore alle posizioni 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.

7 Funzioni operatore

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- 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.

8 Funzioni predittive

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

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 obbligatorio comune per le funzioni che fanno riferimento alla cronologia degli item del host
- `(sec|#num)<:time shift>` è un secondo parametro comune per le funzioni che fanno riferimento alla cronologia degli item del host, dove:
  - **sec** - massimo periodo di valutazione in secondi (è possibile usare i suffissi di tempo), oppure
  - **#num** - massimo intervallo di valutazione negli ultimi valori raccolti (se preceduto da un cancelletto)
  - **time shift** (opzionale) consente di spostare il punto di valutazione indietro nel tempo. Vedi maggiori 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 (è possibile usare suffissi temporali); sono supportati valori negativi;
- **fit** (opzionale; deve essere tra virgolette doppie) - la funzione usata 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 di potenza
Si noti che *polynomial1* è equivalente a *linear*;
- **mode** (opzionale; deve essere tra virgolette doppie) - l'output richiesto. Modalità supportate:
value - valore (predefinito)
max - massimo
min - minimo
delta - *max-min*
avg - media
Si noti che *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 unsupported solo se usata in modo errato nell'espressione (tipo di item errato, parametri non validi), altrimenti restituisce -1 in caso di errori;
- Vedere anche le informazioni aggiuntive sulle funzioni predittive dei trigger.

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 a un'ora di dati di un g
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 20 minuti fa in base agli ultimi due valori (questo
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Il tempo in secondi necessario affinché un item raggiunga la soglia specificata.
 Tipi di valore supportati: *Float*, *Integer*.

Parametri:

- Vedi parametri comuni;
- **threshold** - il valore da raggiungere (è possibile usare i suffissi di unità);
- **fit** (opzionale; deve essere tra virgolette doppie) - vedi 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 la soglia non può essere raggiunta;
- Diventa unsupported solo se usata in modo errato nell'espressione (tipo di item errato, parametri non validi), altrimenti restituisce -1 in caso di errori;
- Vedi anche le informazioni aggiuntive sulle funzioni predittive dei trigger.

Esempi:

```
timeleft(/host/key,#10,0) #il tempo fino a quando il valore dell'item raggiunge zero in base agli ultimi 1
timeleft(/host/key,1h,100) #il tempo fino a quando il valore dell'item raggiunge 100 in base ai dati dell'
timeleft(/host/key,1h:now-1d,100) #il tempo fino a quando il valore dell'item raggiunge 100 in base a un'o
timeleft(/host/key,1h,200,"polynomial2") #il tempo fino a quando il valore dell'item raggiunge 200 in base
```

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 fino a quanto 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 corrente.

Soglia da raggiungere

(solo **timeleft**)
Il parametro `threshold` specifica un valore che l'item analizzato deve raggiungere, indipendentemente dal fatto che lo raggiunga dall'alto o dal basso. Una volta determinata f(t) (vedi sotto), si deve risolvere l'equazione f(t) = `threshold` e restituire la radice più vicina al momento attuale e situata a destra del momento attuale oppure 1.7976931348623158E+308 se non esiste una tale radice.

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 usare 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, anche 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 adesso 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	$\max_{now <= t <= now + time} f(t)$
min	$\min_{now <= t <= now + time} f(t)$
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)`

9 Funzioni stringa

Tutte le funzioni elencate qui sono supportate in:

- Espressioni dei trigger
- Item calcolati
- Macro di espressione

Le funzioni sono elencate senza informazioni aggiuntive. Fare clic sulla funzione per visualizzare tutti i dettagli.

Function	Description
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 di valori costanti.
insert	Inserisce caratteri o spazi specificati nella stringa di caratteri a partire dalla posizione specificata nella stringa.

Function	Description
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'.
repeat	Ripete una stringa.
replace	Trova il modello nel valore e lo sostituisce con il valore di sostituzione.
right	Restituisce i caratteri più a destra del valore.
rtrim	Rimuove i caratteri specificati dalla fine della stringa.
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])
```

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'.
 Tipi di valore supportati: *String*, *Text*, *Log*.

Parametri:

- **value** - il valore da controllare;
- **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
```

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.

4 Best practice per i trigger

I trigger sono uno strumento potente, ma possono anche creare rumore di avvisi indesiderato. Per vedere più segnali reali e meno rumore, segui questi suggerimenti:

1. Riduci la sensibilità dei trigger. Invece di generare un avviso sull'ultimo valore (troppo alto/troppo basso), analizza la media su un periodo più lungo utilizzando funzioni come **avg**, **min** e **max**.
2. Valuta l'uso della funzione **percentile** (impostata al 95% o al 5%) nei trigger, se vuoi evitare avvisi dovuti a picchi e cali casuali.
3. Usa l'isteresi per evitare il flapping dei trigger, ovvero frequenti cambiamenti dello stato del trigger (OK ↔ Problem). È possibile definire una continuità per lo stato di problema aggiungendo un'espressione di ripristino (una condizione separata per la risoluzione del problema).
4. Usa le dipendenze dei trigger per evitare avvisi non correlati alla causa principale.
5. Usa la gravità dei trigger per generare avvisi solo per i 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 va giù. Con trigger configurati per entrambi, potresti ricevere notifiche su due host non disponibili, mentre in realtà il responsabile è solo il router.

È qui che una qualche dipendenza tra host può risultare utile. Con una dipendenza impostata, le notifiche dei dipendenti possono essere trattenute e può essere inviata solo la notifica del problema principale.

Sebbene Zabbix non supporti direttamente le dipendenze tra host, esse possono essere definite con un altro metodo, più flessibile: le dipendenze tra trigger. Un trigger può avere uno o più trigger da cui dipende.

Quindi, nel nostro semplice esempio, apriamo il modulo di configurazione del trigger del server e impostiamo che dipenda dal rispettivo trigger del router. Con questa dipendenza, il trigger del server non cambierà stato finché il trigger da cui dipende si trova nello stato 'PROBLEM'; pertanto non verranno eseguite azioni dipendenti e non verranno inviate notifiche.

Se sia il server sia il router non sono disponibili ed è presente una dipendenza, Zabbix non eseguirà azioni per il trigger dipendente.

Mentre il trigger padre si trova nello stato PROBLEM, i suoi dipendenti possono segnalare valori non affidabili. Pertanto i trigger dipendenti non verranno rivalutati finché il trigger padre (il router nell'esempio sopra):

- non torna dallo stato 'PROBLEM' allo stato 'OK';
- non cambia 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 da un valore di un item non coinvolto nel trigger dipendente;
- non viene disabilitato, oppure non ha un item disabilitato o un host dell'item disabilitato

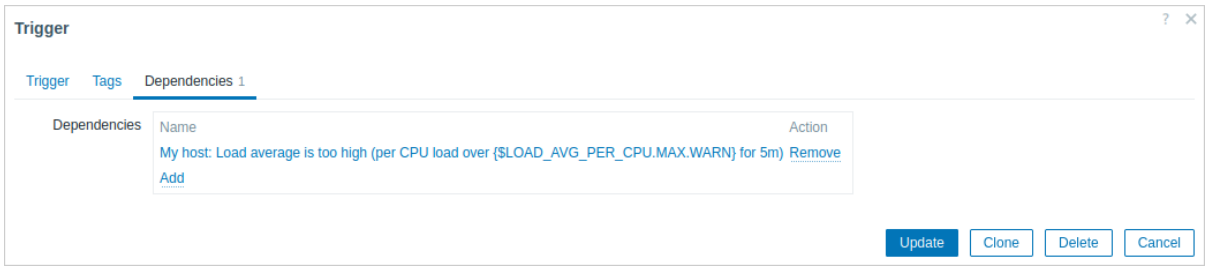
In tutti i casi sopra menzionati, il trigger dipendente (server) verrà rivalutato solo quando verrà ricevuta una nuova metrica per esso. Ciò significa che il trigger dipendente potrebbe non essere aggiornato immediatamente.

Inoltre:

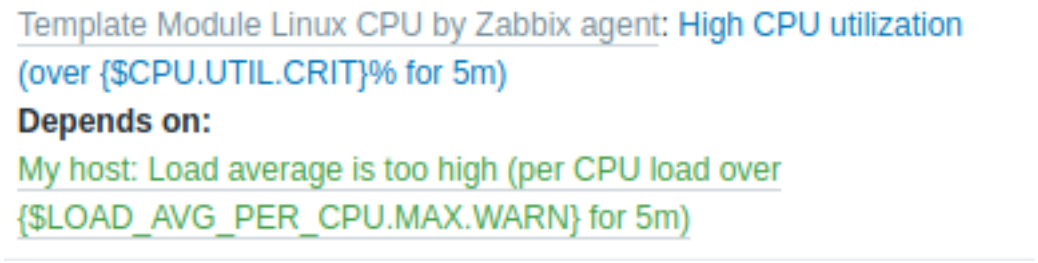
- La dipendenza tra trigger può essere aggiunta da qualsiasi trigger di host a qualsiasi altro trigger di host, purché ciò non comporti una dipendenza circolare.
- La dipendenza tra trigger può essere aggiunta da un template a un altro. Se un trigger del template A dipende da un trigger del template 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 a un host (o a un altro template) da solo.
- La 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 di host che dipende dallo stesso template di trigger da cui il trigger dipendeva. Questo 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 quello specifico router.
- La dipendenza tra trigger non può essere aggiunta da un trigger di host a un trigger di template.
- La 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) oppure a un trigger reale. Un prototipo di trigger non può dipendere da un prototipo di trigger di una regola LLD diversa né da un trigger creato da un prototipo di trigger. Un prototipo di trigger di host non può dipendere da un trigger di un template.

Configurazione

Per definire una dipendenza, aprire la scheda Dependencies nel modulo di configurazione del trigger. Fare clic su *Add* nel blocco 'Dependencies' e selezionare uno o più trigger da cui il trigger dipenderà.



Fare clic su *Update*. Ora il trigger mostra l'indicazione della sua dipendenza nell'elenco.



Esempio di più dipendenze

Per esempio, l'host si trova dietro Router2 e Router2 si trova dietro Router1.

```
Zabbix - Router1 - Router2 - Host
```

Se Router1 non è raggiungibile, allora ovviamente anche l'host e Router2 non sono raggiungibili, tuttavia ricevere tre notifiche relative al fatto che l'host, Router1 e Router2 siano tutti non raggiungibili 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 controllerà le corrispondenti dipendenze del trigger. Se ne vengono trovate e uno di questi trigger si trova nello stato `Problem`, allora lo stato del trigger non verrà modificato, le azioni non verranno eseguite e non verranno inviate notifiche.

Zabbix esegue questo controllo 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.

4 Eventi

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 registrati automaticamente dal server
- eventi interni - quando un item/regola di low-level discovery diventa non supportato oppure un trigger passa in uno stato sconosciuto

Gli eventi sono contrassegnati temporalmente e possono costituire la base per azioni come l'invio di email di notifica, ecc.

Per visualizzare i dettagli degli eventi nel frontend, andare su *Monitoring → Problems*. Qui è possibile fare clic sulla data e sull'ora dell'evento per visualizzarne i dettagli.

Ulteriori informazioni sono disponibili in:

- eventi di trigger
- altre sorgenti di eventi

1 Generazione di eventi trigger

Panoramica

Il cambiamento dello stato di un 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 del cambiamento dello stato del trigger: quando è avvenuto e quale sia il nuovo stato.

Dai trigger vengono creati due tipi di eventi: `Problem` e `OK`.

Eventi di problema

Un evento di problema viene creato:

- quando un'espressione del trigger viene valutata come TRUE se il trigger è nello stato OK;
- ogni volta che un'espressione del trigger viene valutata come TRUE se per il trigger è abilitata la generazione di eventi di problema multipli.

Eventi OK

Un evento OK chiude i relativi eventi di problema e può essere creato da 3 componenti:

- trigger - in base alle impostazioni di 'Generazione evento OK' e 'Chiusura evento OK';
- correlazione degli eventi
- task manager – quando un evento viene chiuso manualmente

Trigger

I trigger hanno un'impostazione "Generazione evento OK" che controlla come vengono generati gli eventi OK:

- *Espressione* - un evento OK viene generato per un trigger nello stato di problema quando la sua espressione viene valutata come FALSE. Questa è l'impostazione più semplice, abilitata per impostazione predefinita.
- *Espressione di ripristino* - un evento OK viene generato per un trigger nello stato di problema quando la sua espressione viene valutata come FALSE e l'espressione di ripristino viene valutata come TRUE. Questo può essere utilizzato se i criteri di ripristino del trigger sono diversi dai criteri del problema.
- *Nessuno* - un evento OK non viene mai generato. Questo può essere utilizzato insieme alla generazione di eventi di problema multipli per inviare semplicemente una notifica quando accade qualcosa.

Inoltre, i trigger hanno un'impostazione "Chiusura da evento OK" che controlla quali eventi di problema vengono chiusi:

- *Tutti i problemi* - un evento OK chiuderà tutti i problemi aperti creati dal trigger
- *Tutti i problemi se i valori dei tag corrispondono* - un evento OK chiuderà i problemi aperti creati dal trigger che hanno almeno un valore di tag corrispondente. Il tag è definito dall'impostazione del trigger "Tag per la corrispondenza". Se non ci sono eventi di problema 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 (detta anche correlazione globale degli eventi) è un modo per configurare regole personalizzate di chiusura degli eventi (che comportano la generazione di eventi OK).

Le regole definiscono come i nuovi eventi di problema vengono associati agli eventi di problema esistenti e consentono di chiudere il nuovo evento o gli eventi corrispondenti generando i relativi eventi OK.

Tuttavia, la correlazione degli eventi deve essere configurata con molta attenzione, poiché può influire negativamente sulle prestazioni dell'elaborazione degli eventi oppure, se configurata in modo errato, chiudere più eventi del previsto (nel caso peggiore potrebbero essere chiusi persino tutti gli eventi di problema). Alcuni suggerimenti per la configurazione:

1. ridurre sempre l'ambito della correlazione impostando un tag univoco per l'evento di controllo (l'evento che viene associato ai vecchi eventi) e utilizzare la condizione di correlazione 'new event tag'
2. non dimenticare di aggiungere una condizione basata sul vecchio evento quando si utilizza l'operazione 'close old event', altrimenti tutti i problemi esistenti potrebbero essere chiusi
3. evitare di utilizzare nomi di tag comuni usati da diverse configurazioni di correlazione

Gestore attività

Se l'impostazione 'Allow manual close' è abilitata per il trigger, allora è possibile chiudere manualmente gli eventi di problema generati dal trigger. Questo viene fatto nel frontend quando si aggiorna un problema. L'evento non viene chiuso direttamente, ma viene invece creata un'attività di 'chiusura evento', che viene gestita poco dopo dal gestore attività. Il gestore attività genererà un corrispondente evento OK e l'evento di problema verrà chiuso.

2 Altre fonti di eventi

Eventi di servizio

Gli eventi di servizio vengono generati solo se le azioni di 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 si è verificata e quale sia il nuovo stato.

Eventi di discovery

Zabbix esegue periodicamente la scansione degli intervalli IP definiti nelle regole di network discovery. La frequenza del controllo è configurabile singolarmente per ciascuna regola. Una volta che un host o un servizio viene rilevato, 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 rilevato per la prima volta.
Service Lost	Se il servizio viene perso dopo essere stato attivo.
Host Discovered	Se l'host torna disponibile dopo un periodo di inattività o viene rilevato per la prima volta.
Host Lost	Se l'host viene perso dopo essere stato attivo.

Eventi di autoregistrazione dell'agent attivo

L'autoregistrazione dell'agent attivo crea eventi in Zabbix.

Se configurato, l'evento di autoregistrazione dell'agent attivo viene creato quando un agent attivo precedentemente sconosciuto richiede i controlli oppure se i metadati del host sono cambiati. Il server aggiunge un nuovo host autoregistrato, utilizzando 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 low-level discovery cambia stato da 'normal' a 'unsupported'
- una regola di low-level discovery cambia stato da 'unsupported' a 'normal'
- un trigger cambia stato da 'normal' a 'unknown'
- un trigger cambia stato da 'unknown' a 'normal'

Lo scopo dell'introduzione degli eventi interni è consentire agli utenti di ricevere notifiche 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 gli item che diventano unsupported), disabilitare tutte le azioni per gli eventi interni in Avvisi → Azioni → Azioni interne.

Note:
Se le azioni interne sono disabilitate, mentre un oggetto si trova nello stato 'unsupported', verrà comunque creato l'evento di ripristino per questo oggetto.

Se le azioni interne sono abilitate, mentre un oggetto si trova nello stato 'unsupported', l'evento di ripristino per questo oggetto verrà creato anche se per l'oggetto non è stato creato alcun 'problem event'.

Vedere anche: Ricezione di notifiche sugli item unsupported

3 Chiusura manuale dei problemi

Panoramica

Sebbene in genere gli eventi di problema vengano risolti automaticamente quando lo stato del trigger passa da `Problem` a `OK`, possono verificarsi 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 ottimizzati per ottenere prestazioni ottimali. In questo caso il problema viene segnalato agli amministratori Linux, che lo correggono 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. Successivamente, il processo *task manager* esegue questa attività e genera un evento OK, chiudendo così l'evento di problema.

L'evento *OK* generato include l'intero insieme di tag evento risolti per quell'evento (compresi 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 nuovamente produrre un problema:

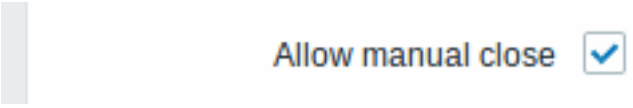
- Quando arrivano nuovi dati per qualsiasi item incluso nell'espressione del trigger (si noti che i valori scartati da una fase di preprocessing di throttling non sono considerati come ricevuti e non causeranno la rivalutazione dell'espressione del trigger);
- Quando nell'espressione vengono utilizzate le funzioni date and time e/o nodata().

Configurazione

Per chiudere manualmente un problema sono necessari due passaggi.

Configurazione del trigger

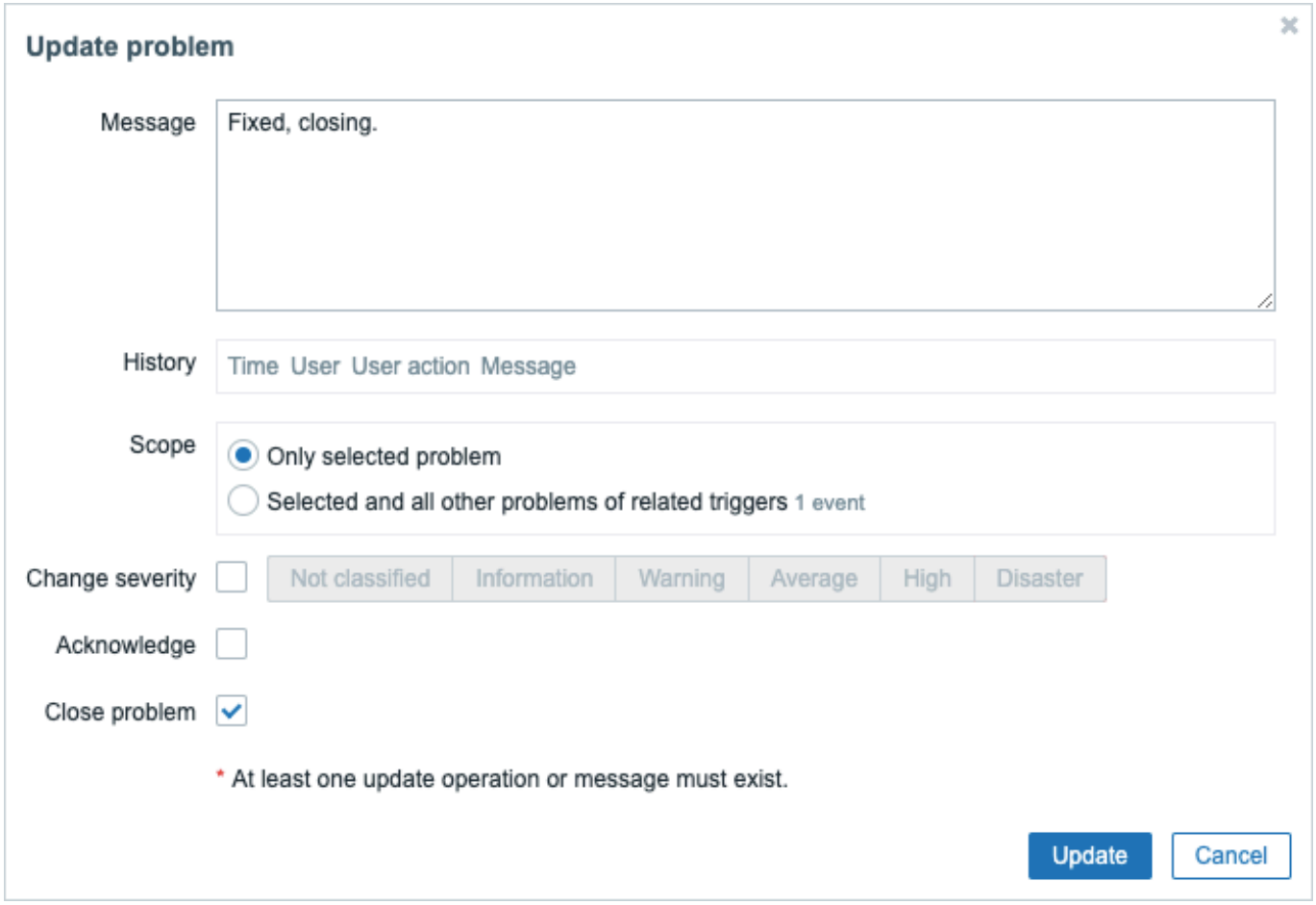
Nella configurazione del trigger, abilitare l'opzione *Consenti chiusura manuale*.



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 saranno necessari alcuni secondi per chiudere il problema. Durante questo processo, in *Monitoring* > *Problems* viene visualizzato *CLOSING* 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 questa informazione.

5 Correlazione degli eventi

Panoramica

La correlazione degli eventi consente di correlare gli eventi di problema alla loro risoluzione in modo molto preciso e flessibile.

La correlazione degli eventi può essere definita:

- a livello di trigger - un trigger può essere utilizzato per collegare problemi separati alla loro soluzione
- globalmente - i problemi possono essere correlati alla loro soluzione da un trigger/metodo di polling diverso utilizzando regole di correlazione globali

1 Correlazione di eventi basata su trigger

Panoramica

La correlazione degli eventi basata sui trigger consente di correlare problemi separati segnalati da un trigger.

Sebbene in generale un evento OK possa chiudere tutti gli eventi problema creati da un trigger, ci sono casi in cui è necessario un approccio più dettagliato. Per esempio, durante il monitoraggio dei file di log potresti voler individuare determinati problemi in un file di log e chiuderli singolarmente anziché tutti insieme.

Questo è il caso dei trigger con il parametro *PROBLEM event generation mode* impostato su *Multiple*. Tali trigger sono normalmente utilizzati per il monitoraggio dei log, l'elaborazione delle trap, ecc.

In Zabbix è possibile mettere in relazione gli eventi problema in base ai tag. I tag vengono utilizzati per estrarre valori e creare un'identificazione per gli eventi problema. Sfruttando questo, i problemi possono anche essere chiusi singolarmente in base alla corrispondenza del 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:

```
Riga1: Service 1 stopped
Riga2: Service 2 stopped
Riga3: Service 1 was restarted
Riga4: Service 2 was restarted
```

L'idea della correlazione degli eventi è poter associare l'evento di problema della Riga1 alla risoluzione della Riga3 e l'evento di problema della Riga2 alla risoluzione della Riga4, e chiudere questi problemi uno per uno:

```
Riga1: Service 1 stopped
Riga3: Service 1 was restarted #problema della Riga 1 chiuso

Riga2: Service 2 stopped
Riga4: Service 2 was restarted #problema della Riga 2 chiuso
```

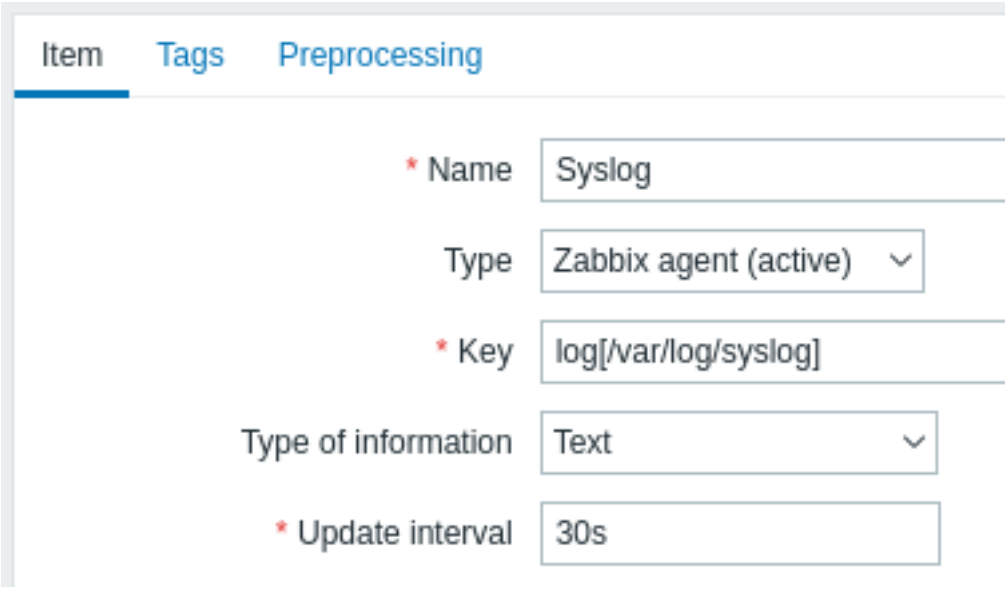
Per fare questo è necessario assegnare a questi eventi correlati dei tag come, ad esempio, "Service 1" e "Service 2". Questo può essere fatto applicando una espressione regolare alla riga di log per estrarre il valore del tag. Quindi, quando gli eventi vengono creati, ricevono rispettivamente i tag "Service 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 recepite, quindi vai a Ultimi dati per assicurarti che l'item abbia iniziato a raccogliere dati.

Trigger

Con l'item funzionante, 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 assicurarsi che ogni riga contenente la stringa ”Stopping” venga considerata un problema, impostare anche la *Modalità di generazione degli eventi di problema* nella configurazione del trigger su ‘Multiple’.

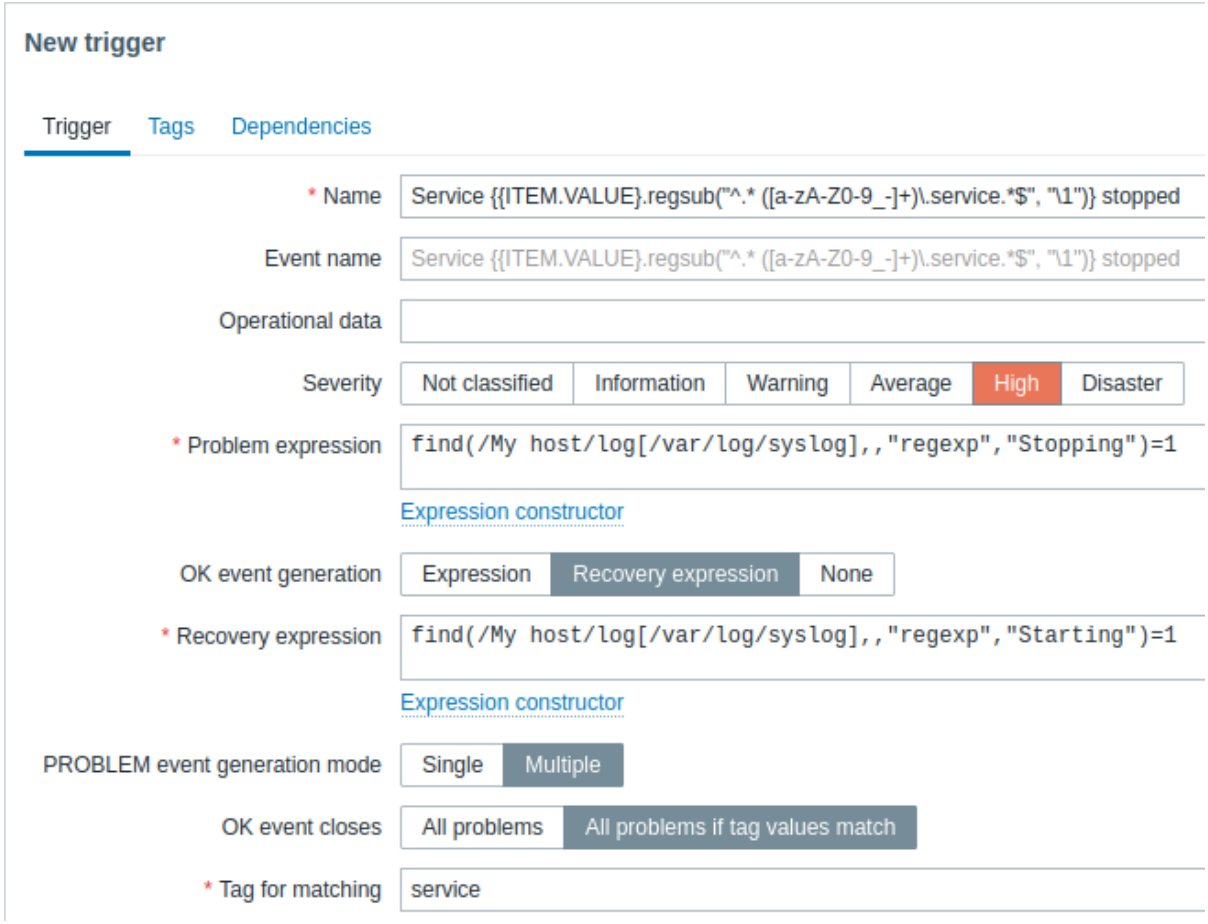
Quindi definire un’espressione di ripristino. La seguente espressione di ripristino risolverà tutti i problemi se viene trovata una riga di log contenente la stringa ”Starting”:

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

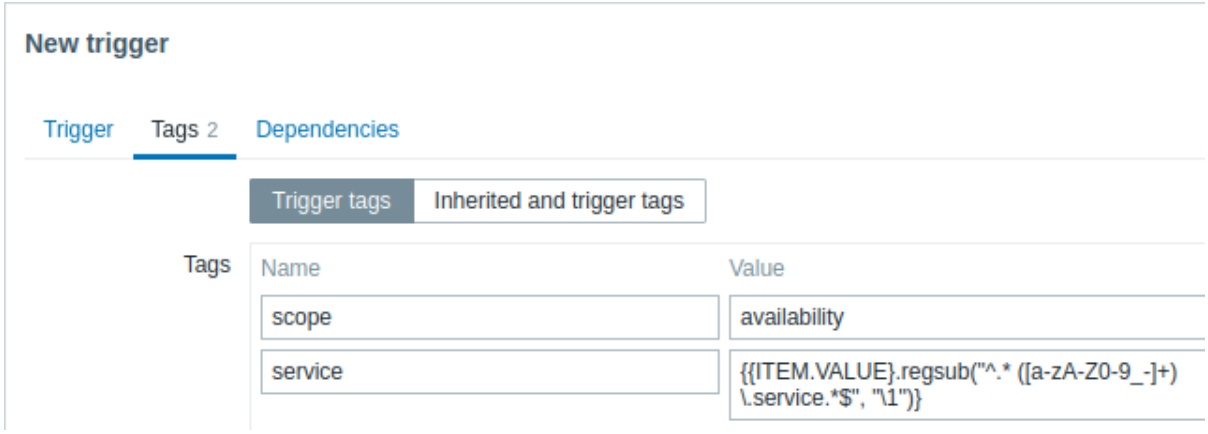
Poiché non vogliamo questo, è importante assicurarsi in qualche modo che vengano chiusi i corrispondenti problemi radice, non semplicemente tutti i problemi. È qui che il tagging può essere d’aiuto.

I problemi e le risoluzioni possono essere associati specificando un tag nella configurazione del trigger. Devono essere effettuate le seguenti impostazioni:

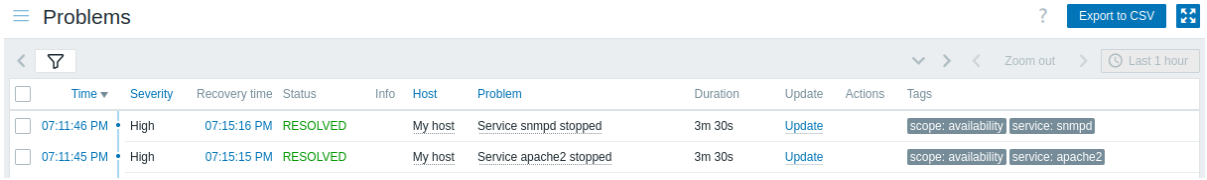
- *Modalità di generazione degli eventi di problema*: Multiple
- *L’evento OK chiude*: Tutti i problemi se i valori dei tag corrispondono
- Inserire il nome del tag per la corrispondenza degli eventi



- configurare i tag per estrarre i valori dei tag dalle righe di log



Se configurato correttamente, sarà possibile vedere gli eventi di problema etichettati per applicazione e associati alla loro risoluzione in *Monitoring* → *Problems*.



Warning:
Poiché è possibile una configurazione errata, quando possono essere creati tag evento simili per problemi **non correlati**, rivedere i casi descritti di seguito!

- Le macro indicizzate fanno sempre riferimento al campo *Expression* della configurazione del trigger, non alla *Recovery expression*. Ad esempio, in un evento di ripristino, **{ITEM.VALUE1}** verrà risolta nel 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 entro il momento del ripristino, gli eventi otterranno 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, ad esempio, del servizio A e del servizio B dalla macro {ITEM.VALUE} (ad es. 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 negli eventi OK è sufficiente per correlarli. Quindi un messaggio di ripristino del servizio A può accidentalmente chiudere un messaggio di errore del servizio 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, possono comparire successivi eventi OK con lo stesso valore di tag *UNKNOWN* che possono chiudere eventi di problema che non avrebbero dovuto chiudere.
- Se un utente usa la macro {ITEM.VALUE} senza funzioni di 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, anche questo può causare tag evento simili per problemi non correlati.

2 Correlazione globale degli eventi

Panoramica

La correlazione globale degli eventi consente di estendersi a 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 effettivamente possibile risparmiarsi 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 problema e risoluzione basata su un singolo trigger. Finora, un singolo evento di problema veniva creato da un trigger e dipendevamo da quello stesso trigger per la risoluzione del problema. Non era possibile risolvere un problema creato da un trigger con un altro trigger. Ma con la correlazione degli eventi basata sul tagging degli eventi, questo 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. Sfruttando i tag evento, è possibile contrassegnare il trigger di log con *status:down* e il trigger di polling con *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 loro la stessa operazione. E se potessimo ricevere un solo report di problema per ogni problema di porta di rete? Non è 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 riportato 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 di elaborazione degli eventi oppure, se configurata in modo errato, chiudere più eventi di quanto previsto (nel caso peggiore potrebbero essere chiusi persino tutti gli eventi di problema).

Per configurare la correlazione globale in modo **sicuro**, osservare i seguenti importanti suggerimenti:

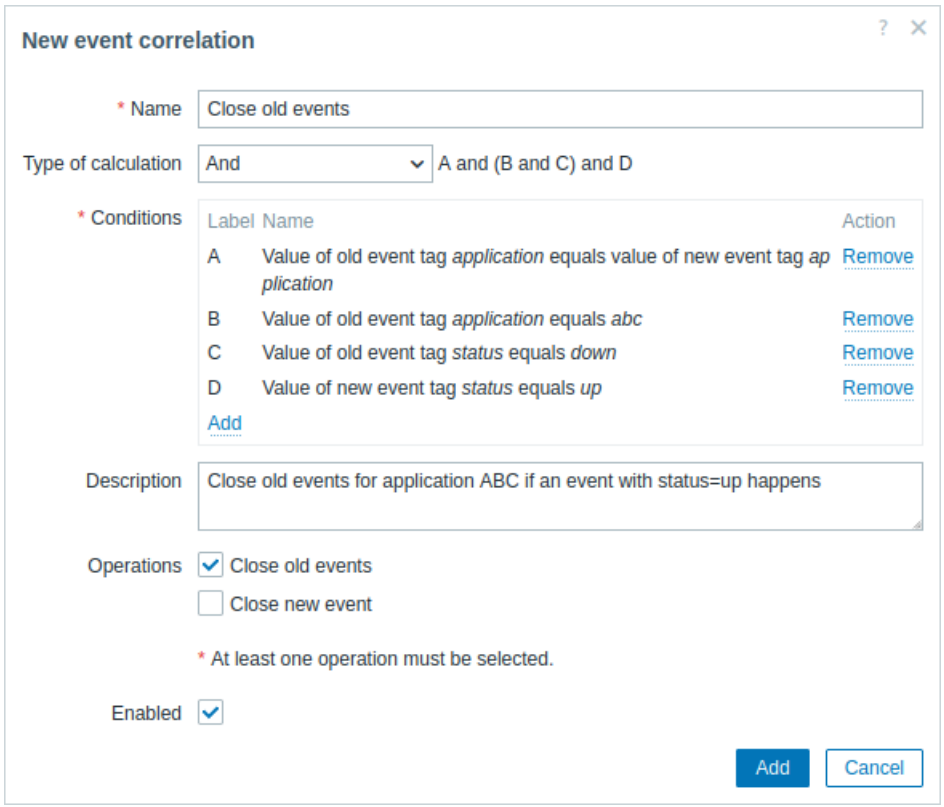
- Ridurre l'ambito della correlazione. Impostare sempre un tag univoco per il nuovo evento che viene associato ai vecchi eventi e utilizzare la condizione di correlazione *New event tag name*.
- Aggiungere una condizione esplicita per i vecchi eventi quando si utilizza **Close old events**. Aggiungere sempre almeno una condizione *Old event* (ad esempio, *Old event tag name*, *Old event tag value* oppure *Event tag pair*) quando si seleziona **Close old events** — altrimenti la regola potrebbe corrispondere e chiudere eventi di problema esistenti non correlati (nel caso peggiore, tutti i problemi). Preferire **Event tag pair** per la corrispondenza di valori di runtime (host:port, session id, ecc.) e restringere ulteriormente la corrispondenza per host o gruppo di host quando possibile.
- Evitare di utilizzare nomi di tag comuni che potrebbero finire per essere usati da diverse configurazioni di correlazione.
- Limitare il numero di regole di correlazione a quelle realmente necessarie.

Vedere anche: problemi noti.

Configurazione

Per configurare globalmente le regole di correlazione degli eventi:

- Vai a *Raccolta dati* > *Correlazione eventi*
- Fai clic su *Crea correlazione eventi* a destra (oppure sul nome della correlazione per modificare una regola esistente)
- Inserisci i parametri della regola di correlazione nel modulo

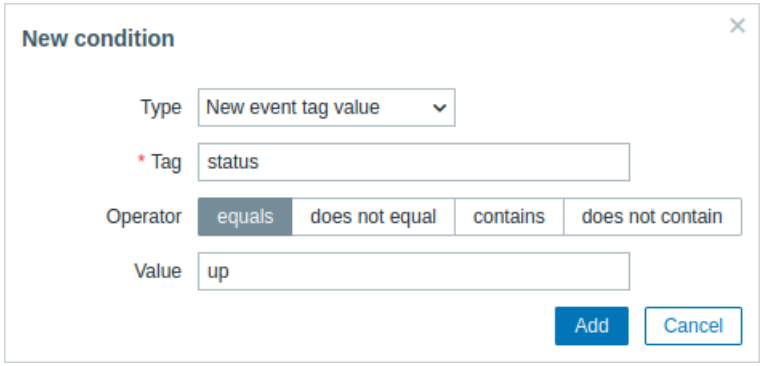


Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

Parametro	Descrizione
Nome	Nome univoco della regola di correlazione.
Tipo di calcolo	Sono disponibili le seguenti opzioni per il calcolo delle condizioni: **And** - tutte le condizioni devono essere soddisfatte **Or** - è sufficiente che sia soddisfatta una condizione **And/Or** - AND con tipi di condizione diversi e OR con lo stesso tipo di condizione **Espressione personalizzata** - 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** (sensibile a maiuscole/minuscole), **or** (sensibile a maiuscole/minuscole), **not** (sensibile a maiuscole/minuscole).

Parametro	Descrizione
Condizioni	Elenco delle condizioni. Vedi sotto per i dettagli sulla configurazione di una condizione.
Descrizione	Descrizione della regola di correlazione.
Operazioni	Seleziona la casella dell'operazione da eseguire quando l'evento viene correlato. Sono disponibili le seguenti operazioni: **Chiudi eventi vecchi** - chiude gli eventi vecchi quando si verifica un nuovo evento. Aggiungi sempre una condizione basata sul vecchio evento quando usi l'operazione *Chiudi eventi vecchi*, altrimenti tutti i problemi esistenti potrebbero essere chiusi. **Chiudi nuovo evento** - chiude il nuovo evento quando si verifica. **Attenzione!** Non lasciare vuote le condizioni del vecchio/nuovo evento quando usi *Chiudi eventi vecchi*/*Chiudi nuovo evento*. Se selezioni l'operazione *Chiudi eventi vecchi* senza aggiungere una condizione che corrisponda al vecchio evento, Zabbix potrebbe far corrispondere tutti gli eventi vecchi esistenti e chiuderli. Aggiungi sempre una condizione esplicita del vecchio evento (ad esempio, *Nome tag del vecchio evento* o *Coppia di tag evento*) quando usi *Chiudi eventi vecchi*. Ad esempio, una regola che usa solo una *Condizione del nuovo evento* e l'operazione *Chiudi eventi vecchi* corrisponderà a tutti gli eventi vecchi che soddisfano i criteri (mancanti) del vecchio evento, chiudendo di fatto i vecchi problemi.
Abilitato	Se selezioni questa casella, la regola di correlazione sarà abilitata.

Per configurare i dettagli di una nuova condizione, fai clic su Add nel blocco Condizioni. Si aprirà una finestra popup in cui potrai modificare i dettagli della condizione.



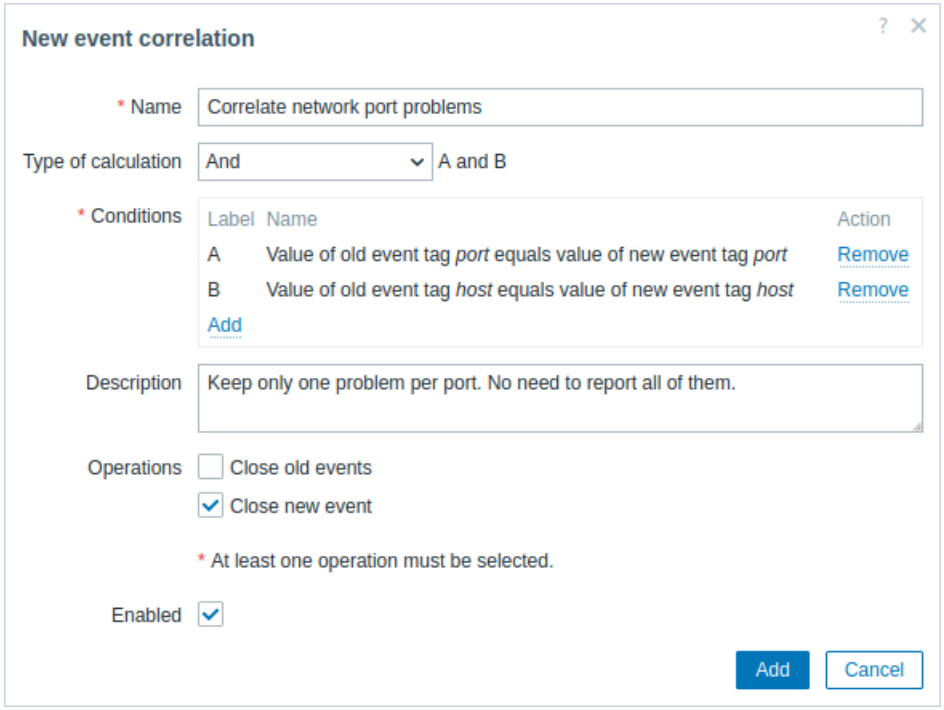
Parametro	Descrizione
Nuova condizione	Seleziona una condizione per correlare gli eventi. *Nota* che se non viene specificata alcuna condizione del vecchio evento, tutti i vecchi eventi potrebbero essere associati e chiusi. Analogamente, se non viene specificata alcuna condizione del nuovo evento, tutti i nuovi eventi potrebbero essere associati e chiusi. Sono disponibili le seguenti condizioni: **Nome tag del vecchio evento** - specifica il nome del tag del vecchio evento per la corrispondenza. **Nome tag del nuovo evento** - specifica il nome del tag del nuovo evento per la corrispondenza. **Gruppo host del nuovo evento** - specifica il gruppo host del nuovo evento per la corrispondenza. **Coppia di tag evento** - 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 corrispondono. I *nomi* dei tag non devono necessariamente corrispondere. Questa opzione è utile per far corrispondere valori di runtime, che potrebbero non essere noti al momento della configurazione (vedi anche Esempio). **Valore tag del vecchio evento** - 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 **Valore tag del nuovo evento** - 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 possono essere creati tag evento simili per problemi **non correlati**, rivedi i casi descritti di seguito!

- 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 iniziale del problema con un valore tag *UNKNOWN* viene perso, potrebbero comparire successivi eventi OK con lo stesso valore 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, anche questo può causare tag evento simili per problemi non correlati.

Esempio

Interrompi gli eventi di problema ripetitivi 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.

6 Etichettatura

Panoramica

I tag sono costituiti da un nome del tag e da un valore del tag. Quando si assegnano tag alle entità, è possibile usare solo il nome oppure abbinarlo 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 di template
- Prototipi di host, item e trigger

Negli elenchi *Templates*, *Hosts*, *Items*, *Triggers*, *Web scenarios* e nei relativi prototipi, la colonna *Tags* mostra sia i tag normali sia quelli ereditati. Passando il puntatore sopra un tag *ereditato* o facendovi clic, viene visualizzato un tooltip JavaScript con il testo "Inherited tag". Se un tag ereditato esiste anche come tag normale, viene mostrato un testo del tooltip diverso a seconda dell'elenco in cui appare (ad esempio, "Inherited and template tag" nell'elenco templates oppure "Inherited and host tag" nell'elenco hosts). Negli altri punti del frontend i tag normali ed ereditati vengono visualizzati insieme (i duplicati vengono rimossi), senza icone e senza testo aggiuntivo nel tooltip.

Note:
Fare riferimento alle linee guida ufficiali di Zabbix per le raccomandazioni generali sulla definizione dei tag, insieme alle indicazioni specifiche per templates, items, triggers e regole di low-level discovery.

I tag hanno molteplici scopi, in particolare quello di contrassegnare gli eventi. Quando alle entità vengono assegnati tag, qualsiasi nuovo evento correlato a un'entità con tag ne erediterà i tag. Ad esempio:

- con template con tag, qualsiasi problema del host (creato dai trigger del template) erediterà i tag del template.
- con host con tag, qualsiasi problema del host erediterà i tag del host.
- con item/scenari web con tag, qualsiasi problema di item/scenario web erediterà i tag dell'item/dello scenario web.
- con trigger con tag, 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 `tag:value` identiche (dopo la risoluzione delle macro) vengono unite in una sola, evitando così duplicazioni.

Gli eventi di ripristino generati da una chiusura manuale includono anch'essi i tag evento risolti 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:

- la correlazione degli eventi può essere configurata in base ai tag evento.
- le 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.

Alle entità possono essere assegnati tag con lo stesso nome ma con valori 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.

Casi d'uso

Alcuni casi d'uso comuni per l'uso dei tag sono i seguenti:

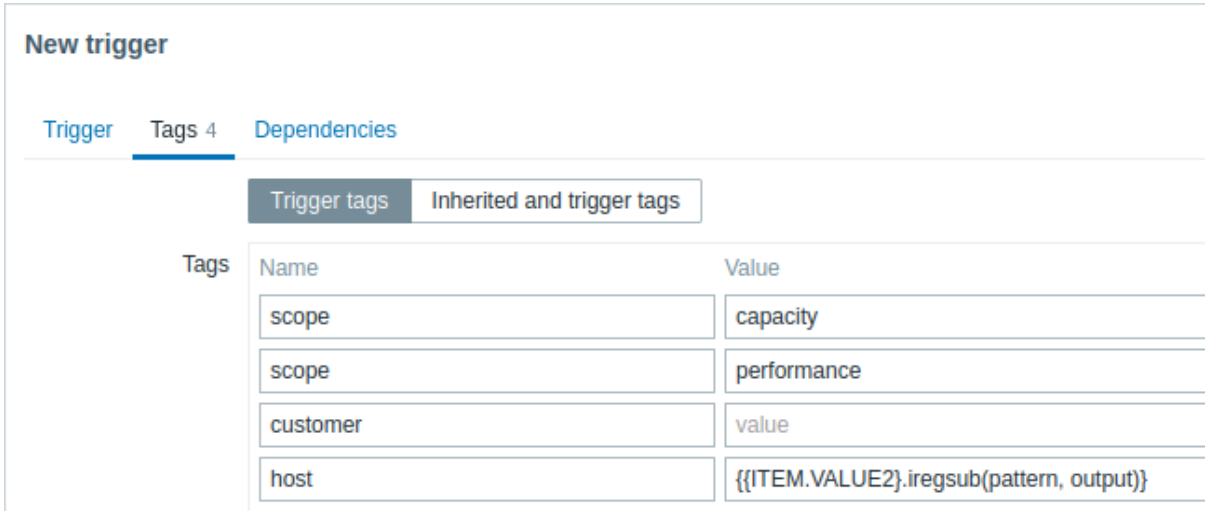
1. Contrassegnare gli eventi dei trigger:
   - Definire un tag del trigger (ad esempio, `scope:performance`).
   - I problemi creati da questo trigger avranno il tag del trigger.
2. Contrassegnare i problemi ereditati dal template:
   - Definire un tag del template (ad esempio, `target:mysql`).
   - I problemi creati dai trigger di questo template avranno il tag del template.
3. Contrassegnare i problemi del host:
   - Definire un tag del host (ad esempio, `service:jira`).
   - I problemi creati dai trigger di questo host avranno il tag del host.
4. Filtrare gli item correlati:
   - Definire un tag dell'item (ad esempio, `component:cpu`).
   - In *Monitoring* > *Latest data*, gli item possono essere filtrati tramite il tag `component:cpu`.
5. Utilizzare 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 log che estrarrà valori dal valore dell'item usando una macro (ad esempio, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Nella configurazione del trigger, configurare la correlazione degli eventi:
     - 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 log avranno il tag del trigger e verranno chiusi singolarmente.
7. Filtrare le notifiche:
   - Definire tag dei trigger (ad esempio, `scope:security` per trigger1 e `scope:availability` per trigger2).
   - Utilizzare il filtraggio per tag nelle condizioni dell'azione per ricevere notifiche solo sugli eventi che corrispondono ai dati dei tag.
8. Identificare i problemi nelle notifiche:
   - Definire tag dei trigger.
   - Utilizzare la macro {EVENT.TAGS} nella notifica del problema.
   - La notifica del problema conterrà i tag del trigger, rendendo più semplice identificare a quale applicazione/servizio appartiene la notifica.
9. Semplificare le attività di configurazione utilizzando 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 dal low-level discovery (LLD):
    - Definire un tag del prototipo di trigger con una macro LLD nel nome o nel valore del tag (ad esempio, `scope:{#FSNAME}`).
    - I trigger creati dal prototipo di trigger avranno il suo tag.
11. Associare i servizi usando i tag di servizio:
    - Definire i tag di servizio.
    - Configurare le azioni di servizio per i servizi con tag corrispondenti.
    - Inoltre, utilizzare i tag di servizio per collegare un servizio a un SLA per i calcoli SLA.
12. Collegare i servizi ai problemi usando i tag problema del servizio:
    - Definire un tag problema nella configurazione del servizio (ad esempio, `target:mysql`).

- I problemi con un tag corrispondente saranno automaticamente correlati al servizio e lo stato del servizio cambierà in base alle regole configurate per il calcolo dello stato del servizio.

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 utenti:
    - Definire i tag nella configurazione del gruppo utenti.
    - Gli utenti del gruppo 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 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 saranno mostrate nel frontend di Zabbix come non risolte.

Le macro di low-level discovery 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 host da uno o più host in un'espressione trigger)
- macro utente e macro utente con contesto (il contesto può includere macro di low-level discovery)
- macro di low-level discovery (solo nei tag dei prototipi di trigger)

Le seguenti macro possono essere utilizzate nei nomi e nei valori dei tag di template, host e item/scenari 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 low-level discovery (solo nei tag dei prototipi 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 di tag evento o tag evento di ripristino separati da virgole)
- macro integrate {EVENT.TAGSJSON} e {EVENT.RECOVERY.TAGSJSON} (queste macro verranno risolte in un array JSON contenente oggetti tag evento o oggetti tag evento di ripristino)

Estrazione di sottostringhe nei tag dei trigger

L'estrazione di sottostringhe è supportata per popolare il nome o il valore del tag, utilizzando una funzione di macro. La funzione applica un'espressione regolare al valore ottenuto dalla macro supportata. Per 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.

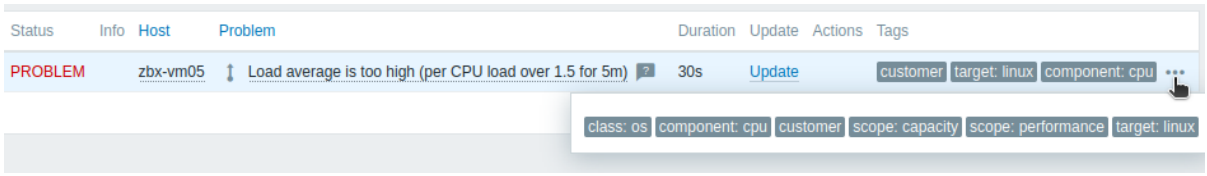
Vedi anche: utilizzo delle funzioni macro nelle macro di low-level discovery per il tagging degli eventi.

Visualizzazione dei tag evento

I tag, se definiti, possono essere visualizzati con i nuovi eventi in:

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Dettagli evento*
- *Dashboards* > widget *Problems*

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 dashboard *Problems*. Si noti che è possibile visualizzare un massimo di tre tag; se sono presenti più tag, passando il cursore sui tre puntini vengono mostrati tutti i tag in una finestra pop-up.



7 Visualizzazione

Utilizzare la barra laterale per accedere ai contenuti della sezione Visualizzazione.

1 Grafici

Panoramica

Con una grande quantità di dati che fluisce in Zabbix, per gli utenti diventa molto più semplice osservare una rappresentazione visiva di ciò che sta accadendo piuttosto che soltanto numeri.

È qui che entrano in gioco i grafici. I grafici permettono di comprendere il flusso dei dati a colpo d'occhio, correlare i problemi, scoprire quando qualcosa è iniziato o presentare quando qualcosa potrebbe trasformarsi in un problema.

Zabbix offre agli utenti:

- grafici semplici integrati dei dati di un 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

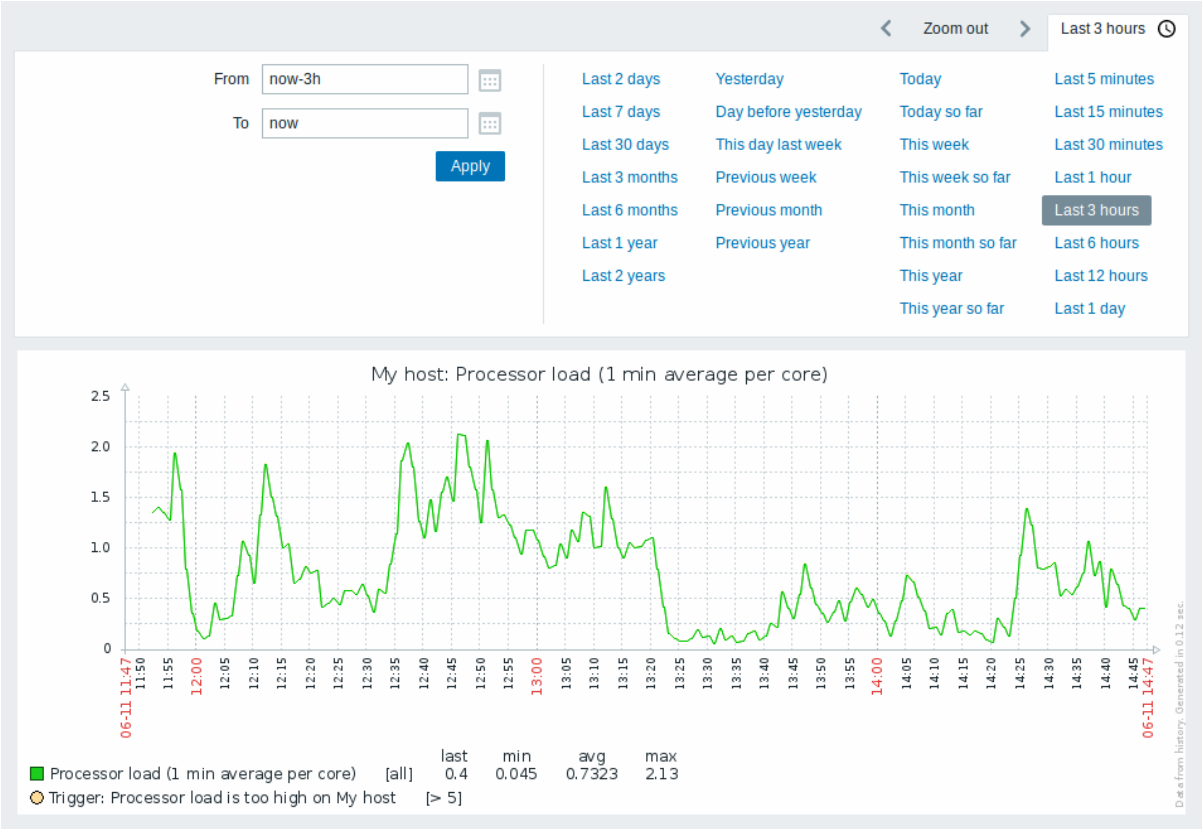
1 Grafici semplici

Panoramica

I grafici semplici sono disponibili per la visualizzazione dei dati raccolti dagli item.

Non è richiesto alcuno sforzo di configurazione da parte dell'utente per visualizzare i grafici semplici. Sono messi liberamente a disposizione da Zabbix.

È sufficiente andare in *Monitoring* > *Latest data* e fare clic sul collegamento Graph per il rispettivo item; verrà visualizzato 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

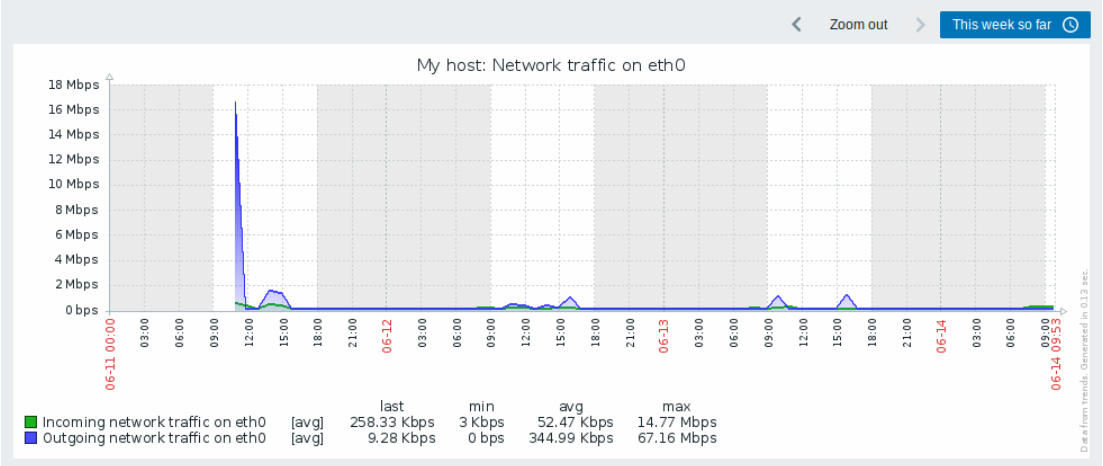
Il selettore del periodo di tempo sopra il grafico consente di selezionare i periodi richiesti più spesso con un clic del mouse. Per ulteriori informazioni, vedere Selettori del periodo di tempo e dell'host.

Dati recenti vs periodi più lunghi

Per i dati molto recenti viene tracciata una linea **singola** che collega ogni valore ricevuto. La linea singola viene tracciata finché è disponibile almeno un pixel orizzontale per un valore.

Per i dati che mostrano 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 massimi e minimi in quel momento. Lo spazio tra i massimi e i minimi viene 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 è una preferenza 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 sopra il colore della gravità del trigger; si noti la linea blu nel grafico e le informazioni del trigger visualizzate nella legenda. Nel grafico possono essere visualizzate fino a 3 linee di trigger; se ci sono più trigger, vengono prioritizzati i trigger con gravità inferiore. I trigger vengono sempre visualizzati nei grafici semplici, mentre la loro visualizzazione nei grafici personalizzati è una preferenza dell’utente.



Generazione da storico/trend

I grafici possono essere disegnati in base allo storico o ai trend di un item.

Per gli utenti che hanno la modalità di debug del frontend attivata, nell’angolo inferiore destro di un grafico viene visualizzata una didascalia verticale grigia che indica da dove provengono i dati.

Diversi fattori influenzano se viene utilizzato lo storico o i trend:

- durata della conservazione dello storico dell’item. Ad esempio, lo storico di un item può essere conservato per 14 giorni. In tal caso, tutti i dati più vecchi di quattordici giorni proverranno dai trend.
- congestione dei dati nel grafico. Se il numero di secondi da visualizzare in un pixel orizzontale del grafico supera 3600/16, vengono visualizzati i dati di trend (anche se lo storico dell’item è ancora disponibile per lo stesso periodo).
- se i trend sono disabilitati, per la creazione del grafico viene utilizzato lo storico 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 ultimi valori*. Questo può essere utile per visualizzare i valori numerici che compongono il grafico.

I valori qui rappresentati sono grezzi, cioè non vengono utilizzate né unità né post-elaborazione dei valori. La mappatura dei valori, tuttavia, viene applicata.

Problemi conosciuti

Consulta problemi noti per i grafici.

2 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.

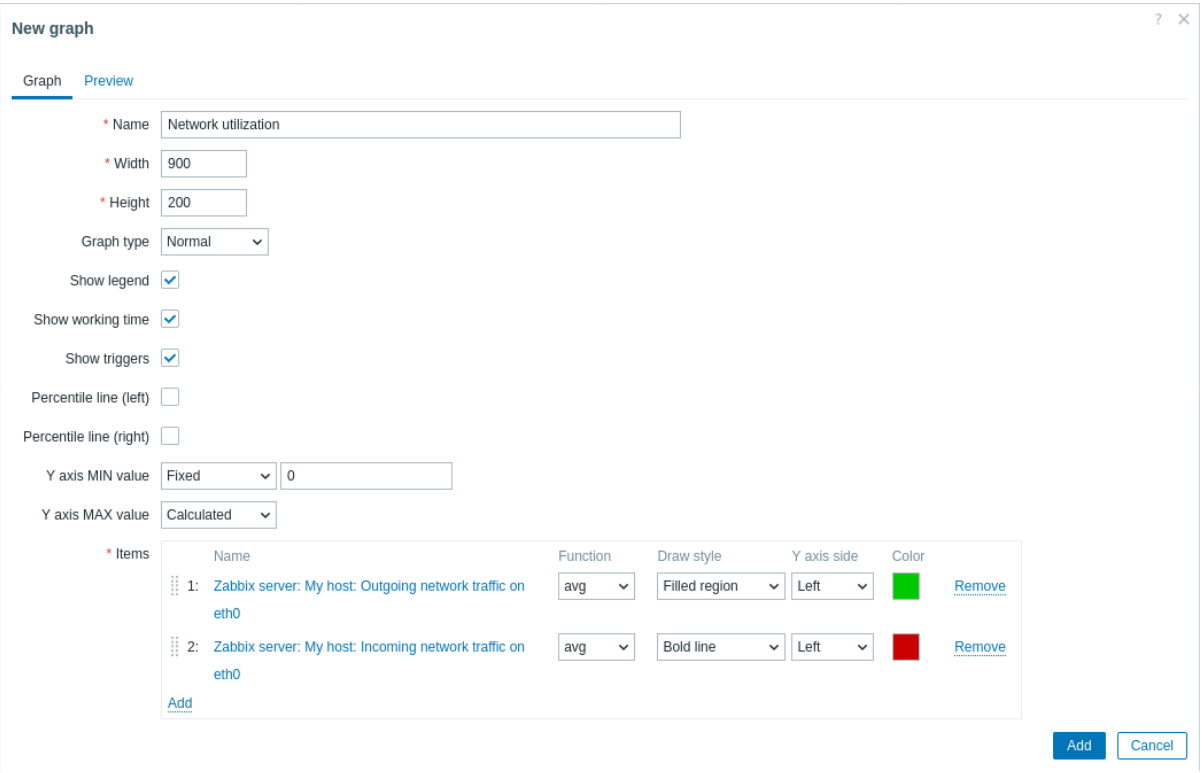
I grafici personalizzati vengono configurati manualmente.

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:

- Andare su *Raccolta dati* > *Host (o Template)*
- Fare clic su *Grafici* nella riga accanto all'host o al template desiderato
- Nella schermata Grafici fare clic su *Crea grafico*
- Modificare gli attributi del grafico



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

Attributi del grafico:

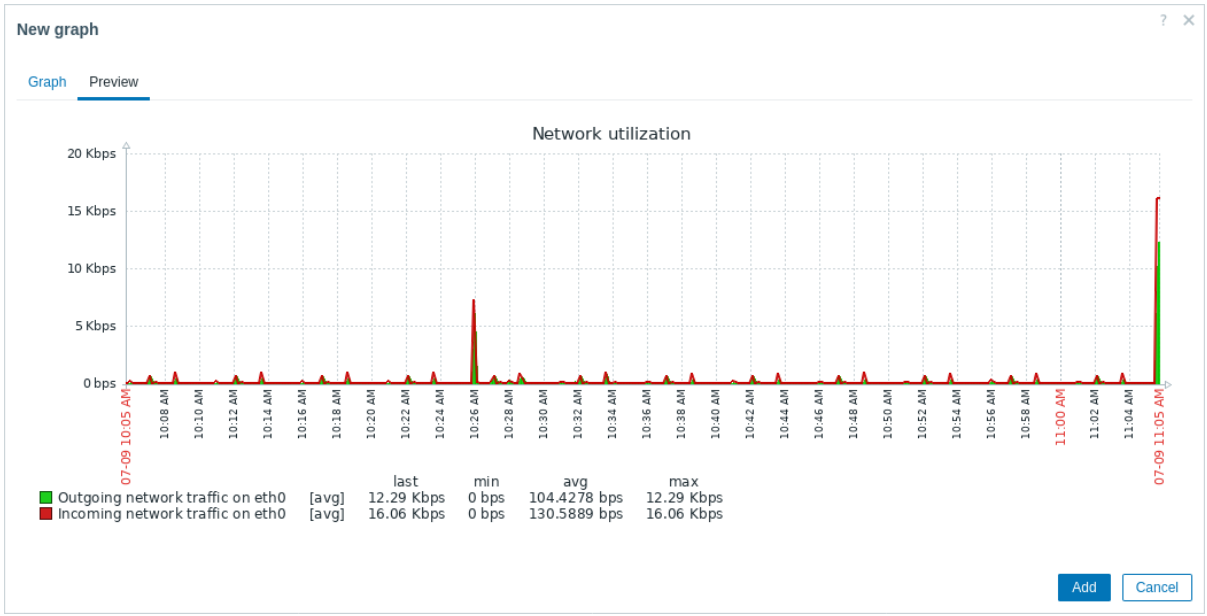
Parametro	Descrizione
Nome	Nome univoco del grafico. In questo campo sono supportate le macro di espressione, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`). Le macro `{HOST.HOST<1-9>}` sono supportate per l'uso all'interno di questa macro, facendo riferimento al primo, secondo, terzo, ecc. host nel grafico, ad esempio `{?avg(/{HOST.HOST2}/key,1h)}`. Si noti che fare riferimento al primo host con questa macro è ridondante, poiché al primo host si può fare riferimento implicitamente, ad esempio `{?avg(//key,1h)}`.
Larghezza	Larghezza del grafico in pixel (solo per anteprima e grafici a torta/esplosi).
Altezza	Altezza del grafico in pixel.

Parametro	Descrizione
Tipo di grafico	Tipo di grafico: **Normale** - grafico normale, i valori sono visualizzati come linee **Impilato** - grafico impilato, vengono visualizzate aree riempite **Torta** - grafico a torta **Esploso** - grafico a torta ”esploso”, con porzioni visualizzate come ”staccate” dalla torta
Mostra legenda	Selezionando questa casella verrà visualizzata la legenda del grafico.
Mostra orario di lavoro	Se selezionato, le ore non lavorative verranno mostrate con uno sfondo grigio. Questo parametro non è disponibile per i grafici a torta ed esplosi.
Mostra trigger	Se selezionato, i trigger semplici verranno visualizzati come linee con trattini neri sopra il colore di gravità del trigger. Questo parametro non è disponibile per i grafici a torta ed esplosi.
Linea percentile (sinistra)	Visualizza il percentile per l’asse Y sinistro. Se, ad esempio, è impostato il 95° percentile, la linea del percentile sarà al livello sotto il quale ricade il 95 percento dei valori. Visualizzata come una linea verde brillante. Disponibile solo per i grafici normali.
Linea percentile (destra)	Visualizza il percentile per l’asse Y destro. Se, ad esempio, è impostato il 95° percentile, la linea del percentile sarà al livello sotto il quale ricade il 95 percento dei valori. Visualizzata come una linea rosso brillante. Disponibile solo per i grafici normali.
Valore MIN asse Y	Valore minimo dell’asse Y: **Calcolato** - il valore minimo dell’asse Y verrà calcolato automaticamente. **Fisso** - 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 ed esplosi.
Valore MAX asse Y	Valore massimo dell’asse Y: **Calcolato** - il valore massimo dell’asse Y verrà calcolato automaticamente. **Fisso** - 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 ed esplosi.
Vista 3D	Abilita lo stile 3D. Solo per grafici a torta ed esplosi.
Item	Item i cui dati devono essere visualizzati in questo grafico. Fare clic su *Aggiungi* per selezionare gli item. È inoltre possibile selezionare varie opzioni di visualizzazione (funzione, stile di disegno, visualizzazione asse sinistro/destro, colore).
Ordine di ordinamento (0→100)	Ordine di disegno. 0 verrà elaborato per primo. Può essere usato per disegnare linee o aree dietro (o davanti a) un altro elemento. È possibile trascinare gli item usando l’icona all’inizio di una riga per impostare l’ordine di ordinamento o quale item viene visualizzato davanti all’altro.
Nome	Il nome dell’item selezionato viene visualizzato come link. Facendo clic sul link si apre l’elenco degli altri item disponibili.
Tipo	Tipo (disponibile solo per grafici a torta ed esplosi): **Semplice** - il valore dell’item è rappresentato proporzionalmente nella torta **Somma del grafico** - il valore dell’item rappresenta l’intera torta Si noti che il colore dell’item ”somma del grafico” sarà visibile solo nella misura in cui non è occupato dagli item ”proporzionali”.

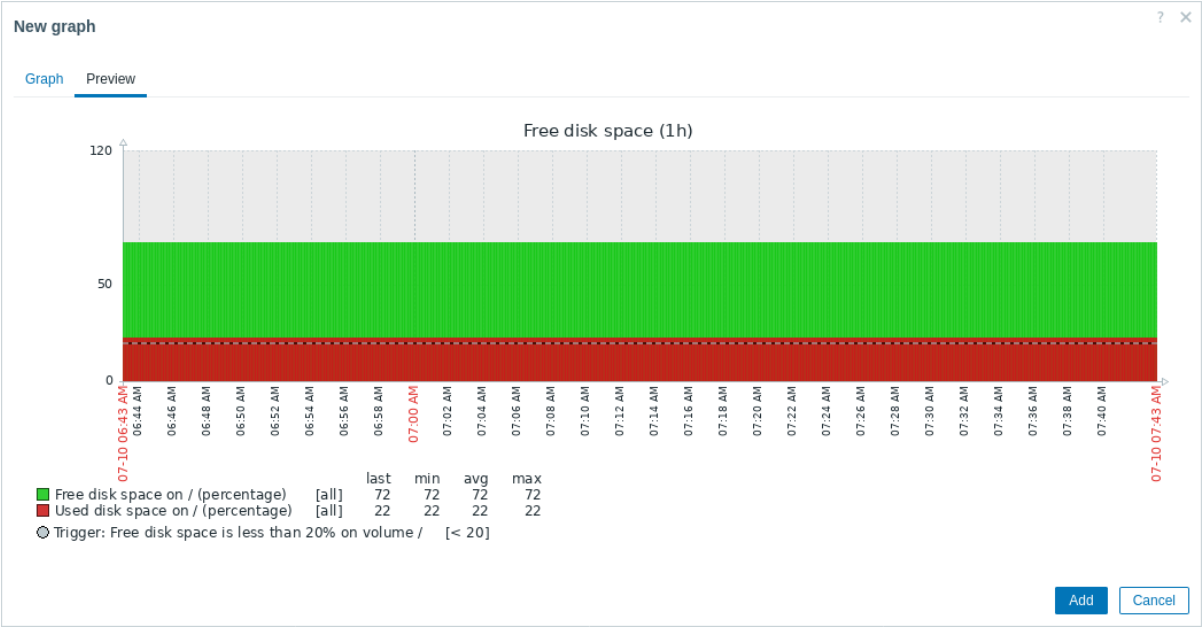
Parametro	Descrizione
Funzione	Selezionare quali valori verranno visualizzati quando esiste più di un valore per pixel verticale del grafico per un item: **all** - visualizza tutti i valori possibili (minimo, massimo, media) nel grafico. Si noti che per periodi più brevi questa impostazione non ha effetto; solo per periodi più lunghi, quando aumenta la congestione dei dati in un pixel verticale del grafico, 'all' inizia a visualizzare i valori minimo, massimo e medio. Questa funzione è disponibile solo per il tipo di grafico *Normale*. Vedere anche: Generazione di grafici da history/trends. **avg** - visualizza i valori medi **last** - visualizza i valori più recenti. Questa funzione è disponibile solo se come tipo di grafico è selezionato *Torta/Torta esplosa*. **max** - visualizza i valori massimi **min** - visualizza i valori minimi
Stile di disegno	Selezionare lo stile di disegno (disponibile solo per grafici normali; per i grafici impilati viene sempre usata un'area riempita) da applicare ai dati dell'item - *Linea*, *Linea spessa*, *Area riempita*, *Punto*, *Linea tratteggiata*, *Linea sfumata*.
Lato asse Y	Selezionare il lato dell'asse Y su cui mostrare i dati dell'item - *Sinistra*, *Destra*.
Colore	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.

3 Grafici ad hoc

Panoramica

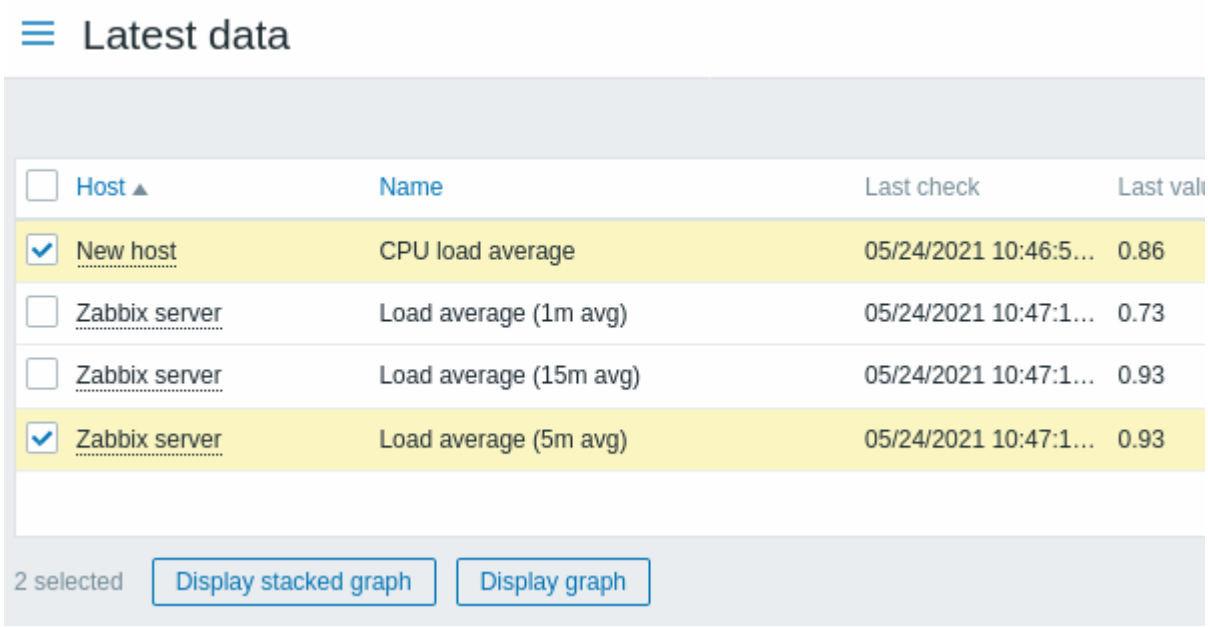
Mentre un grafico semplice è ottimo per accedere ai dati di un item e i grafici personalizzati offrono opzioni di personalizzazione, nessuno dei due consente di creare rapidamente un grafico di confronto per più item con poco sforzo e senza manutenzione.

Per risolvere questo problema, è possibile creare grafici ad hoc per diversi item in modo molto rapido.

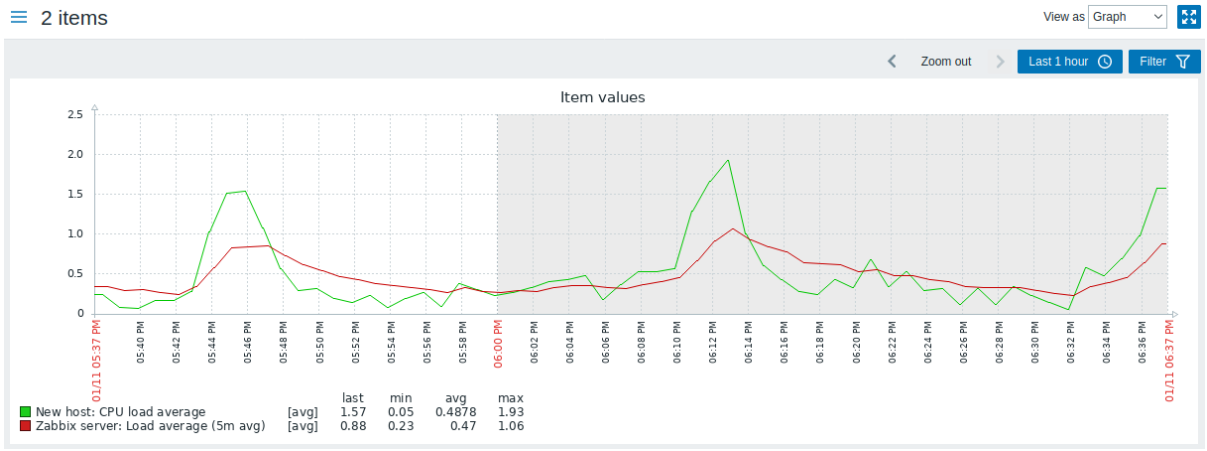
Configurazione

Per creare un grafico ad hoc, procedere come segue:

- Andare in *Monitoraggio* → *Ultimi dati*
- Utilizzare il filtro per visualizzare gli item desiderati
- Selezionare le caselle di controllo degli item che si desidera rappresentare nel grafico
- Fare clic sui pulsanti *Visualizza grafico in pila* o *Visualizza grafico*

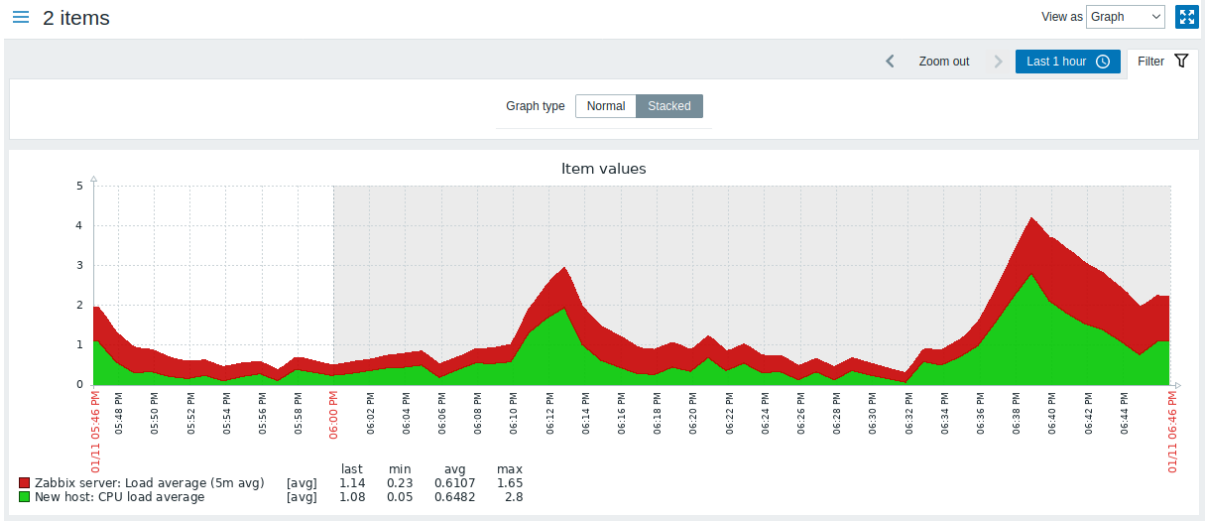


Il grafico viene creato immediatamente:



Si noti che, per evitare di visualizzare troppe linee nel grafico, viene mostrato solo il valore medio per 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 sono disponibili il selettore dell'intervallo di tempo e la possibilità di passare dal grafico a linee "normale" a quello in pila (e viceversa).



4 Aggregazione nei grafici

Panoramica

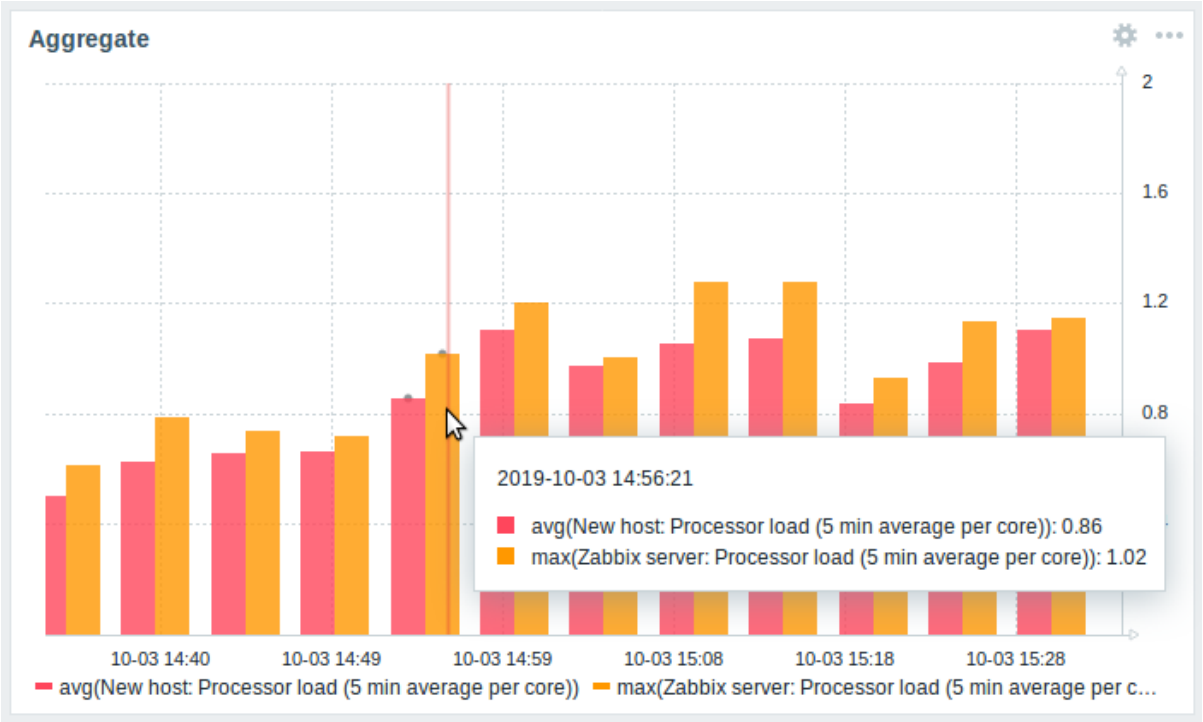
Le funzioni di aggregazione, disponibili nei widget grafico e grafico a torta della dashboard, consentono di visualizzare un valore aggregato per l'intervallo scelto (5 minuti, un'ora, un giorno), invece di tutti i valori.

Questa sezione fornisce maggiori dettagli sulle opzioni di aggregazione nel widget grafico.

Le opzioni di aggregazione sono le seguenti:

- min
- max
- avg
- count
- sum
- first (primo valore visualizzato)
- last (ultimo valore visualizzato)

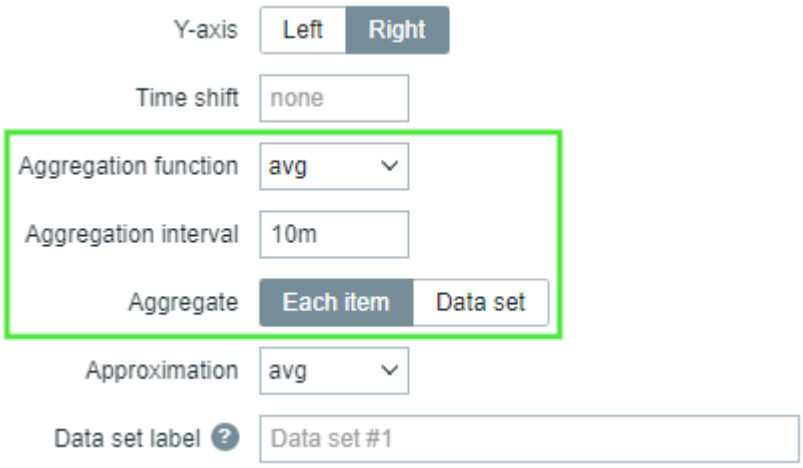
L'utilizzo più interessante dell'aggregazione dei dati è la possibilità di creare utili confronti affiancati dei dati per un determinato periodo:



Passando il cursore su un punto temporale nel grafico, vengono visualizzati la data e l'ora, oltre agli item e ai relativi valori aggregati. Gli item sono visualizzati tra parentesi, preceduti dalla funzione di aggregazione utilizzata. Se nel widget grafico è configurata un'etichetta *Data set label*, l'etichetta viene visualizzata tra parentesi, preceduta dalla funzione di aggregazione utilizzata. Si noti che questa è la data e l'ora del punto nel grafico, non dei valori effettivi.

Configurazione

Le opzioni per l'aggregazione sono disponibili nelle impostazioni del set di dati durante la configurazione di un widget grafico.



È possibile selezionare la funzione di aggregazione e l'intervallo di tempo. Poiché il set di dati può comprendere diversi item, è disponibile anche un'altra opzione che consente di mostrare i dati aggregati per ciascun item separatamente oppure per tutti gli item del set di dati come un unico valore aggregato.

Casi d'uso

Conteggio medio delle richieste al server Nginx

Visualizza il conteggio medio delle richieste al secondo per giorno verso il server Nginx:

- aggiungi l'item del conteggio delle richieste al secondo al set di dati
- seleziona la funzione di aggregazione `avg` e specifica l'intervallo `1d`
- viene visualizzato un grafico a barre, in cui ogni barra rappresenta il numero medio di richieste al secondo per giorno

Spazio minimo settimanale su disco tra i cluster

Visualizza lo spazio su disco più basso tra i cluster nell'arco di una settimana.

- aggiungere al set di dati: host `cluster*`, chiave "Spazio libero su disco su /data"
- selezionare la funzione di aggregazione `min` e specificare l'intervallo `1w`
- viene visualizzato un grafico a barre, in cui ogni barra rappresenta lo spazio minimo su disco settimanale per ciascun volume /data del cluster

2 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.

1 Configurazione di una mappa di rete

Panoramica

La configurazione di una mappa in Zabbix richiede innanzitutto la creazione della mappa, definendone i parametri generali; successivamente si procede a popolare la mappa vera e propria con elementi e relativi collegamenti.

È possibile popolare la mappa con elementi quali 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 da visualizzare con le icone e impostare la visualizzazione dei problemi recenti in modo speciale. È inoltre possibile collegare le icone e definire le informazioni da visualizzare sui collegamenti.

È possibile aggiungere URL personalizzati accessibili 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.

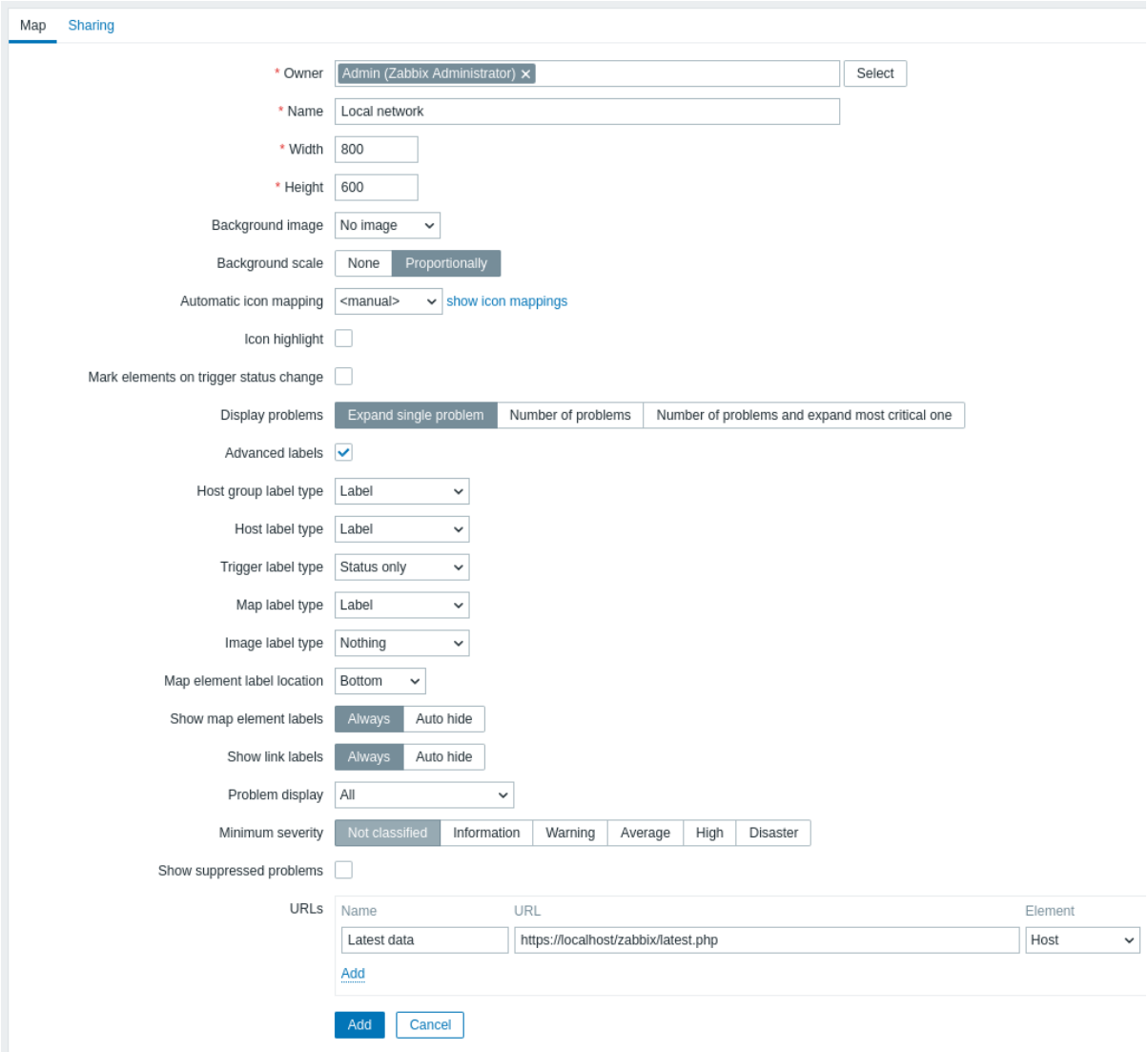
Creazione di una mappa

Per creare una mappa, procedere come segue:

1. Andare in *Monitoring* > *Maps*.
2. Aprire la vista con tutte le mappe.
3. Fare clic su *Create map*.

È anche possibile usare 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, nonché gli elementi della mappa esistente.

La scheda **Map** contiene gli attributi generali della mappa:



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

Attributi generali della mappa:

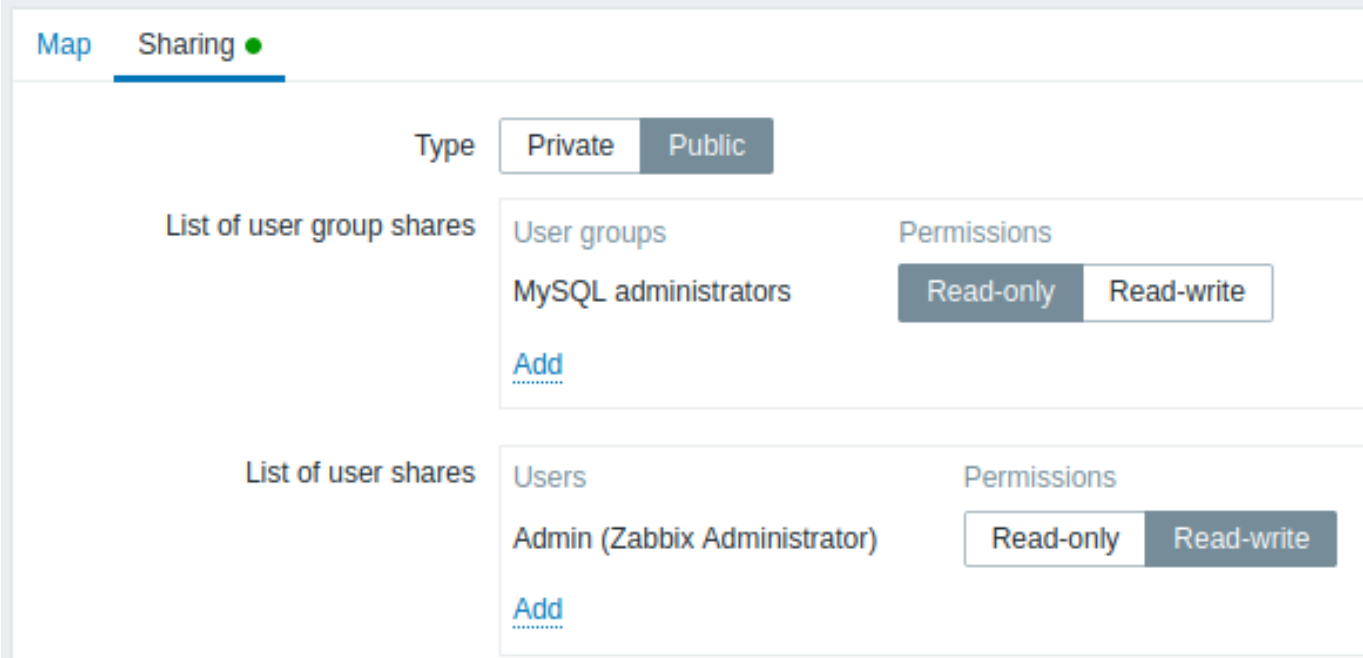
Parameter	Description
Owner	Nome del proprietario della mappa.
Name	Nome univoco della mappa.
Width	Larghezza della mappa in pixel.

Parameter	Description
Height	Altezza della mappa in pixel.
Background image	Usa immagine di sfondo: **No image** - nessuna immagine di sfondo (sfondo bianco) **Image** - l'immagine selezionata verrà usata come immagine di sfondo. È possibile usare una mappa geografica o qualsiasi altra immagine per migliorare la mappa.
Background scale	Usa ridimensionamento dello sfondo: **None** - nessun ridimensionamento; **Proportionally** - copre l'intero sfondo della mappa, ma mantiene le proporzioni dell'immagine (predefinito).
Automatic icon mapping	È possibile impostare l'uso 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 host.
Icon highlight	Se si seleziona questa casella di controllo, gli elementi della mappa verranno evidenziati. Gli elementi con un trigger attivo riceveranno uno sfondo rotondo, dello stesso colore del trigger con severità più alta. Inoltre, verrà visualizzata una spessa linea verde attorno al cerchio se tutti i problemi sono stati confermati. Gli elementi con stato "disabled" o "in maintenance" riceveranno uno sfondo quadrato, rispettivamente grigio e arancione. Vedi 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 che sono liberi dall'etichetta. Gli indicatori vengono visualizzati per 30 minuti.
Display problems	Selezionare come visualizzare i problemi con un elemento della mappa: **Expand single problem** - se c'è un solo problema, viene visualizzato il nome del problema. Altrimenti, 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 "più critico" viene determinato in base alla severità del problema e, in caso di parità, all'ID evento del problema (viene visualizzato per primo l'ID più alto o il problema più recente). Per un *trigger map element* si basa sulla severità del problema e, in caso di parità, sulla posizione del trigger nell'elenco dei trigger. In caso di più problemi dello stesso trigger, verrà visualizzato il più recente.
Advanced labels	Se si seleziona questa casella di controllo, sarà possibile definire tipi di etichetta separati per tipi di elemento separati.
Host group label type	Tipo di etichetta usato per l'elemento della mappa: **Label** - etichetta dell'elemento della mappa **IP address** - indirizzo IP **Element name** - nome dell'elemento (ad esempio, nome host) **Status only** - solo stato (OK o PROBLEM) **Nothing** - non viene visualizzata alcuna etichetta
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 (predefinito); **Auto hide** - nasconde l'etichetta se non è selezionata o se il puntatore non vi passa sopra.
Show link labels	Selezionare come visualizzare le etichette dei collegamenti: **Always** - sempre visualizzate (predefinito); **Auto hide** - nasconde l'etichetta se non è selezionata o se il puntatore non vi passa sopra.
Problem display	Visualizza il conteggio dei problemi come: **All** - verrà visualizzato il conteggio completo dei problemi **Separated** - il conteggio dei problemi non confermati verrà visualizzato separatamente come numero rispetto al conteggio totale dei problemi **Unacknowledged only** - verrà visualizzato solo il conteggio dei problemi non confermati

Parameter	Description
Minimum severity	I problemi al di sotto del livello minimo di severità selezionato non verranno visualizzati sulla mappa. Ad esempio, con *Warning* selezionato, le modifiche con trigger di livello *Information* e *Not classified* non verranno riportate nella mappa.
Show suppressed problems	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione dell'host.
URLs	Qui è possibile definire gli URL per ciascun 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 usate nei nomi e nei valori degli URL della mappa. Per un elenco completo, vedere macro supportate e cercare 'map URL names and values'.

Condivisione

La scheda **Condivisione** contiene il tipo di mappa e le opzioni di condivisione (gruppi di utenti, utenti) per le mappe private:



Parametro	Descrizione
Tipo	Selezionare il tipo di mappa: **Privata** - la mappa è visibile solo ai gruppi di utenti e agli utenti selezionati **Pubblica** - la mappa è visibile a tutti
Elenco delle condivisioni dei gruppi di utenti	Selezionare i gruppi di utenti a cui la mappa è accessibile. È possibile consentire l'accesso in sola lettura o in lettura-scrittura.
Elenco delle condivisioni degli utenti	Selezionare gli utenti a cui la mappa è accessibile. È possibile consentire l'accesso in sola lettura o in lettura-scrittura.

Quando si fa clic su *Aggiungi* per salvare questa mappa, viene creata una mappa vuota con un nome, dimensioni e determinate preferenze. Ora è necessario aggiungere alcuni elementi. Per farlo, fare clic su *Modifica* nell'elenco delle mappe per aprire l'area modificabile.

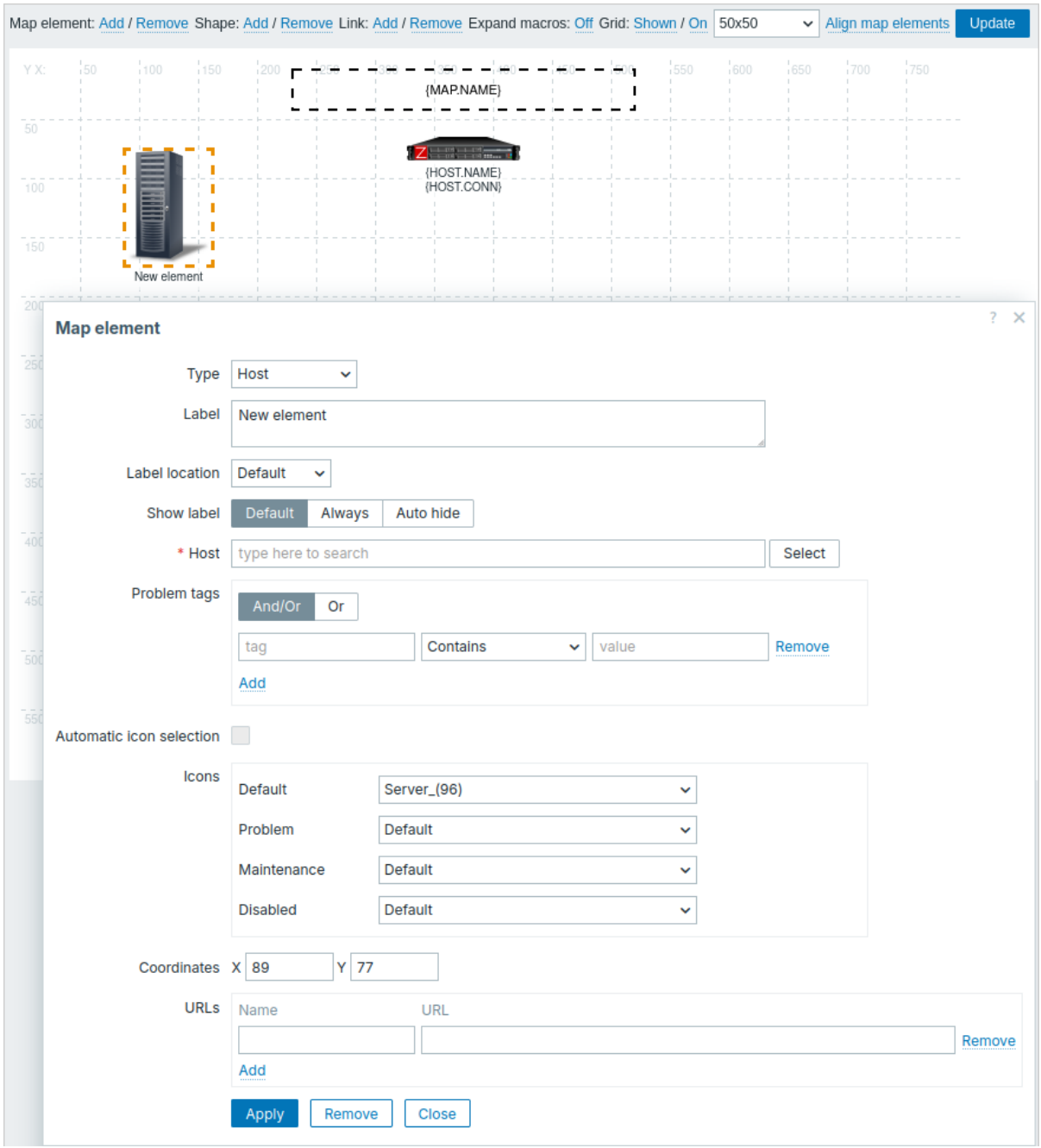
Aggiunta di elementi

Per aggiungere un elemento, fare clic su *Add* accanto a `Map element`. Il nuovo elemento apparirà nell'angolo superiore sinistro della mappa. Trascinarlo nella posizione desiderata.

Si noti che con l'opzione Grid impostata su "On", gli elementi si allineeranno sempre alla griglia (è possibile scegliere varie dimensioni della griglia dal menu a discesa, nonché nascondere/mostrare la griglia). Se si desidera posizionare gli elementi liberamente

senza allineamento, impostare l'opzione su "Off". (È possibile allineare in seguito elementi casuali alla griglia facendo clic su *Align map elements*.)

Ora che alcuni elementi sono stati posizionati, si potrebbe voler iniziare a differenziarli assegnando nomi, ecc. Facendo clic sull'elemento, viene visualizzato un modulo in cui è possibile 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 del 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 i host appartenenti al gruppo selezionato **Image** - un'icona, non collegata ad alcuna risorsa

Parameter	Description
Label	Etichetta dell'icona, una stringa qualsiasi. È possibile utilizzare macro e stringhe multilinea. Le macro di espressione sono supportate in questo campo, 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, vedere supported macros e cercare 'map element labels'.
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	Selezionare come visualizzare l'etichetta dell'elemento: **Default** - usa le impostazioni *Show map element labels* della configurazione della mappa (predefinito); **Always** - sempre visualizzata (predefinito); **Auto hide** - nasconde l'etichetta se non è evidenziata dal puntatore o selezionata.
Host	Selezionare un host; in alternativa, inserire il nome del host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un host verrà visualizzato un menu a discesa con i host corrispondenti. Scorrere verso il basso per selezionarne uno. Fare clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Host*.
Map	Selezionare una mappa; in alternativa, inserire il nome della mappa. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di una mappa verrà visualizzato un menu a discesa con le mappe corrispondenti. Scorrere verso il basso per selezionarne una. Fare clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Map*.
Triggers	Selezionare uno o più trigger nel campo *New triggers* sottostante e fare clic su *Add*. In alternativa, iniziare a digitare il nome di un trigger e selezionarne uno dall'elenco a discesa dei trigger corrispondenti. Fare 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 dello stesso livello di gravità dei trigger. La selezione di più trigger influisce anche sulla risoluzione della macro {HOST.*} sia in modalità di modifica che 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 è selezionata la modalità *Expand single problem*, le prime macro {HOST.*} visualizzate verranno risolte in base al trigger del problema rilevato più di recente (indipendentemente dalla gravità) oppure al primo trigger nell'elenco (se non viene rilevato alcun problema). - Se è selezionata 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	Selezionare un gruppo di host; in alternativa, inserire il gruppo di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo di host verrà visualizzato un menu a discesa con i gruppi di host corrispondenti. Scorrere verso il basso per selezionarne uno. Fare clic sull'icona di rimozione (x) per rimuovere la selezione. Questo campo è disponibile solo per elementi di tipo *Host group*.

Parameter	Description
Problem tags	Specificare i tag 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. 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) 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 Questo campo è disponibile solo per elementi di tipo *Host* e *Host group*.
Automatic icon selection	Quando la casella di controllo è selezionata, la mappatura delle icone verrà utilizzata per determinare quale icona visualizzare.
Icons	È possibile 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 collegamenti quando un utente fa clic sull'elemento nella modalità di visualizzazione della mappa. Se l'elemento ha URL propri e sono definiti URL a livello di mappa per il suo tipo, verranno combinati nello stesso menu. Le macro possono essere utilizzate nei nomi e nei valori degli elementi della mappa. Per un elenco completo, vedere supported macros e cercare 'map URL names and values'.

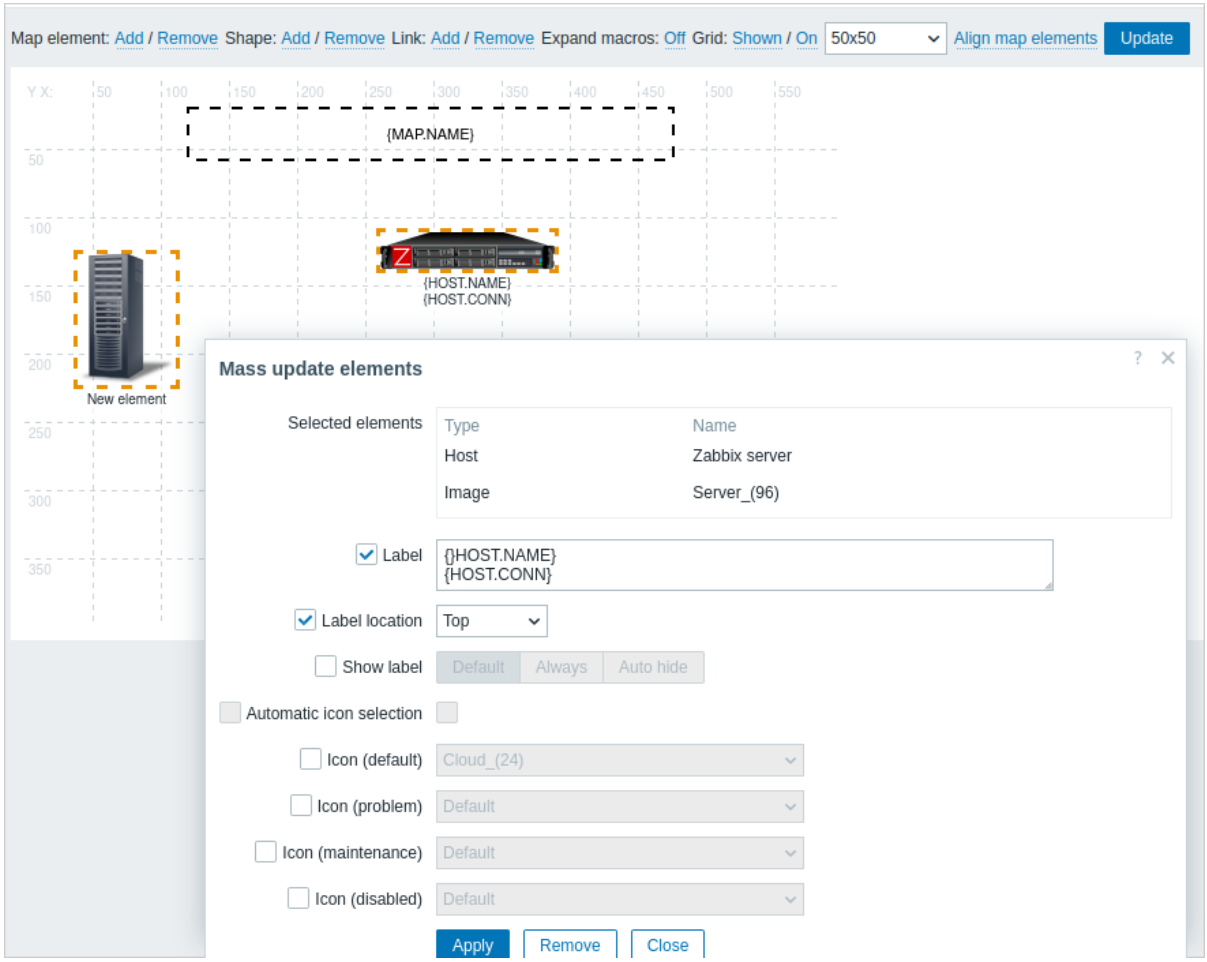
Attention:
Gli elementi aggiunti non vengono salvati automaticamente. Se si lascia la pagina, tutte le modifiche potrebbero andare perse.
Pertanto, è consigliabile fare clic sul pulsante **Update** nell'angolo superiore destro. Una volta fatto clic, le modifiche vengono salvate indipendentemente da ciò che si sceglie nel popup successivo.
Anche le opzioni della griglia selezionate vengono salvate con ciascuna 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.

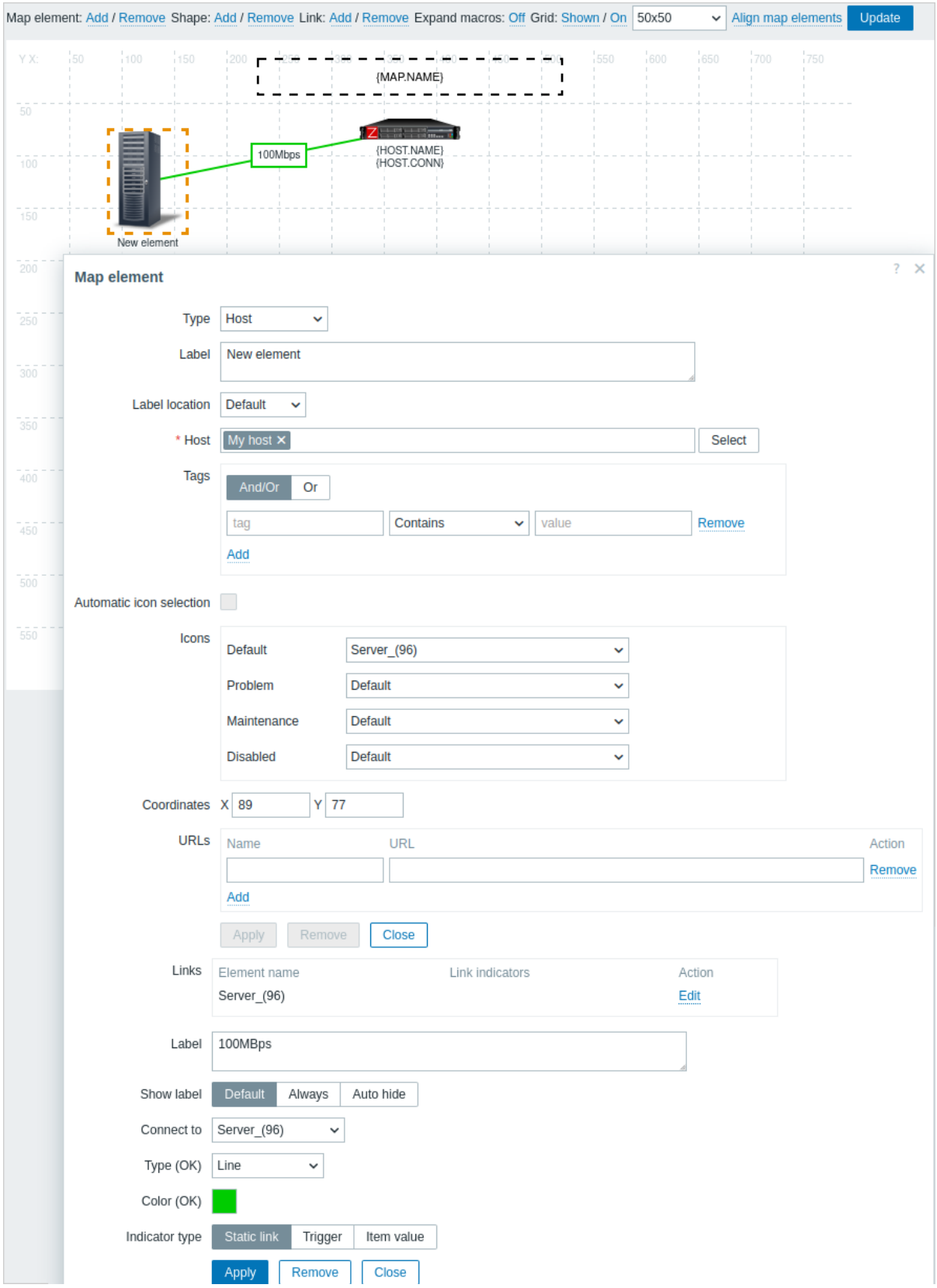
Una volta selezionato più di un elemento, il modulo delle proprietà dell'elemento passa alla modalità di aggiornamento di massa, così puoi modificare in un'unica volta gli attributi degli elementi selezionati. Per farlo, contrassegna l'attributo usando la casella di controllo e inserisci un nuovo valore. Qui puoi usare le macro (ad esempio, {HOST.NAME} per l'etichetta dell'elemento).



Collegamento degli elementi

Una volta inseriti alcuni elementi nella mappa, è il momento di iniziare a collegarli. Per collegare due elementi, è necessario prima selezionarli. Con gli elementi selezionati, fare clic su *Add* accanto a Link.

Una volta creato un collegamento, il modulo del singolo elemento contiene ora un'ulteriore sezione *Links*. Fare clic su *Edit* per modificare gli attributi del collegamento.



Attributi del collegamento:

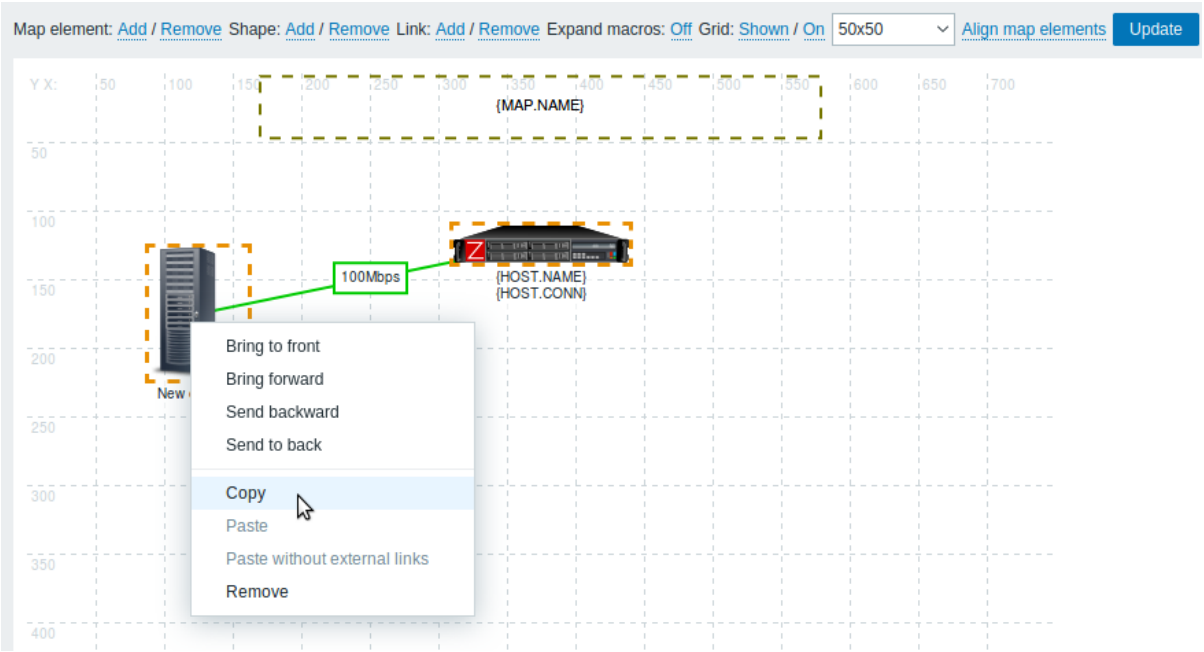
Parametro	Descrizione
Label	Etichetta che verrà visualizzata sopra il collegamento. In questo campo sono supportate le macro di espressione, ma solo con le funzioni `avg`, `last`, `min` e `max`, con il tempo come parametro (ad esempio, `{?avg(/host/key,1h)}`).

Parametro	Descrizione
Show label	Selezionare come visualizzare l'etichetta del collegamento: **Default** - usa le impostazioni *Show map element labels* della configurazione della mappa (predefinito); **Always** - sempre visualizzata (predefinito); **Auto hide** - nasconde l'etichetta se non è evidenziata al passaggio del mouse o selezionata.
Connect to	L'elemento a cui il collegamento si connette.
Type (OK)	Stile predefinito del collegamento. Selezionare lo stile: *Line*, *Bold line*, *Dot* oppure *Dashed line*.
Color (OK)	Colore predefinito del collegamento. Fare clic sul campione di colore per selezionare un altro colore.
Indicator type	Selezionare il tipo di indicatore del collegamento: **Static link** - nessun indicatore; **Trigger** - consente indicatori di collegamento basati su trigger; **Item value** - consente indicatori di collegamento basati sul valore dell'item.
Item	Selezionare l'item. Il valore di questo item influenzerà lo stile del collegamento. Questo campo è disponibile quando per *Indicator type* è selezionato *Item value*.
Indicators	L'elenco degli indicatori di collegamento. 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 di collegamento. Quando un trigger è nello stato *Problem*, il suo stile (colore selezionato e tipo di linea) verrà applicato al collegamento. Se impostato su *Item value* e nel campo *Item* è selezionato un item, i valori dell'item possono essere assegnati come indicatori di collegamento. Quando il valore dell'item raggiunge la soglia specificata (per un item numerico) o corrisponde al modello specificato (per un item di testo), il suo stile (colore selezionato e tipo di linea) viene applicato al collegamento.

Spostamento e copia-incolla degli elementi

Diversi elementi selezionati possono essere **spostati** in un altro punto della 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 con il pulsante destro del mouse su un elemento selezionato e scegliendo *Copy* dal menu.

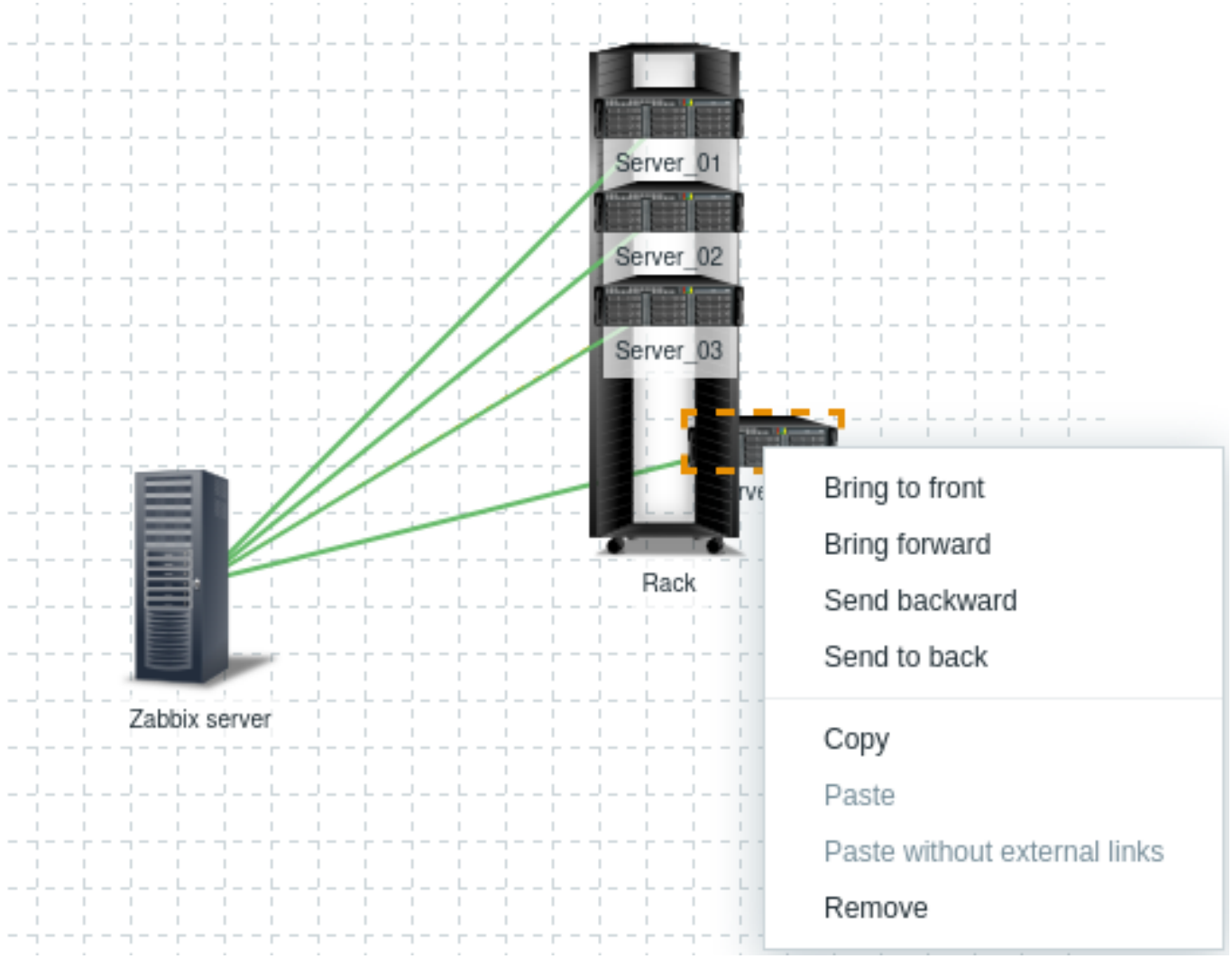


Per incollare gli elementi, fare clic con il pulsante destro del mouse su un'area della mappa e selezionare *Paste* dal menu. L'opzione *Paste without external links* incollerà gli elementi mantenendo solo i collegamenti tra gli elementi selezionati.

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*.

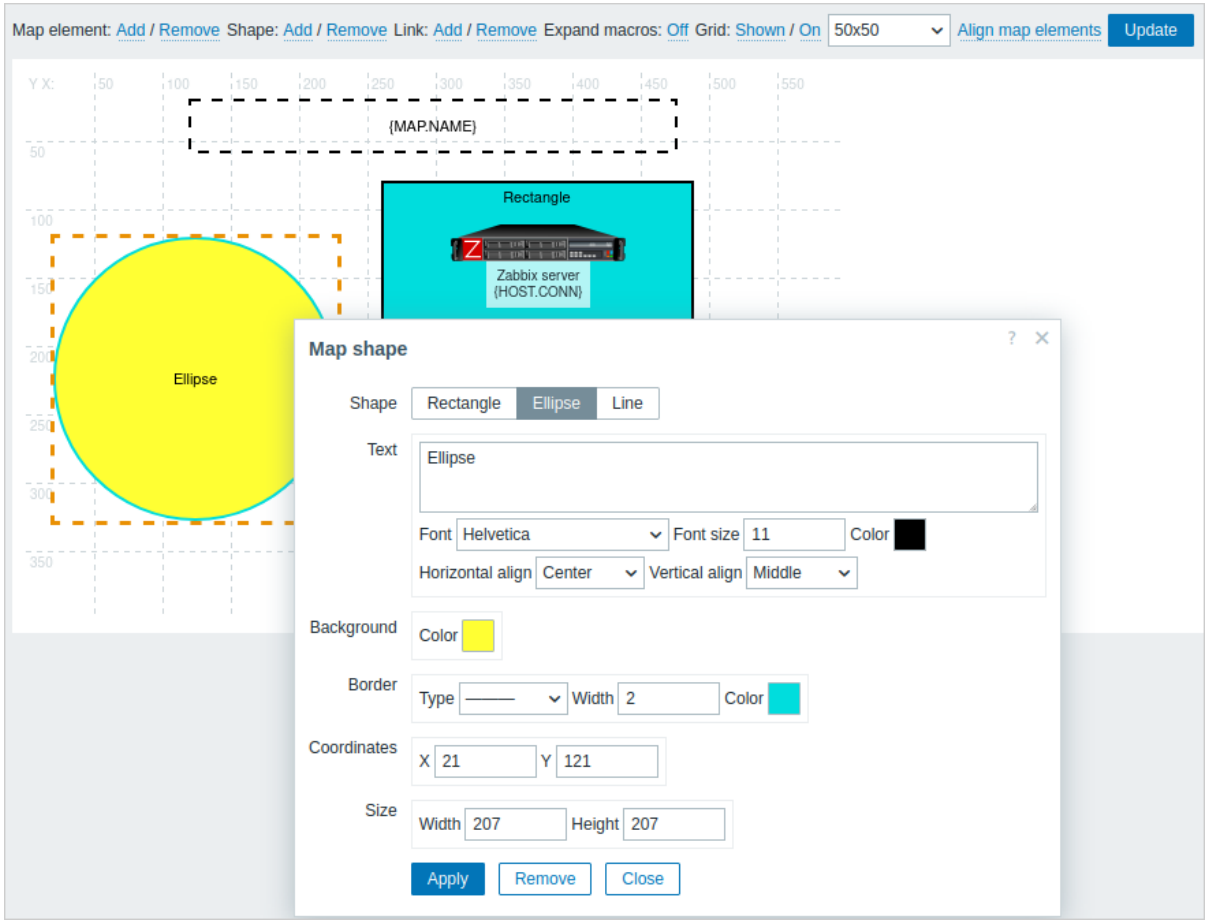


Aggiunta di forme

Oltre agli elementi della mappa, è anche possibile aggiungere alcune forme. Le forme non sono elementi della mappa; sono solo una rappresentazione visiva. Ad esempio, una forma rettangolare può essere usata come sfondo per raggruppare alcuni host. È possibile aggiungere forme rettangolari ed ellittiche.

Per aggiungere una forma, fare clic su *Add* accanto a Shape. La nuova forma apparirà nell'angolo superiore sinistro della mappa. Trascinarla dove si preferisce.

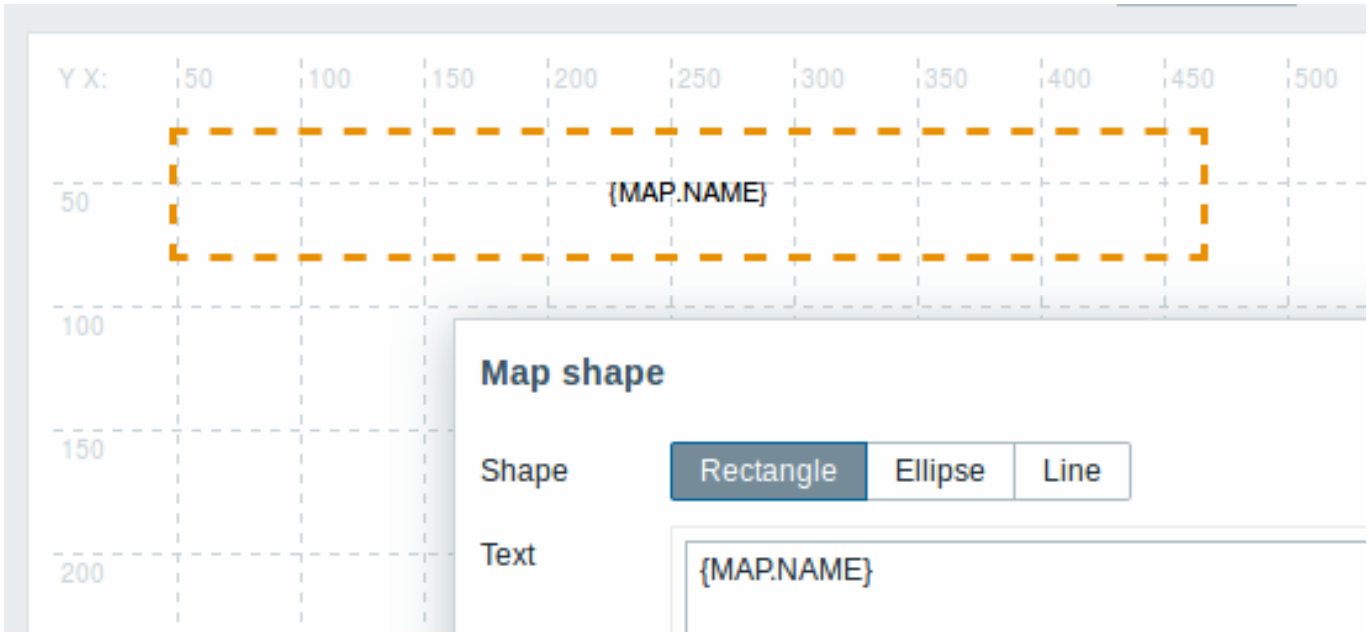
Una nuova forma viene aggiunta con i colori predefiniti. Facendo clic sulla forma, viene visualizzato un modulo ed è possibile personalizzare l'aspetto della forma, aggiungere testo, ecc.



Per selezionare le forme, selezionarne una e poi tenere premuto *Ctrl* per selezionare le altre. Con più forme selezionate, le proprietà comuni possono essere aggiornate in massa, in modo simile a quanto avviene con gli elementi.

È possibile aggiungere testo nelle forme. Le macro di espressione sono supportate nel testo, 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 (selezionare 'None' nel campo *Border*). Ad esempio, si noti come la macro {MAP.NAME}, visibile nello screenshot sopra, sia in realtà una forma rettangolare con testo, come si può vedere facendo clic sulla macro:



{MAP.NAME} viene risolta nel nome della mappa configurato durante la visualizzazione della mappa.

Se nel testo vengono utilizzati collegamenti ipertestuali, questi diventano cliccabili durante la visualizzazione della mappa.

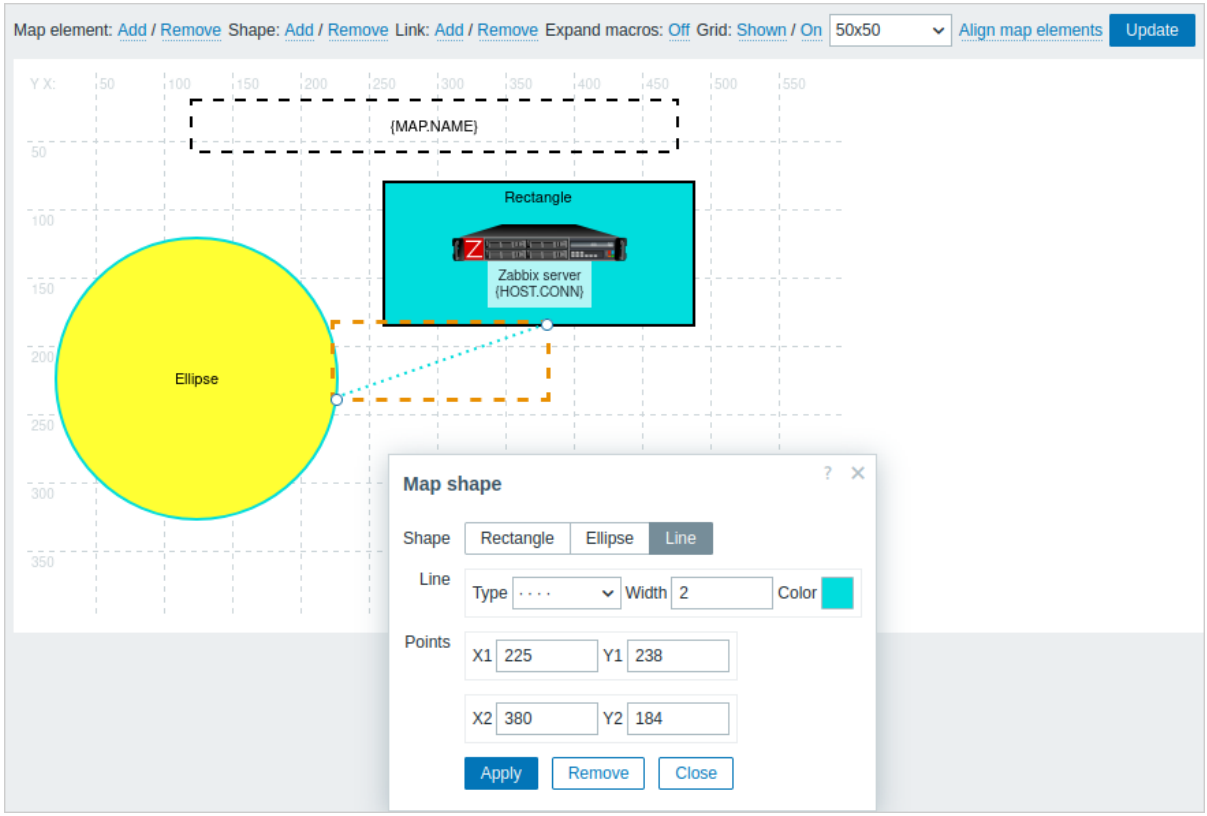
L'a capo automatico del testo è sempre attivo all'interno delle forme. Tuttavia, all'interno di un'ellisse, le righe vanno a capo come se l'ellisse fosse un rettangolo. L'a capo automatico delle parole non è implementato, quindi le parole lunghe (parole che non

entrano nella forma) non vanno a capo, ma vengono mascherate (nella pagina di modifica della mappa) o tagliate (altre pagine con mappe).

Aggiunta di linee

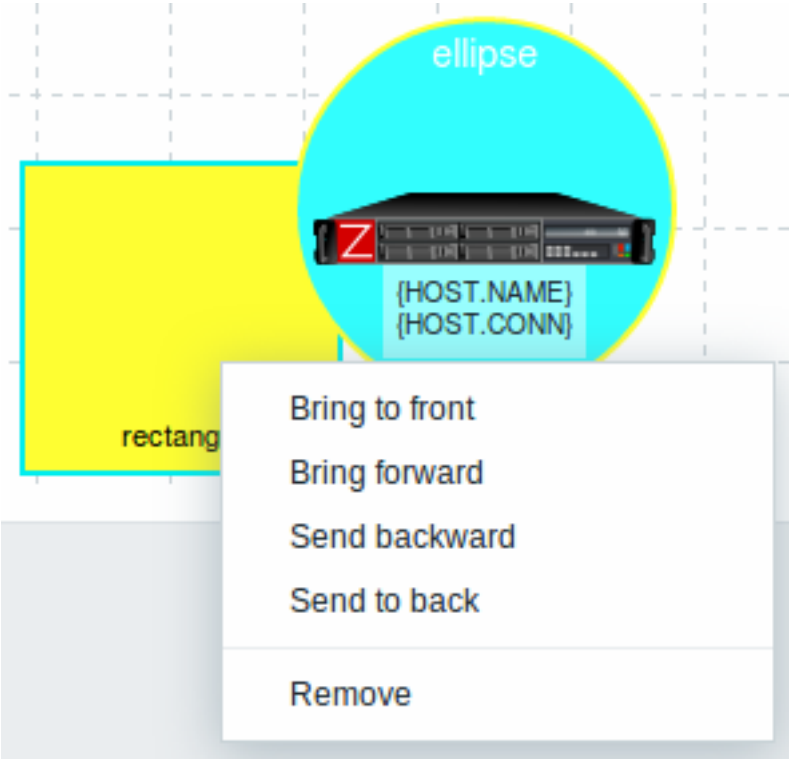
Oltre alle forme, è anche possibile aggiungere delle linee. Le linee possono essere utilizzate per collegare elementi o forme in una mappa.

Per aggiungere una linea, fare clic su *Add* accanto a Shape. Una nuova forma apparirà nell'angolo superiore sinistro della mappa. Selezionarla e fare clic su *Line* nel modulo di modifica per trasformare la forma in una linea. Quindi regolare le proprietà della linea, come il tipo di linea, la larghezza, il 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.

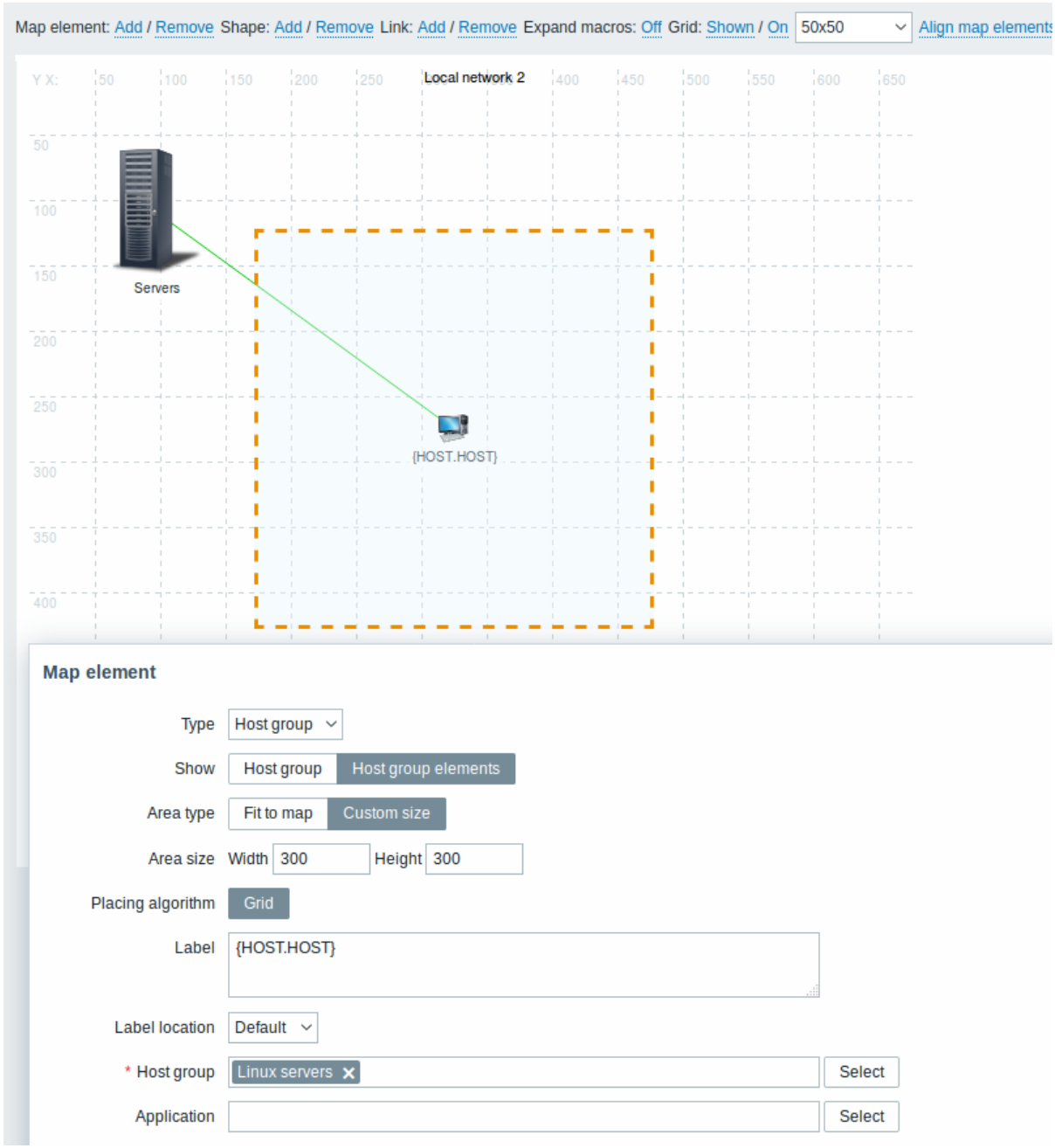


2 Elementi del gruppo host

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 con un asterisco rosso.

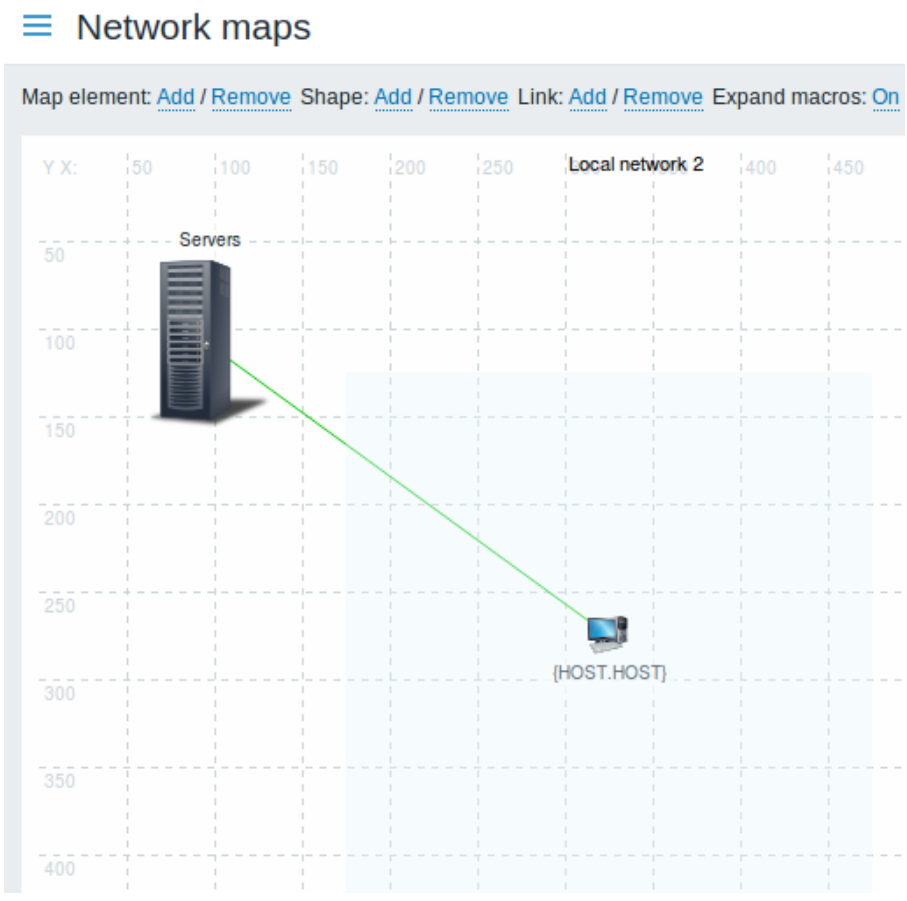
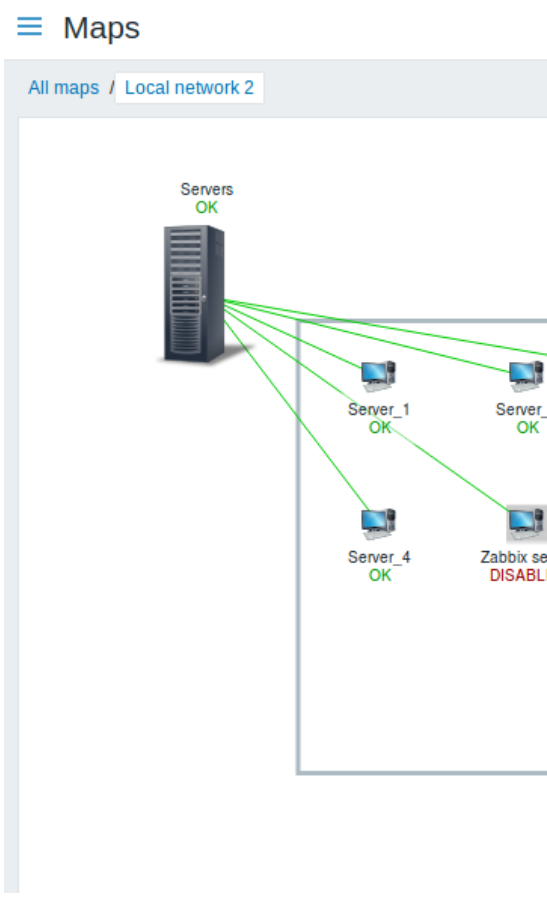
Questa tabella contiene i parametri tipici del tipo di elemento *Host group*:

Parameter	Description
Type	Selezionare il tipo dell'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 mostrata una singola icona che visualizza le informazioni corrispondenti sullo specifico gruppo di host. **Host group elements** - Selezionando questa opzione verranno mostrate più icone che visualizzano le informazioni corrispondenti su ogni singolo elemento (host) dello specifico 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 uniformemente all'interno della mappa. **Custom size** - Impostazione manuale dell'area della mappa in cui visualizzare tutti gli elementi del gruppo di host.

Parameter	Description
Area size	Questa impostazione è disponibile se *Area type* è impostato su *Custom size*: **Width** - Valore in pixel per specificare la larghezza dell'area della mappa. **Height** - Valore in pixel per specificare l'altezza dell'area della mappa.
Placing algorithm	**Grid** - Unica opzione disponibile per visualizzare tutti gli elementi del gruppo di host.
Label	Etichetta dell'icona, qualsiasi stringa. Nelle etichette possono essere utilizzate macro e stringhe multilinea. Se il tipo dell'elemento della mappa è *Host group*, la specifica di determinate macro influisce sulla vista mappa, visualizzando le informazioni corrispondenti per ogni singolo host. Ad esempio, se viene utilizzata la macro {HOST.IP}, la vista di modifica della mappa visualizzerà solo la macro {HOST.IP} stessa, mentre la vista mappa includerà e visualizzerà l'indirizzo IP univoco di ciascun host.

Visualizzazione degli elementi del gruppo host

Questa opzione è disponibile se è selezionata l'opzione di visualizzazione *Elementi del gruppo host*.
Se si seleziona *Elementi del gruppo host* come opzione *mostra*, inizialmente verrà visualizzata una sola icona per il gruppo host. Tuttavia, dopo aver salvato la mappa e aperto la vista mappa, si vedrà che la mappa include tutti gli elementi (host) del determinato gruppo host:

Vista di modifica della mappa	Vista mappa
	

Si noti come venga utilizzata la macro {HOST.HOST}.
Durante la modifica della mappa, il nome della macro non viene risolto, mentre nella vista mappa vengono visualizzati tutti i nomi univoci degli host.

3 Indicatori di collegamento

Panoramica

È possibile 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 entrano in 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 si configura un collegamento, si impostano 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 entra in uno stato di problema, lo stile e il colore del collegamento cambieranno per rifletterlo. Quindi, ad esempio, il collegamento predefinito poteva essere una linea verde. Ora, con il trigger nello stato di problema, il collegamento può diventare rosso e in grassetto (se lo si è definito così).

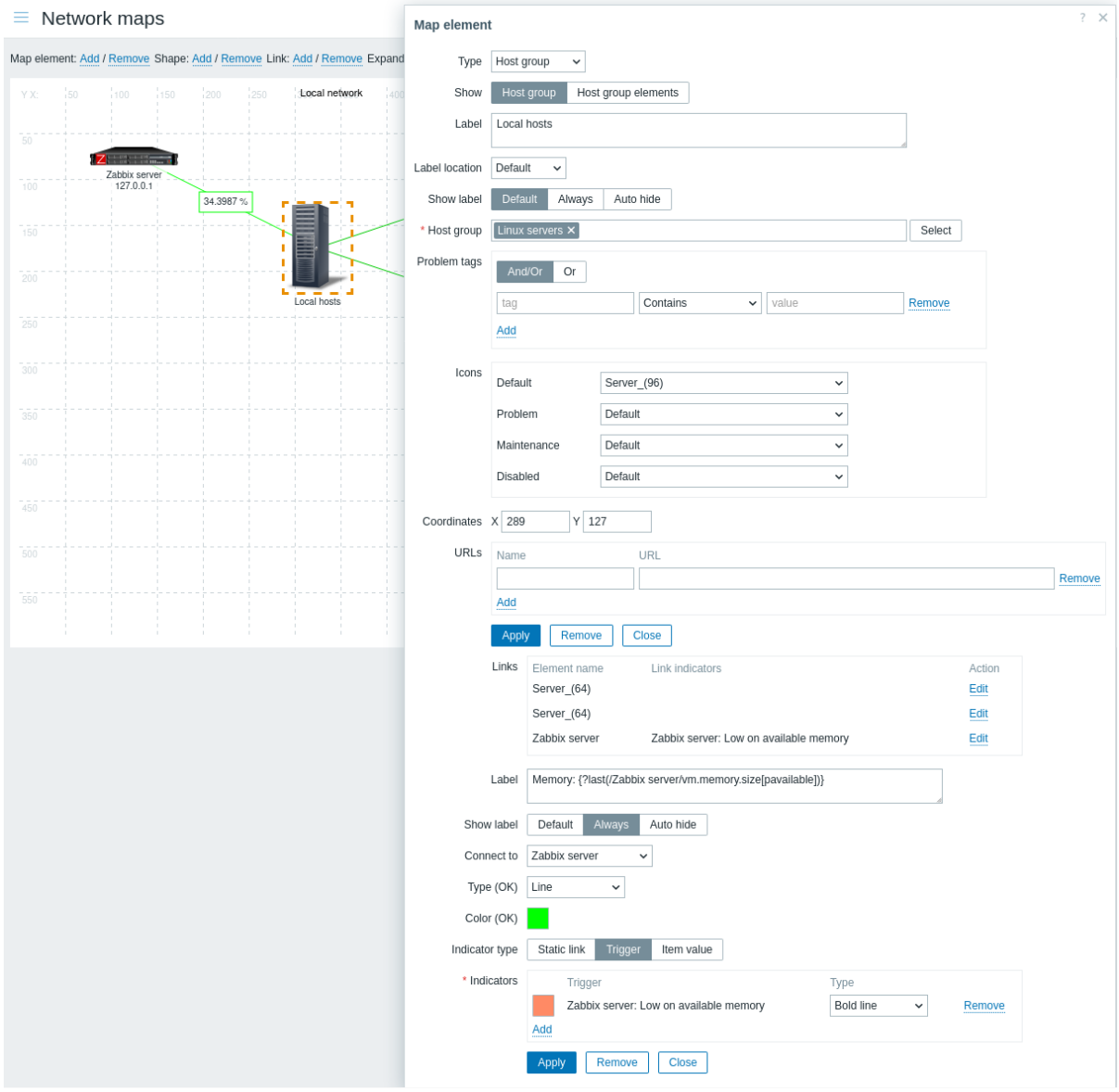
Allo stesso modo, se il valore di un item raggiunge una soglia specificata o corrisponde a un'espressione regolare specificata, lo stile del collegamento può rifletterlo.

Configurazione

Trigger

Per assegnare i trigger come indicatori dei collegamenti, procedere come segue:

- selezionare un elemento della mappa
- fare clic su *Modifica* nella sezione *Collegamenti* per il collegamento appropriato
- selezionare *Trigger* come tipo di indicatore
- fare clic su *Aggiungi* nel blocco *Indicatori* e selezionare uno o più trigger

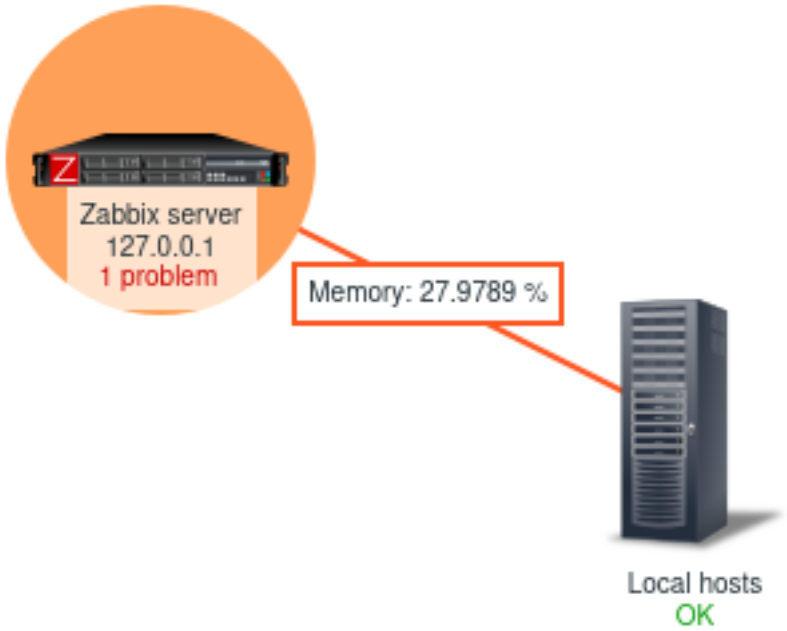


Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

I trigger aggiunti possono essere visualizzati nell'elenco *Indicatori*.

È possibile impostare il tipo e il colore del collegamento per ciascun indicatore direttamente dall'elenco. Al termine, fare clic su *Applica*, chiudere il modulo e fare clic su *Aggiorna* per salvare le modifiche alla mappa.

In *Monitoraggio* → *Mappe* verranno visualizzati il tipo e il colore del collegamento corrispondenti se il **trigger** passa a uno stato di problema.



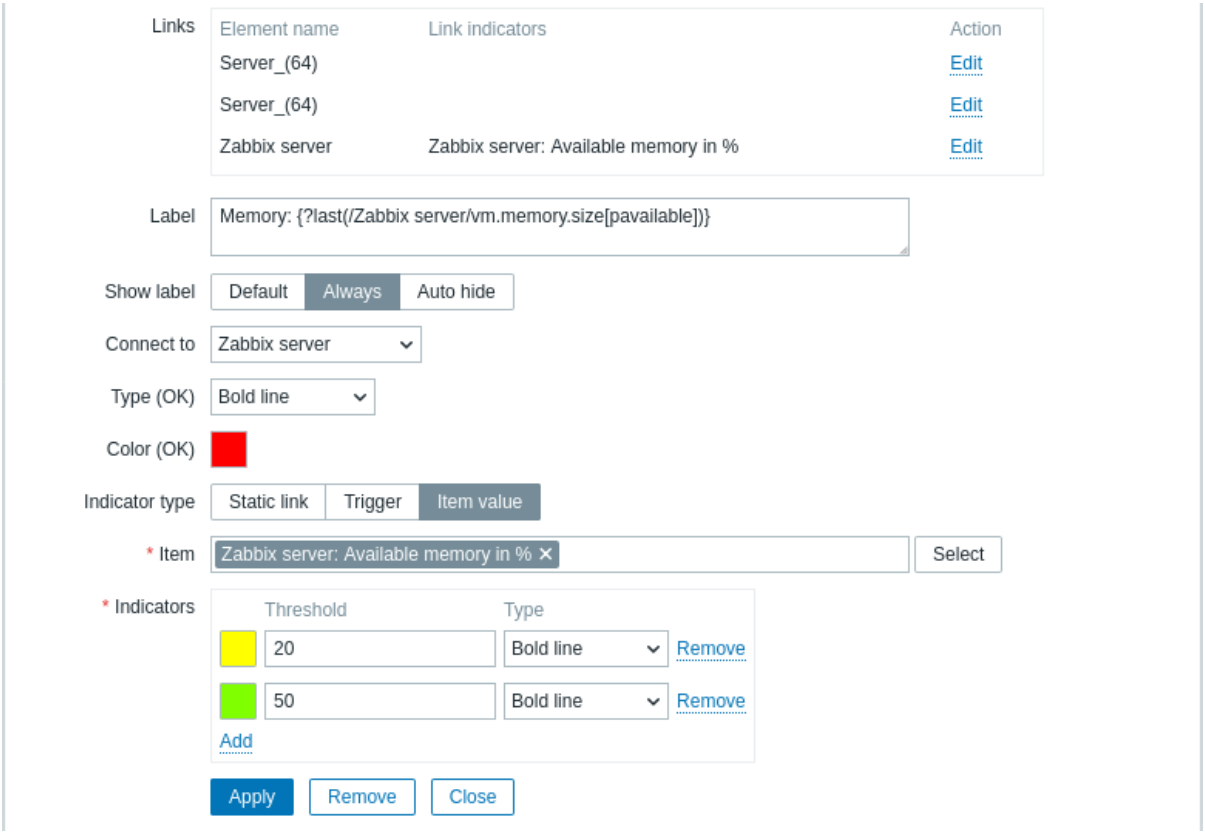
Note:
Se più trigger passano a uno stato di problema, il problema con la gravità più alta determinerà lo stile e il colore del collegamento. Se più trigger con la stessa gravità sono assegnati allo stesso collegamento della mappa, ha la precedenza quello con l'ID più basso. Si noti inoltre che:

1. Le impostazioni *Gravità minima del trigger* e *Mostra problema soppresso* della configurazione della mappa influiscono su quali problemi vengono presi in considerazione.
2. Nel caso di trigger con più problemi (generazione di problemi multipli), ogni problema può avere una gravità diversa da quella 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 dei collegamenti, procedere come segue:

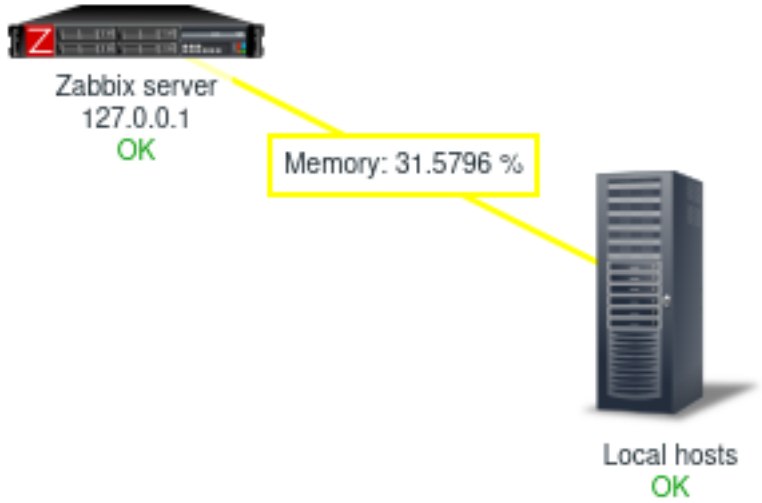
- selezionare un elemento della mappa
- fare clic su *Modifica* nella sezione *Collegamenti* per il collegamento appropriato
- selezionare *Valore item* come tipo di indicatore
- selezionare l'item
- aggiungere una o più soglie o pattern di valore dell'item nel blocco *Indicatori*



Le soglie/i pattern degli item aggiunti possono essere visualizzati nell'elenco *Indicatori*.

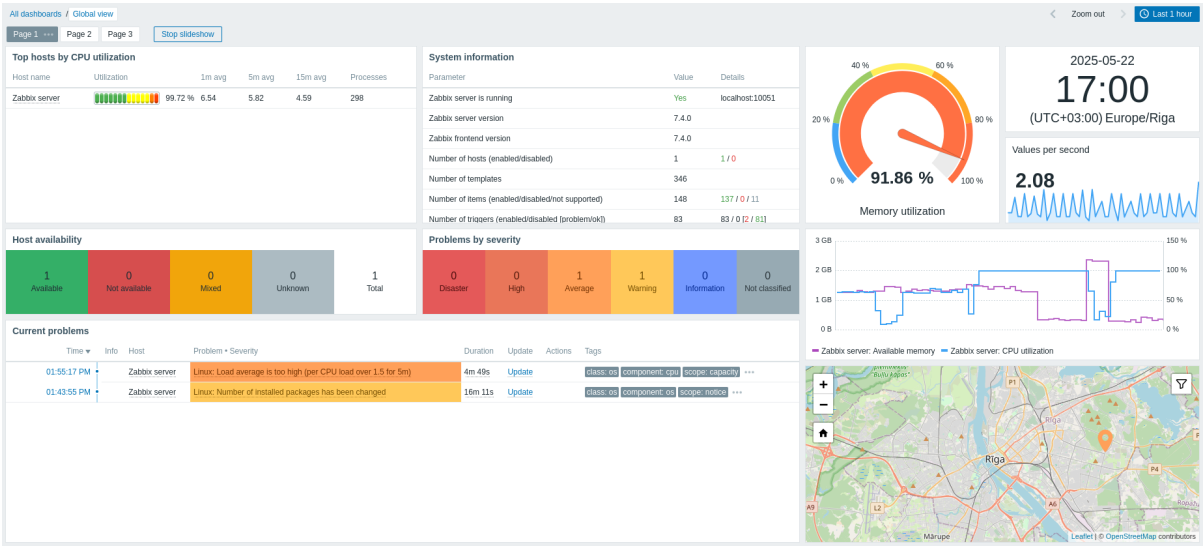
È possibile impostare il tipo e il colore del collegamento per ciascun indicatore direttamente dall'elenco. Al termine, fare clic su *Applica*, chiudere il modulo e fare clic su *Aggiorna* per salvare le modifiche alla mappa.

In *Monitoraggio* → *Mappe* verranno visualizzati il tipo e il colore del collegamento corrispondenti se il **valore dell'item** raggiunge la soglia impostata (per i tipi di dati numerici) o corrisponde al pattern di espressione regolare (per i tipi di dati testuali).



3 Dashboard

Le dashboard - sia i dashboard globali che i dashboard host - offrono una potente piattaforma di visualizzazione con widget e strumenti quali grafici moderni, mappe, presentazioni e altro.



8 Modelli e gruppi di modelli

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.

9 Modelli pronti all'uso

Panoramica

Zabbix fornisce un insieme in continua crescita di template preconfigurati per semplificare e velocizzare la configurazione dei target di monitoraggio.

Tutti i template disponibili out-of-the-box sono accessibili in *Raccolta dati* > *Template*.

Durante l'aggiornamento di Zabbix, i template esistenti non vengono aggiornati automaticamente per evitare la sovrascrittura di modifiche personalizzate. Per aggiornare i template esistenti o aggiungerne di nuovi, vedere Aggiornamento dei template.

Utilizzare la barra laterale per accedere alle informazioni sui tipi specifici di template e sui requisiti operativi.

Vedere anche:

- Collegamento di un template
- Problemi noti dei template

Aggiornamento dei template

Per aggiornare un singolo template o aggiungerne uno nuovo dopo un aggiornamento di Zabbix:

1. Vai al repository Git di Zabbix.
2. Nel selettore del branch (sotto l'etichetta *Source*), scegli il branch che corrisponde alla tua versione di Zabbix (ad esempio, *master* per la versione di sviluppo più recente).
3. Apri il template che vuoi aggiungere o aggiornare (ad esempio, MySQL by Zabbix agent).
4. Copia il contenuto del file del template (ad esempio, `template_db_mysql_agent.yaml`) e salvalo in un file locale.
5. In Zabbix frontend, vai su *Data collection* > *Templates* e importa il file locale.

Puoi anche aggiornare tutti i template e aggiungere quelli nuovi in un'unica volta:

1. Scarica il bundle dei template in base al tuo web server: `templates_1M_8.0.7z` (Nginx) oppure `templates_2M_8.0.7z` (Apache).
2. Estrai il bundle. Contiene file YAML (ciascuno con più template) e un indice (`index-1M_80.md`) che elenca i template inclusi in ciascun file YAML.
3. In Zabbix frontend, vai su *Data collection* > *Templates* e importa i file YAML uno per uno.

In alternativa, puoi importare tutti i file YAML in una sola volta usando lo script `import_templates.sh`. Lo script richiede le utility `curl` e `jq`, e utilizza il metodo API di Zabbix `configuration.import`. Quando esegui lo script, fornisci la directory contenente i file YAML, nonché URL e token di autenticazione della tua API Zabbix.

> **Warning:**
> Lo script `import_templates.sh` sovrascrive tutti i template esistenti, senza alcuna opzione per rivedere o annullare le modifiche ai template.

Esempio:

```
./import_templates.sh /tmp/templates_2M_8.0 https://example.com/zabbix/api_jsonrpc.php d8d6b5c78ee2a8333db
```

Compatibilità dei template con Host Wizard

Il Host Wizard è compatibile con tutti i template forniti out-of-the-box.

Puoi anche rendere compatibili con Host Wizard i tuoi template personalizzati:

1. Nel frontend di Zabbix, vai su *Raccolta dati* > *Template* ed esporta il template personalizzato.
2. Modifica il file esportato:
   - Aggiungi l'elemento del template `wizard_ready` per indicare la compatibilità con Host Wizard.
   - Facoltativamente, aggiungi l'elemento del template `readme` o `config`, secondo necessità.
3. Nel frontend di Zabbix, vai su *Raccolta dati* > *Template* e importa il file del template aggiornato.

1 Funzionamento dei template di 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 prima importare il template).
3. Se necessario, regolare i valori delle macro del template.
4. Configurare l'istanza monitorata in modo da 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 di un template).

Sono disponibili i seguenti template:

- Apache tramite Zabbix agent

- Apache tramite Zabbix agent active
- HAProxy tramite Zabbix agent
- IIS tramite Zabbix agent
- IIS tramite Zabbix agent active
- Microsoft Exchange Server 2016 tramite Zabbix agent
- Microsoft Exchange Server 2016 tramite Zabbix agent active
- MySQL tramite Zabbix agent
- MySQL tramite Zabbix agent active
- Nginx tramite Zabbix agent
- Nginx tramite Zabbix agent active
- PHP-FPM tramite Zabbix agent
- PHP-FPM tramite Zabbix agent active
- PostgreSQL tramite Zabbix agent
- PostgreSQL tramite Zabbix agent active
- Cluster RabbitMQ tramite Zabbix agent

2 Funzionamento dei template di 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 sul host e che la versione installata contenga il plugin richiesto. In alcuni casi, potrebbe essere necessario prima aggiornare agent 2.

2. Collegare il template a un host di destinazione (se il template non è disponibile nella propria installazione di Zabbix, potrebbe essere necessario prima importare il template).

3. Se necessario, regolare i valori delle macro del template. Si noti che le macro utente possono essere utilizzate per sovrascrivere i parametri di configurazione.

4. Configurare l'istanza monitorata in modo da consentire la condivisione dei dati con Zabbix.

> **Attention:**
> I template di Zabbix agent 2 funzionano insieme ai plugin. Sebbene la configurazione di base possa essere eseguita semplicemente regolando le macro utente, una personalizzazione più approfondita può essere ottenuta configurando il plugin stesso. 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, 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:

- Ceph by Zabbix agent 2
- Docker
- Memcached
- MongoDB cluster by Zabbix agent 2
- MongoDB node by Zabbix agent 2
- MySQL by Zabbix agent 2
- MySQL by Zabbix agent 2 active
- Oracle by Zabbix agent 2
- PostgreSQL by Zabbix agent 2
- PostgreSQL by Zabbix agent 2 active
- Redis
- SMART by Zabbix agent 2
- SMART by Zabbix agent 2 active
- Systemd by Zabbix agent 2
- Website certificate by Zabbix agent 2

3 Funzionamento dei template HTTP

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche con HTTP agent:

1. Creare un host in Zabbix che corrisponda alla destinazione di monitoraggio. Specificare l'indirizzo IP/nome DNS della destinazione di monitoraggio 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 nella propria installazione di Zabbix, potrebbe essere necessario prima importare il template).

3. Se necessario, regolare i valori delle macro a livello di template.

4. Configurare l'istanza monitorata in modo da 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 di un 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
- 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
- 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
- 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
- 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

4 Funzionamento dei 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, incluso l'elenco completo di macro, item e trigger, è disponibile nel file README del template (accessibile facendo clic sul nome del template).

Template disponibile:

- Chassis by IPMI

5 Funzionamento dei template JMX

Passaggi per garantire il corretto funzionamento dei template che raccolgono metriche tramite JMX:

1. Assicurarsi che Java gateway di Zabbix sia installato e configurato correttamente.
2. Collegare il template al host di destinazione. Il host deve avere l'interfaccia JMX configurata. Se il template non è disponibile nella propria installazione di Zabbix, potrebbe essere necessario prima importare il template.

3. Se necessario, modificare i valori delle macro del template.

4. Configurare l'istanza monitorata in modo da 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 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

6 Funzionamento dei 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 sullo Zabbix server o proxy.

2. Collegare il template a un host di destinazione (se il template non è disponibile nella propria installazione di Zabbix, potrebbe essere necessario prima importare il template).

3. Se necessario, modificare i valori delle macro del template.

4. Configurare l'istanza monitorata in modo da 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:

- MariaDB by ODBC
- MSSQL by ODBC
- MySQL by ODBC
- Oracle by ODBC
- Percona by ODBC
- PostgreSQL by ODBC

7 Modelli standardizzati per dispositivi di rete

Panoramica

Per fornire il monitoraggio dei dispositivi di rete come switch e router, abbiamo creato due cosiddetti modelli: per il dispositivo di rete stesso (in pratica il suo chassis) e per l'interfaccia di rete.

Sono forniti template per molte famiglie di dispositivi di rete. Tutti i template coprono (ove possibile ottenere questi item dal dispositivo):

- Monitoraggio dei guasti dello chassis (alimentatori, ventole e temperatura, stato complessivo)
- 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 di traffico dell'interfaccia, stato duplex per Ethernet)

Se il template non è disponibile nella tua installazione di Zabbix, potrebbe essere necessario prima importare il template.

Se stai importando i nuovi template predefiniti, potresti voler aggiornare anche l'espressione regolare globale `@Network interfaces for discovery` in:

```
Result is FALSE: ^Software Loopback Interface
Result is FALSE: ^(In)?[lL]oop[bB]ack[0-9._]*$
Result is FALSE: ^NULL[0-9.]*$
```

```
Result is FALSE: ^[lL]o[0-9.]*$
Result is FALSE: ^[sS]ystem$
Result is FALSE: ^Nu[0-9.]*$
```

per escludere 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
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

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
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
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

Nome del template	Fornitore	Famiglia di dispositivi	Modelli noti	OS	MIB utilizzati	**Tag**
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

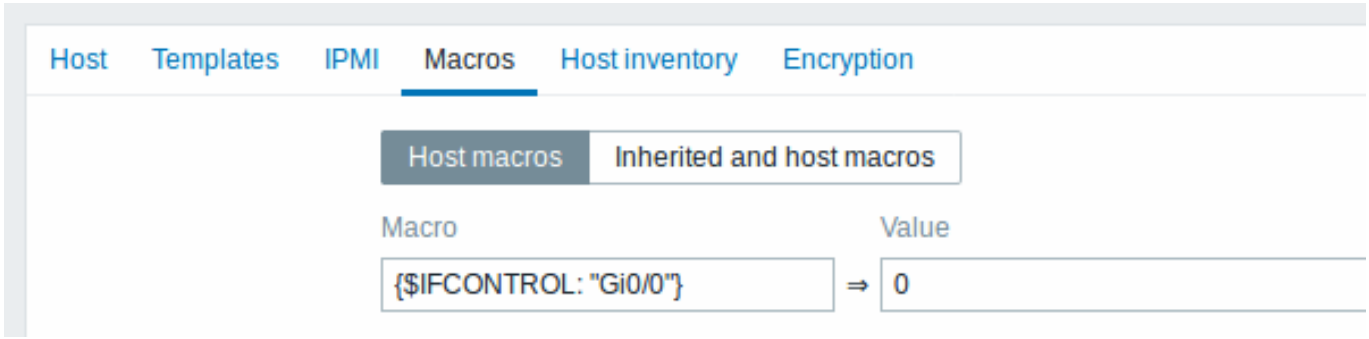
I template sono stati progettati tenendo presenti i seguenti aspetti:

- 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:

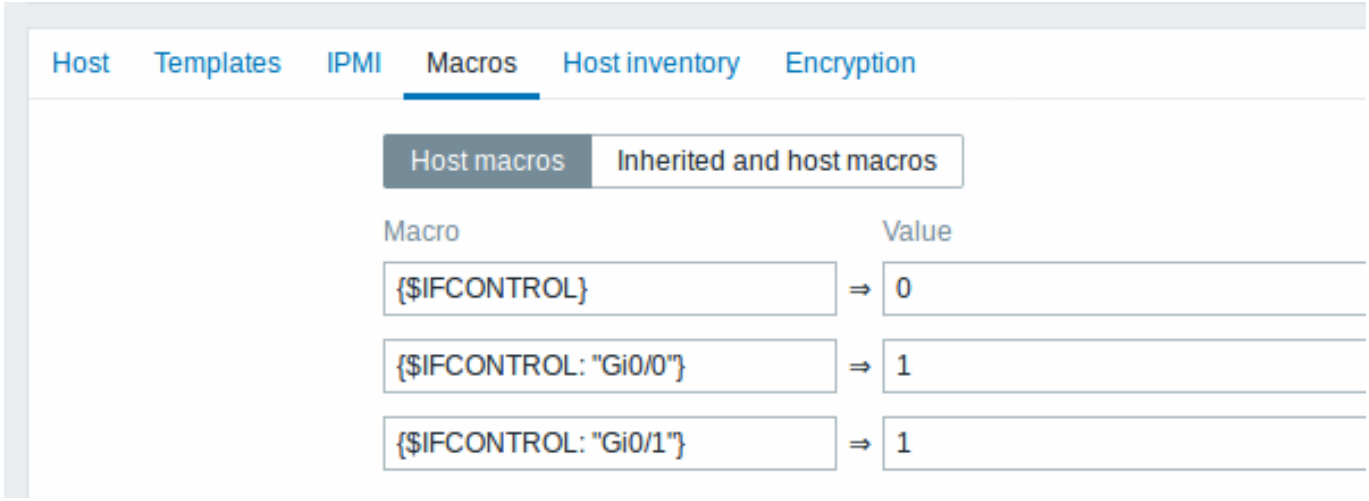
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- 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 – i MIB della famiglia di dispositivi forniscono un modo per monitorare gli item di CPU e memoria;
- Guasti - i MIB della famiglia di dispositivi forniscono un modo per monitorare almeno un sensore di temperatura;
- Inventario – i 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 principali categorie sopra indicate sono coperte.

8 Funzionamento 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

10 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.

1 Tipi di supporto

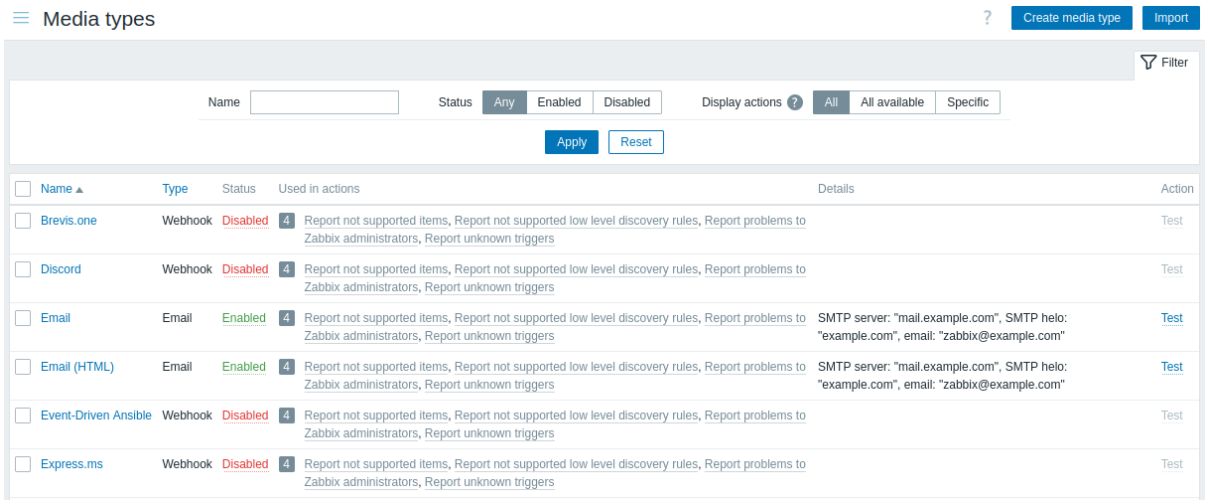
Panoramica

I tipi di supporto sono i canali di consegna utilizzati per inviare notifiche e avvisi da Zabbix.

I tipi di supporto supportano i seguenti metodi di consegna:

- Email
- SMS
- Script personalizzato
- Webhook

I tipi di supporto sono gestiti in *Avvisi* > *Tipi di supporto*. Alcuni tipi di supporto sono predefiniti nel dataset predefinito.
È sufficiente ottimizzarne i parametri per renderli funzionanti.



Per vedere come i tipi di supporto si inseriscono nel processo di avviso, esaminiamo i tre requisiti per recapitare notifiche da Zabbix agli utenti finali:

1. Deve essere definita un'operazione di azione che invii notifiche
2. Deve essere definito un tipo di supporto funzionante (ad esempio *Email*, che invia avvisi tramite SMTP)
3. I dettagli di consegna a livello utente (come indirizzi e-mail, numeri di telefono, ecc.) devono essere definiti nei supporti utente

Test del tipo di supporto

Per verificare se un tipo di supporto configurato funziona, fai clic su *Test* nell'elenco dei tipi di supporto.

La richiesta di test verrà inviata al server Zabbix. Il server Zabbix tenterà di inviare un avviso utilizzando il tipo di supporto specificato e restituirà il risultato al frontend. Il frontend attenderà che il server restituisca i risultati. Il test del tipo di supporto ha un timeout predefinito di 65 secondi (configurabile in *Administration* > *General* > *Timeouts*).

Vedi anche i dettagli del test per:

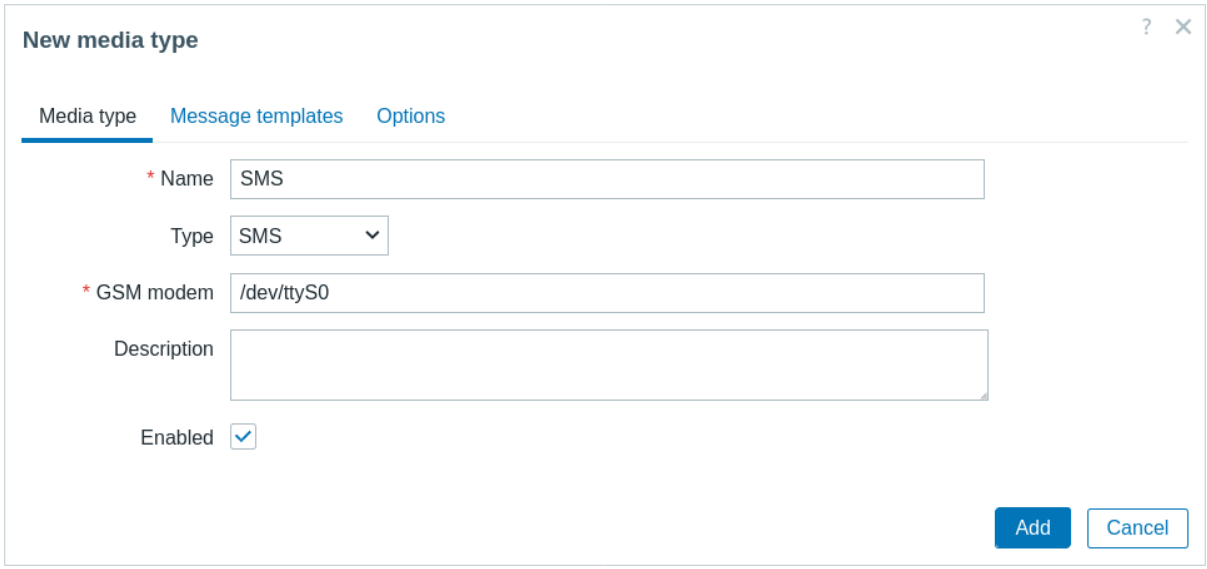
- Email
- Webhook
- Script

Configurazione

Per creare un tipo di supporto nel frontend di Zabbix:

- Vai a: *Avvisi* > *Tipi di supporto*
- Fai clic su *Crea tipo di supporto*
- Inserisci i parametri del tipo di supporto nel modulo

Alcuni parametri sono **comuni** a tutti i metodi di consegna.

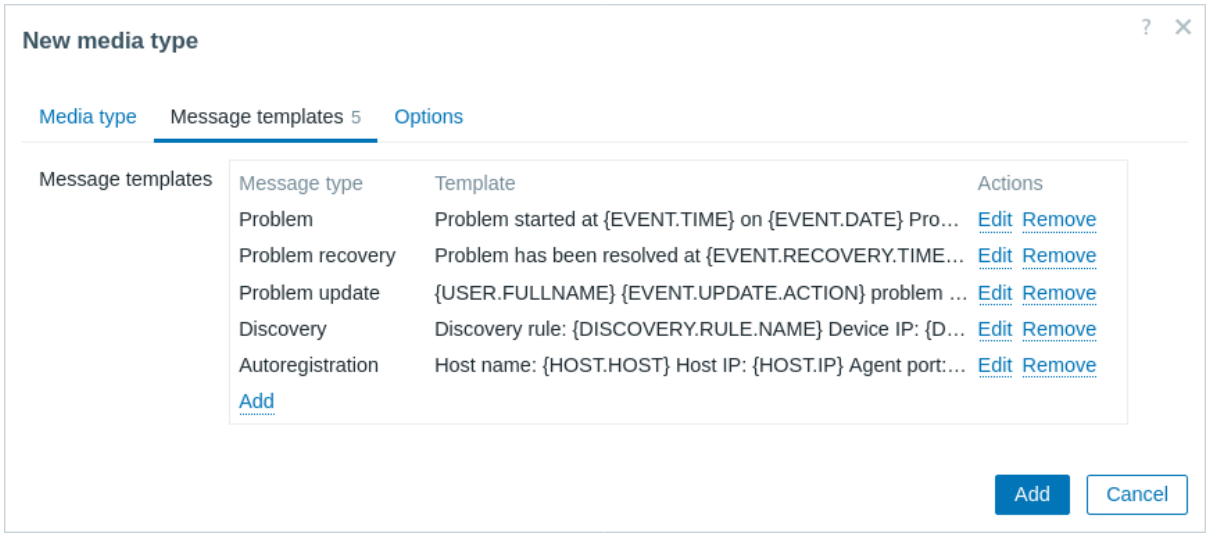


Parametro	Descrizione
Nome	Nome del tipo di supporto.
Tipo	Seleziona il metodo di consegna per il tipo di supporto.
Descrizione	Inserisci una descrizione del tipo di supporto.
Abilitato	Seleziona la casella di controllo per abilitare il tipo di supporto.

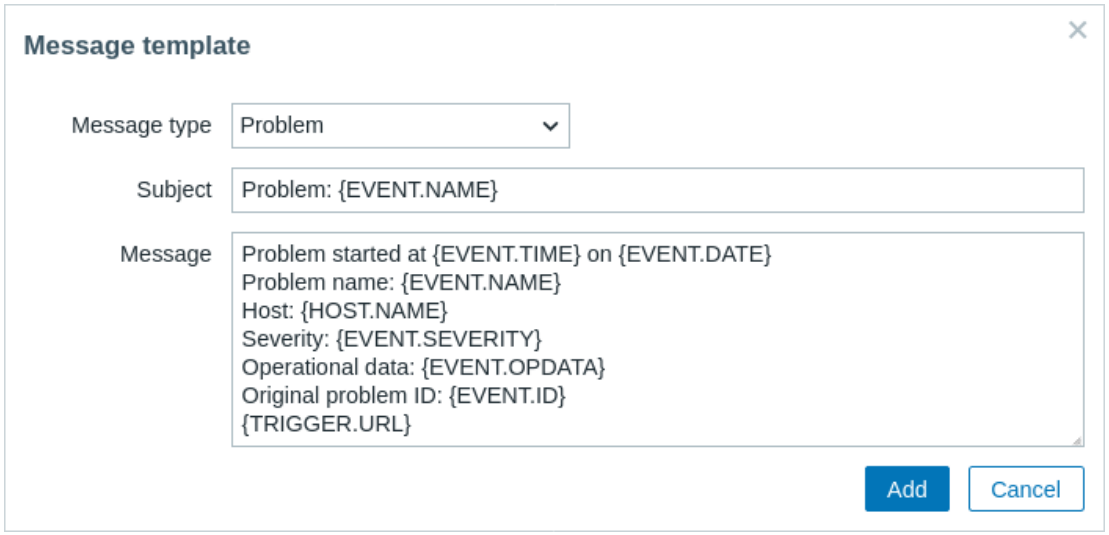
Per i parametri specifici del metodo, consulta le pagine email, SMS, script di avviso personalizzato o webhook.

Modelli di messaggio

La scheda **Modelli di messaggio** contiene i messaggi predefiniti per tipo di evento (problema, ripristino del problema, rilevamento, ecc.).



Fare clic su **Aggiungi** per definire un messaggio predefinito (oppure su **Modifica** per aggiornare un messaggio esistente):



Parametro	Descrizione
Tipo di messaggio	Tipo di evento per il quale deve essere utilizzato il messaggio predefinito. Per ogni tipo di evento può essere definito un solo messaggio predefinito.
Oggetto	Oggetto del messaggio predefinito. L'oggetto può contenere macro. È limitato a 255 caratteri. L'oggetto non è disponibile per il tipo di supporto SMS.
Messaggio	Il messaggio predefinito. È limitato a un certo numero di caratteri a seconda del tipo di database (per maggiori informazioni, vedere Invio di messaggi). 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)}`).

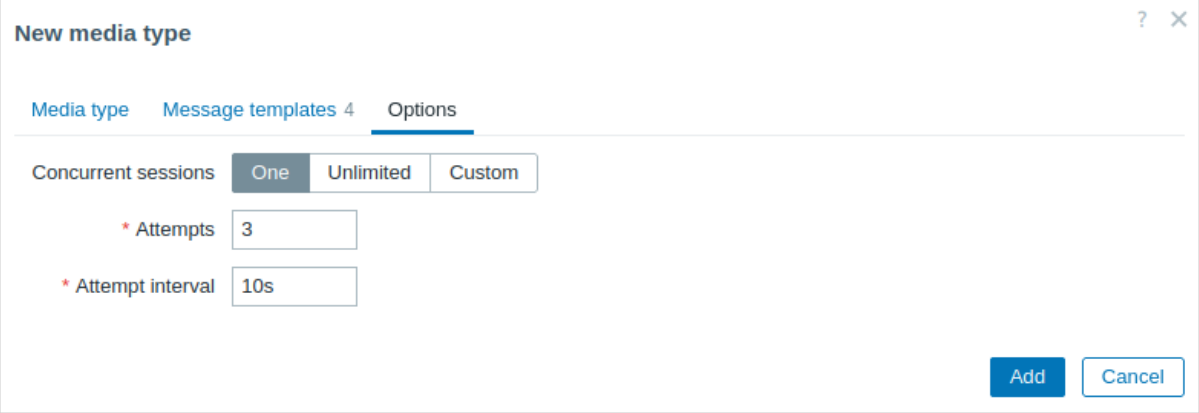
Si noti che i messaggi predefiniti vengono sovrascritti dai messaggi personalizzati, se definiti nelle operazioni dell'azione.

Warning:
La definizione dei modelli di messaggio è obbligatoria per tutti i metodi di consegna, inclusi webhook o script di avviso personalizzati che non utilizzano i messaggi predefiniti per le notifiche. Ad esempio, l'azione "Invia messaggio al webhook Pushover" 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 di elaborazione degli avvisi.
Lo stesso insieme di opzioni è configurabile per ciascun tipo di supporto.

Tutti i tipi di supporto vengono elaborati in parallelo.
Mentre il numero massimo di sessioni simultanee è configurabile per ciascun tipo di supporto,
il numero totale di processi alerter sul server può essere limitato solo dal parametro StartAlerters.
Gli avvisi generati da un trigger vengono elaborati in modo sequenziale.
Pertanto, più notifiche possono essere elaborate contemporaneamente solo se sono generate da più trigger.



Parametro	Descrizione
Sessioni simultanee	Selezionare il numero di sessioni alerter parallele per il tipo di supporto: **Una** - una sessione **Illimitate** - numero illimitato di sessioni **Personalizzate** - selezionare un numero personalizzato di sessioni Valori illimitati/elevati significano più sessioni parallele e una maggiore capacità di invio delle notifiche. I valori illimitati/elevati dovrebbero essere utilizzati in ambienti di grandi dimensioni in cui potrebbe essere necessario inviare molte notifiche contemporaneamente. Se devono essere inviate più notifiche rispetto al numero di sessioni simultanee disponibili, le notifiche rimanenti verranno messe in coda; non andranno perse.
Tentativi	Numero di tentativi di 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 riproverà in caso di errore di invio.
Intervallo tra i tentativi	Frequenza con cui si tenta di reinviare una notifica in caso di errore di invio, in secondi (0-3600). Se viene specificato '0', Zabbix ritenterà immediatamente. Sono supportati i suffissi temporali, ad esempio 5s, 3m, 1h.

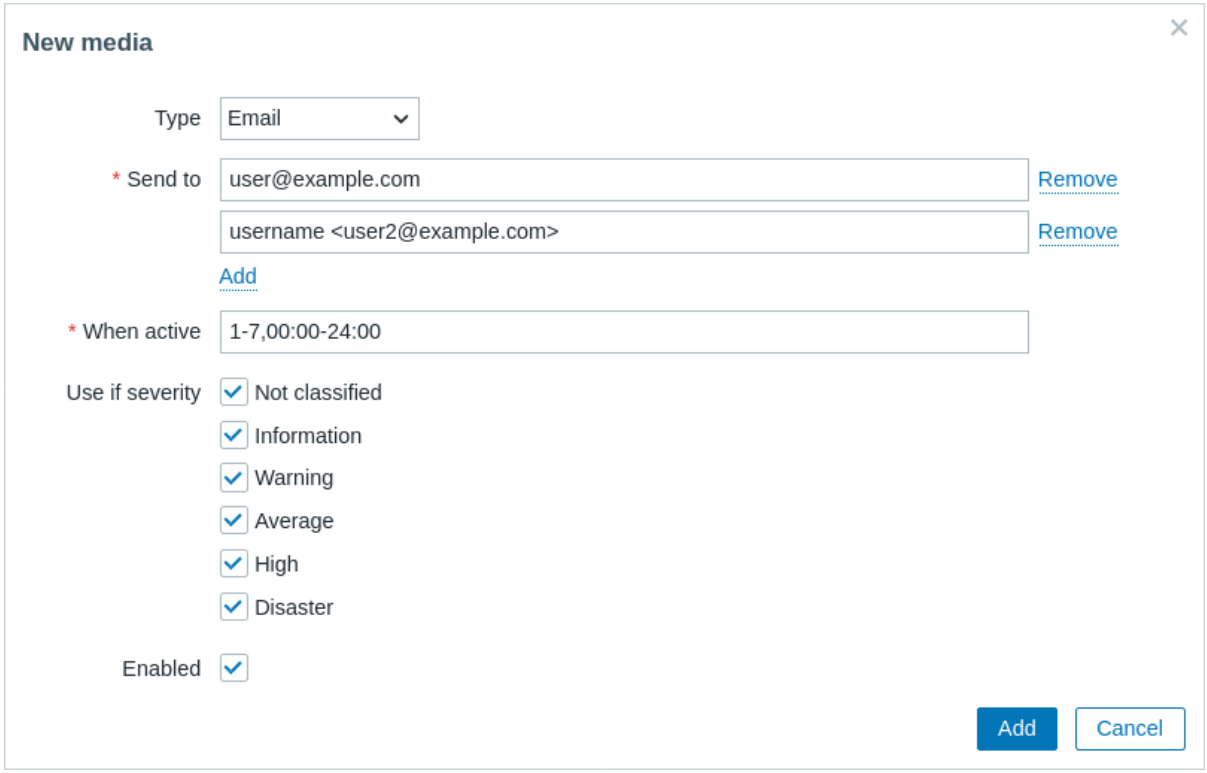
Supporti utente

Mentre i tipi di supporto definiscono **come** verrà inviata una notifica, i supporti utente definiscono **dove** la notifica deve essere inviata.

I supporti utente (ad esempio indirizzo email, ID utente webhook, ecc.) devono essere definiti nel profilo 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 utente di Admin.

Per definire i supporti utente:

- Vai su *Users* > *Users* e apri il modulo delle proprietà dell'utente (oppure vai su *User settings* > *Notifications* del tuo profilo utente)
- Fai clic su *Add* nella scheda Media



Parametro	Descrizione
Type	L'elenco a discesa contiene i nomi dei tipi di supporto abilitati. Nota che quando si modifica un supporto di un tipo di supporto disabilitato, il tipo verrà visualizzato in rosso.
Send to	Inserisci le informazioni di contatto a cui devono essere inviati i messaggi. Per il tipo di supporto 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, vedi la descrizione del parametro *Email* per il tipo di supporto email.
When active	Puoi limitare l'orario in cui i messaggi vengono inviati, 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 dal fuso orario del sistema, potrebbe essere necessario regolare di conseguenza questo limite per non perdere messaggi importanti. Vedi la pagina Specifica del periodo di tempo per la descrizione del formato. Le macro utente sono supportate.
Use if severity	Seleziona le caselle di controllo delle severità dei trigger per le quali desideri ricevere notifiche. *Nota* che la severità predefinita ('Not classified') **deve essere** selezionata se vuoi ricevere notifiche per eventi non relativi ai trigger. Dopo il salvataggio, le severità dei trigger selezionate verranno visualizzate con i colori di severità corrispondenti, mentre quelle non selezionate saranno visualizzate in grigio.
Status	Stato del supporto utente. **Enabled** - è in uso. **Disabled** - non è in uso.

1 E-mail

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.

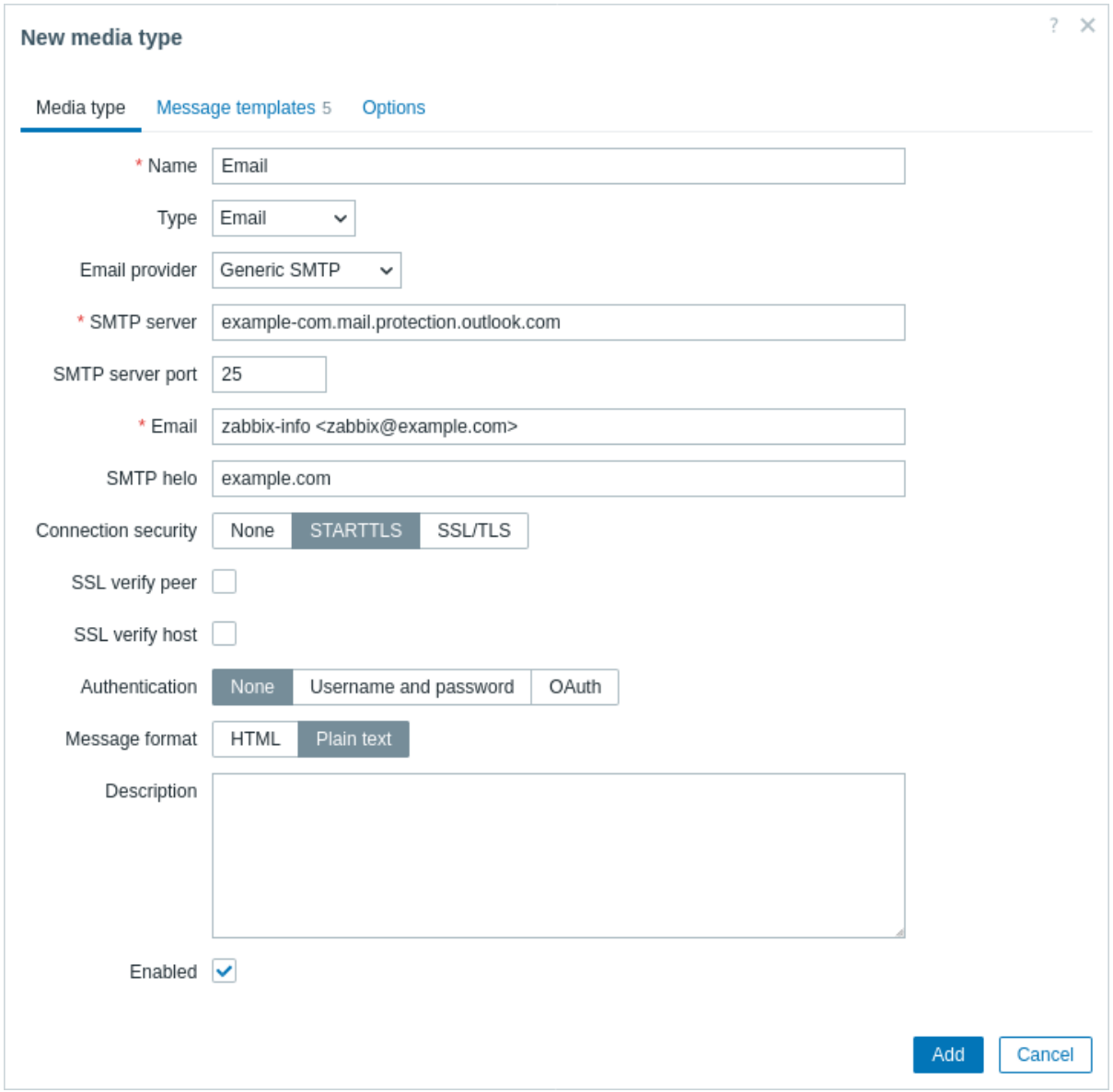
> **Note:**
> Le notifiche multiple per un singolo evento verranno raggruppate nello stesso thread email.

Configurazione

Per configurare l'email come tipo di supporto:

1. Vai in *Avvisi* > *Tipi di supporto*.
2. Fai clic su *Crea tipo di supporto* (oppure fai clic su *Email* nell'elenco dei tipi di supporto predefiniti).

La scheda **Tipo di supporto** contiene gli attributi generali del tipo di supporto:



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

Warning:
La password viene esportata in chiaro quando si esporta la configurazione del tipo di supporto email.

I seguenti parametri sono specifici del tipo di supporto email:

Parametro	Descrizione
Provider email	Seleziona il provider email: *SMTP generico*, *Gmail*, *Relay Gmail*, *Office365* oppure *Relay Office365*. Se selezioni le opzioni relative a Gmail/Office365, dovrai fornire solo l'indirizzo email del mittente e la password; opzioni come *Server SMTP*, *Porta 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 come provider email è selezionato *SMTP generico*.
Porta server SMTP	Imposta la porta del server SMTP per gestire i messaggi in uscita. Questo campo è disponibile se come provider email è selezionato *SMTP generico*.

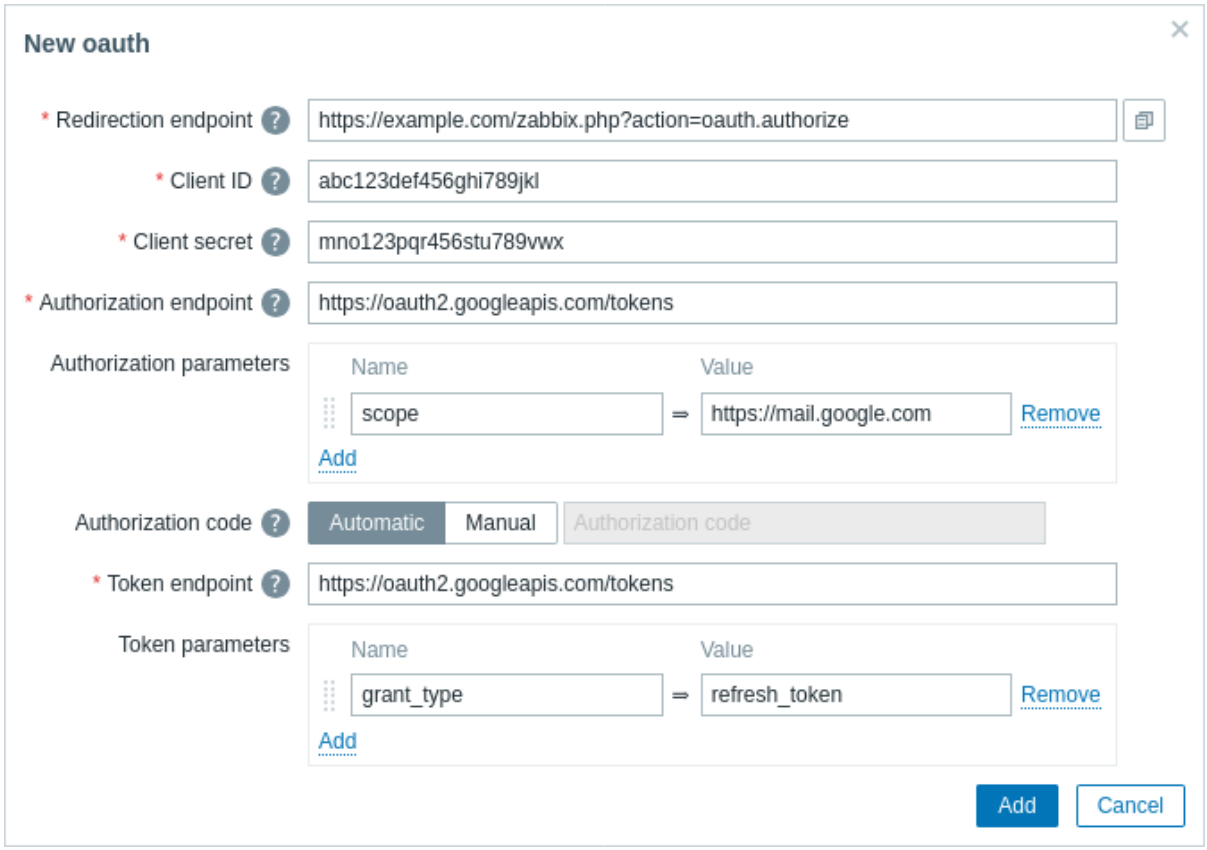
Parametro	Descrizione
Email	L'indirizzo inserito qui verrà usato come indirizzo **Da** per i messaggi inviati. È supportata l'aggiunta di un nome visualizzato del mittente (come "Zabbix_info" in *Zabbix_info <zabbix@company.com>* nello screenshot sopra) insieme all'indirizzo email effettivo. Esistono alcune limitazioni per i nomi visualizzati nelle email di Zabbix rispetto a quanto consentito da RFC 5322, come illustrato dagli esempi: Esempi validi: *zabbix@company.com* (solo indirizzo email, non è necessario usare le parentesi angolari) *Zabbix_info <zabbix@company.com>* (nome visualizzato e indirizzo email tra parentesi angolari) *∑Ω-monitoring <zabbix@company.com>* (caratteri UTF-8 nel nome visualizzato) Esempi non validi: *Zabbix HQ zabbix@company.com* (nome visualizzato presente ma senza parentesi angolari attorno 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 di Zabbix)
SMTP helo	Imposta un valore SMTP helo corretto, normalmente un nome di dominio. Se vuoto, verrà inviato il nome di dominio dell'email (cioè ciò che segue @ nel campo *Email*). Se non è possibile recuperare il nome di dominio, verrà registrato un avviso a livello debug e il nome host del server verrà inviato come dominio per il comando HELO. Questo campo è disponibile se come provider email è selezionato *SMTP generico*.
Sicurezza della connessione	Seleziona il livello di sicurezza della connessione: **Nessuna** - non usare l'opzione CURLOPT_USE_SSL **STARTTLS** - usa l'opzione CURLOPT_USE_SSL con valore CURLUSESSL_ALL **SSL/TLS** - l'uso 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. Questo 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. Questo imposta l'opzione cURL CURLOPT_SSL_VERIFYHOST.
Autenticazione	Seleziona il livello di autenticazione: **Nessuna** - non viene impostata alcuna opzione 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 *Relay Office365*.
Nome utente	Nome utente da usare per l'autenticazione. Questo imposta il valore di CURLOPT_USERNAME. Macro utente supportate.
Password	Password da usare per l'autenticazione. Questo imposta il valore di CURLOPT_PASSWORD. Macro utente supportate. 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, usa invece una macro utente (ad esempio `{$EMAIL_SMTP_PASSWORD}`) al posto di una password letterale: verrà esportato il riferimento alla macro, mentre il valore segreto dovrà essere (ri)configurato 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, Zabbix server deve essere compilato con l'opzione di compilazione `--with-libcurl`, che richiede libcurl (vedi i requisiti di server per i dettagli sulla versione).

Vedi anche parametri comuni del tipo di supporto per i dettagli su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi.

Token OAuth

Per recuperare i token OAuth sono richiesti i seguenti parametri:



Parametro	Descrizione
Endpoint di reindirizzamento	Inserire l'URL del frontend di Zabbix a cui il servizio OAuth reindirizzerà l'autorizzazione OAuth (utilizzando il formato `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Viene impostato automaticamente per un nuovo tipo di supporto email quando è definito l'URL del frontend.
ID client	Inserire l'identificatore univoco dell'app client registrata nel server di autorizzazione OAuth.
Segreto client	Inserire il segreto privato dell'app client registrata nel server di autorizzazione OAuth.
Endpoint di autorizzazione	Inserire l'URL del server di autorizzazione OAuth per richiedere l'autorizzazione dell'utente.
Parametri di autorizzazione	Inserire i parametri per l'endpoint di autorizzazione.
Codice di autorizzazione	Inserire il codice di autorizzazione: **Automatico** - il codice verrà recuperato automaticamente tramite una pagina di reindirizzamento **Manuale** - inserire il codice manualmente se il recupero automatico non riesce
Endpoint del token	Inserire l'URL del server di autorizzazione OAuth per scambiare il codice di autorizzazione con token di accesso e di aggiornamento.
Parametri del token	Inserire 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 *Endpoint di reindirizzamento*, *ID client* e *Segreto client*. Zabbix compilerà automaticamente gli altri valori richiesti (vedere Valori predefiniti degli URL OAuth per 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.

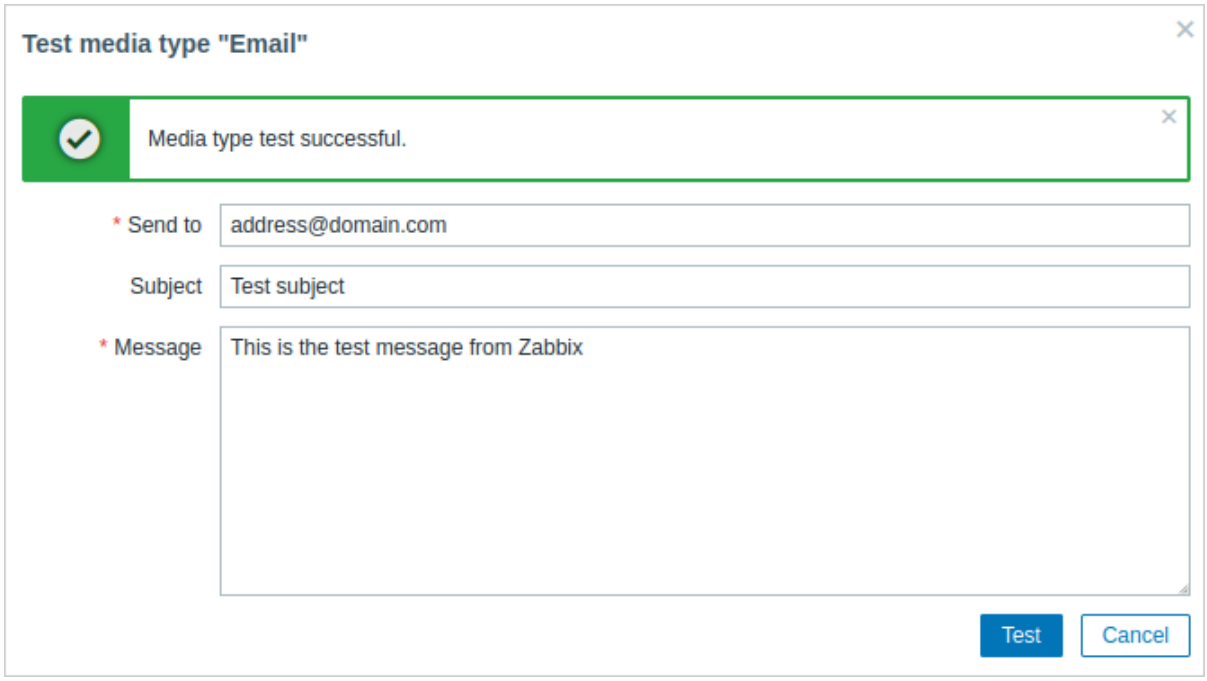
3. Il servizio OAuth reindirizza l'utente all'azione `oauth.authorize` del frontend di Zabbix con il codice di autorizzazione e il valore dello scope.
4. In risposta, Zabbix invierà una richiesta all'*endpoint del token* per scambiare il codice di autorizzazione con i token di accesso e di aggiornamento.

Test

Per verificare se un tipo di supporto email configurato funziona correttamente:

1. Individua l'email pertinente nell'elenco dei tipi di supporto.
2. Fai clic su *Test* nell'ultima colonna dell'elenco (si aprirà una finestra di test).
3. Inserisci un indirizzo del 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:



Supporti utente

Una volta configurato il tipo di supporto email, vai alla sezione *Users* > *Users* e modifica il profilo utente per assegnare il supporto email all'utente. I passaggi per configurare i supporti utente, comuni a tutti i tipi di supporto, sono descritti nella pagina Media types.

1 Tipi di supporto Gmail/Office365 automatizzati

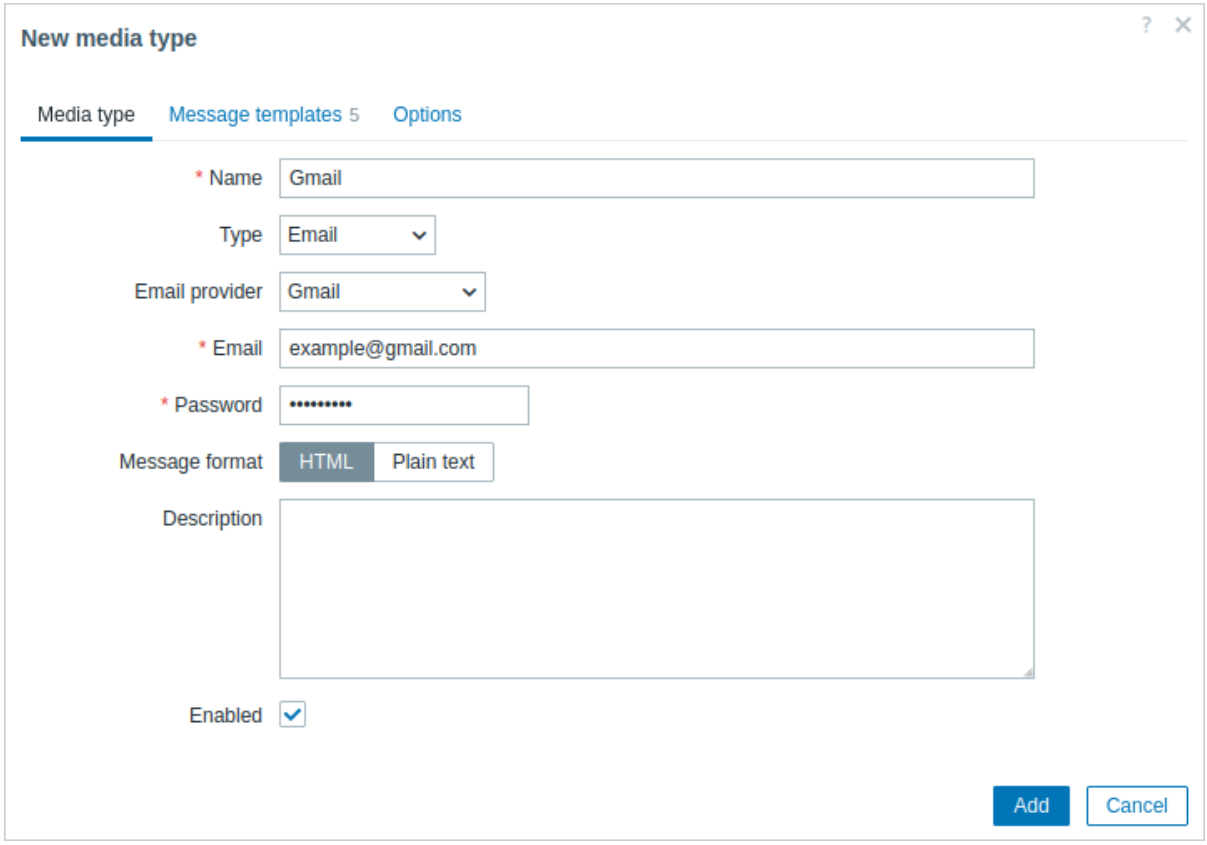
Panoramica

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.

Se si selezionano le opzioni relative a Gmail/Office365, è necessario fornire solo l'indirizzo email/la password del mittente per creare un tipo di supporto funzionante.



Non appena vengono forniti l'indirizzo email e la password, Zabbix sarà in grado di compilare automaticamente tutte le impostazioni richieste per i tipi di supporto Gmail/Office365 con i valori effettivi/consigliati, cioè *Server SMTP*, *Porta del server SMTP*, *SMTP helo* e *Sicurezza della connessione*. Grazie a questa automazione, questi campi non vengono nemmeno mostrati; tuttavia, è possibile vedere il server SMTP e i dettagli dell'email nell'elenco dei tipi di supporto (vedere la colonna *Dettagli*).

Si noti inoltre che:

- La password non è richiesta 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 (cioè sostituire "example.com" in `example-com.mail.protection.outlook.com` con il valore reale).

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.

2 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 conservi dopo un 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 recapito 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 SMS come tipo di supporto:

- Vai a *Avvisi* → *Tipi di supporto*
- Fai clic su *Crea tipo di supporto* (oppure fai clic su *SMS* nell'elenco dei tipi di supporto predefiniti).

I seguenti parametri sono specifici per il tipo di supporto SMS:

Parameter	Description
Modem GSM	Imposta il nome del dispositivo seriale del modem GSM. Il percorso inserito qui verrà convalidato rispetto al parametro del server SMSDevices (se specificato).

Per i dettagli su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi, consulta i parametri comuni del tipo di supporto. Tieni presente che l'elaborazione parallela dell'invio delle notifiche SMS non è possibile.

Supporti utente

Una volta configurato il tipo di supporto SMS, vai alla sezione *Utenti* → *Utenti* e modifica il profilo utente per assegnare il supporto SMS all'utente. I passaggi per configurare i supporti utente, comuni a tutti i tipi di supporto, sono descritti nella pagina Tipi di supporto.

3 Script di avviso personalizzati

Panoramica

Se non sei soddisfatto dei tipi di supporto esistenti per l'invio degli avvisi, esiste un modo alternativo per farlo. Puoi creare uno script che gestisca la notifica nel modo che preferisci.

Gli script di avviso personalizzati vengono eseguiti su Zabbix server. Questi script devono trovarsi nella directory specificata dal parametro AlertScriptsPath nel file di configurazione del server.

Ecco un esempio di script di avviso 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.

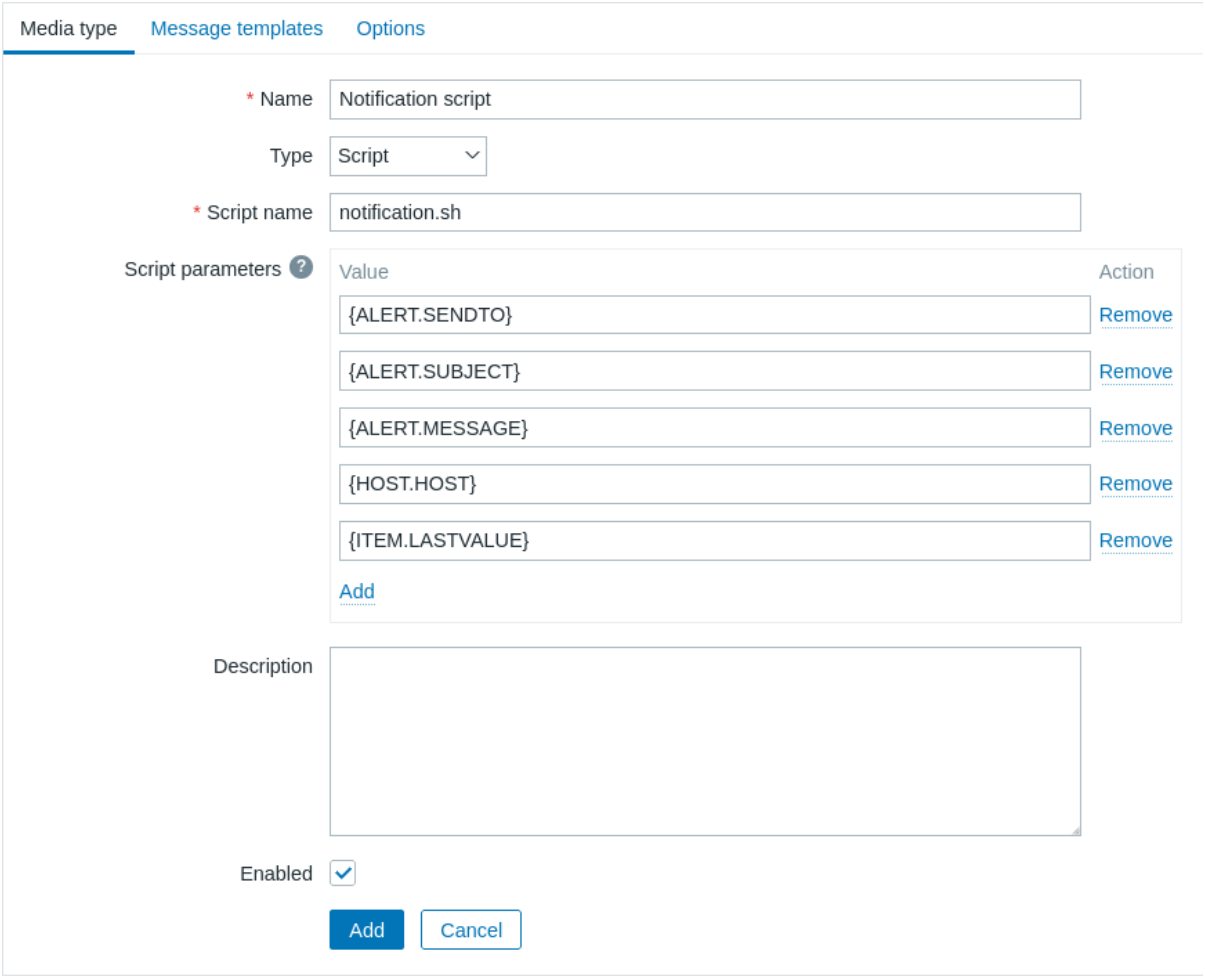
Le variabili d'ambiente non vengono mantenute né create per lo script, quindi devono essere gestite esplicitamente.

Configurazione

Per configurare script di avviso personalizzati come tipo di supporto:

1. Andare su *Avvisi → Tipi di supporto*.
2. Fare clic su *Crea tipo di supporto*.

La scheda **Tipo di supporto** contiene gli attributi generali del tipo di supporto:



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

I seguenti parametri sono specifici per il tipo di supporto script:

Parameter	Description
Nome script	Inserire il nome del file di script (ad esempio, notification.sh) che si trova nella directory specificata nel parametro di configurazione del server AlertScriptsPath.
Parametri script	Aggiungere parametri opzionali dello script 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 i dettagli su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi, vedere i parametri comuni del tipo di supporto.

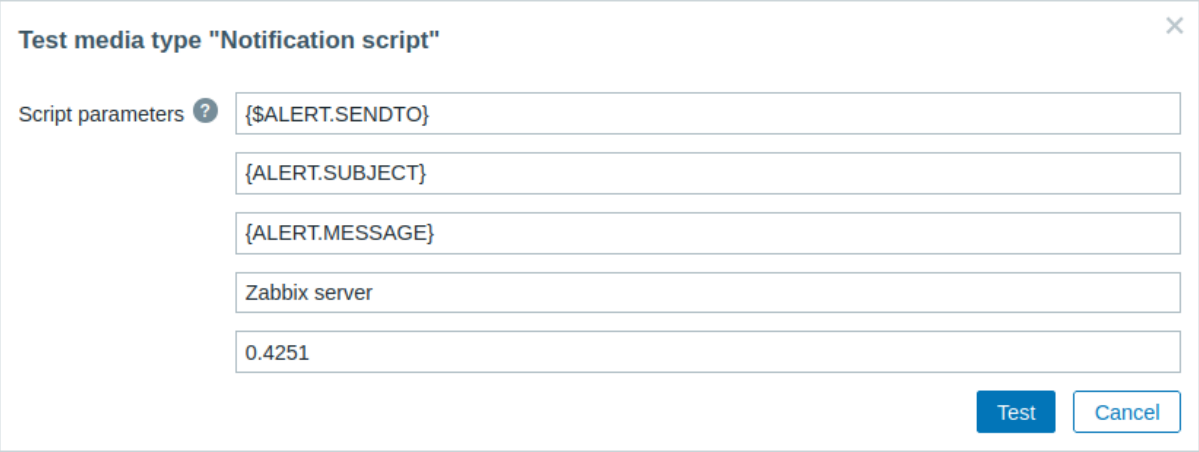
Warning:
Anche se uno script di avviso non utilizza messaggi predefiniti, i template di messaggio per i tipi di operazione utilizzati da questo tipo di supporto devono comunque essere definiti. In caso contrario, una notifica non verrà inviata.

Attention:
Se è configurato più di un tipo di supporto script, questi script possono essere elaborati in parallelo dai processi alerter. Il numero totale di processi alerter è limitato dal parametro `StartAlerters` del file di configurazione del server.

Test

Per testare un tipo di supporto script configurato:

1. Individuare lo script pertinente nell'elenco dei tipi di supporto.
2. Fare 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 tipo di supporto script.
3. Modificare i valori dei parametri dello script, se necessario. La modifica influisce solo sulla procedura di test; i valori effettivi non verranno modificati.
4. Fare clic su *Test*.



Note:
Quando si testa un tipo di supporto script configurato, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} e le macro utente verranno risolte nei rispettivi valori, ma le macro relative agli eventi (ad esempio, {HOST.HOST}, {ITEM.LASTVALUE}, ecc.) non verranno risolte, poiché durante il test non esiste alcun evento correlato da cui ottenere i dettagli. Si noti 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 "Problema: {EVENT.NAME}", la macro {EVENT.NAME} non verrà risolta.

Media utente

Una volta configurato il tipo di supporto, vai alla sezione *Utenti* → *Utenti* e modifica il profilo di un utente assegnando questo tipo di supporto all'utente. I passaggi per configurare i media utente, comuni a tutti i tipi di supporto, sono descritti nella pagina Tipi di supporto.

Tieni presente che, quando definisci i media utente, il campo *Invia a* non può essere vuoto. Se questo campo non viene utilizzato nello script di avviso, inserisci una qualsiasi combinazione di caratteri supportati per aggirare i requisiti di convalida.

4 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 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
- 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 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

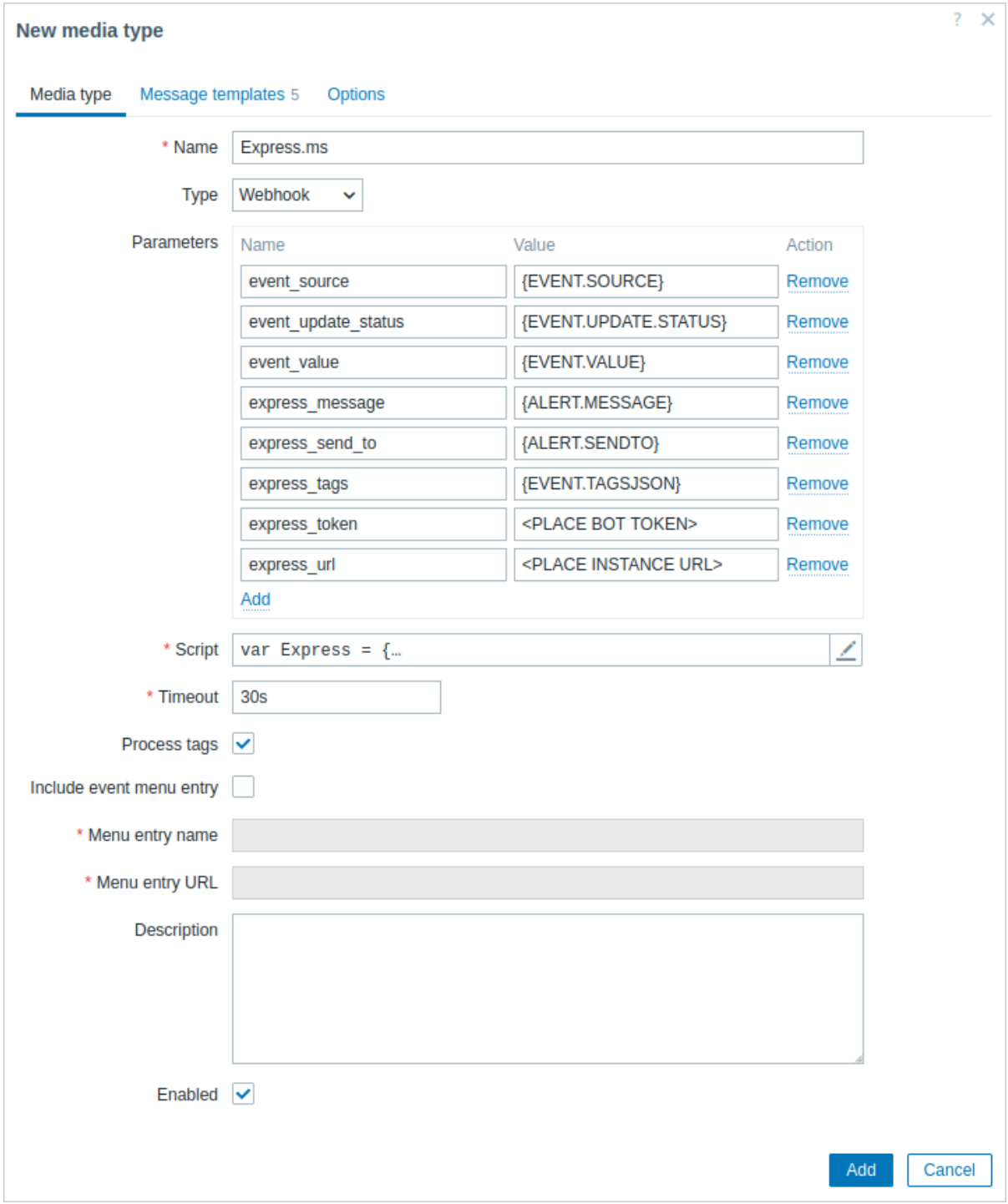
Per iniziare a utilizzare un'integrazione webhook:

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 nella propria installazione di Zabbix. Il webhook apparirà nell'elenco dei tipi di supporto.
3. Configurare il webhook secondo le istruzioni nel file *Readme.md* (è possibile fare clic sul nome di un webhook sopra per accedere rapidamente a *Readme.md*).

Per creare un webhook personalizzato da zero:

1. Andare in *Alerts* > *Media types*.
2. Fare clic su *Create media type*.

La scheda **Media type** contiene vari attributi specifici per questo tipo di supporto:



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

I seguenti parametri sono specifici per il tipo di supporto webhook:

Parameter	Description
Parameters	Specificare le variabili del webhook come coppie attributo-valore. Per i webhook preconfigurati, l'elenco dei parametri varia a seconda del 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 liberamente. 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 la stessa funzionalità del campo HTTP proxy nella configurazione dell'item. La stringa del proxy può essere preceduta da `[scheme]://` per specificare quale tipo di proxy viene 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 della matita accanto ad esso. Questo codice eseguirà l'operazione del webhook. Lo script è un codice 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, supporta metodi aggiuntivi come CONNECT, PATCH, HEAD, OPTIONS e TRACE e consente il controllo delle intestazioni HTTP e del corpo della richiesta. Lo script deve contenere un operatore return, altrimenti non sarà valido. Può restituire uno stato OK insieme a un elenco facoltativo di tag e valori di tag (vedere l'opzione *Process tags*) oppure una stringa di errore. Gli eventi di ripristino (sia generati automaticamente sia come risultato di una chiusura manuale) vengono creati dal server e includono i tag dell'evento risolto (compresi i tag ereditati da templates, hosts e triggers). 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 collisioni tra chiamate simultanee (vedere i problemi noti). Vedere anche: Linee guida per lo sviluppo di webhook, Esempi di script webhook, Oggetti JavaScript aggiuntivi.
Timeout	Timeout di esecuzione di JavaScript (1-60s, predefinito 30s). I suffissi temporali sono supportati, ad esempio 30s, 1m.
Process tags	Selezionare la casella per elaborare i valori delle proprietà JSON restituite come tag. Questi tag vengono aggiunti a eventuali tag di problema esistenti. Si noti che 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>*
Include event menu entry	Selezionare la casella per includere una voce nel menu evento che colleghi a un ticket esterno creato. Verrà inclusa una voce per ogni webhook abilitato che abbia questa casella selezionata. Si noti che se i parametri *Menu entry name* e *Menu entry URL* contengono macro {EVENT.TAGS.<tag name>}, una voce verrà inclusa solo se queste macro possono essere risolte (ossia se per l'evento sono definiti questi tag). Se selezionata, il webhook non deve essere utilizzato per inviare notifiche a utenti diversi (valutare invece la creazione di un utente dedicato) e non deve essere utilizzato in più azioni di avviso per un singolo evento di problema.
Menu entry name	Specificare il nome della voce di menu. La macro {EVENT.TAGS.<tag name>} è supportata. Questo campo è obbligatorio solo se *Include event menu entry* è selezionato.

Parameter	Description
Menu entry URL	Specificare l'URL sottostante della voce di menu. La macro {EVENT.TAGS.<tag name>} è supportata. Questo campo è obbligatorio solo se *Include event menu entry* è selezionato.

Per i dettagli su come configurare i messaggi predefiniti e le opzioni di elaborazione degli avvisi, vedere parametri comuni del tipo di supporto.

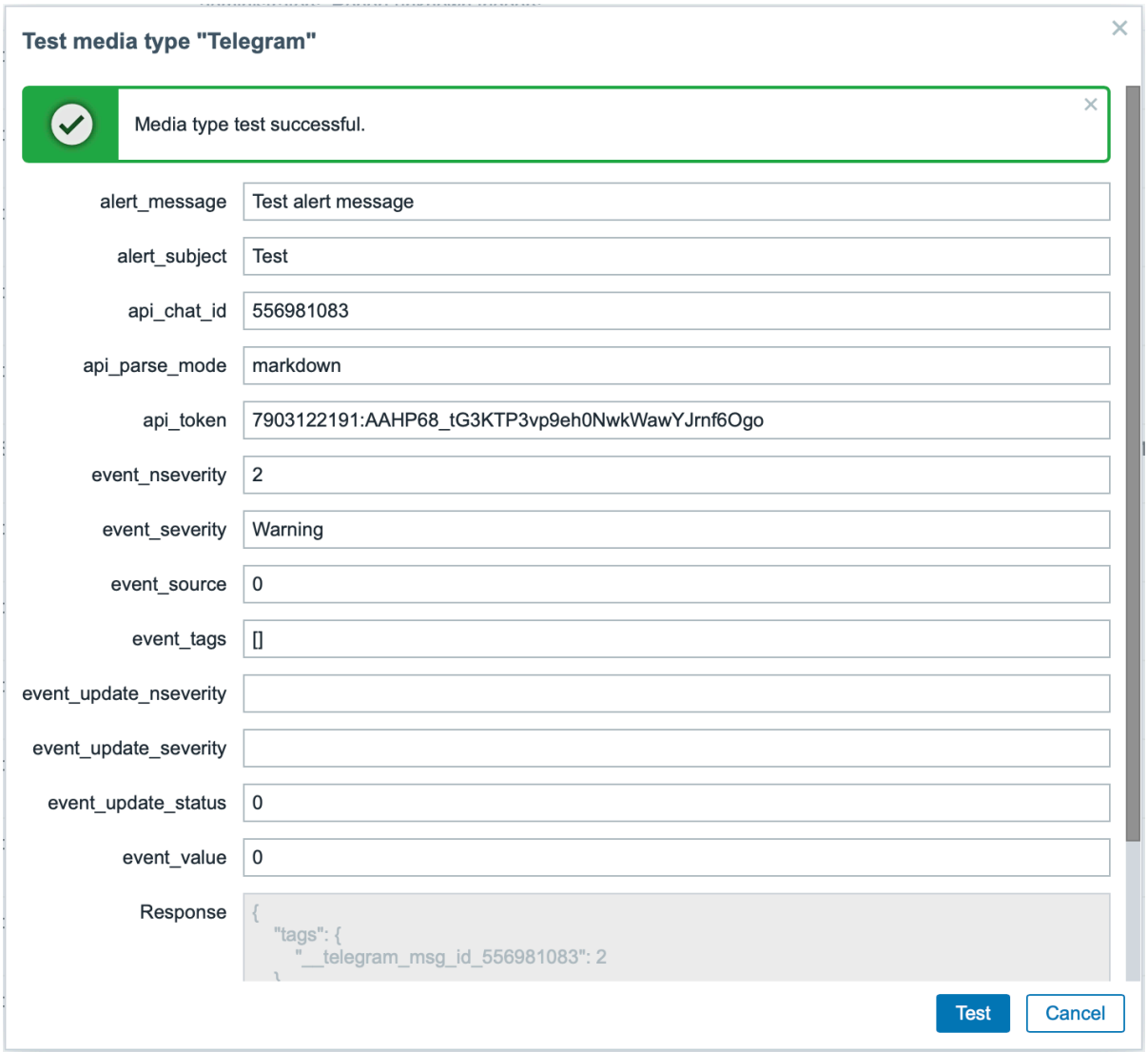
Warning:
Anche se un webhook non utilizza i messaggi predefiniti, i template di messaggio per i tipi di operazione utilizzati da questo webhook devono comunque essere definiti.

Test

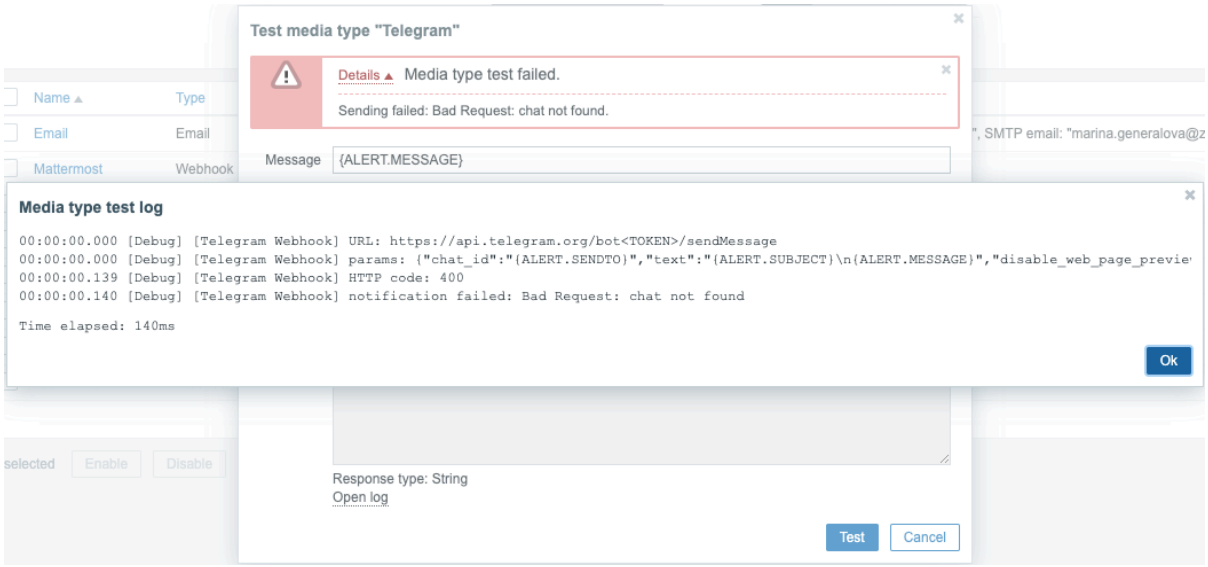
Per testare un tipo di supporto webhook configurato:

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 non riuscirà.
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 log di test del tipo di supporto senza uscire dalla finestra di test, fare clic su *Open log* (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 utente

Una volta configurato il tipo di supporto, vai alla sezione *Utenti* > *Utenti* e assegna il supporto webhook a un utente esistente oppure crea un nuovo utente che rappresenti il webhook. I passaggi per configurare i media utente per un utente esistente, comuni a tutti i tipi di supporto, sono descritti nella pagina Tipi di supporto.

Se un webhook utilizza i tag per memorizzare l'ID del ticket\messaggio, evita di assegnare lo stesso webhook come supporto a utenti diversi, poiché ciò potrebbe causare errori del webhook (si applica alla maggior parte dei webhook che utilizzano l'opzione *Include event menu entry*). In questo caso, la procedura consigliata è creare un utente dedicato che rappresenti il webhook:

1. Dopo aver configurato il tipo di supporto webhook, vai alla sezione *Utenti* > *Utenti* e crea un utente Zabbix dedicato che rappresenti il webhook, ad esempio con il nome utente *Slack* per il webhook Slack. Tutte le impostazioni, eccetto i media, possono essere lasciate ai valori predefiniti, poiché questo utente non accederà a Zabbix.
2. Nel profilo utente, vai alla scheda *Media* e aggiungi un webhook con le informazioni di contatto richieste. Se il webhook non utilizza un campo *Send to*, inserisci una qualsiasi combinazione di caratteri supportati per aggirare i requisiti di convalida.
3. Concedi a questo utente almeno i permessi di lettura per tutti gli host per i quali deve inviare gli avvisi.

Quando configuri un'azione di avviso, aggiungi questo utente nel campo *Send to users* nei dettagli dell'operazione: questo indicherà a Zabbix di utilizzare il webhook per le notifiche di questa azione.

Configurazione delle azioni di avviso

Le azioni determinano quali notifiche devono essere inviate tramite il webhook. I passaggi per la configurazione delle azioni che coinvolgono i webhook sono gli stessi di tutti gli altri tipi di supporto, con le seguenti eccezioni:

- Se un webhook utilizza i tag del webhook per memorizzare l'ID del ticket\messaggio e gestire le operazioni di aggiornamento\risoluzione, evitare di utilizzare lo stesso webhook in più azioni di avviso per un singolo evento di problema. Se {EVENT.TAGS.<tag name>} esiste e viene aggiornato nel webhook, il valore risultante sarà indefinito. Per evitare ciò, utilizzare un nuovo nome di tag nel webhook per memorizzare i valori aggiornati. Questo vale per i webhook Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad e Zendesk forniti da Zabbix e per la maggior parte dei webhook che utilizzano l'opzione *Include event menu entry*. Si noti tuttavia che un singolo webhook può essere utilizzato in più operazioni o passaggi di escalation della stessa azione, così come in azioni diverse che non verranno attivate dallo stesso evento di problema grazie a condizioni differenti.
- Quando si utilizza un webhook nelle azioni per eventi interni, assicurarsi di selezionare la casella *Custom message* e definire un messaggio personalizzato nella configurazione dell'operazione dell'azione. In caso contrario, la notifica non verrà inviata.

1 Esempi di script webhook

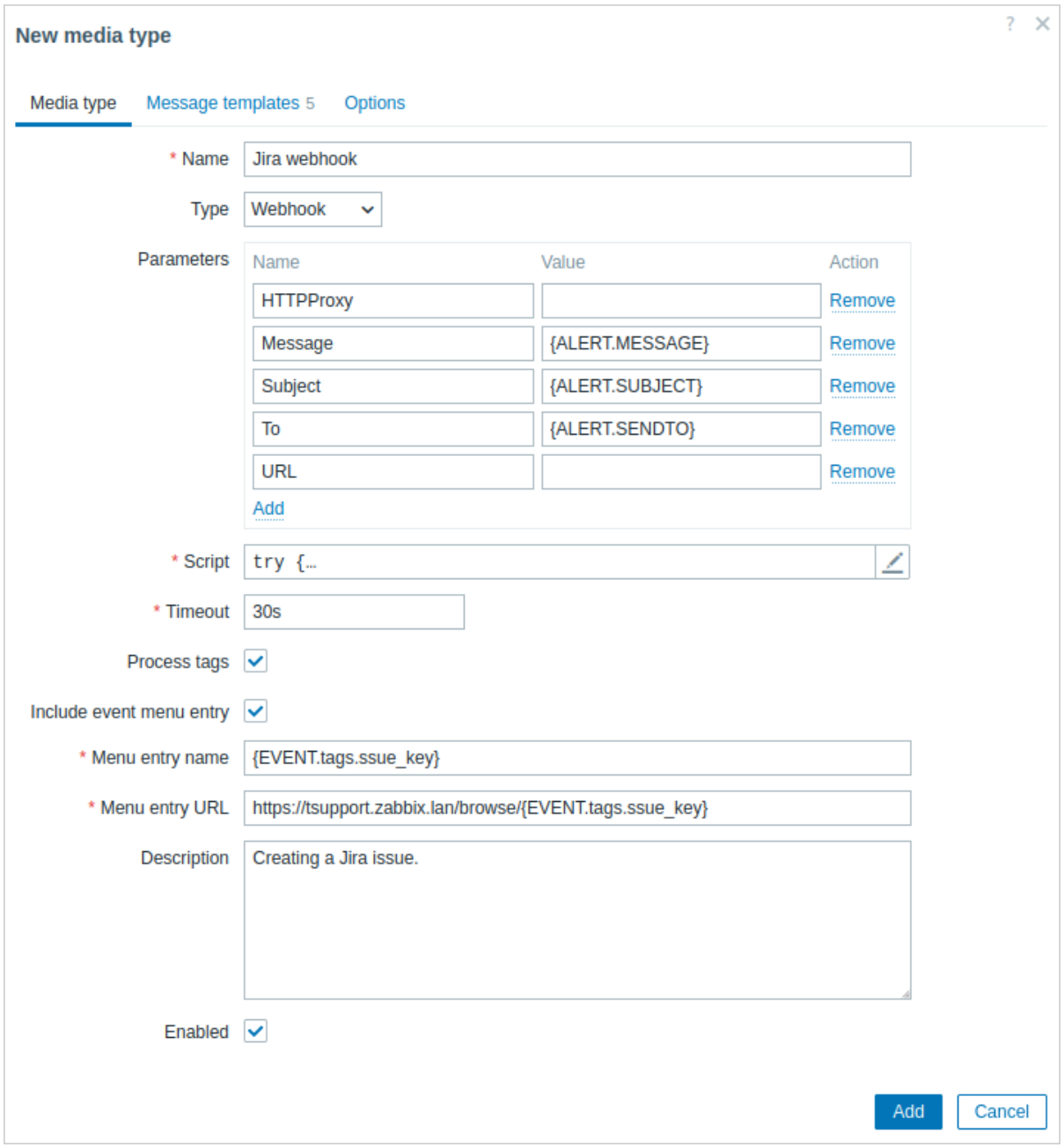
Panoramica

Sebbene Zabbix offra un gran numero di integrazioni webhook disponibili immediatamente, potresti voler creare invece i tuoi webhook personalizzati. Questa sezione fornisce esempi di script webhook personalizzati (utilizzati nel parametro *Script*). Vedi la sezione webhook per la descrizione degli altri parametri del webhook.

Attention:
Non utilizzare assegnazioni non dichiarate nel preprocessing JavaScript. Usa `var` per dichiarare le variabili locali.

Gli eventi di ripristino (sia generati automaticamente sia in seguito a una chiusura manuale) includono i tag degli eventi risolti (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, utilizzando l'API Event) oppure memorizza i tag prodotti dal webhook in un archivio persistente esterno ed esegui lì la correlazione.

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();
    }

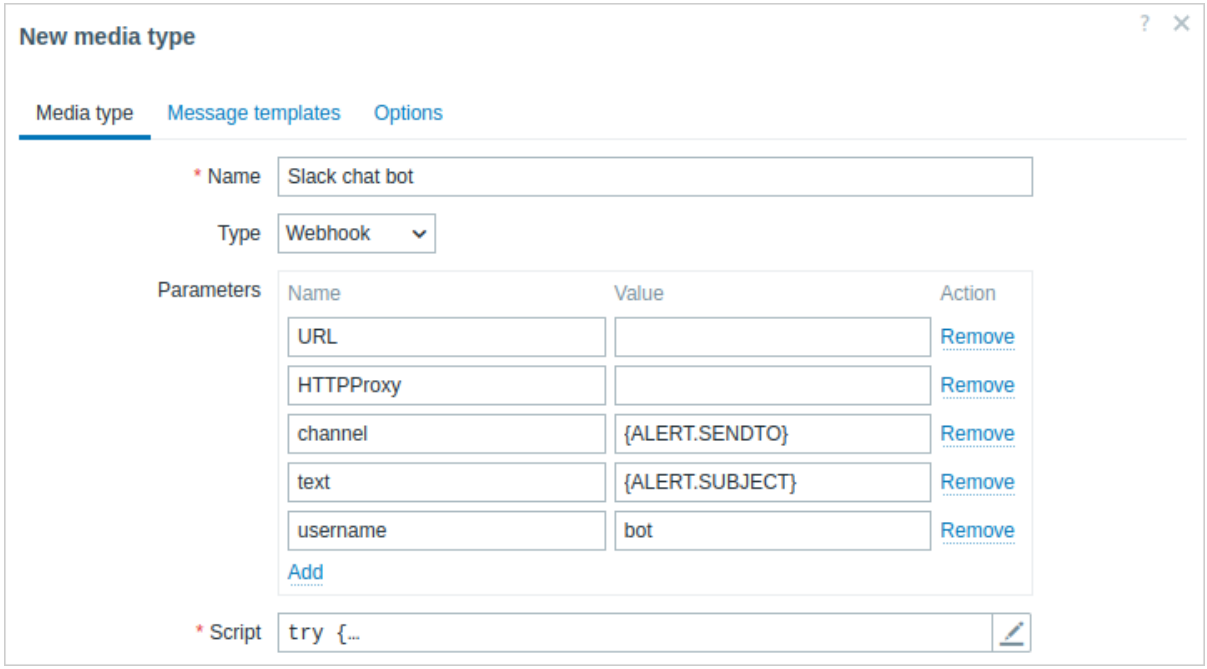
    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    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;
}
```

webhook Slack (personalizzato)

Questo webhook inoltra 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.hook_url, 'payload=' + encodeURIComponent(value));
    Zabbix.log(4, '[ Slack webhook ] Responded with code: ' + req.getStatus() + '. Response: ' + response)

    try {
        response = JSON.parse(response);
    }
    catch (error) {
        if (req.getStatus() < 200 || req.getStatus() >= 300) {
            throw 'Request failed with status code ' + req.getStatus();
        }
        else {
            throw 'Request success, but response parsing failed.';
        }
    }

    if (req.getStatus() !== 200 || !response.ok || response.ok === 'false') {
        throw response.error;
    }

    return 'OK';
}
catch (error) {
    Zabbix.log(3, '[ Slack webhook ] Sending failed. Error: ' + error);

    throw 'Failed with error: ' + error;
}
```

2 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:

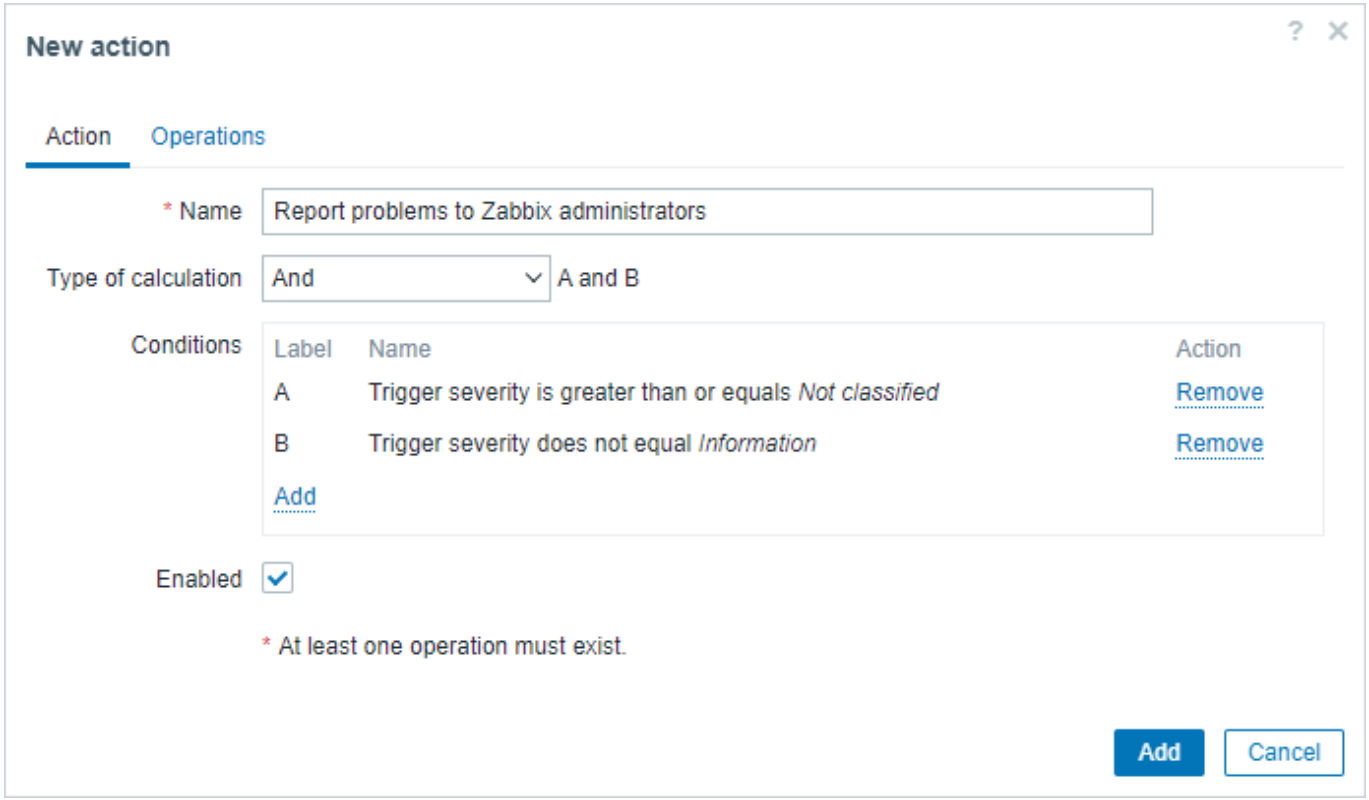
- L'accesso dell'utente alle azioni di servizio dipende dai diritti di accesso ai servizi concessi dal ruolo dell'utente
- Le azioni di servizio supportano un insieme diverso di condizioni

Configurazione di un'azione

Per configurare un'azione, procedere come segue:

- Andare in *Avvisi* > *Azioni* e selezionare dal sottomenu il tipo di azione richiesto (è possibile passare a un altro tipo in seguito, utilizzando il menu a discesa nel titolo).
- Fare clic su *Crea azione*.
- Assegnare un nome all'azione.
- Scegliere le condizioni in base alle quali vengono eseguite le operazioni.
- Scegliere le operazioni da eseguire.

Attributi generali dell'azione:



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

Parametro	Descrizione
Nome	Nome univoco dell'azione.

Parametro	Descrizione
Tipo di calcolo	Selezionare l'opzione di valutazione per le condizioni dell'azione (quando è presente 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. **Espressione personalizzata** - una formula di calcolo definita dall'utente per valutare le condizioni dell'azione.
Condizioni	Elenco delle condizioni dell'azione. Fare clic su *Aggiungi* per aggiungere una nuova condizione. Se non è configurata alcuna condizione, l'azione verrà eseguita per ogni evento che corrisponde al tipo di azione in fase di configurazione.
Abilitato	Selezionare la casella di controllo per abilitare l'azione. In caso contrario, sarà disabilitata.

1 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 dei servizi

Le seguenti condizioni possono essere utilizzate nelle azioni dei servizi:

Tipo di condizione	Operatori supportati	Descrizione
Servizio	uguale a diverso da	Specificare un servizio o un servizio da escludere. **uguale a** - l'evento appartiene a questo servizio. **diverso da** - l'evento non appartiene a questo servizio. Specificando un servizio padre vengono selezionati implicitamente tutti i servizi figli. Per specificare solo il servizio padre, tutti i servizi annidati devono essere impostati aggiuntivamente con l'operatore **diverso da**.
Nome del servizio	contiene non contiene	Specificare una stringa nel nome del servizio o una stringa da escludere. **contiene** - l'evento è generato da un servizio che contiene questa stringa nel nome. **non contiene** - questa stringa non può essere trovata nel nome del servizio.
Nome del tag del servizio	uguale a diverso da contiene non contiene	Specificare un tag evento o un tag evento da escludere. I tag evento del servizio possono essere definiti nella sezione di configurazione del servizio *Tag*. **uguale a** - 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 del tag del servizio	uguale a diverso da contiene non contiene	Specificare una combinazione di tag evento e valore oppure una combinazione di tag e valore da escludere. I tag evento del servizio possono essere definiti nella sezione di configurazione del servizio *Tag*. **uguale a** - 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.

Attention:
Assicurarsi di definire i modelli di messaggio per le azioni dei servizi nel menu *Avvisi → Tipi di supporto*. 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
IP host	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 un 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, ICMP ping, 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, eccetto 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, eccetto 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'.
Tempo di attività/inattività	maggiore o uguale a minore o uguale a	Tempo di attività per gli eventi 'Host Up' e 'Service Up'. Tempo di inattività per gli eventi 'Host Down' e 'Service Down'. **maggiore o uguale a** - è maggiore o uguale a. Il parametro è espresso in secondi. **minore o uguale a** - è minore o uguale a. Il parametro è espresso in secondi.

Tipo di condizione	Operatori supportati	Descrizione
Valore ricevuto	uguale diverso da maggiore o uguale a minore o uguale a contiene non contiene	Specificare il valore ricevuto da un controllo agent (Zabbix, SNMP) in una regola di discovery. Confronto tra stringhe. Se per una regola sono configurati più controlli Zabbix agent 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 fornito come stringa. **non contiene** - non contiene la sottostringa. Il parametro è fornito come stringa.
Proxy	uguale diverso da	Specificare un proxy o un proxy da escludere. **uguale** - utilizza questo proxy. **diverso da** - utilizza qualsiasi altro proxy eccetto questo.

Note:
I controlli di servizio in una regola di discovery che producono eventi di discovery non vengono eseguiti simultaneamente. Pertanto, se nell'azione sono configurati valori **multipli** per le condizioni `Service type`, `Service port` o `Received value`, essi verranno confrontati con un evento di discovery alla volta, ma **non** con più eventi simultaneamente. Di conseguenza, le azioni con valori multipli 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 attiva dell'agent:

Tipo di condizione	Operatori supportati	Descrizione
Metadati host	contiene non contiene corrisponde non corrisponde	Specificare i metadati host o i metadati host da escludere. **contiene** - i metadati host contengono la stringa. **non contiene** - i metadati host non contengono la stringa. I metadati host possono essere specificati in un file di configurazione dell'agent. **corrisponde** - i metadati host corrispondono a un'espressione regolare. **non corrisponde** - i metadati host non corrispondono a un'espressione regolare.
Nome host	contiene non contiene corrisponde non corrisponde	Specificare un nome host o un nome host da escludere. **contiene** - il nome host contiene la stringa. **non contiene** - il nome host non contiene la stringa. **corrisponde** - il nome host corrisponde a un'espressione regolare. **non corrisponde** - il nome host non corrisponde a un'espressione regolare.
Proxy	uguale a diverso da	Specificare un proxy o un proxy da escludere. **uguale a** - utilizza questo proxy. **diverso da** - utilizza qualsiasi altro proxy tranne 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 dei 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 da lettere maiuscole A, B, C, ...) e può includere spazi, tabulazioni, parentesi ( ), **and** (con distinzione tra maiuscole e minuscole), **or** (con distinzione tra maiuscole e minuscole), **not** (con distinzione tra maiuscole e 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).".

2 Operazioni

Panoramica

È possibile definire le seguenti operazioni per tutti gli eventi:

- Inviare un messaggio
- Eseguire un comando remoto

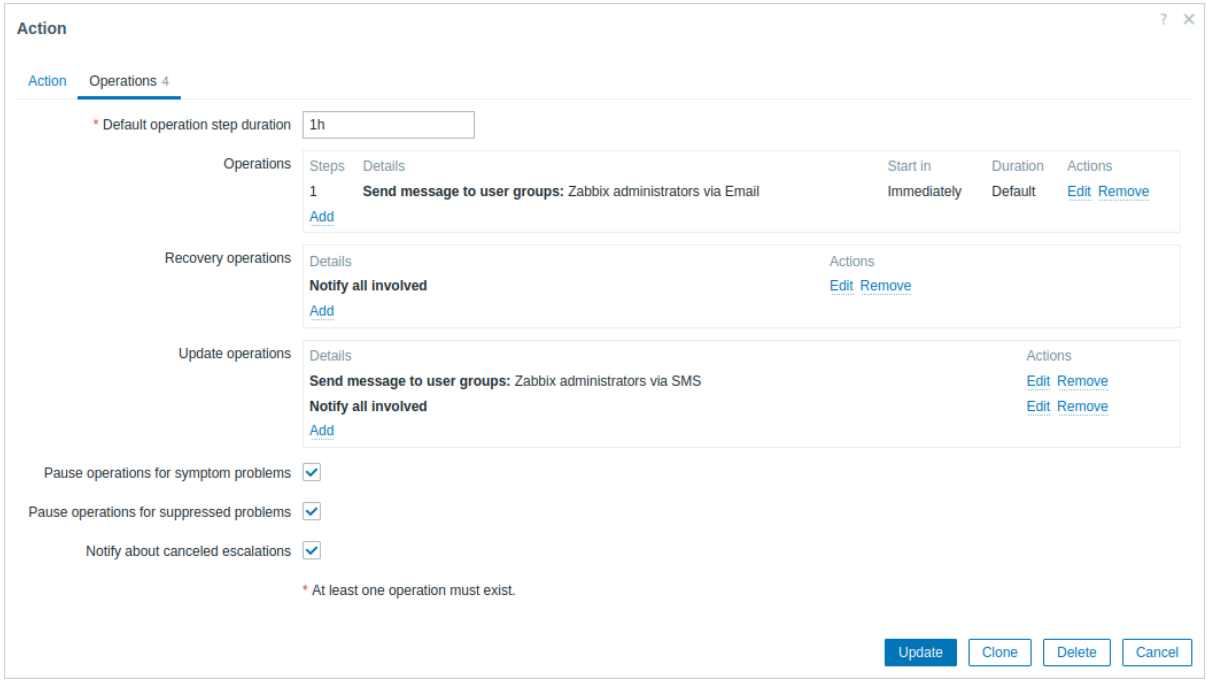
> **Attention:**
> Zabbix server non crea avvisi se l'accesso al host è esplicitamente "negato" per l'utente definito come destinatario dell'operazione dell'azione oppure se per l'utente non sono definiti alcun diritto sul host.

Per gli eventi di discovery e autoregistrazione, sono disponibili operazioni aggiuntive:

- Aggiungi host
- Rimuovi host
- Abilita host
- Disabilita host
- Aggiungi a un gruppo di host
- Rimuovi da un gruppo di host
- Aggiungi tag host
- Rimuovi tag host
- Collega template
- Scollega template
- Imposta la modalità dell'inventario host

Configurazione di un'operazione

Per configurare un'operazione, vai alla scheda *Operations* nella configurazione di action.



Attributi generali dell'operazione:

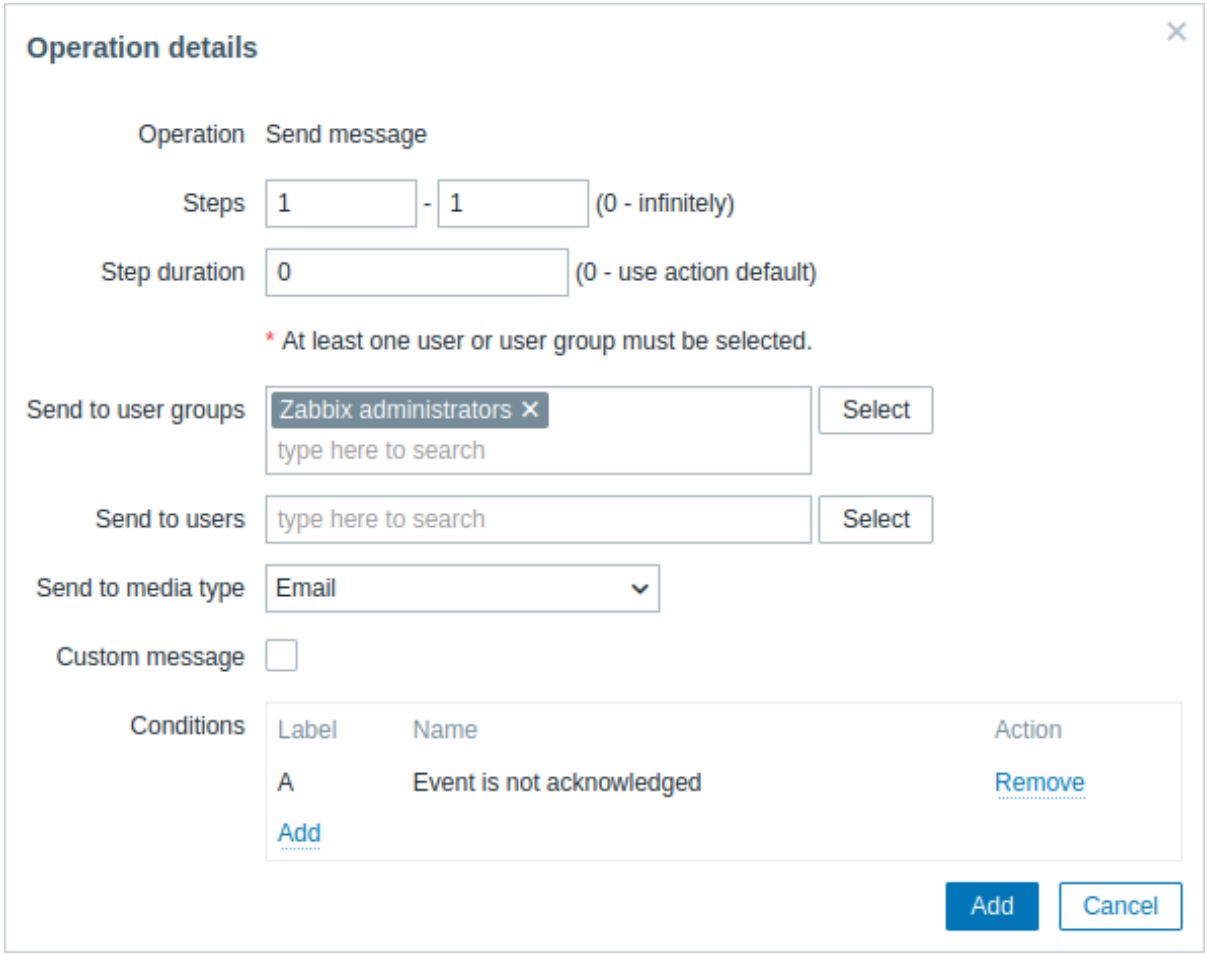
Parameter	Description
Default operation step duration	Durata predefinita di un passaggio dell'operazione (da 60 secondi a 1 settimana). Ad esempio, una durata del passaggio di un'ora significa che, se viene eseguita un'operazione, passerà un'ora prima del passaggio successivo. Sono supportati i suffissi temporali, ad esempio 60s, 1m, 2h, 1d. Sono supportate le macro utente.
Operations	Vengono visualizzate le operazioni dell'azione (se presenti), con questi dettagli: **Steps** - passaggio/i di escalation a cui è assegnata l'operazione. **Details** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di supporto (email, SMS o script) utilizzato, nonché il nome e il cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Start in** - quanto tempo dopo un evento viene eseguita l'operazione. **Duration (sec)** - viene visualizzata la durata del passaggio. Viene visualizzato *Default* se il passaggio utilizza la durata predefinita, mentre viene visualizzato un valore temporale se viene utilizzata una durata personalizzata. **Actions** - vengono visualizzati i link per modificare e rimuovere un'operazione.
Recovery operations	Vengono visualizzate le operazioni dell'azione (se presenti), con questi dettagli: **Details** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di supporto (email, SMS o script) utilizzato, nonché il nome e il cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Actions** - vengono visualizzati i link per modificare e rimuovere un'operazione.
Update operations	Vengono visualizzate le operazioni dell'azione (se presenti), con questi dettagli: **Details** - tipo di operazione e relativo destinatario/target. L'elenco delle operazioni mostra anche il tipo di supporto (email, SMS o script) utilizzato, nonché il nome e il cognome (tra parentesi dopo il nome utente) del destinatario della notifica. **Actions** - vengono visualizzati i link per modificare e rimuovere un'operazione.
Pause operations for symptom problems	Seleziona questa casella di controllo per mettere in pausa le operazioni (dopo la prima operazione) per i problemi sintomo. Nota che questa impostazione influisce solo sulle escalation dei problemi; le operazioni di ripristino e aggiornamento non saranno interessate. Questa opzione è disponibile solo per *Trigger actions*.
Pause operations for suppressed problems	Seleziona questa casella di controllo per ritardare l'avvio delle operazioni per la durata di un periodo di manutenzione. Quando le operazioni vengono avviate, dopo la manutenzione, vengono eseguite tutte le operazioni, comprese quelle relative agli eventi durante la manutenzione. Nota che questa impostazione influisce solo sulle escalation dei problemi; le operazioni di ripristino e aggiornamento non saranno interessate. Se deselezioni questa casella di controllo, le operazioni verranno eseguite senza ritardo anche durante un periodo di manutenzione. Questa opzione non è disponibile per *Service actions*.

Parameter	Description
Notify about canceled escalations	Deseleziona questa casella di controllo per disabilitare le notifiche sulle escalation annullate (quando host, item, trigger o action sono disabilitati).

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

Per configurare i dettagli di una nuova operazione, fai clic su Add nel blocco *Operations*. Per modificare un'operazione esistente, fai clic su Edit accanto all'operazione. Si aprirà una finestra pop-up in cui potrai modificare i dettagli del passaggio dell'operazione.

Dettagli dell'operazione



Parametro	Descrizione
Operazione	Selezionare l'operazione: **Invia messaggio** - invia un messaggio all'utente. **<nome comando remoto>** - esegue un comando remoto. I comandi sono disponibili per l'esecuzione se precedentemente definiti negli script globali con *Operazione azione* selezionato come ambito. Sono disponibili ulteriori operazioni per eventi basati su discovery e autoregistrazione (vedere sopra).
Passi	Selezionare il/i passo/i a cui assegnare l'operazione in una pianificazione di escalation: **Da** - esegui a partire da questo passo. **A** - esegui fino a questo passo (0=infinito, l'esecuzione non sarà limitata).
Durata del passo	Durata personalizzata per questi passi (0=usa la durata predefinita del passo). I suffissi temporali sono supportati, ad esempio 60s, 1m, 2h, 1d. Le macro utente sono supportate. È possibile assegnare più operazioni allo stesso passo. Se queste operazioni hanno una durata del passo diversa definita, viene presa in considerazione la più breve e applicata al passo.

Parametro	Descrizione
Tipo di operazione: invia messaggio	
Invia a gruppi utenti	Selezionare i gruppi utenti a cui inviare il messaggio. Il gruppo utenti deve avere almeno i permessi di ”lettura” sul host per poter ricevere la notifica.
Invia a utenti	Selezionare gli utenti a cui inviare il messaggio. L’utente deve avere almeno i permessi di ”lettura” sul 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 sugli 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	Il messaggio personalizzato. Il messaggio può contenere macro. È limitato a un certo numero di caratteri a seconda del tipo di database (vedere Invio messaggio per ulteriori informazioni).
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 problema. Questa opzione non funzionerà se nel trigger sono presenti più host. **Host** - selezionare il/i host su cui eseguire il comando. **Gruppo host** - selezionare il/i gruppo/i host su cui eseguire il comando. Specificando un gruppo host padre vengono implicitamente selezionati tutti i gruppi host nidificati. Pertanto il comando remoto verrà eseguito anche sui host dei gruppi nidificati. Un comando su un host viene eseguito una sola volta, anche se l’host corrisponde più di una volta (ad esempio da più gruppi host; individualmente e da un gruppo host). L’elenco destinazioni non ha significato se uno script personalizzato viene eseguito su Zabbix server. In questo caso, selezionare più destinazioni comporta solo che lo script venga eseguito più volte sul server. Si noti che, per gli script globali, la selezione della destinazione dipende anche dall’impostazione *Gruppo host* nella configurazione dello script globale. L’opzione *Elenco destinazioni* non è disponibile per le *Azioni servizio* perché in questo caso i comandi remoti vengono sempre eseguiti su Zabbix server.

Parametro	Descrizione
Condizioni	Condizione per l'esecuzione dell'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 trigger*.

Al termine, fare clic su *Aggiungi* per aggiungere l'operazione all'elenco delle *Operazioni*.

1 Invio 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 i supporti a cui inviare un messaggio

Se l'operazione avviene al di fuori dell'intervallo di tempo When active definito per il supporto selezionato nella configurazione utente, il messaggio non verrà inviato.

La gravità predefinita del trigger ('Not classified') **deve** essere selezionata nella configurazione dei supporti utente se si desidera ricevere notifiche per eventi non relativi ai trigger, come discovery, autoregistrazione attiva dell'agent o eventi interni.

- configurare un'operazione dell'azione che invii un messaggio a uno dei supporti definiti

Attention:
Zabbix invia notifiche solo agli utenti che dispongono almeno dei permessi di 'read' per il host che ha generato l'evento. Almeno un host di un'espressione trigger deve essere accessibile.

È possibile configurare scenari personalizzati per l'invio di messaggi utilizzando le escalation.

Per ricevere e leggere correttamente le email da Zabbix, i server/client di posta 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 sono codificati in base64 per rispettare lo standard del formato 'SMTP/MIME email'.

Il limite del messaggio dopo l'espansione di tutte le macro è lo stesso del limite del messaggio per i comandi remoti.

Tracciamento dei messaggi

È possibile visualizzare lo stato dei messaggi inviati in *Monitoring* → *Problems*.

Nella colonna *Actions* è possibile vedere informazioni riepilogative sulle azioni eseguite. Qui 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.

Facendo clic sull'ora dell'evento per visualizzarne i dettagli, sarà possibile vedere i dettagli dei messaggi inviati (o non inviati) a causa dell'evento nel blocco *Actions*.

In *Reports* → *Action log* saranno visibili i dettagli di tutte le azioni eseguite per gli eventi per i quali è configurata un'azione.

2 Comandi remoti

Panoramica

Con i comandi remoti è possibile definire che un determinato comando predefinito venga eseguito automaticamente sul host monitorato al verificarsi di una certa condizione.

Pertanto, i comandi remoti rappresentano un potente meccanismo per un monitoraggio proattivo intelligente.

Negli utilizzi più evidenti di questa funzionalità è possibile, ad esempio:

- 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 file meno recenti, pulendo /tmp) in caso di esaurimento dello spazio disponibile
- Migrare una VM da una macchina fisica a un'altra in base al carico della CPU
- Aggiungere nuovi nodi a un ambiente cloud in caso di risorse CPU (disco, memoria o altro) insufficienti

La configurazione di un'azione per i comandi remoti è simile a quella per l'invio di un messaggio; l'unica differenza è che Zabbix eseguirà un comando invece di inviare un messaggio.

I comandi remoti possono essere eseguiti da Zabbix server, proxy o agent. I comandi remoti su Zabbix agent possono essere eseguiti direttamente da Zabbix server oppure tramite Zabbix proxy. Sia su Zabbix agent sia su Zabbix proxy i comandi remoti sono disabilitati per impostazione predefinita. Possono essere abilitati:

- aggiungendo un parametro `AllowKey=system.run[*]` nella configurazione dell'agent;
- impostando il parametro EnableRemoteCommands a '1' nella configurazione del 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 il host di destinazione è in manutenzione.

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.

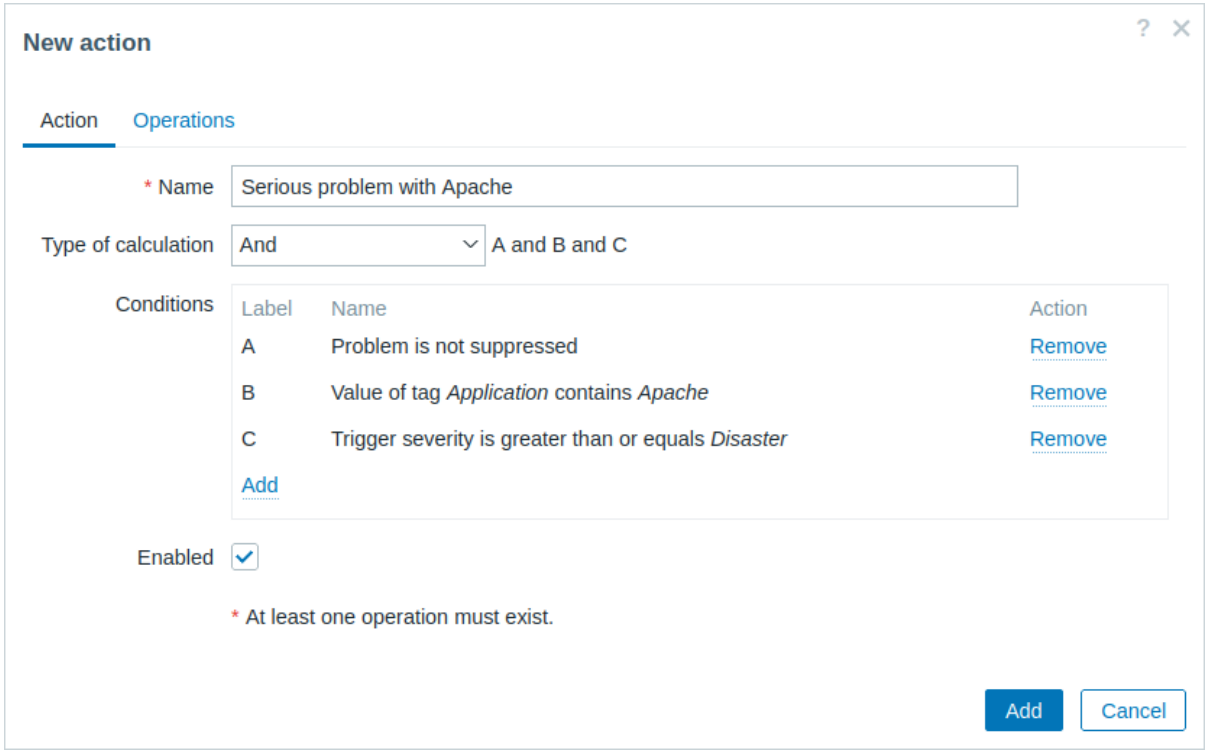
Configurazione

Quei comandi remoti che vengono eseguiti su Zabbix agent (script personalizzati) devono essere prima abilitati nella configurazione dell'agent.

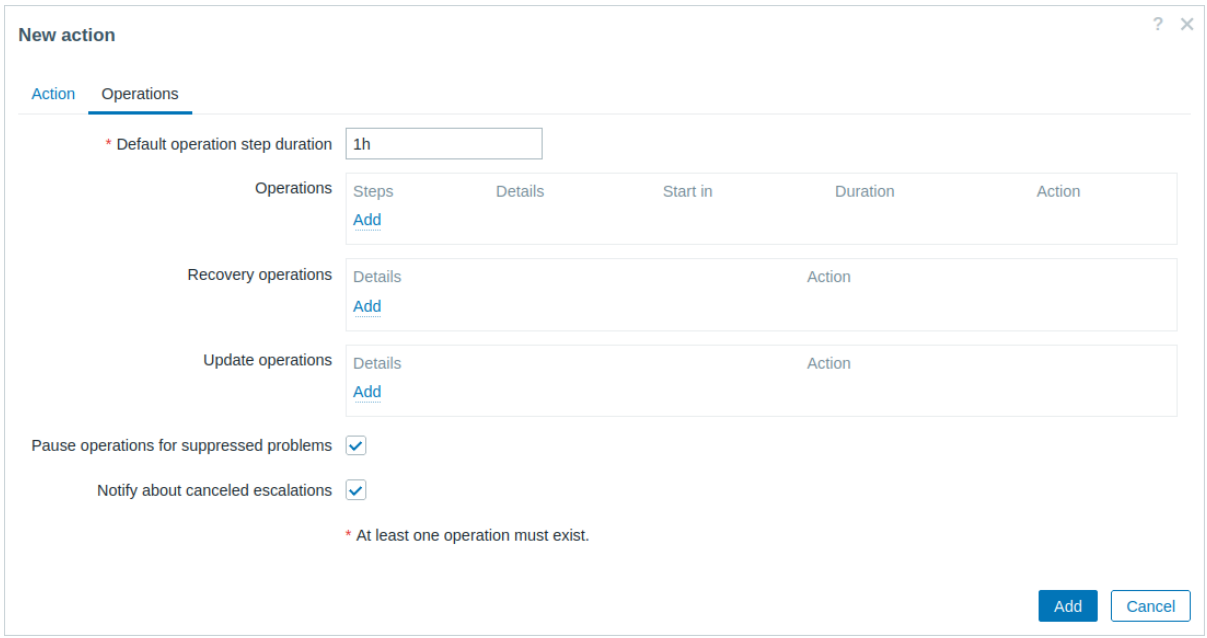
Assicurarsi che il parametro AllowKey=system.run[<command>,*] sia aggiunto per ogni comando consentito nella configurazione dell'agent, per permettere il comando specifico con la modalità nowait. Riavviare il demone agent se si modifica questo parametro.

Quindi, durante la configurazione di una nuova azione in *Avvisi → Azioni → Azioni trigger*:

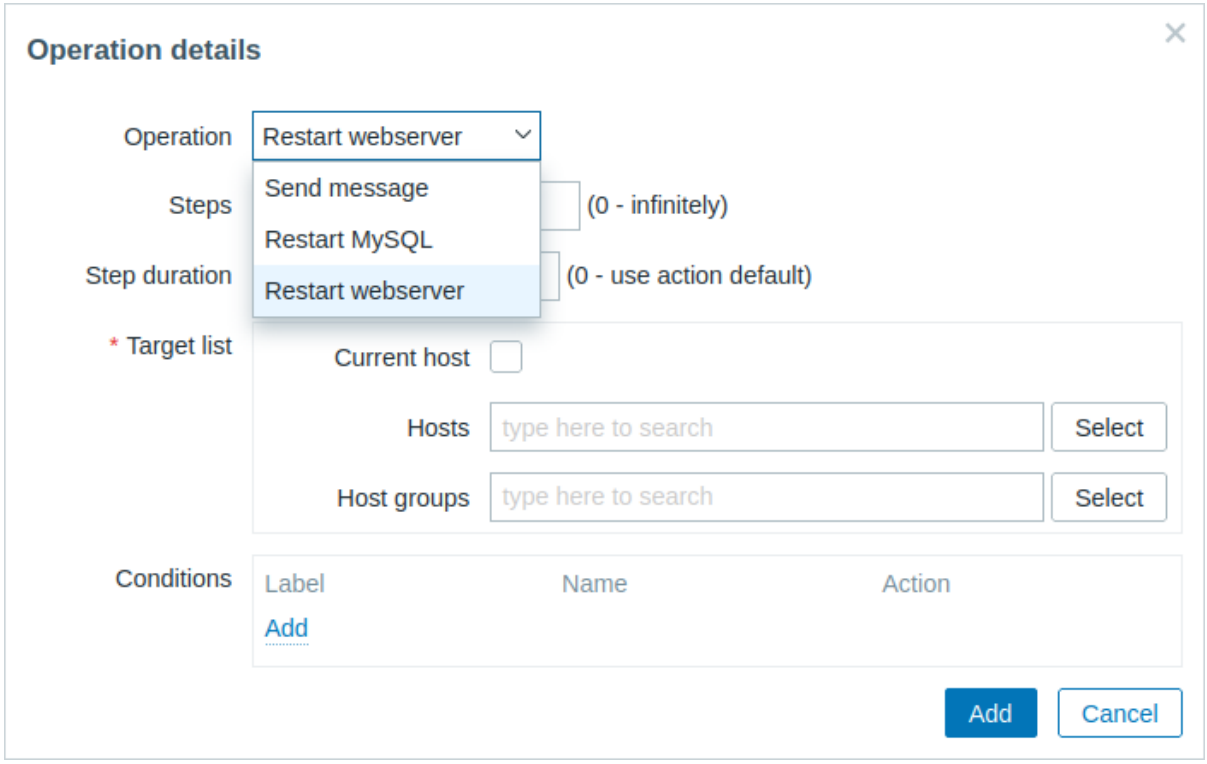
1. Definire le condizioni appropriate, ad esempio impostare che l'azione venga attivata in presenza di qualsiasi problema di gravità disaster in una delle applicazioni Apache.



2. Nella scheda *Operazioni*, fare clic su *Aggiungi* nel blocco *Operazioni*, *Operazioni di ripristino* o *Operazioni di aggiornamento*.



3. Selezionare uno degli script predefiniti dall'elenco a discesa *Operazione* e impostare l'*Elenco target* per lo script.



Script predefiniti

Gli script disponibili per le operazioni delle azioni (webhook, script, SSH, Telnet, IPMI) sono definiti negli script globali.

Per esempio:

```
sudo /etc/init.d/apache restart
```

In questo caso, Zabbix tenterà di riavviare un processo Apache. Con questo comando, assicurarsi che il comando venga eseguito su Zabbix agent (fare clic sul pulsante *Zabbix agent* accanto a *Execute on*).

Attention:
Notare l'uso di **sudo** - per impostazione predefinita l'utente Zabbix non dispone delle autorizzazioni per riavviare i servizi di sistema. Vedere 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 - attivo o passivo - deve essere in esecuzione sul host remoto ed esegue i comandi in background.

I comandi remoti su Zabbix agent vengono eseguiti senza timeout tramite la chiave system.run[,nowait] e i risultati dell'esecuzione non vengono controllati. 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 controllati per verificarne i risultati di esecuzione. Per ulteriori informazioni, vedere *Timeout dello script*.

Permessi di accesso

Assicurarsi che l'utente 'zabbix' disponga dei permessi di esecuzione per i comandi configurati. Potrebbe essere utile utilizzare **sudo** per concedere l'accesso a comandi privilegiati. Per configurare l'accesso, eseguire come root:

```
visudo
```

Esempi di righe che possono essere utilizzate 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 a una determinata condizione.

Per riavviare automaticamente Windows in seguito a un problema rilevato da Zabbix, definire il seguente script:

Parametro dello script	Valore
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

3 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.

4 Utilizzo di 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
```

3 Operazioni di recupero

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

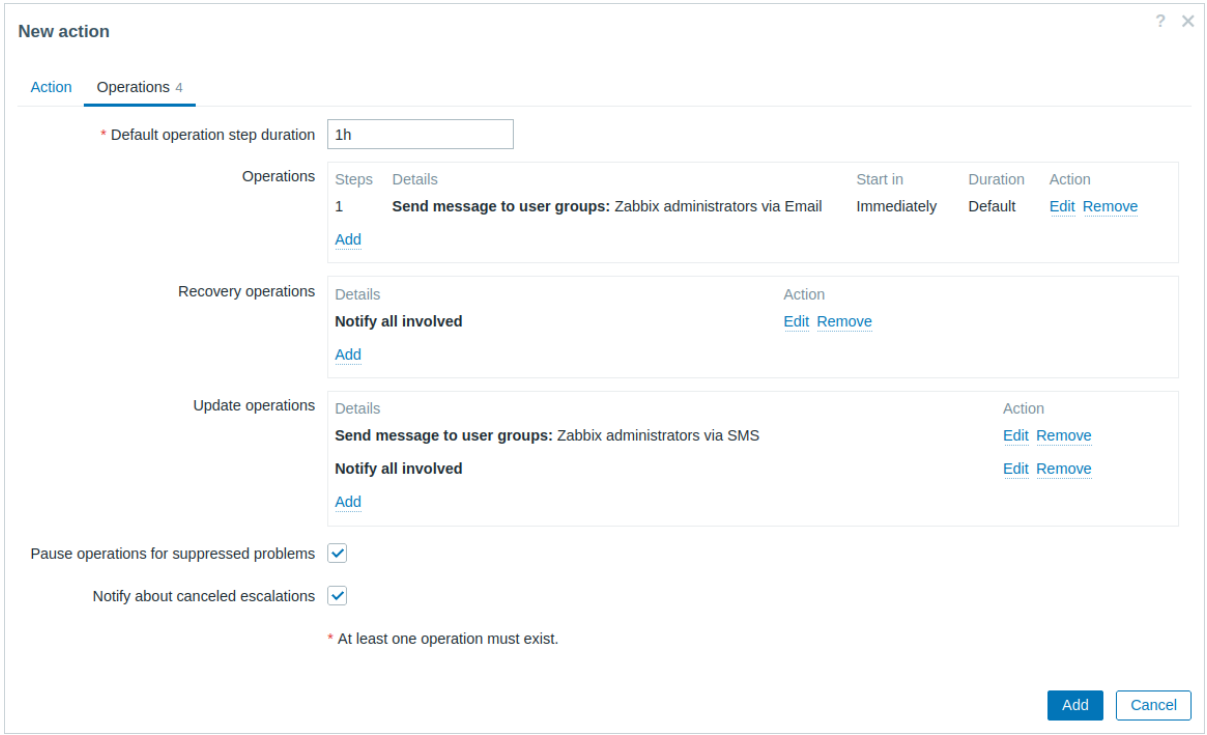
Alcuni casi d'uso per le operazioni di ripristino sono i seguenti:

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.

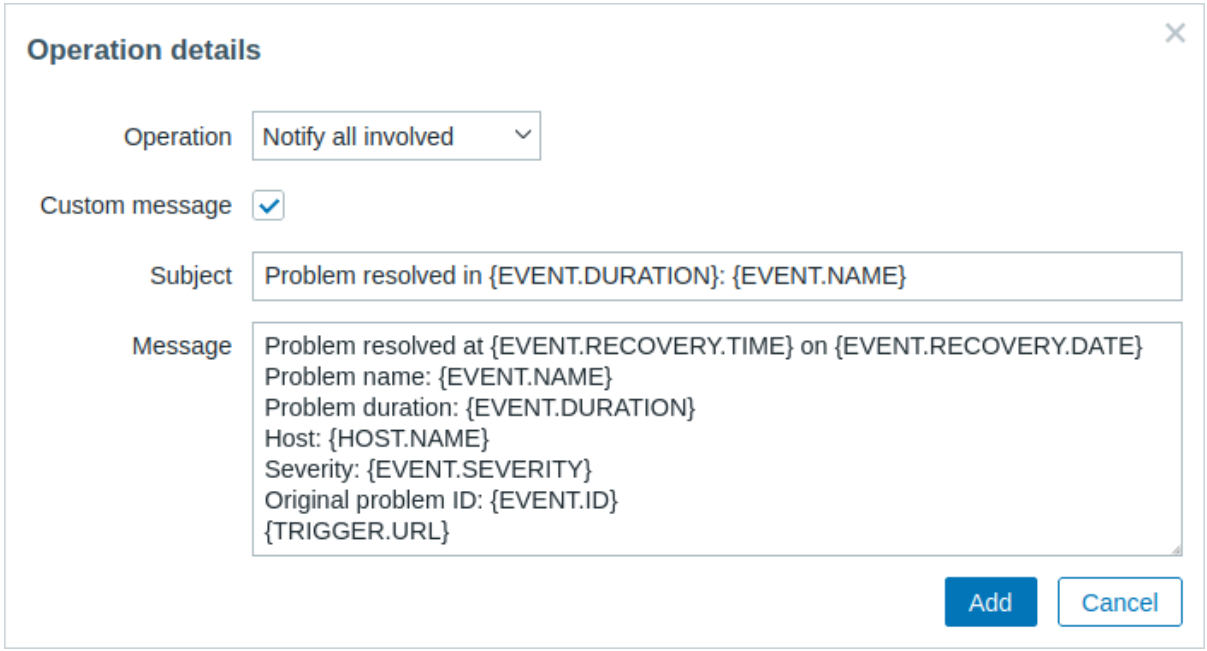
Configurazione di un'operazione di ripristino

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 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 l'host o gli host su cui eseguire il comando. **Gruppo di host** - selezionare il gruppo o i gruppi di host su cui eseguire il comando. Specificando un gruppo di host padre vengono selezionati implicitamente anche tutti i gruppi di host nidificati. Pertanto, il comando remoto verrà eseguito anche sugli host dei gruppi nidificati. Un comando su un host viene eseguito una sola volta, anche se l'host corrisponde più di una volta (ad esempio, da più gruppi di host; singolarmente e da un gruppo di host). L'elenco delle destinazioni non ha significato se il comando viene eseguito su Zabbix server. In questo caso, selezionare più destinazioni comporta solo che il comando venga eseguito sul server più volte. Si noti che, 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.

4 Operazioni di aggiornamento

Panoramica

Le operazioni di aggiornamento sono disponibili nelle azioni con le seguenti origini evento:

- *Trigger* - quando i problemi vengono aggiornati da altri utenti, cioè quando viene aggiunto un commento, vengono riconosciuti, la gravità viene modificata, oppure vengono chiusi (manualmente);
- *Servizi* - quando la gravità di un servizio è cambiata ma il servizio non è ancora stato ripristinato.

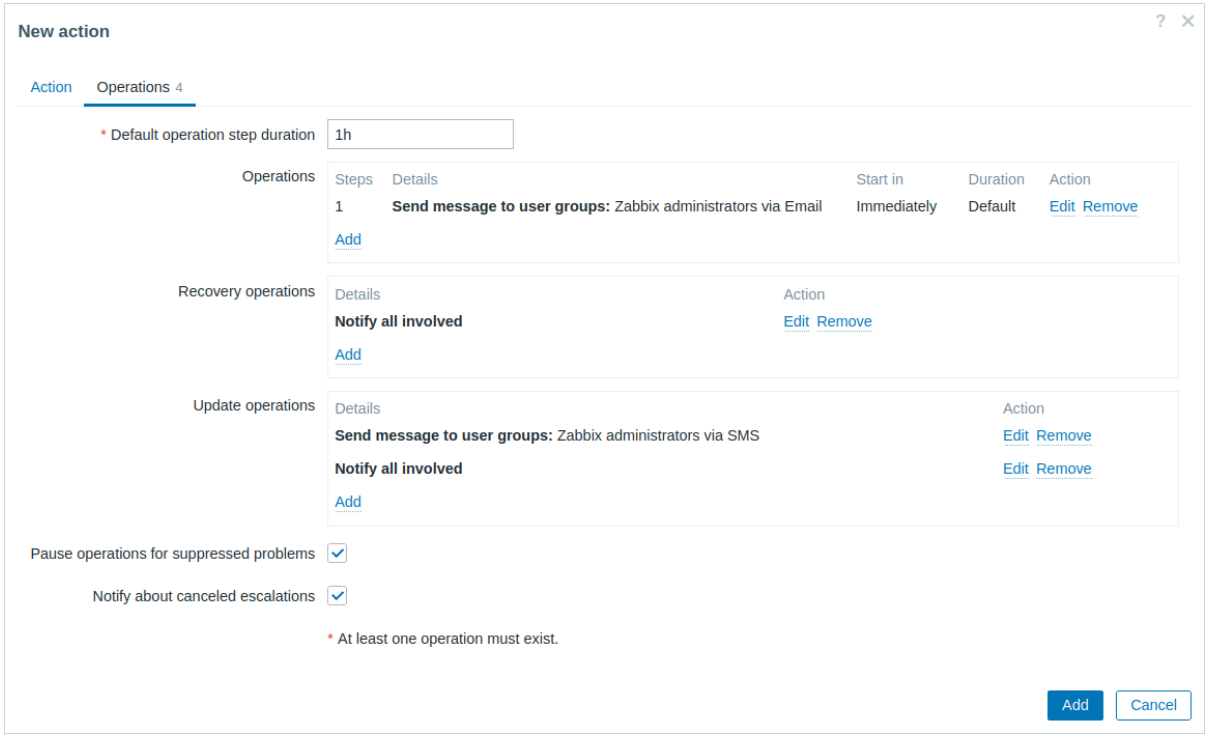
Si noti che gli utenti non ricevono notifiche sui propri aggiornamenti.

Nelle operazioni di aggiornamento sono supportati sia i messaggi sia i comandi remoti. Sebbene sia possibile aggiungere più operazioni, l'escalation non è supportata: tutte le operazioni vengono assegnate a un singolo passaggio e pertanto verranno eseguite simultaneamente.

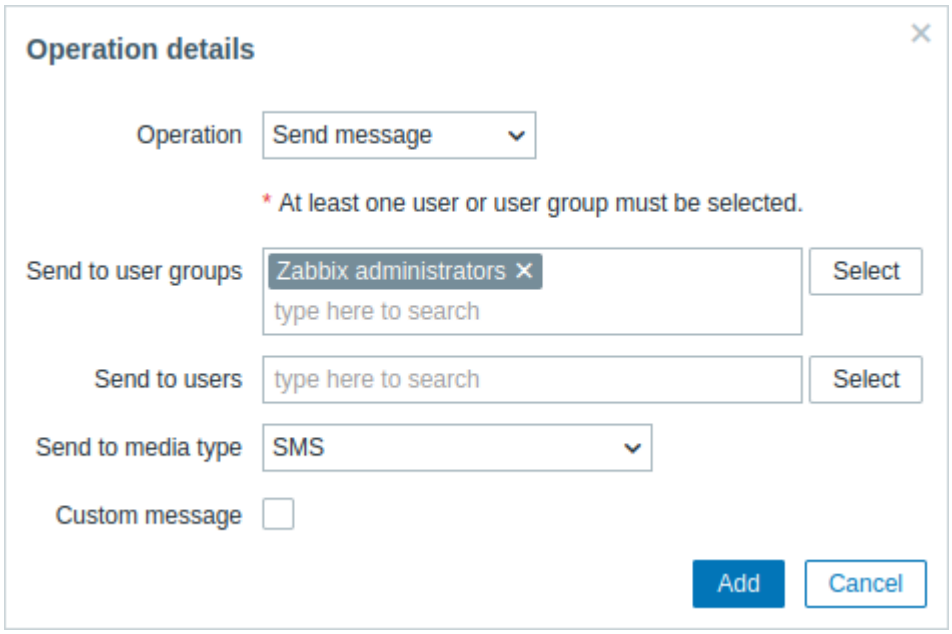
Configurazione di un'operazione di aggiornamento

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.

5 Escalation

Overview

With escalations you can create custom scenarios for sending notifications or executing remote commands.

In practical terms it means that:

- Users can be informed about new problems immediately
- Notifications can be repeated until the problem is resolved
- Sending a notification can be delayed
- Notifications can be escalated to another "higher" user group
- Remote commands can be executed immediately or when a problem is not resolved for a lengthy period

Actions are escalated based on the **escalation step**. Each step has a duration in time.

You can define both the default duration and a custom duration of an individual step. The minimum duration of one escalation step is 60 seconds.

You can start actions, such as sending notifications or executing commands, from any step. Step one is for immediate actions. If you want to delay an action, you can assign it to a later step. For each step, several actions can be defined.

The number of escalation steps is not limited.

Escalations are defined when configuring an operation. Escalations are supported for problem operations only, not recovery.

Vari aspetti del comportamento dell'escalation

Consideriamo cosa succede in diverse circostanze se un'azione contiene diversi passaggi di escalation.

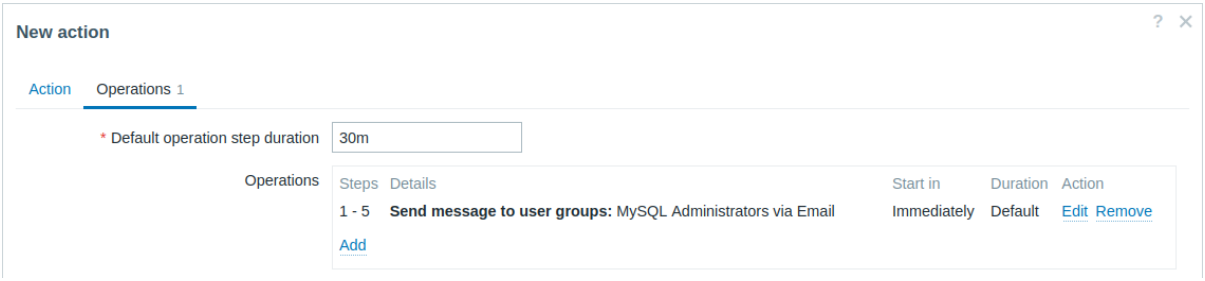
Situazione	Comportamento
L'host in questione entra in manutenzione dopo l'invio della notifica iniziale del problema	A seconda dell'impostazione *Pause operations for suppressed problems* nella configurazione dell'azione, tutti i restanti passaggi di escalation vengono eseguiti con un ritardo causato dal periodo di manutenzione oppure senza ritardo. Un periodo di manutenzione non annulla le operazioni.
Il periodo di tempo definito nella condizione dell'azione ***Time period*** *termina dopo l'invio della notifica iniziale*	Tutti i restanti passaggi di escalation vengono eseguiti. La condizione *Time period* non può interrompere le operazioni; ha effetto su quando 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 *Pause operations for suppressed problems* nella configurazione dell'azione, tutti i passaggi di escalation vengono eseguiti o dal momento in cui termina la manutenzione oppure immediatamente.
Un problema inizia durante una manutenzione no-data e continua (non viene risolto) dopo la fine della manutenzione	È necessario attendere che il trigger si attivi prima che vengano eseguiti tutti i passaggi di escalation.
Diverse escalation si susseguono a breve distanza e si sovrappongono	L'esecuzione di ogni nuova escalation sostituisce la precedente, ma almeno un passaggio di escalation della precedente viene sempre eseguito. Questo comportamento è rilevante nelle azioni sugli eventi creati a OGNI valutazione del problema del trigger.
Durante un'escalation in corso (ad esempio, durante l'invio di un messaggio), in base a qualsiasi tipo di evento: - l'azione viene disabilitata In base a un evento trigger: - il trigger viene disabilitato - l'host o l'item vengono disabilitati In base a un evento interno relativo ai trigger: - il trigger viene disabilitato In base a un evento interno relativo a item/regole di low-level discovery: - l'item viene disabilitato - l'host viene disabilitato	Il messaggio in corso viene inviato e poi viene inviato ancora un altro messaggio nell'escalation. Il messaggio successivo conterrà il testo di annullamento all'inizio del corpo del messaggio (*NOTE: Escalation canceled*), indicando il motivo (ad esempio, *NOTE: Escalation canceled: action '<Action name>' disabled*). 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 in precedenza. Il motivo dell'annullamento viene anche registrato nel file di log del server (a partire da Debug Level 3=Warning). Si noti che il messaggio *Escalation canceled* 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, durante l'invio di un messaggio) l'azione viene eliminata	Non vengono inviati altri messaggi. L'informazione viene registrata nel file di log del server (a partire da Debug Level 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).

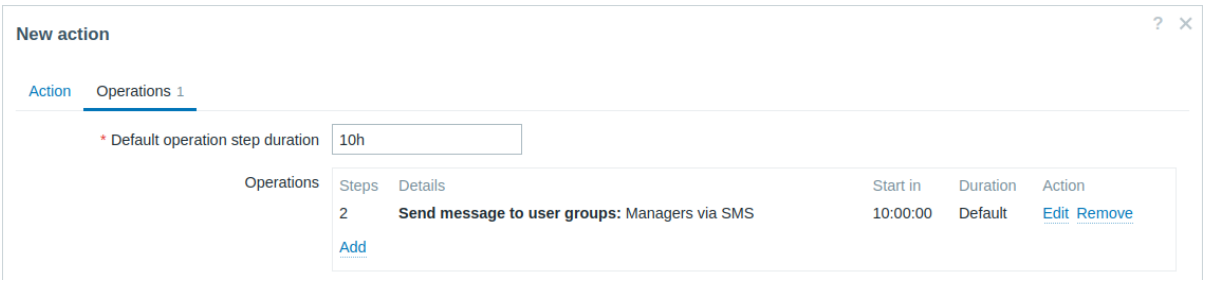
Se il problema viene risolto ed è configurato un messaggio di ripristino, esso verrà inviato a coloro che hanno ricevuto almeno un messaggio di problema all'interno di questo scenario di escalation.

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".



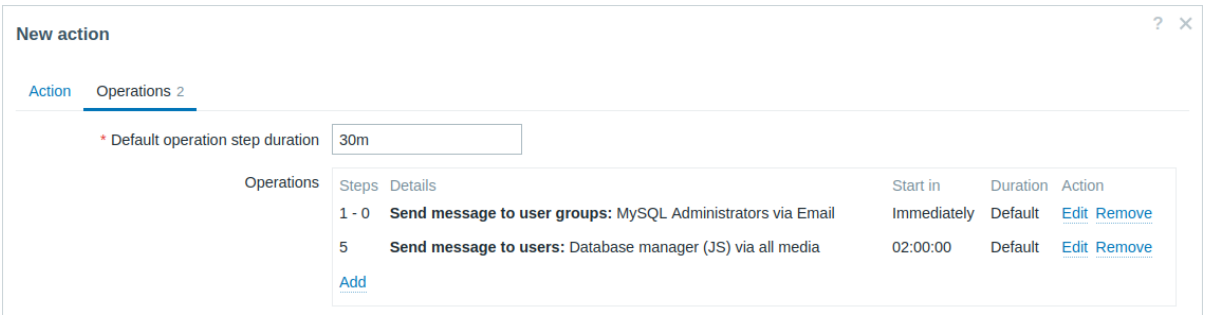
Una notifica verrà inviata solo allo Step 2 dello scenario di escalation, ovvero 10 ore dopo l'inizio del problema.

È possibile personalizzare il testo del messaggio con qualcosa come "Il problema dura da più di 10 ore".

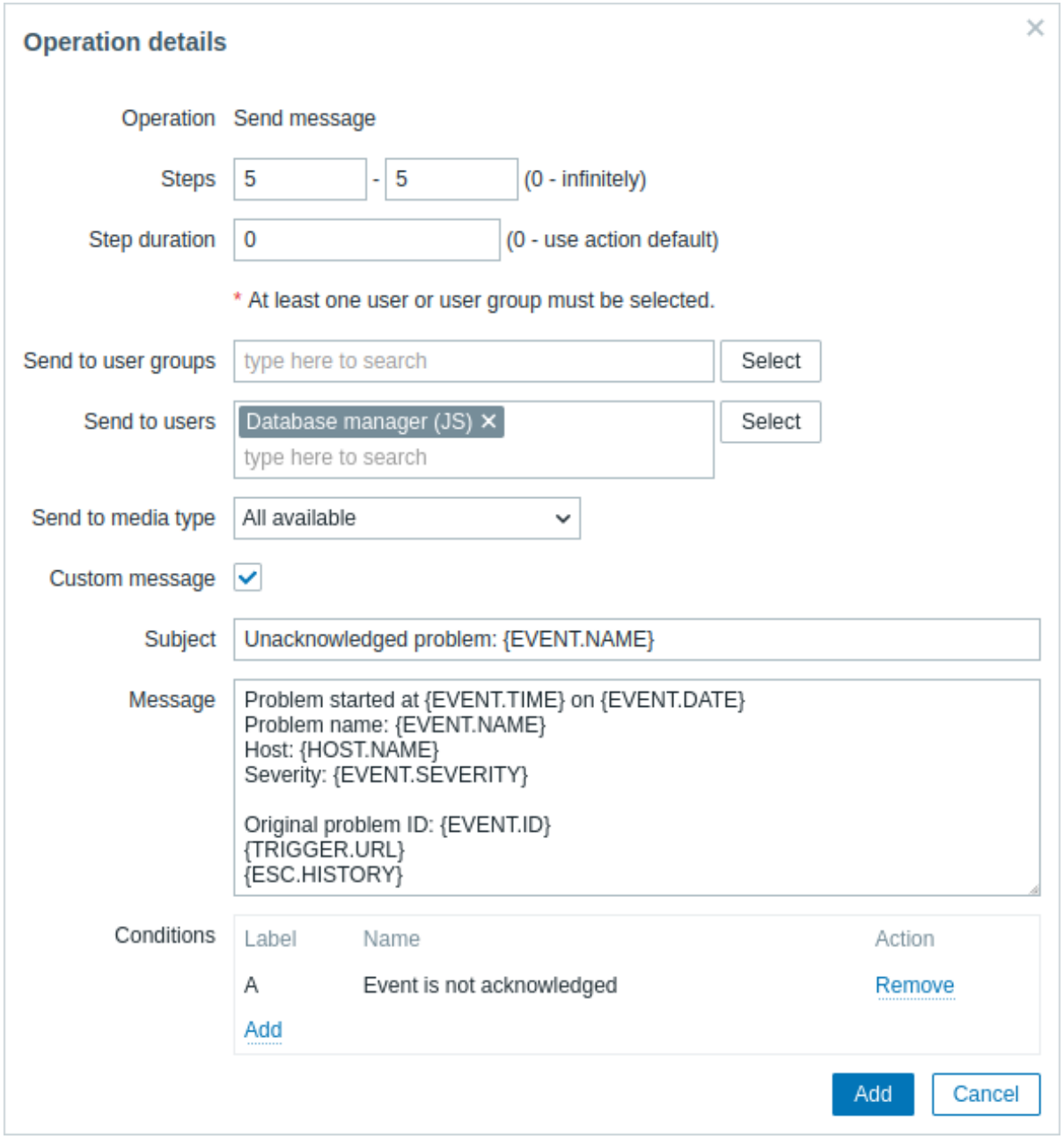
Esempio 3

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:



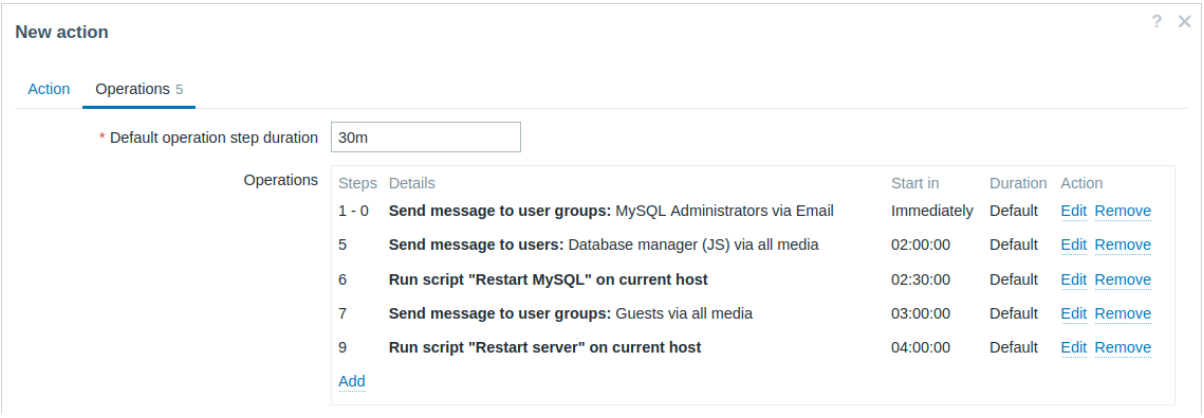
Si noti l'uso della macro {ESC.HISTORY} nel messaggio personalizzato. La macro conterrà informazioni su tutti i passaggi eseguiti in precedenza in questa escalation, come le notifiche inviate e i comandi eseguiti.

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.

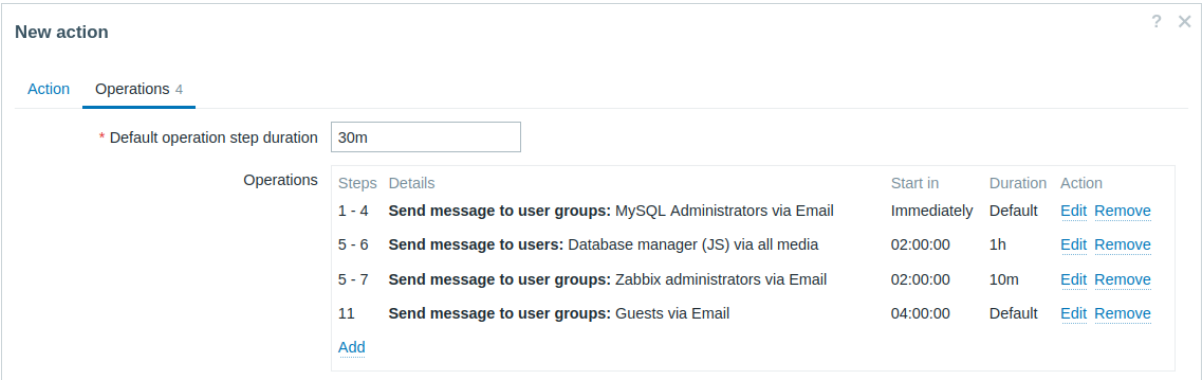
Se il problema esiste ancora, dopo altri 30 minuti Zabbix invierà un messaggio a tutti gli utenti guest.

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).

3 Ricezione di notifiche su elementi non supportati

Panoramica

In Zabbix è possibile ricevere notifiche sugli item non supportati.

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 low-level discovery passano da 'normale' a 'non supportato' (e viceversa).

Questa sezione presenta una procedura 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, come 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 (’Non classificato’), quindi lasciarla selezionata durante la configurazione dei supporti utente se si desidera ricevere notifiche per gli eventi interni.

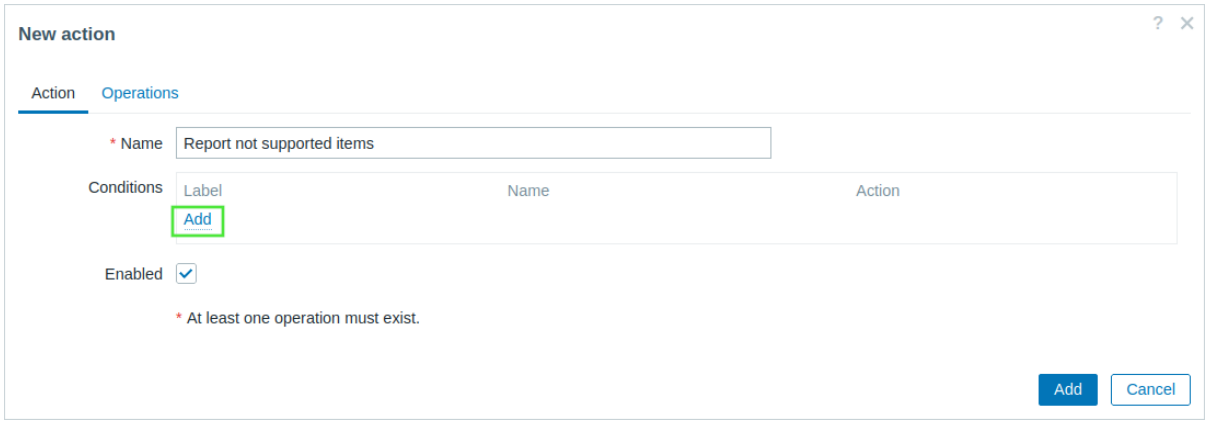
Passaggio 2

Andare su *Avvisi* → *Azioni* → *Azioni interne*.

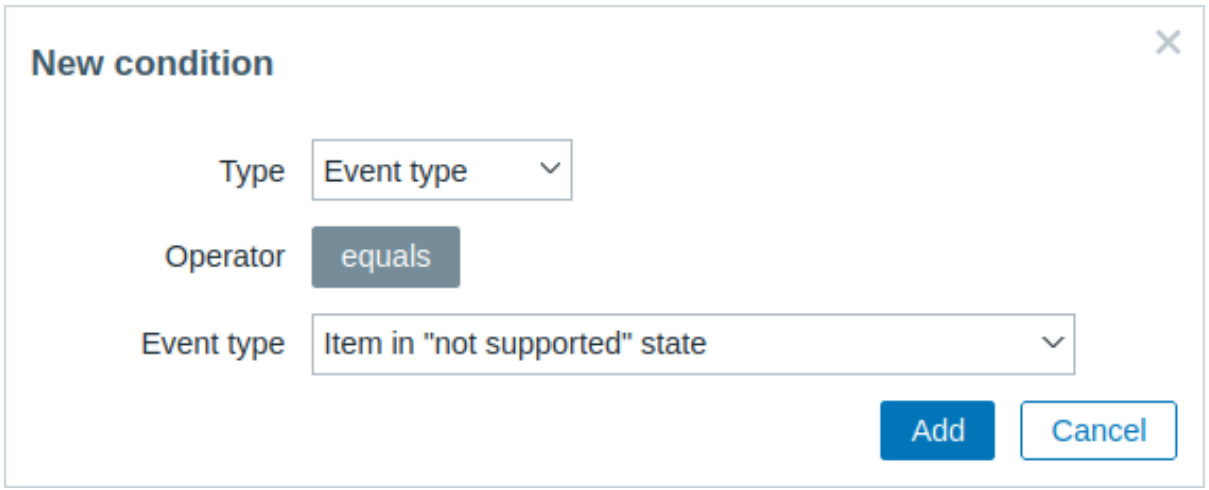
Fare clic su *Crea azione* nell’angolo superiore destro della pagina per aprire un modulo di configurazione dell’azione.

Passaggio 3

Nella scheda *Action* inserisci un nome per l’azione. Quindi fai clic su *Add* nel blocco *Conditions* per aggiungere una nuova condizione.



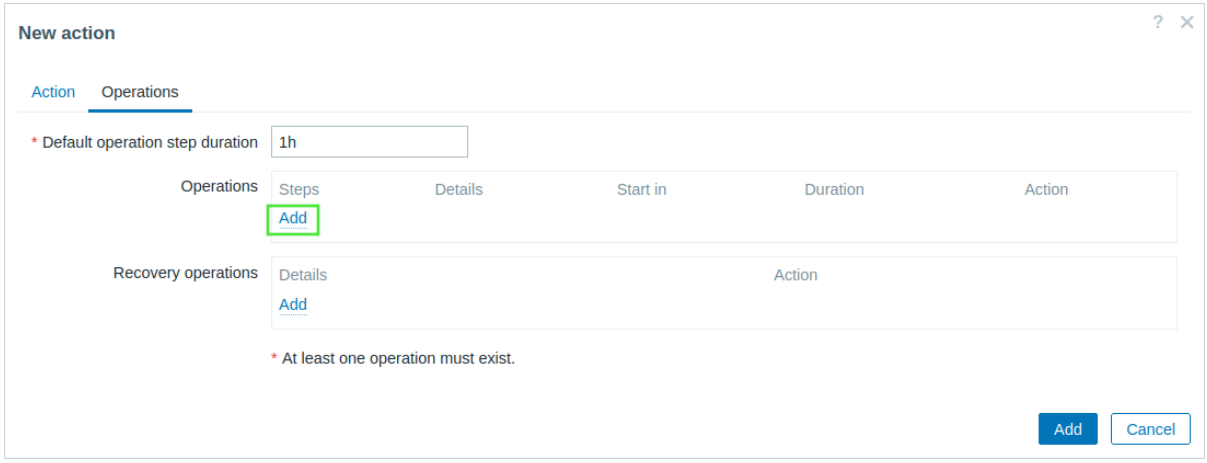
Nella finestra pop-up *New condition* seleziona ”Event type” come tipo di condizione e quindi seleziona ”Item in ’not supported’ state” come tipo di evento.



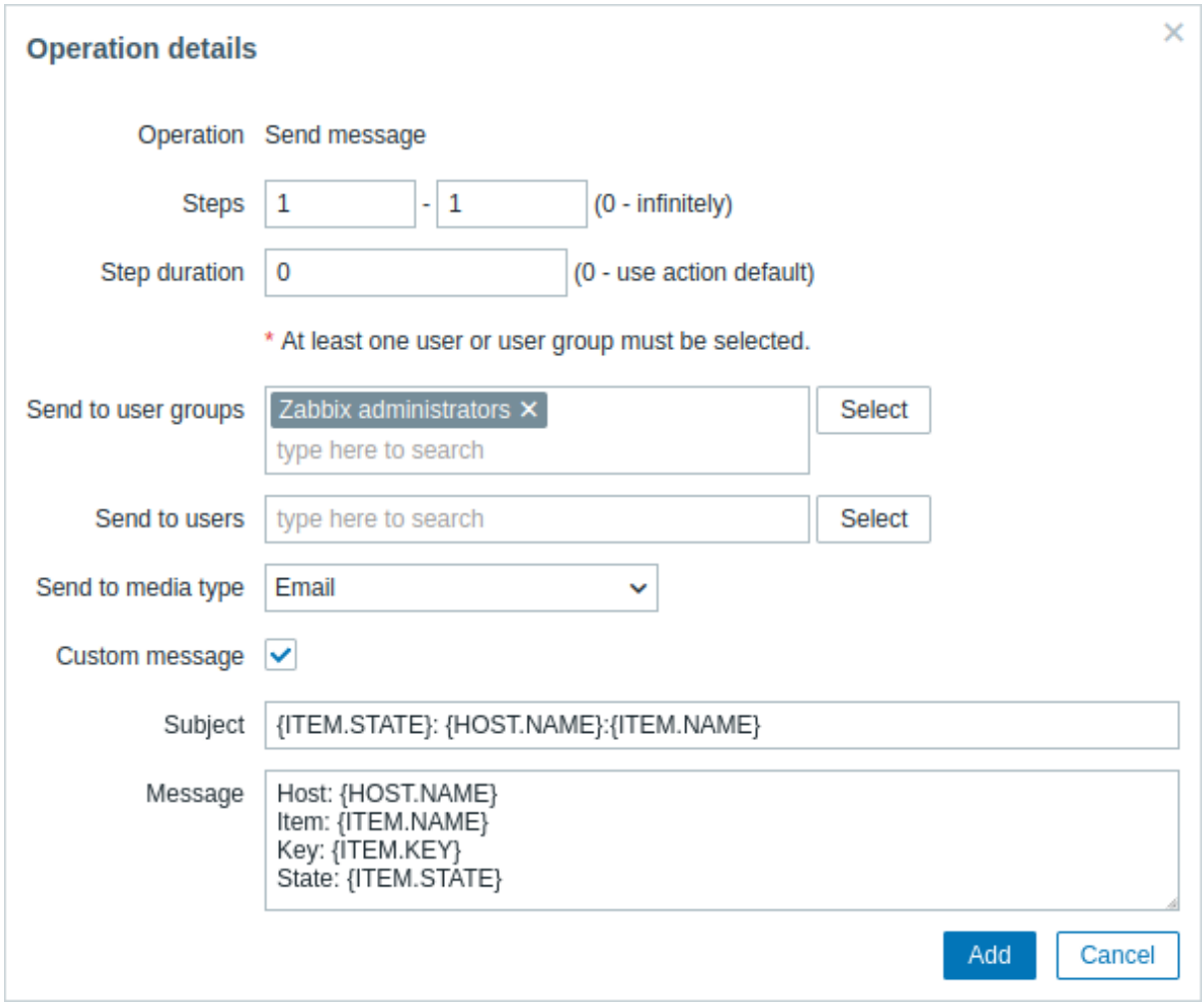
Non dimenticare di fare clic su *Add* per elencare effettivamente la condizione nel blocco *Conditions*.

Passaggio 4

Nella scheda *Operations*, fai clic su *Add* nel blocco *Operations* per aggiungere una nuova operazione.



Seleziona alcuni destinatari del messaggio (gruppi utenti/utenti) e il tipo di supporto (oppure "All") da utilizzare per la consegna. Se desideri inserire l'oggetto/il contenuto personalizzato del messaggio di problema, seleziona la casella di controllo *Custom message*.

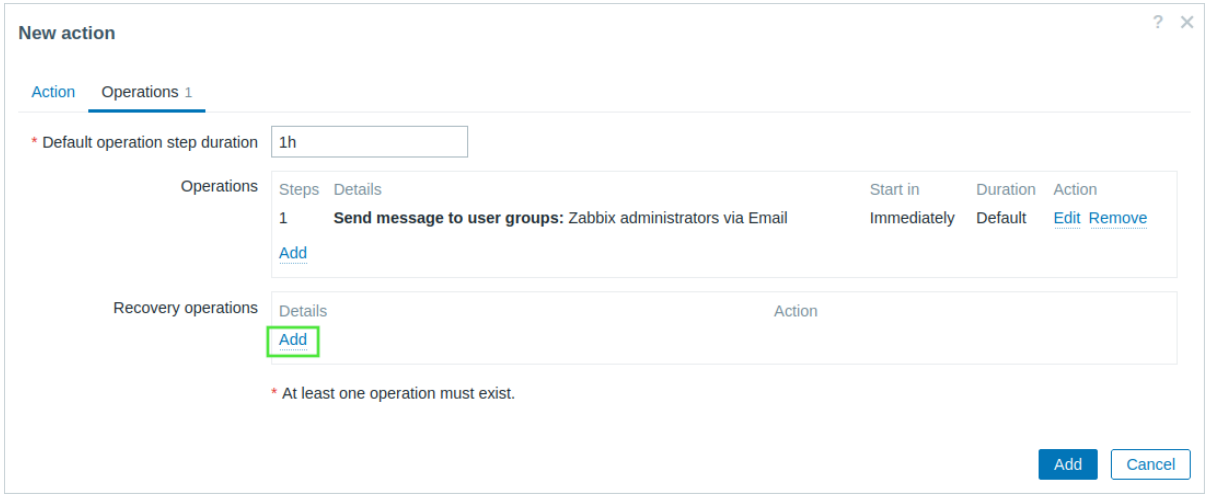


Fai clic su *Add* per elencare effettivamente l'operazione nel blocco *Operations*.

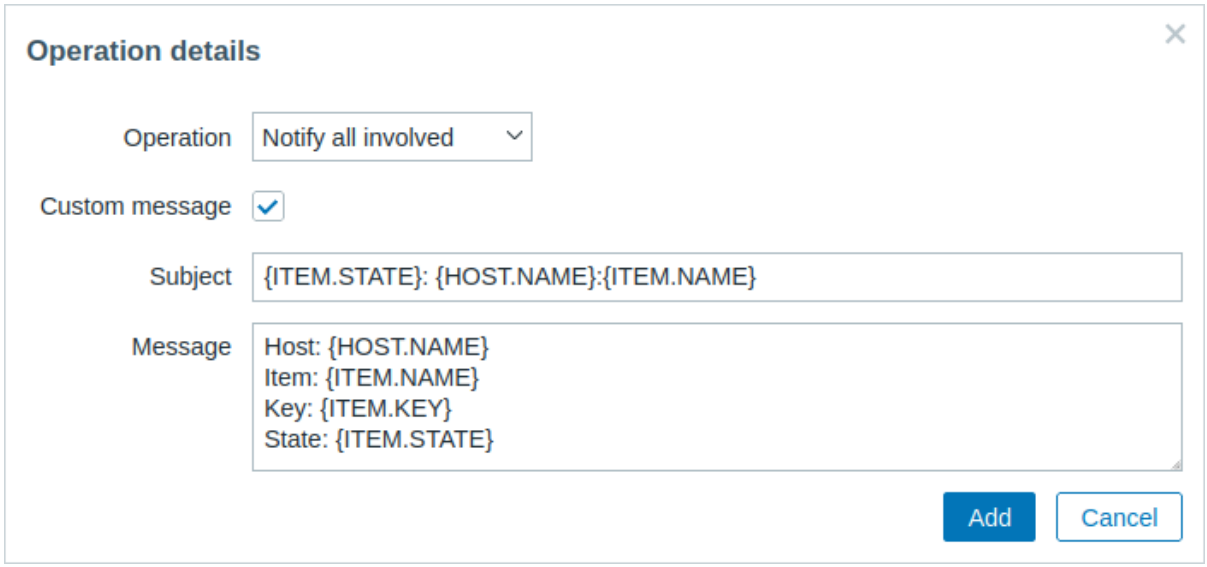
Se desideri ricevere più di una notifica, imposta la durata del passaggio dell'operazione (intervallo tra i messaggi inviati) e aggiungi un altro passaggio.

Passaggio 5

Il blocco *Operazioni di ripristino* consente di configurare una notifica di ripristino quando un item torna allo stato normale. Fare clic su *Aggiungi* nel blocco *Operazioni di ripristino* per aggiungere una nuova operazione di ripristino.



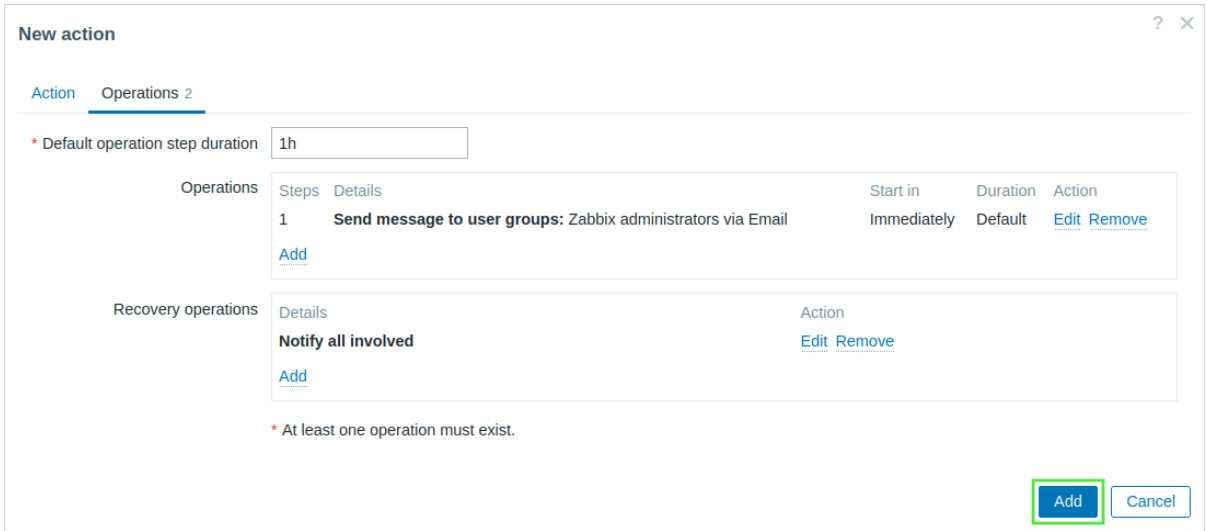
Selezionare il tipo di operazione ”Notifica tutti i coinvolti”. Se si desidera inserire l’oggetto/il contenuto personalizzato del messaggio di ripristino, selezionare la casella di controllo *Messaggio personalizzato*.



Fare clic su *Aggiungi* nella finestra pop-up *Dettagli operazione* per elencare effettivamente l’operazione nel blocco *Operazioni di ripristino*.

Passaggio 6

Al termine, fai clic sul pulsante *Aggiungi* in fondo al modulo.



E questo è tutto, hai finito! Ora non ti resta che attendere la tua prima notifica da Zabbix se qualche item diventa non supportato.

11 Macro

Panoramica

Zabbix supporta un certo numero di 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 usi tipici, una macro può essere utilizzata in un template. Pertanto, 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 Zabbix server, il nome verrà risolto in "Il carico del processore è troppo elevato su Zabbix server" quando il trigger viene visualizzato nella sezione Monitoring.

Le macro possono essere utilizzate nei parametri della chiave item. Una macro può essere usata solo per una parte del parametro, ad esempio `item.key[server_{HOST.HOST}_local]`. Non è necessario racchiudere il parametro tra doppi apici, poiché Zabbix gestirà eventuali simboli speciali ambigui, se presenti nella macro risolta.

Esistono altri tipi di macro in Zabbix.

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 low-level discovery
- `{?EXPRESSION}` - macro di espressione

1 Funzioni macro

Panoramica

Le funzioni macro offrono la possibilità di personalizzare i valori delle macro (ad esempio, abbreviare o estrarre sottostringhe specifiche), rendendoli più facili da utilizzare.

La sintassi di una funzione macro è:

```
{macro.func(params)}
```

dove

- **macro** - la macro da personalizzare;
- **func** - la funzione da applicare (vedere 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 chiuse 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 low-level discovery
- Macro di espressione

Le funzioni macro possono essere utilizzate in tutte le posizioni che supportano le macro elencate. Questo vale salvo nei casi in cui sia esplicitamente indicato che è prevista solo una macro (ad esempio, durante la configurazione delle macro host o dei filtri delle regole di low-level discovery).

È supportata una sola funzione per macro; non sono supportate più funzioni macro concatenate.

Note:
Consultare gli esempi di escaping per i casi in cui le funzioni macro vengono utilizzate 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.

Function	Description
btoa	Codifica del valore della macro in codifica 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 corrispondenza con un'espressione regolare (senza distinzione tra maiuscole e minuscole).
lowercase	Trasformazione dei caratteri del valore della macro in minuscolo.
regrepl	Sostituzione di carattere/sottostringa nel valore della macro.
regsub	Estrazione di una sottostringa tramite corrispondenza con un'espressione regolare (con distinzione tra maiuscole e minuscole).
tr	Traslitterazione dei caratteri del valore della macro.
uppercase	Trasformazione 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.

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 utilizzata con macro che vengono risolte in un valore in uno dei seguenti formati temporali:

- `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) oppure y (anno).

Commenti:

- Il parametro `time_shift` supporta operazioni temporali in più passaggi e può includere `/<time_unit>` per spostarsi all'inizio dell'unità di tempo (`/d` - mezzanotte, `/w` - 1° giorno della settimana (lunedì), `/M` - 1° 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:

```
{{TIME}.fmttime(%B)} - restituirà "October" da un valore ricevuto di "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} - restituirà "1 September" da un valore ricevuto di "1633098961"
```

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
```

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 caratteri/sottostringhe nel valore della macro.

Parametri:

- **pattern** - l'espressione regolare da corrispondere;
- **replacement** - la stringa di sostituzione. I segnaposto **\1 - \9** sono supportati nelle stringhe di sostituzione per i gruppi catturati.

Commenti:

- 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 preceduti da una bar
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - 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"
```

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_11	
{{#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"}

Funzione macro	Valore ricevuto	Output
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$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.

2 Macro utente

Panoramica

Le macro utente sono supportate in Zabbix per una maggiore flessibilità, oltre alle macro supportate disponibili out-of-the-box.

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 il seguente ordine di precedenza:

1. macro a livello di host (controllate per prime)
2. macro definite per i template di primo livello dell'host (cioè 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, e così via
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 con profondità crescente. Se ancora non viene trovata, verrà utilizzata una macro globale, se esiste.

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 è in grado di trovare una macro, la macro non verrà risolta.

Attention:
Le macro (incluse le macro utente) vengono lasciate non risolte nella sezione Configuration (ad esempio, nell'elenco dei trigger) per scelta progettuale, così da rendere più trasparente la configurazione complessa.

Le macro utente possono essere utilizzate in:

- nome item
- parametro della chiave item
- intervalli di aggiornamento 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.

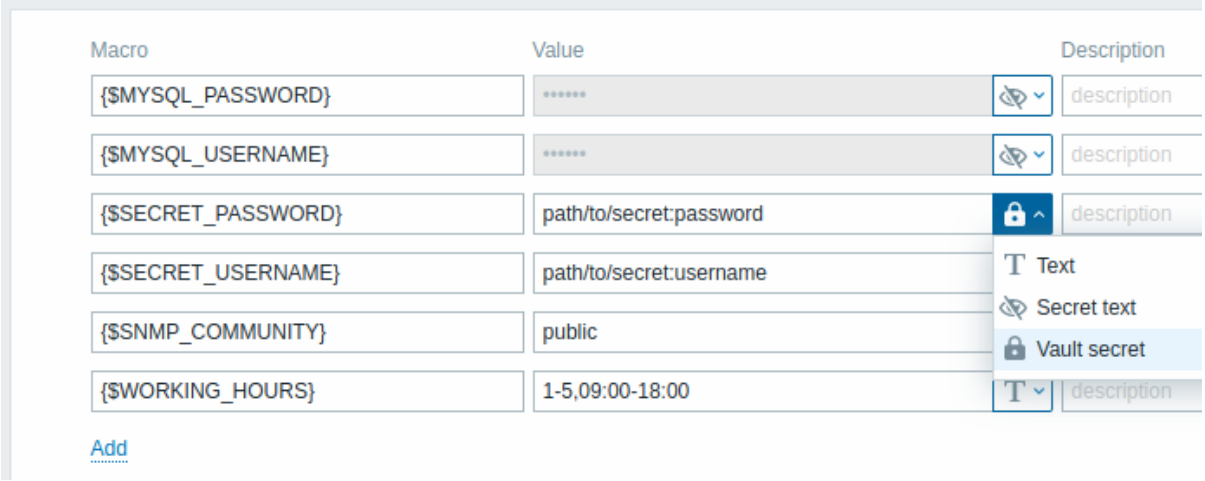
Note:
Si consiglia di utilizzare le macro host invece delle macro globali, perché l'aggiunta, l'aggiornamento o l'eliminazione delle macro globali forza un aggiornamento incrementale della configurazione per tutti gli host. Per ulteriori informazioni, vedere Controlli agent passivi e attivi.

Configurazione

Per definire le macro utente, vai alla posizione corrispondente nel frontend:

- per le macro globali, vai in *Administration* → *Macros*
- per le macro a livello di host e template, apri le proprietà dell'host o del template e cerca la scheda *Macros*

Una macro utente ha i seguenti attributi:



Parametro	Descrizione
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 è mascherato con asterischi **Vault secret** - il valore contiene un percorso/query a un vault secret. Per modificare 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 utilizzato per fornire maggiori informazioni su questa macro.

Quando configuri le macro utente, **nota** i seguenti comportamenti specifici del contesto:

- se le macro utente sono utilizzate in item o trigger di template, valuta di aggiungere tali macro anche al template stesso (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 sono utilizzate nelle espressioni dei 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 item, a una funzione, a un operatore o a un'altra espressione di trigger (le macro segrete non possono essere utilizzate nelle espressioni dei trigger)
- se le macro utente sono utilizzate su un host che dispone di una regola di low-level discovery con prototipi di host, gli host rilevati erediteranno tutte le macro utente definite su quell'host

Esempi

Esempio 1

Uso della macro a livello di host nella chiave 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

Uso di una macro a livello di host nel trigger "CPU load is too high":

```
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 usare il numero di valori come parametro della funzione (ad esempio, **max(/host/key,#3)**), includere il simbolo cancelletto nella definizione della macro in questo modo: SOME_PERIOD => #3

Esempio 3

Uso di due macro nel trigger "CPU load is too high":

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Si noti che una macro può essere utilizzata come parametro di una 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:

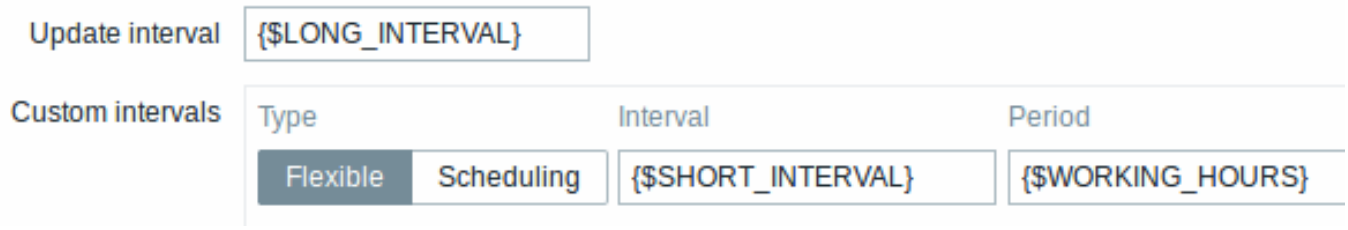
- definire la macro {$INTERVAL} e utilizzarla nell'intervallo di aggiornamento dell'item;
- utilizzare {$INTERVAL} come parametro del trigger di indisponibilità dell'agent:

```
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;
- usarla nel campo *Working time* in *Administration → General → GUI*;
- usarla nel campo *When active* in *Users → Users*, scheda *Media* di un utente;
- usarla per impostare un polling degli item più frequente durante l'orario di lavoro:

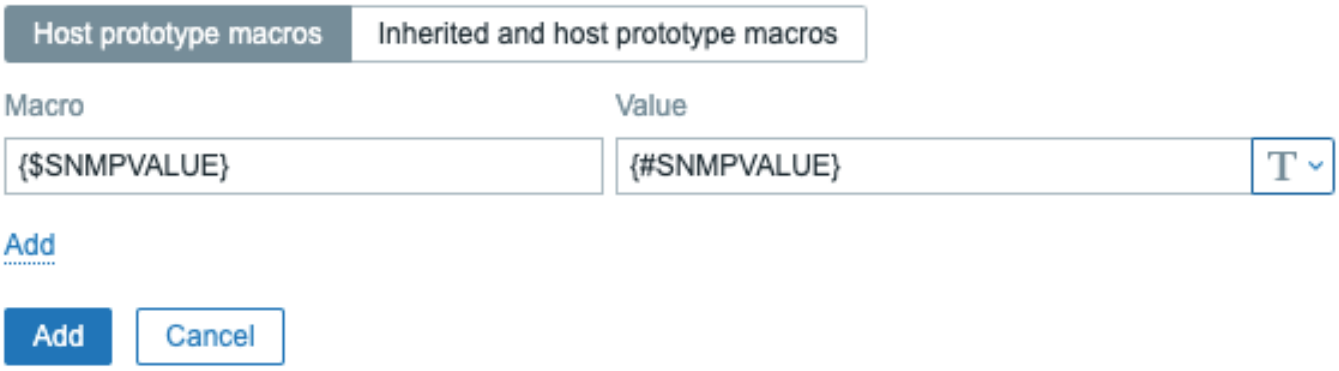


- usarla nella condizione dell'azione *Time period*;
- modificare l'orario di lavoro in *Administration → Macros*, se necessario.

Esempio 6

Utilizzare la macro del prototipo di host per configurare gli item per gli host rilevati:

- in un prototipo di host definire la macro utente {$SNMPVALUE} con la macro di low-level discovery {#SNMPVALUE} come valore:



- assegnare il template *Generic SNMPv2* al prototipo di host;
- utilizzare {$SNMPVALUE} nel campo *SNMP OID* degli item del template *Generic SNMPv2*.

Contesto macro utente

Vedi macro utente con contesto.

3 Macro utente con contesto

Panoramica

È possibile utilizzare un contesto opzionale nelle macro utente, che consente di sovrascrivere il valore predefinito con uno specifico del 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 di espressione regolare può essere definita solo nella configurazione delle macro utente. Se il prefisso `regex:` viene utilizzato altrove come contesto della 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 contesto della 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 con prefisso /var/log/.

Casi d'uso

Le macro utente con contesto possono essere definite per ottenere soglie più flessibili nelle espressioni dei trigger (in base ai valori recuperati dal low-level discovery). 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 low-level discovery può essere utilizzata come contesto macro in un prototipo di trigger per il rilevamento dei file system montati:

```
last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}
```

Dopo il rilevamento, nei trigger verranno applicate soglie di spazio libero differenti 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.

4 Macro utente segrete

Panoramica

Zabbix offre due opzioni per proteggere le informazioni sensibili nei valori delle macro utente:

- Testo segreto
- Segreto del vault

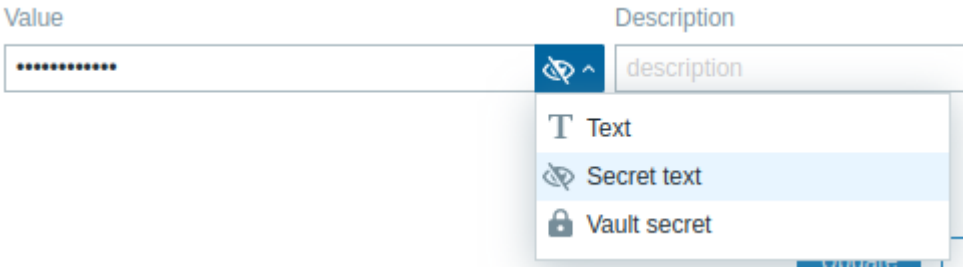
Note:
Sebbene il valore di una macro segreta sia nascosto, può essere rivelato tramite l'uso negli item. Ad esempio, in uno script esterno, un'istruzione echo che fa riferimento a una macro segreta può essere usata per rivelare il valore della macro nel frontend, perché Zabbix server ha accesso al valore 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 il valore di una macro, fare clic sul pulsante alla fine del campo *Valore* e selezionare l'opzione *Testo segreto*:



Una volta salvata la configurazione, non sarà più possibile visualizzare il valore.

Per modificare il valore della macro, passare con il mouse sul campo *Valore* e fare clic sul pulsante *Imposta nuovo valore* (compare al passaggio del mouse):



Quando si fa clic sul pulsante *Imposta nuovo valore* (o si cambia il tipo di valore della macro), il valore corrente verrà cancellato.

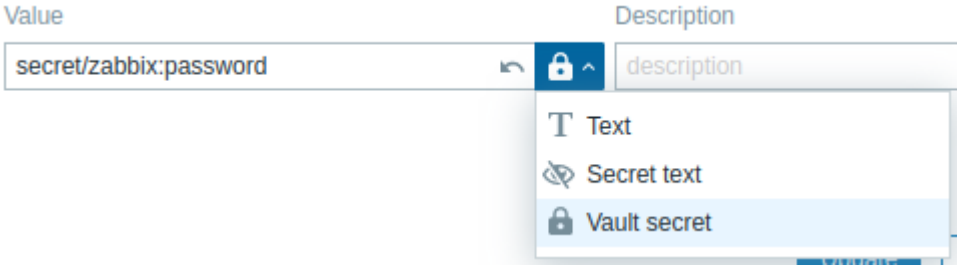
È possibile ripristinare il valore originale facendo clic sulla freccia alla fine del campo *Valore* (disponibile solo prima di salvare la nuova configurazione). Si noti che il ripristino del valore originale non lo renderà visibile.

Note:
Gli URL che contengono una macro segreta non funzioneranno, poiché la macro al loro interno verrà risolta come "******".

Segreto del vault

Con le macro di segreto del vault, il valore della macro viene memorizzato in un software esterno di gestione dei segreti (vault).

Per configurare una macro di segreto del vault, fare clic sul pulsante alla fine del campo *Value* e selezionare 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, vedere:

- HashiCorp
- CyberArk

I valori delle macro segrete di 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*, i segreti del vault vengono recuperati solo dal server e Zabbix proxy riceve i valori delle macro segrete di 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 di controllo runtime `secrets_reload` (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 utilizzate nei tipi di supporto (parametri Script o webhook) verranno risolte in chiavi item contenenti macro segrete mascherate, come `net.tcp.port[******,******]` invece di `net.tcp.port[192.0.2.0,80]`.

Contesto	Parametro
Item, prototipi di item, regole LLD	
Item	*Parametri della chiave item*
Prototipo di item	*Parametri della chiave del prototipo di item*
Regola di low-level discovery	*Parametri della chiave dell'item di discovery*
agent SNMP	*Community SNMP* *Nome del contesto* (SNMPv3) *Nome di sicurezza* (SNMPv3) *Passphrase di autenticazione* (SNMPv3) *Passphrase di privacy* (SNMPv3)
agent HTTP	*URL*

Contesto	Parametro
	Campi della query
	Corpo della richiesta
	Header
	Nome utente
	Password
	Password della chiave SSL
Script	*Parametri*
	Script
Browser	*Parametri*
	Script
Monitor database	*Query SQL*
agent TELNET	*Script*
	Nome utente
	Password
agent SSH	*Script*
	Nome utente
	Password
Controllo semplice	*Nome utente*
	Password
agent JMX	*Nome utente*
	Password
Pre-elaborazione del valore dell'item	
Passaggio di pre-elaborazione JavaScript	*Script*
Scenari web	
Scenario web	*Valore della variabile*
	Valore dell'header
	URL
	Valore del campo query
	Valore del campo POST
	POST grezzo
Autenticazione dello scenario web	*Utente*
	Password
	Password della chiave SSL
Connettori	
Connettore	*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)
Script globali	

Contesto	Parametro
Webhook	*Script JavaScript*
	Valore del parametro dello script JavaScript
Telnet	*Nome utente*
	Password
SSH	*Nome utente*
	Password
Script	*Script*
Tipi di supporto	
Script	*Parametri dello script*
Webhook	*Parametri*
Gestione IPMI	
Host	*Nome utente*
	Password

5 Macro di rilevamento di 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.

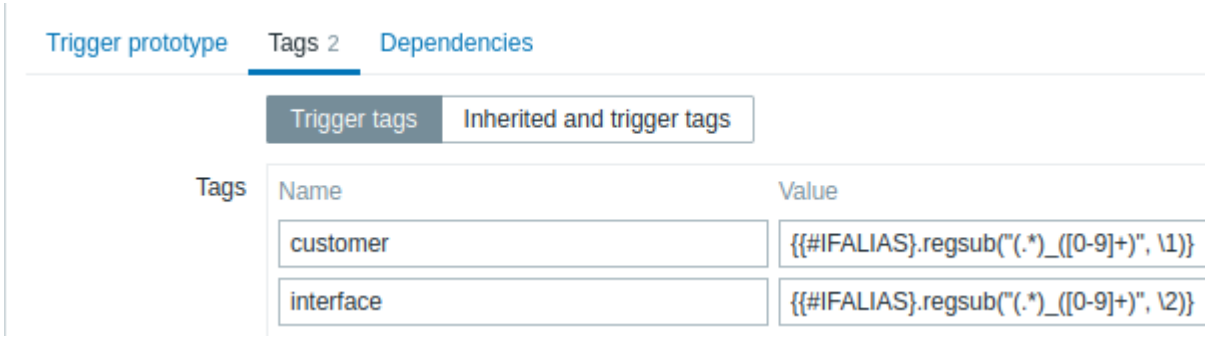
Utilizzo delle funzioni macro

Le funzioni macro sono supportate con le macro di low-level discovery (tranne che nel filtro della regola di low-level discovery), consentendo di estrarre una determinata parte del valore della macro utilizzando un'espressione regolare.

Ad esempio, potresti voler estrarre il nome del cliente e il numero dell'interfaccia dalla seguente macro LLD ai fini del tagging degli eventi:

```
{#IFALIAS}=customername_1
```

Per farlo, è possibile utilizzare la funzione macro `regsub` con la macro nel campo valore del tag evento di un prototipo di trigger:



Si noti che le virgole non sono consentite nei parametri key non tra virgolette di un item, quindi il parametro contenente una funzione macro deve essere racchiuso tra virgolette. Il carattere barra rovesciata (\) 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.

6 Espressioni macro

Panoramica

Le macro di espressione consentono di eseguire calcoli nei campi.

Il loro valore viene calcolato risolvendo prima eventuali macro interne e valutando poi l'espressione risultante.

Sintassi:

```
{?EXPRESSION}
```

`EXPRESSION` utilizza 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, cercare "{?EXPRESSION}" nella tabella delle macro supportate.

Note sull'utilizzo:

- Utilizzare le macro {FUNCTION.*} per fare riferimento ai valori funzione delle espressioni trigger/espressioni di ripristino.
- Utilizzare le macro {HOST.HOST<1-9>} e {ITEM.KEY<1-9>} per fare riferimento a host e item.
- Nei template, utilizzare le macro {HOST.HOST<1-9>} oppure omettere completamente 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 host durante il collegamento dei template.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Vedere anche Esempi di espressioni trigger per un esempio di utilizzo di una macro di espressione in un nome evento.

12 Utenti e gruppi di utenti

Panoramica

Tutti gli utenti in Zabbix accedono all'applicazione Zabbix tramite il frontend web. A ciascun utente vengono assegnati un nome di accesso univoco e una password.

Tutte le password degli utenti sono crittografate e memorizzate nel database di Zabbix. Gli utenti non possono utilizzare il proprio ID utente e la password per accedere direttamente al server UNIX, a meno che non siano stati configurati di conseguenza anche in UNIX. La comunicazione tra il server web 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 host
- utilizzare specifici metodi API

1 Configurazione di un utente

Panoramica

L'installazione iniziale di Zabbix include due utenti predefiniti:

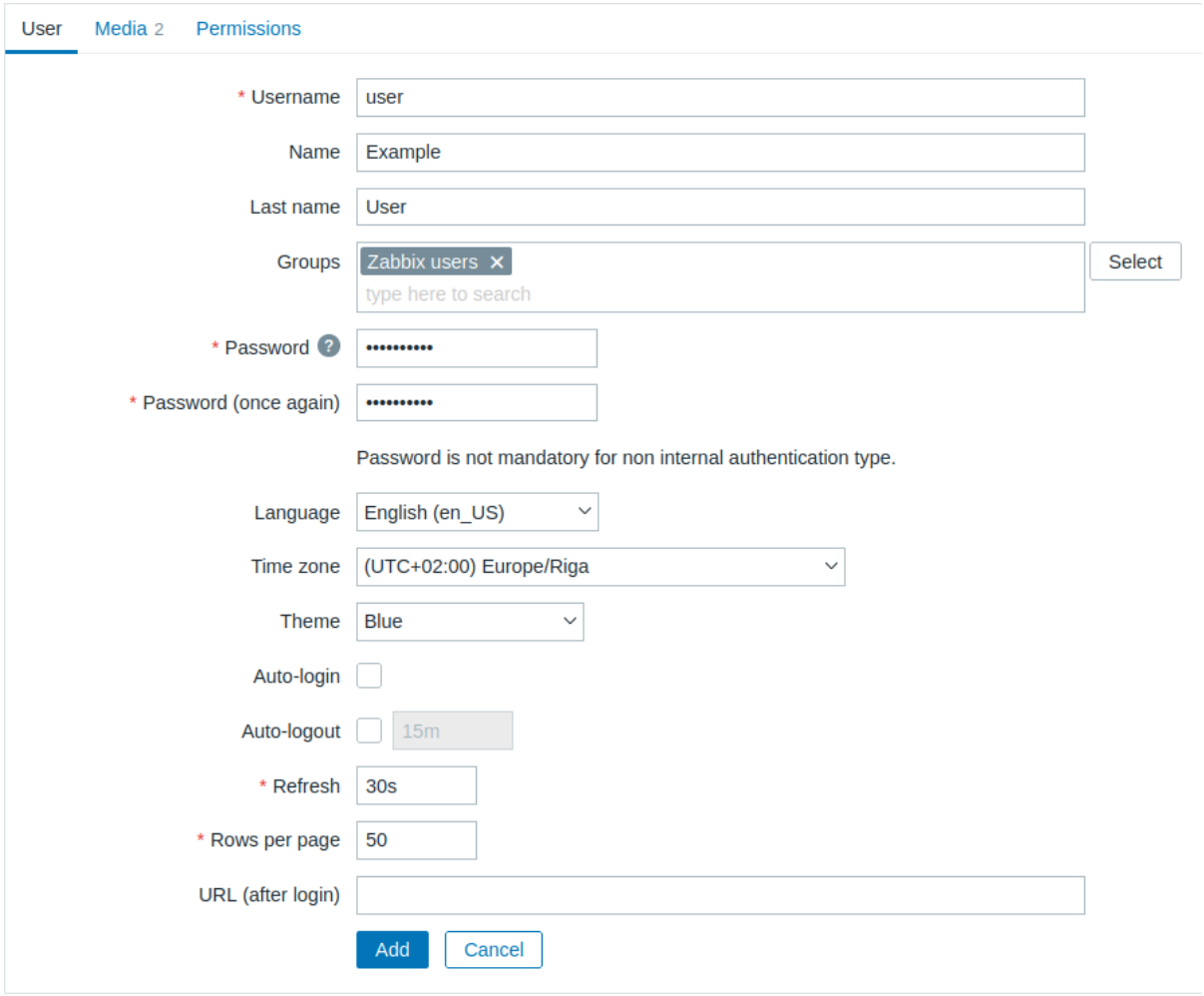
- *Admin* - un superutente di Zabbix con autorizzazioni complete.
- *guest* - uno speciale utente di Zabbix. L'utente 'guest' è disabilitato per impostazione predefinita. Se lo si aggiunge al gruppo utenti Guests, è possibile accedere con questo utente e visualizzare le pagine di monitoraggio in Zabbix. Si noti che per impostazione predefinita 'guest' non dispone di autorizzazioni sugli oggetti di Zabbix.

Per configurare un utente:

- Andare in *Users* → *Users*.
- Fare clic su *Create user* (oppure sul nome di un utente per modificare un utente esistente).
- Modificare gli attributi dell'utente nel modulo.

Attributi generali

La scheda *Utente* contiene gli attributi generali dell'utente:

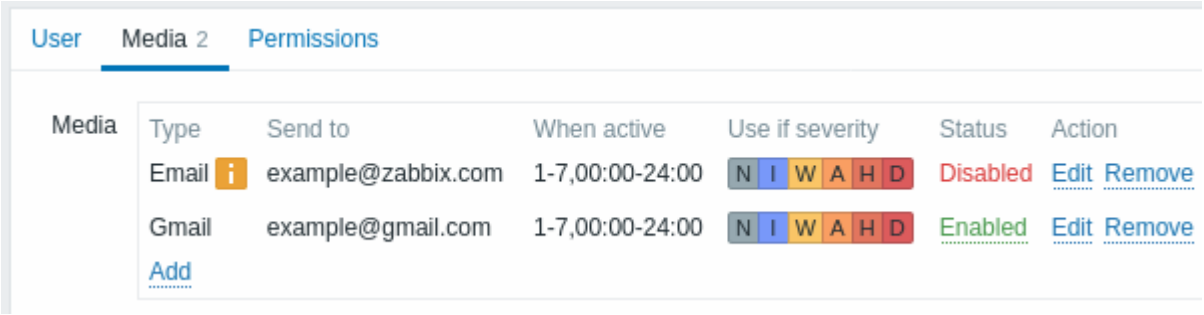


Tutti i campi di input obbligatori sono contrassegnati con un asterisco rosso.

Parameter	Description
Username	Nome utente univoco, utilizzato come nome di accesso.
Name	Nome dell'utente (opzionale). Se non è vuoto, è visibile nelle informazioni di conferma e nelle informazioni sul destinatario delle notifiche.
Last name	Cognome dell'utente (opzionale). Se non è vuoto, è visibile nelle informazioni di conferma e nelle informazioni sul destinatario delle notifiche.
Groups	Selezionare i gruppi utente a cui appartiene l'utente. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo utente verrà mostrato un menu a discesa con i gruppi corrispondenti. Scorrere verso il basso per selezionare. In alternativa, fare clic su *Select* per aggiungere gruppi. Fare clic su 'x' per rimuovere l'elemento selezionato. L'appartenenza ai gruppi utente determina a quali gruppi di host e host l'utente avrà accesso.
Password	Due campi per inserire la password dell'utente, oppure un pulsante *Change password* se l'utente esiste già. Facendo clic sul pulsante *Change password* si aprono due campi per inserire una nuova password. Per l'utente con il *ruolo Super admin* che modifica la propria password, facendo clic sul pulsante *Change password* si apre un campo aggiuntivo per inserire la password corrente (vecchia). In caso di modifica riuscita della password, l'utente per il quale la password è stata modificata verrà disconnesso da tutte le sessioni attive. Si noti che la password può essere modificata solo per gli utenti che utilizzano l'autenticazione interna di Zabbix.
Language	Lingua del frontend di Zabbix. Per il funzionamento delle traduzioni è richiesta l'estensione php gettext.
Time zone	Selezionare il fuso orario per sovrascrivere il fuso orario globale a livello utente oppure selezionare **System default** per utilizzare le impostazioni globali del fuso orario.
Theme	Definisce l'aspetto del frontend: **System default** - usa le impostazioni di sistema predefinite **Blue** - tema blu standard **Dark** - tema scuro alternativo **High-contrast light** - tema chiaro ad alto contrasto **High-contrast dark** - tema scuro ad alto contrasto
Auto-login	Selezionare questa casella per fare in modo che Zabbix ricordi l'utente ed effettui automaticamente l'accesso per 30 giorni. Quando si accede con *Remember for 30 days*: - La disconnessione automatica viene reimpostata (la sessione persiste per 30 giorni). - L'accesso automatico viene abilitato per una nuova autenticazione senza interruzioni. Quando si accede senza *Remember for 30 days*: - L'accesso automatico viene abilitato per una nuova autenticazione senza interruzioni. - La disconnessione automatica continua a essere regolata dall'impostazione standard di timeout. Per questo vengono utilizzati i cookie del browser.
Auto-logout	Con questa casella selezionata, l'utente verrà disconnesso automaticamente dopo il numero di secondi impostato (minimo 90 secondi, massimo 1 giorno). Si noti che questa impostazione viene ignorata se *Remember for 30 days* è abilitato, poiché la sessione viene estesa per l'intero periodo. Sono supportati i suffissi temporali, ad esempio 90s, 5m, 2h, 1d. Si noti che questa opzione non funzionerà: * Se è abilitata l'opzione di configurazione globale "Show warning if Zabbix server is down" e il frontend di Zabbix viene mantenuto aperto. * Quando le pagine del menu Monitoring eseguono aggiornamenti delle informazioni in background. * Se si accede con l'opzione *Remember me for 30 days* selezionata.
Refresh	Impostare la frequenza di aggiornamento utilizzata per grafici, dati in testo semplice, ecc. Può essere impostata su 0 per disabilitarla. Sono supportati i suffissi temporali, ad esempio 90s, 5m, 1h.
Rows per page	È possibile determinare quante righe per pagina verranno visualizzate negli elenchi.
URL (after login)	È possibile fare in modo che Zabbix reindirizzi l'utente a un URL specifico dopo un accesso riuscito, ad esempio alla pagina *Problems*.

Supporti utente

La scheda *Media* contiene un elenco di tutti i supporti definiti per l'utente. I supporti vengono utilizzati per l'invio delle notifiche.



Fare clic su *Add* per assegnare un supporto all'utente.

Se il tipo di supporto è stato disabilitato:

- Dopo il nome viene visualizzata un'icona informativa gialla.
- Nella colonna Status viene visualizzato *Disabled*.

Note:
Le autorizzazioni dell'utente per modificare i dettagli dei propri supporti 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 altri utenti possono essere concesse/revocate in base al relativo ruolo utente (vedere l'opzione *Create and edit user media*)

Per i dettagli sulla configurazione dei supporti utente, vedere la sezione Tipi di supporto.

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:
Gli utenti non possono essere creati senza un ruolo utente (tranne che con la User API di Zabbix). Gli utenti creati in precedenza che non hanno un ruolo possono comunque essere modificati senza assegnare loro un ruolo. Tuttavia, una volta assegnato un ruolo, questo può solo essere modificato, non rimosso.

 Si noti che 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 utenti 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.
  - Gli utenti di tipo *User* e *Admin*, per impostazione predefinita, non hanno accesso ad alcun gruppo, template e host. Per concedere tale accesso, gli utenti devono essere inclusi in gruppi utenti configurati con permessi per le entità pertinenti.
- Diritti di accesso alle sezioni e agli elementi del frontend di Zabbix, ai moduli e ai metodi API.
  - Gli elementi con accesso consentito sono visualizzati in verde, mentre quelli con accesso negato in grigio chiaro.
- Diritti per eseguire azioni specifiche.
  - Le azioni che l'utente è autorizzato a eseguire sono visualizzate in verde, mentre quelle negate in grigio chiaro.

Per i dettagli, vedere la pagina Permessi.

2 Autorizzazioni

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 utente

Le autorizzazioni in Zabbix dipendono principalmente dal tipo di utente:

- *User* - dispone di diritti di accesso limitati alle sezioni del menu (vedi sotto) e, per impostazione predefinita, non ha accesso ad alcuna risorsa. Eventuali autorizzazioni ai gruppi di host o template devono essere assegnate esplicitamente;
- *Admin* - dispone di diritti di accesso non completi alle sezioni del menu (vedi sotto). Per impostazione predefinita, l'utente non ha accesso ad alcun gruppo di host. Eventuali autorizzazioni ai gruppi di host o template devono essere concesse esplicitamente;

- *Super admin* - ha accesso a tutte le sezioni del menu. L'utente dispone di 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 di quella pagina, ma 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		User	Admin	Super admin
Dashboard		+	+	+
Monitoraggio		+	+	+
	Problemi	+	+	+
	Host	+	+	+
	Ultimi dati	+	+	+
	Mappe	+	+	+
	Discovery		+	+
Servizi		+	+	+
	Servizi	+	+	+
	SLA		+	+
	Report SLA	+	+	+
Inventario		+	+	+
	Panoramica	+	+	+
	Host	+	+	+
Report		+	+	+
	Informazioni di sistema			+
	Report pianificati		+	+
	Report di disponibilità	+	+	+
	Top 100 trigger	+	+	+
	Log di audit			+
	Log delle azioni			+
	Notifiche		+	+
Raccolta dati			+	+
	Gruppi di template		+	+
	Gruppi di host		+	+
	Template		+	+
	Host		+	+
	Manutenzione		+	+
	Correlazione eventi			+
	Discovery		+	+
Avvisi			+	+
	Azioni trigger		+	+
	Azioni di servizio		+	+
	Azioni di discovery		+	+
	Azioni di autoregistrazione		+	+
	Azioni interne		+	+
	Tipi di supporto			+
	Script			+
Utenti				+
	Gruppi di utenti			+
	Ruoli utente			+
	Utenti			+
	Token API			+
	Autenticazione			+
Amministrazione				+
	Generale			+
	Log di audit			+
	Housekeeping			+
	Gruppi proxy			+
	Proxy			+
	Macro			+

Sezione del menu	User	Admin	Super admin
Coda			+

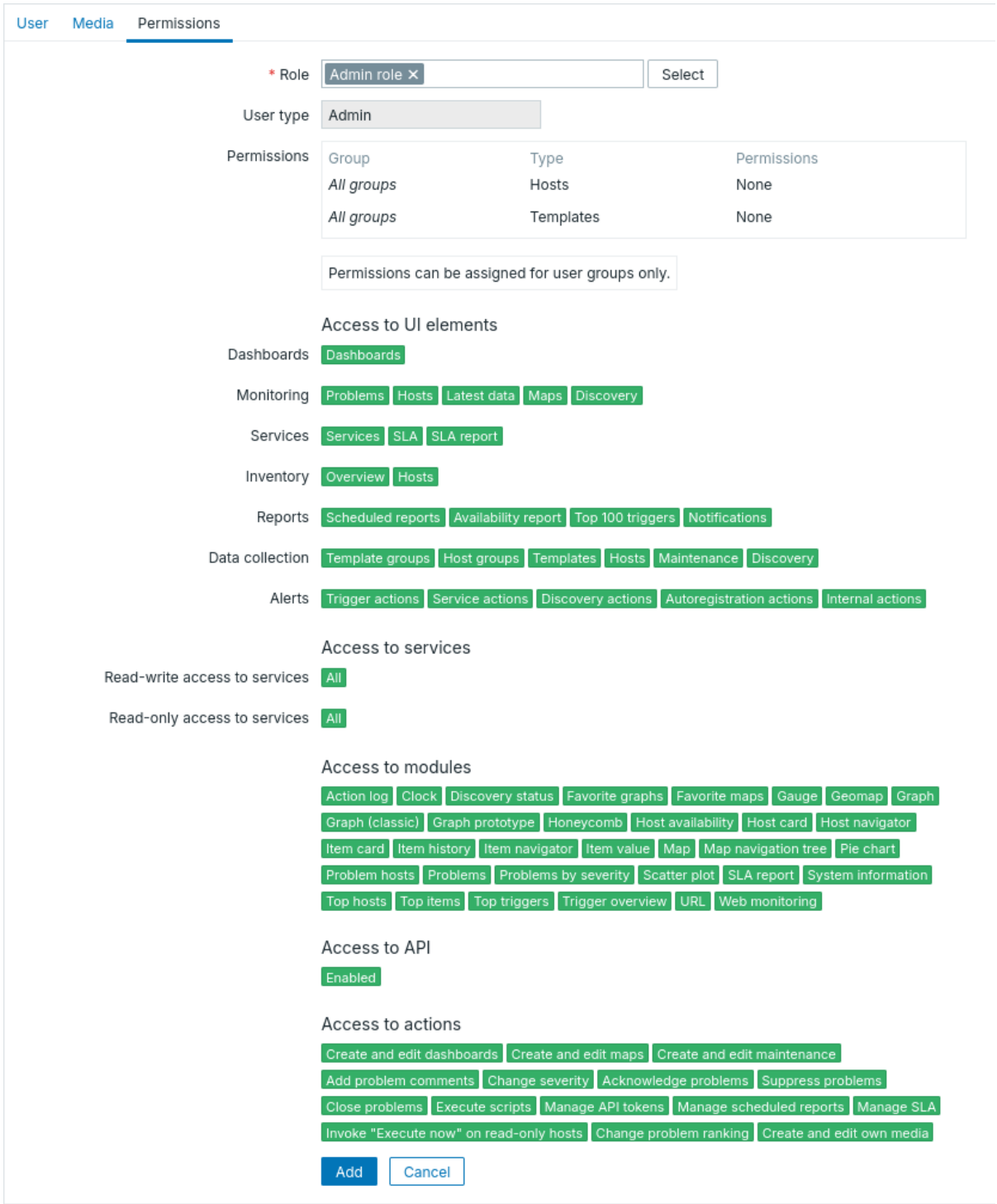
Ruoli utente

I ruoli utente consentono di apportare modifiche personalizzate ai permessi definiti dal tipo di utente. Sebbene non sia possibile aggiungere permessi (poiché ciò supererebbe quelli del tipo di utente), alcuni permessi possono essere revocati.

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 nel modulo di configurazione utente, scheda *Permessi*, dagli utenti Super admin.



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 di host/template.

Ciò significa che a un singolo utente non può essere concesso direttamente l'accesso a un host (o gruppo di host). L'accesso a un host può essere concesso solo facendolo parte di un gruppo di utenti a cui è stato concesso l'accesso al gruppo di host che contiene l'host.

Allo stesso modo, a un utente può essere concesso l'accesso a un template solo facendolo parte di un gruppo di utenti a cui è stato concesso l'accesso al gruppo di template che contiene il template.

3 Gruppi di utenti

Panoramica

I gruppi di utenti consentono di raggruppare gli utenti sia per scopi organizzativi sia per l'assegnazione dei permessi ai dati. I permessi per visualizzare e configurare i dati dei gruppi di host e dei gruppi di template vengono assegnati ai gruppi di utenti, non ai singoli utenti.

Spesso può essere opportuno separare quali informazioni siano disponibili per un gruppo di utenti e quali per un altro. Questo può essere ottenuto raggruppando gli utenti e assegnando poi permessi differenti ai gruppi di host e di template.

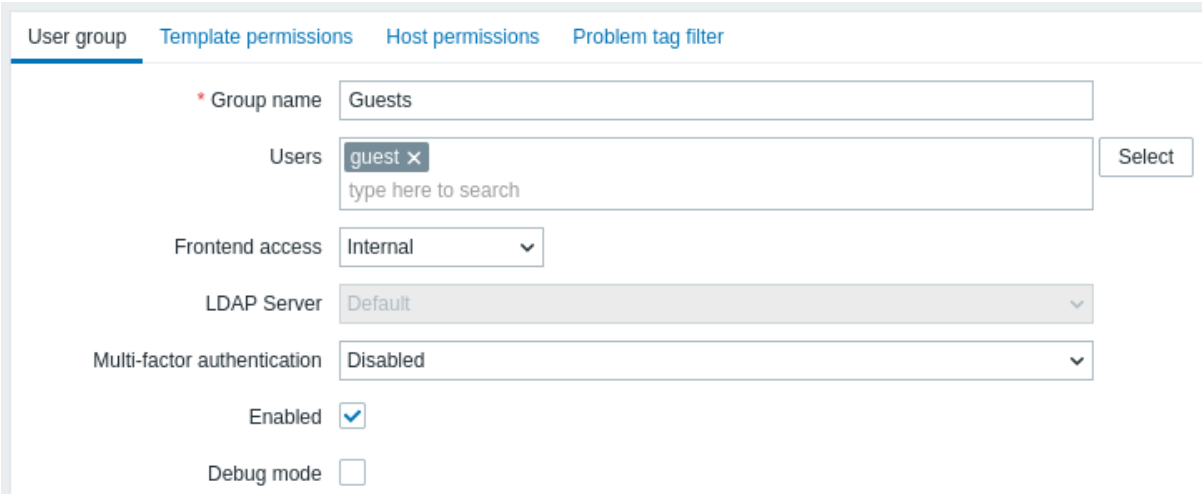
Un utente può appartenere a un numero qualsiasi di gruppi.

Configurazione

Per configurare un gruppo utenti:

- Andare in *Utenti* > *Gruppi utenti*
- Fare clic su *Crea gruppo utenti* (oppure sul nome del gruppo per modificare un gruppo esistente)
- Modificare gli attributi del gruppo nel modulo

La scheda **Gruppo utenti** contiene gli attributi generali del gruppo:

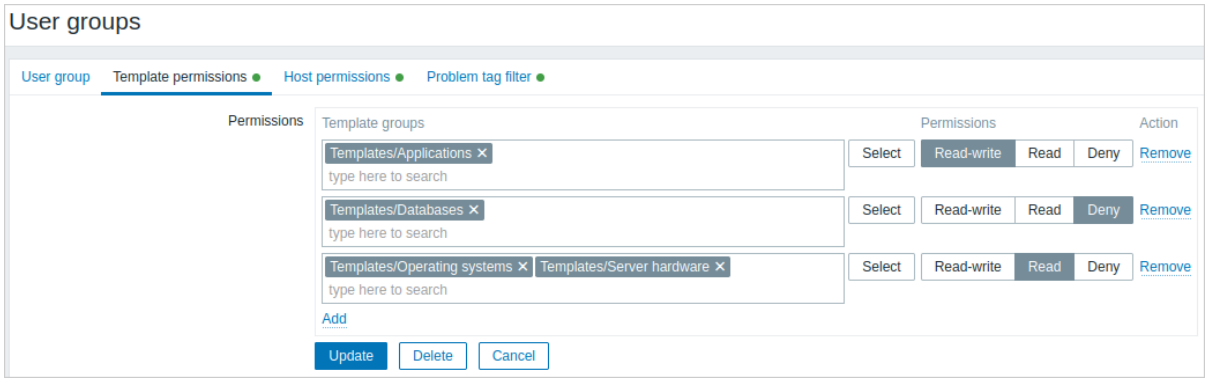


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

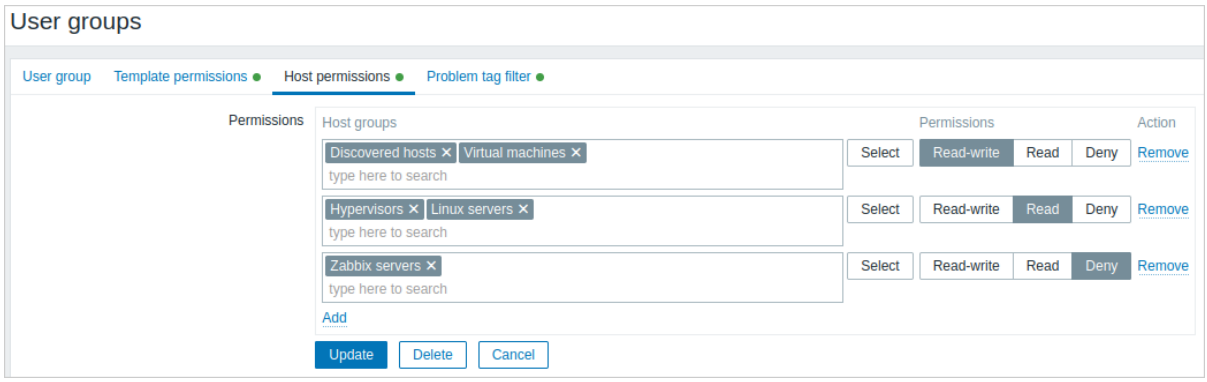
Parametro	Descrizione
Nome gruppo	Nome univoco del gruppo.
Utenti	Per aggiungere utenti al gruppo, iniziare a digitare il nome di un utente esistente. Quando appare il menu a discesa con i nomi utente corrispondenti, scorrere verso il basso per selezionare. In alternativa, è possibile fare clic sul pulsante *Seleziona* per selezionare gli utenti in una finestra popup.
Accesso al frontend	Come vengono autenticati gli utenti del gruppo. **Predefinito del sistema** - utilizza il metodo di autenticazione predefinito (impostato globalmente) **Interno** - utilizza l'autenticazione interna di Zabbix (anche se globalmente viene utilizzata l'autenticazione LDAP). Ignorato se l'autenticazione HTTP è quella predefinita globale. **LDAP** - utilizza l'autenticazione LDAP (anche se globalmente viene utilizzata l'autenticazione interna). Ignorato se l'autenticazione HTTP è quella predefinita globale. **Disabilitato** - l'accesso al frontend di Zabbix è vietato per questo gruppo

Parametro	Descrizione
Server LDAP	Selezionare quale server LDAP utilizzare per autenticare l'utente. Questo campo è abilitato solo se *Accesso al frontend* è impostato su LDAP o Predefinito del sistema.
Autenticazione a più fattori	Selezionare 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 utenti se MFA è abilitata; **<Nome 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 utenti se MFA è disabilitata. Si noti che se un utente appartiene a più gruppi utenti con MFA abilitata (o almeno un gruppo ha MFA abilitata), si applicano le seguenti regole di autenticazione: se un qualsiasi gruppo utilizza il metodo MFA "Predefinito", questo autenticherà l'utente; altrimenti, per l'autenticazione verrà utilizzato il primo metodo (in ordine alfabetico).
Abilitato	Stato del gruppo utenti e dei membri del gruppo. *Selezionato* - il gruppo utenti e gli utenti sono abilitati *Non selezionato* - il gruppo utenti e gli utenti sono disabilitati
Modalità di debug	Selezionare questa casella per attivare la modalità di debug per gli utenti.

La scheda **Permessi template** consente di specificare l'accesso del gruppo utenti ai dati del gruppo template (e quindi del template):



La scheda **Permessi host** consente di specificare l'accesso del gruppo utenti ai dati del gruppo host (e quindi dell'host):



Fare clic su Add per scegliere i gruppi template/host (sia un gruppo padre sia un gruppo nidificato) e assegnare loro i permessi. Iniziare a digitare il nome del gruppo (apparirà un menu a discesa con i gruppi corrispondenti) oppure fare clic su *Seleziona* per aprire una finestra popup con l'elenco di tutti i gruppi.

Quindi utilizzare i pulsanti di opzione per assegnare i permessi ai gruppi scelti. I permessi possibili sono i seguenti:

- **Lettura-scrittura** - accesso in lettura e scrittura a un gruppo
- **Lettura** - accesso in sola lettura a un gruppo
- **Nega** - accesso a un gruppo negato

Se lo stesso gruppo template/host viene aggiunto in più righe con permessi diversi, verrà applicato il permesso più restrittivo.

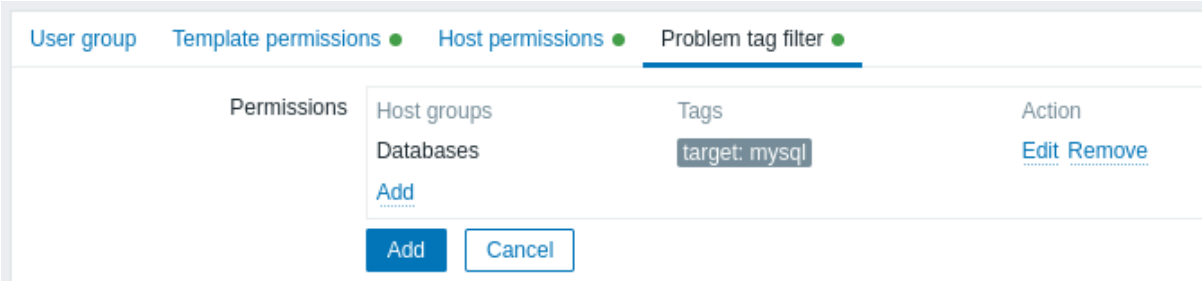
Si noti che un utente *Super admin* può imporre ai gruppi nidificati di avere lo stesso livello di permessi del gruppo padre; ciò può essere fatto nel modulo di configurazione del gruppo host/template.

Le schede **Permessi template** e **Permessi host** supportano lo stesso insieme di parametri.

I permessi correnti per i gruppi sono visualizzati nel blocco *Permessi* e possono essere modificati o rimossi.

Note:
Se un gruppo utenti ha permessi **Lettura-scrittura** su un host e permessi **Nega** o nessun permesso su un template collegato a questo host, gli utenti di tale gruppo non potranno modificare gli item derivati dal template sull'host e il nome del template verrà visualizzato come *Template inaccessibile*.

La scheda **Filtro tag problemi** consente di impostare permessi basati su tag per i gruppi utenti, in modo da visualizzare i problemi filtrati per nome e valore del tag:

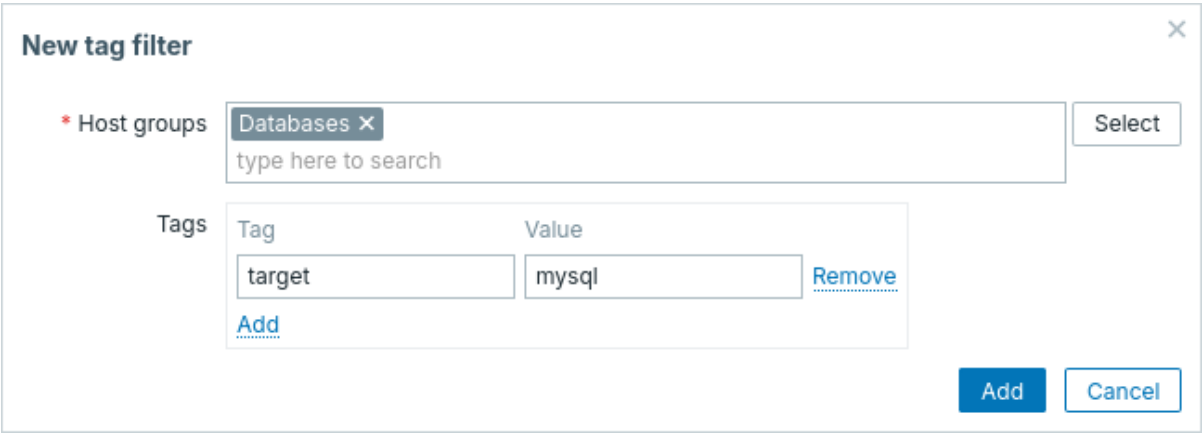


Fare clic su Add per scegliere i gruppi host. Per selezionare un gruppo host a cui applicare un filtro tag, fare clic su *Seleziona* per ottenere l'elenco completo dei gruppi host esistenti oppure iniziare a digitare il nome di un gruppo host per ottenere un menu a discesa con i gruppi corrispondenti. Verranno visualizzati solo i gruppi host, perché il filtro tag problemi non può essere applicato ai gruppi template.

È possibile aggiungere nomi di tag senza valori, ma non valori senza nomi. Nel blocco *Permessi* vengono visualizzati solo i primi tre tag (con i relativi valori, se presenti); se ce ne sono altri, possono essere visualizzati facendo clic o passando il puntatore sull'icona ••• .

Il filtro tag consente di separare l'accesso al gruppo host dalla possibilità di vedere i problemi.

Ad esempio, se un amministratore di database deve vedere solo i problemi del database "MySQL", è necessario prima creare un gruppo utenti per gli amministratori di database, quindi specificare il nome tag "target" e il valore "mysql".



Se viene specificato il nome tag "target" e il campo valore viene lasciato vuoto, il gruppo utenti vedrà tutti i problemi con nome tag "target" per il gruppo host selezionato.

Assicurarsi che il nome del tag e il valore del tag siano specificati correttamente, altrimenti il gruppo utenti non vedrà alcun problema.

Esaminiamo un esempio in cui un utente è membro di diversi gruppi utenti selezionati. In questo caso il filtro utilizzerà la condizione OR per i tag.

Gruppo utenti A			**Gruppo utenti B**			**Risultato visibile per un utente (membro) di entrambi i gruppi**
Filtro tag						
Gruppo host	*Nome tag*	*Valore tag*	*Gruppo host*	*Nome tag*	*Valore tag*	

Databases	target	mysql	Databases	target	oracle	problemi target:mysql o target:oracle visibili
Non configurato nel **Filtro tag problemi**			Databases	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".

13 Conservazione dei segreti

Panoramica

Zabbix può essere configurato per recuperare informazioni sensibili da un vault sicuro. Sono supportati i seguenti servizi di gestione dei segreti: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

I segreti possono essere utilizzati per recuperare:

- valori delle macro utente
- credenziali di accesso al database

Zabbix fornisce accesso in sola lettura ai segreti in un vault, presupponendo che i segreti siano gestiti da qualcun altro.

For information about specific vault provider configuration, see: - HashiCorp configuration - CyberArk configuration

Cache dei valori segreti

Per impostazione predefinita, i valori delle macro segrete del vault vengono recuperati da Zabbix server a ogni aggiornamento dei dati di configurazione e quindi memorizzati nella cache di configurazione. 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.

Attention:
La crittografia deve essere abilitata tra Zabbix server e proxy; in caso contrario viene registrato un messaggio di avviso del server.

È inoltre possibile configurare che i valori delle macro vengano recuperati in modo indipendente da Zabbix server e Zabbix proxy.

Per attivare manualmente l'aggiornamento dei valori segreti memorizzati nella cache da un vault, utilizzare l'opzione da riga di comando 'secrets_reload'.

Per le credenziali del database di Zabbix frontend, la cache è disabilitata per impostazione predefinita, ma può essere abilitata impostando l'opzione `$DB['VAULT_CACHE'] = true` in zabbix.conf.php. Le credenziali verranno memorizzate in una cache locale utilizzando la directory dei file temporanei del filesystem. Il server web 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 con cui la cache dei dati viene aggiornata/invalida, utilizzare la costante ZBX_DATA_CACHE_TTL .

Configurazione TLS

Per configurare TLS per la comunicazione tra i componenti di Zabbix e il vault, aggiungere al repository CA predefinito a livello di sistema un certificato firmato da un'autorità di certificazione (CA). Per utilizzare un'altra posizione, specificare la directory nel parametro di configurazione SSLCALocation di Zabbix server/proxy, inserire il file del certificato all'interno di tale directory, quindi eseguire il comando CLI:

```
c_rehash .
```

1 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 con le credenziali del database viene configurato separatamente per ciascun componente di Zabbix.

Server e proxy

Per ottenere le credenziali del database dal vault per Zabbix server o proxy, specificare i seguenti parametri di configurazione nel file di configurazione:

- `Vault` - quale provider vault deve essere utilizzato;
- `VaultURL` - URL HTTP[S] del server 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 e della chiave SSL; 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 adatto.

Attention:
I parametri di configurazione `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` e `VaultPrefix` vengono utilizzati anche per l'autenticazione al vault durante l'elaborazione delle macro del vault segreto da parte di Zabbix server (e di Zabbix proxy, se configurato). Zabbix server e i proxy non apriranno le macro del secret del vault che contengono credenziali DB da VaultDBPath.

Zabbix server e Zabbix proxy 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>
}
```

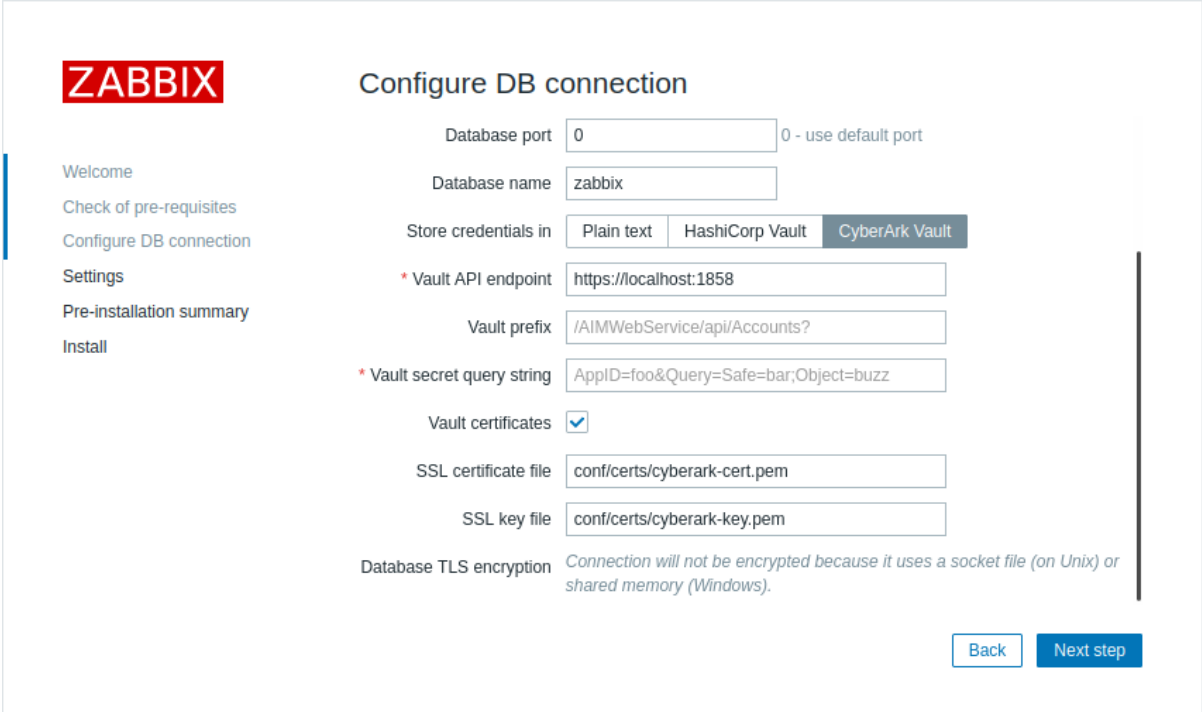
4. Di conseguenza, Zabbix utilizzerà le seguenti credenziali per l’autenticazione al database:

- Nome utente: <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. Quindi, compilare i parametri aggiuntivi:

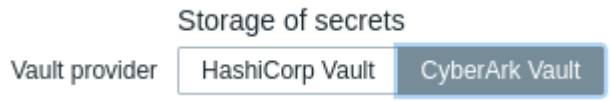
Parametro	Obbligatorio	Valore predefinito	Descrizione
Endpoint API del vault	sì	https://localhost:1858	Specificare l’URL per la connessione al vault nel formato `scheme://host:port`
Prefisso del vault	no	/AIMWebService/api/Accounts?	Fornire un prefisso personalizzato per il percorso o la query del vault. Se non specificato, viene utilizzato quello predefinito.
Stringa di query del segreto del vault	sì		Una query che specifica da dove devono essere recuperate le credenziali del database. Esempio: `AppID=foo&Query=Safe=bar;Object=buzz`

Parametro	Obbligatorio	Valore predefinito	Descrizione
Certificati del vault	no		Dopo aver selezionato la casella di controllo, verranno visualizzati parametri aggiuntivi che consentono di configurare l'autenticazione del client. Sebbene questo parametro sia facoltativo, è altamente consigliato abilitarlo per la comunicazione con il CyberArk Vault.
File del certificato SSL	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 del file della chiave SSL.
File della chiave SSL	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 utilizzare CyberArk Vault per memorizzare i valori delle macro utente *Vault secret*, assicurarsi che:

- il server Zabbix sia configurato per funzionare con CyberArk Vault;
- il parametro *Vault provider* in *Amministrazione → Generale → Altro* sia impostato su "CyberArk Vault".



Note:
Il server Zabbix (e il proxy Zabbix, se configurato) richiedono l'accesso ai valori delle macro *Vault secret* dal vault. Il frontend Zabbix non necessita di tale accesso.

Il valore della macro deve contenere una query (nel formato `query:key`).

Per informazioni dettagliate sull'elaborazione dei valori delle macro da parte di Zabbix, vedere *Macro Vault secret*.

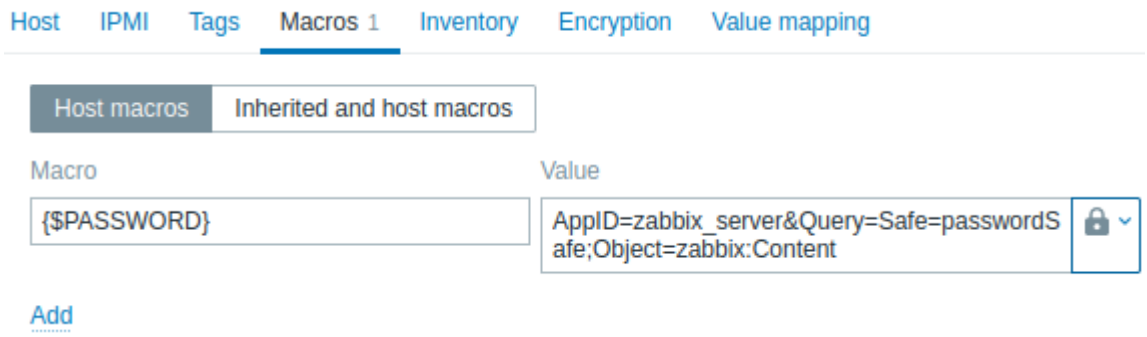
Sintassi della query

Il simbolo dei due punti (":") è riservato per separare la query dalla chiave.

Se una query contiene a sua volta una barra obliqua o i due punti, questi simboli devono essere codificati tramite URL ("/" viene codificato come "%2F", ":" viene codificato come "%3A").

Esempio

1. In Zabbix, aggiungere una macro utente {$PASSWORD} di tipo *Vault secret* con il valore `AppID=zabbix_server&Query=Safe=passw`



2. Zabbix invierà al vault la seguente richiesta API:

```
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 una configurazione esistente

Per aggiornare una configurazione esistente per il recupero dei segreti da un Vault CyberArk:

1. Aggiornare i parametri del file di configurazione del server o del proxy Zabbix come descritto nella sezione *Credenziali del database*.
2. Aggiornare le impostazioni di connessione al DB riconfigurando il frontend Zabbix e specificando i parametri richiesti come descritto nella sezione *Frontend*. Per riconfigurare il frontend Zabbix, aprire nel browser l’URL di configurazione del frontend:

- per Apache: http://<server_ip_or_name>/zabbix/setup.php
- per Nginx: http://<server_ip_or_name>/setup.php

In alternativa, questi parametri possono essere impostati nel file di configurazione del frontend (*zabbix.conf.php*):

```
$DB['VAULT']                    = 'CyberArk';
$DB['VAULT_URL']                = 'https://127.0.0.1:1858';
$DB['VAULT_DB_PATH']            = 'AppID=foo&Query=Safe=bar;Object=buzz';
$DB['VAULT_TOKEN']              = '';
$DB['VAULT_CERT_FILE']          = 'conf/certs/cyberark-cert.pem';
$DB['VAULT_KEY_FILE']           = 'conf/certs/cyberark-key.pem';
$DB['VAULT_PREFIX']             = '';
```

3. 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 Vault HashiCorp, vedere *Configurazione HashiCorp*.

2 Configurazione di HashiCorp

Panoramica

Questa sezione spiega come configurare Zabbix per recuperare segreti da HashiCorp Vault KV Secrets Engine - Versione 2.

Il vault deve essere distribuito e configurato come descritto nella documentazione ufficiale di HashiCorp.

Per informazioni sulla configurazione di TLS in Zabbix, vedere *Archiviazione dei segreti*.

Recupero delle credenziali del database

Per recuperare correttamente un segreto contenente le credenziali del database, è necessario configurare entrambi:

- Zabbix server/proxy
- Zabbix frontend

Server/proxy

Per configurare Zabbix server o proxy, specificare i seguenti parametri di configurazione nel file di configurazione:

- `Vault` - quale provider vault deve essere utilizzato;
- `VaultToken` - token di autenticazione del vault (vedere il file di configurazione di Zabbix server/proxy per i dettagli);
- `VaultURL` - URL HTTP[S] del server vault;
- `VaultDBPath` - percorso del segreto vault contenente le credenziali del database (questa opzione può essere utilizzata solo se DBUser e DBPassword non sono specificati); Zabbix server o proxy recupererà le credenziali tramite le chiavi ”password” e ”username”;
- `VaultPrefix` - prefisso personalizzato per il percorso o la query del vault, a seconda del vault; se non specificato, verrà utilizzato il valore predefinito più adatto.

Attention:
I parametri di configurazione `Vault`, `VaultToken`, `VaultURL` e `VaultPrefix` vengono utilizzati anche per l'autenticazione al vault durante l'elaborazione delle macro del vault segreto da parte di Zabbix server (e di Zabbix proxy, se configurato). Zabbix server e i proxy non apriranno le macro del vault segreto che contengono credenziali DB da VaultDBPath.

Si consiglia vivamente di utilizzare token diversi per proxy diversi.

Zabbix server e Zabbix proxy leggono i parametri di configurazione relativi al vault da *zabbix_server.conf* e *zabbix_proxy.conf* all'avvio. Inoltre, Zabbix server e Zabbix proxy leggeranno la variabile d'ambiente `VAULT_TOKEN` una volta durante l'avvio e la rimuoveranno in modo che non sia disponibile tramite script forkati; è considerato un errore se sia `VaultToken` sia `VAULT_TOKEN` contengono un valore.

Esempio

1. In *zabbix_server.conf*, specificare i seguenti parametri:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
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 percor
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 verificare con Curl; notare che "data" deve essere aggiunto manualmente dopo il punto di mount
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Di conseguenza, Zabbix server recupererà le seguenti credenziali per l'autenticazione al database:

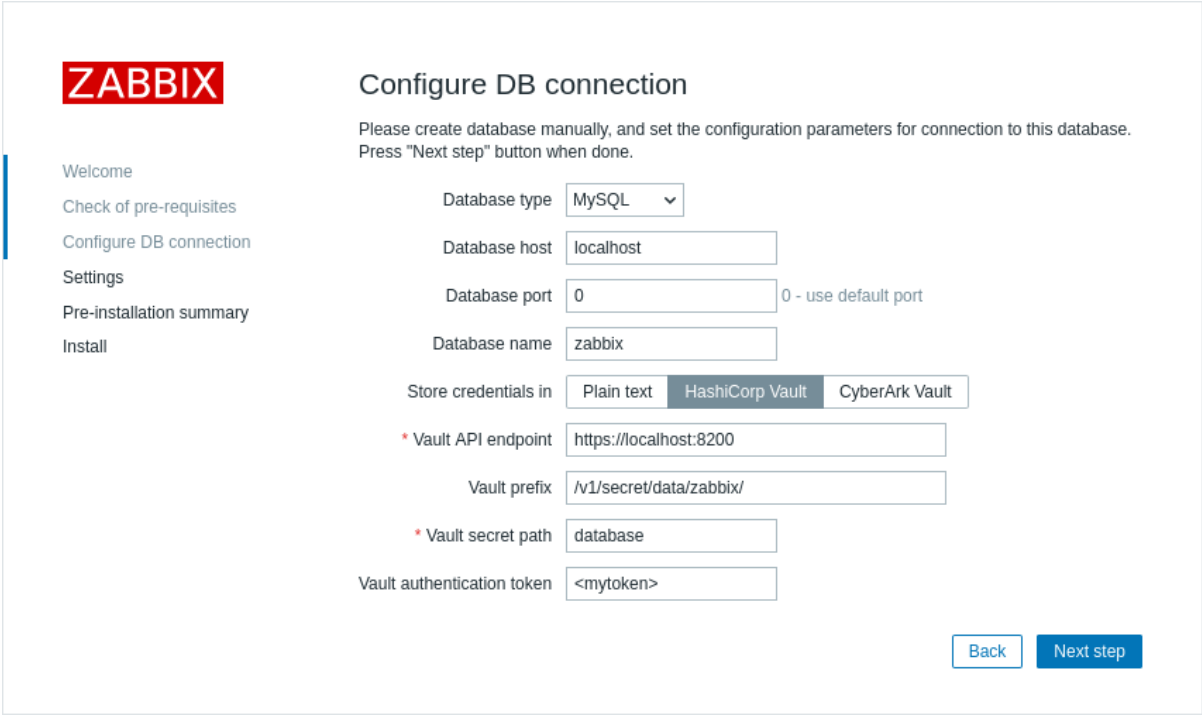
- Nome utente: zabbix
- Password: <password>

Frontend

Il frontend di Zabbix può essere configurato per recuperare le credenziali del database dal vault sia durante l'installazione del frontend sia aggiornando il file di configurazione del frontend (`zabbix.conf.php`).

Attention:
Se le credenziali del vault sono state modificate dopo la precedente installazione del frontend, eseguire nuovamente l'installazione del frontend oppure aggiornare `zabbix.conf.php`. Vedere anche: Aggiornamento della configurazione esistente.

Durante l'**installazione del frontend** i parametri di configurazione devono essere specificati nel passaggio *Configure DB Connection*:



- Impostare il parametro *Store credentials in* su ”HashiCorp Vault”.
- Specificare i parametri di connessione:

Parameter	Mandatory	Default value	Description
Vault API endpoint	yes	https://localhost:8200	Specificare l’URL per la connessione al vault nel formato `scheme://host:port`
Vault prefix	no	/v1/secret/data/	Fornire 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	no		Un percorso del segreto da cui recuperare le credenziali per il database tramite le chiavi ”password” e ”username”. Esempio: `database`
Vault authentication token	no		Fornire un token di autenticazione con accesso in sola lettura al percorso del segreto. Consultare la documentazione di HashiCorp per informazioni sulla creazione di token e policy del vault.

Recupero dei valori delle macro utente

Per utilizzare HashiCorp Vault per memorizzare i valori delle macro utente *Vault secret*, assicurati 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 deve 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, vedi *Vault secret macros*.

Sintassi del percorso

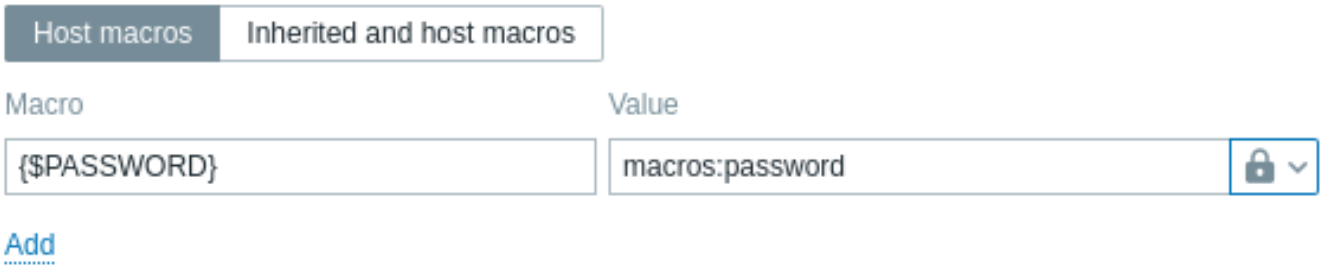
I simboli barra obliqua ("/") e due punti (":") sono riservati.

Una barra obliqua può essere utilizzata solo per separare un punto di mount da un percorso (ad esempio, *secret/zabbix*, dove il punto di mount è "secret" e il percorso è "zabbix"). Nel caso delle macro Vault, i due punti possono essere utilizzati 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 punto di mount con un nome separato da una barra obliqua (ad esempio, *foo/bar/zabbix*, dove il punto di mount è "foo/bar" e il percorso è "zabbix", può essere codificato come "foo%2Fbar/zabbix") e se un nome di punto di mount o un percorso devono contenere i due punti.

Esempio

1. In Zabbix, aggiungere una macro utente {$PASSWORD} di tipo "Vault secret" e con il valore `macros:password`



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 un nuovo secret con chiave "password" sotto il punto di mount "secret/" e il percorso "zabbi
vault kv put -mount=secret zabbix/macros password=<password>

#### Verificare che il secret sia stato aggiunto correttamente.
vault kv get secret/zabbix/macros

#### Infine verificare con Curl; notare che "data" deve essere aggiunto manualmente dopo il punto di mount
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 di una configurazione esistente

Per aggiornare una configurazione esistente per il recupero dei segreti da un HashiCorp Vault:

1. Aggiornare i parametri del file di configurazione di Zabbix server o proxy come descritto nella sezione *Credenziali del database*.
2. Aggiornare le impostazioni di connessione al DB riconfigurando Zabbix frontend e specificando i parametri richiesti come descritto nella sezione *Frontend*. Per riconfigurare Zabbix frontend, aprire nel browser l'URL di configurazione del frontend:

- per Apache: http://<server_ip_or_name>/zabbix/setup.php
- per Nginx: http://<server_ip_or_name>/setup.php

In alternativa, questi parametri possono essere impostati nel file di configurazione del frontend (*zabbix.conf.php*):

```
$DB['VAULT']                    = 'HashiCorp';
$DB['VAULT_URL']                = 'https://localhost:8200';
$DB['VAULT_DB_PATH']            = 'database';
$DB['VAULT_TOKEN']              = '<mytoken>';
$DB['VAULT_CERT_FILE']          = '';
$DB['VAULT_KEY_FILE']           = '';
$DB['VAULT_PREFIX']             = '/v1/secret/data/zabbix/';
```

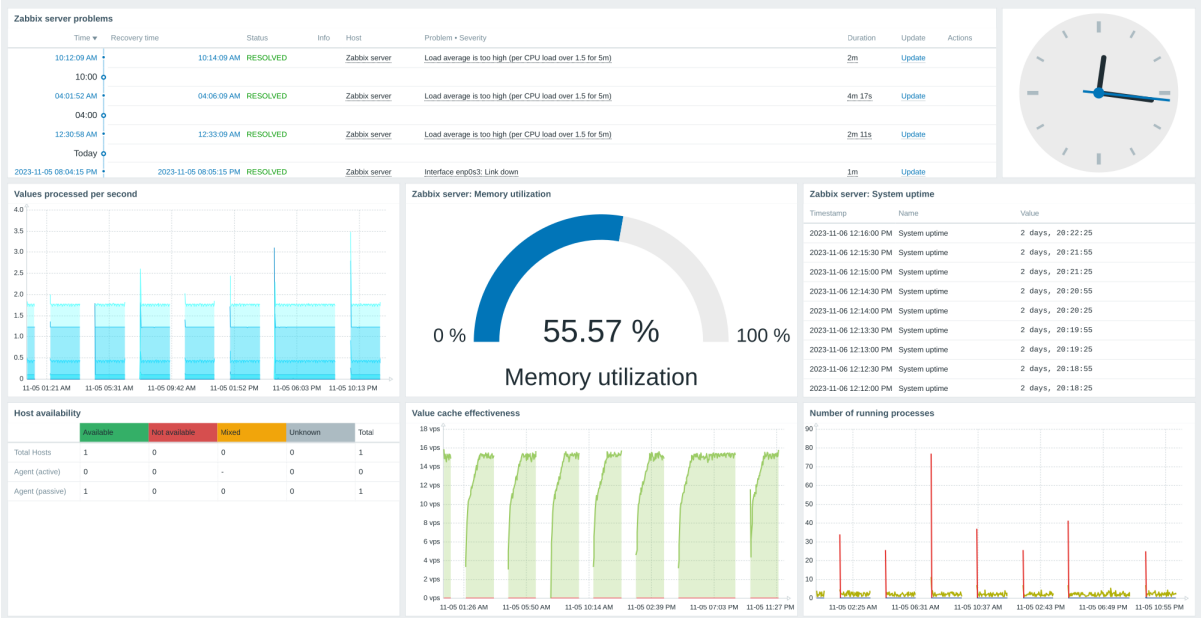
3. Configurare le macro utente come descritto nella sezione *Valori delle macro utente*, se necessario.

Per aggiornare una configurazione esistente per il recupero dei segreti da un CyberArk Vault, vedere *Configurazione di CyberArk*.

14 Rapporti pianificati

Panoramica

Con la funzionalità *Report pianificati*, puoi 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 dei report pianificati - per le istruzioni, vedi 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 del report)
  - *Gestisci report pianificati* nel blocco *Accesso alle azioni* (per creare/modificare i report)

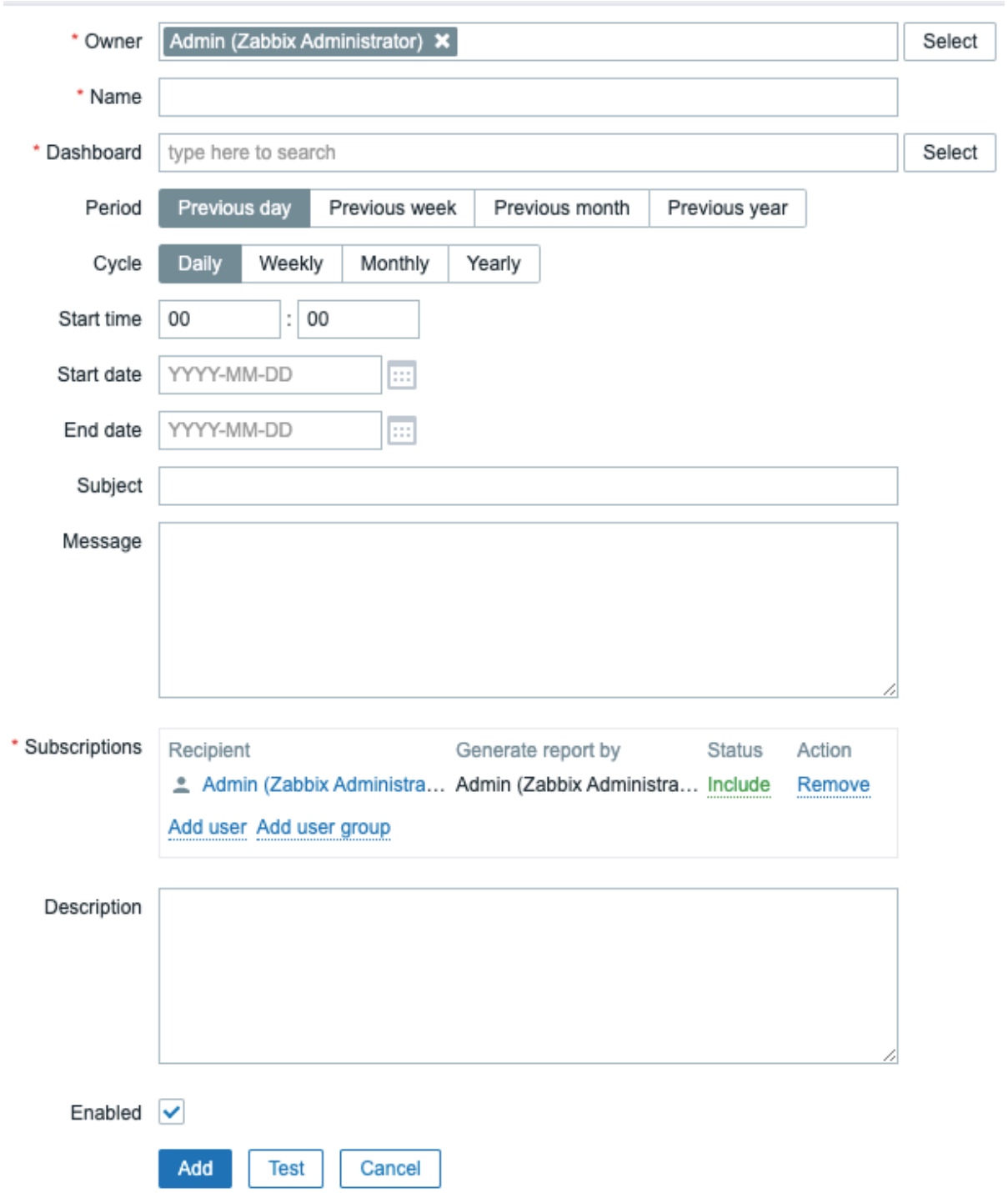
Per creare un report pianificato nel frontend di Zabbix, procedi come segue:

- Vai a: *Report* > *Report pianificati*.
- Fai clic su *Crea report* nell'angolo superiore destro dello schermo.
- Inserisci i parametri del report nel modulo.

Puoi anche creare un report aprendo un report esistente, facendo clic sul pulsante *Clona* e quindi salvandolo con un nome diverso.

Configurazione

La scheda *Report pianificati* contiene gli attributi generali del report.



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

Parametro	Descrizione
Owner	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.
Name	Nome del report; deve essere univoco.
Dashboard	Dashboard su cui 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 selezionare. In alternativa, è possibile fare clic su *Select* accanto al campo e selezionare una dashboard dall'elenco visualizzato.
Period	Periodo di tempo per il quale verrà preparato il report. Selezionare il giorno, la settimana, il mese o l'anno precedente.
Cycle	Frequenza di generazione del report. I report possono essere inviati giornalmente, settimanalmente, mensilmente o annualmente. La modalità "Weekly" consente di selezionare i giorni della settimana in cui il report verrà inviato.
Start time	Ora del giorno nel formato hh:mm in cui il report verrà preparato. Si noti che verrà utilizzato il fuso orario del server Zabbix.

Parametro	Descrizione
Repeat on	Giorni della settimana in cui il report verrà inviato. Questo campo è disponibile solo se *Cycle* è impostato su ”Weekly”.
Start date	Data in cui deve iniziare la generazione regolare del report.
End date	Data in cui deve terminare la generazione regolare del report.
Subject	Oggetto dell’email del report. Macro supportate: {TIME}, {TIMESTAMP}.
Message	Corpo dell’email del report. Macro supportate: {TIME}, {TIMESTAMP}.
Subscriptions	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 *Add user* o *Add user group* per aggiungere altri destinatari. Fare clic sul nome utente per modificare le impostazioni: *Generate report by* - indica se i dati del report devono essere generati in base ai permessi della dashboard dell’utente corrente o del destinatario. *Status* - selezionare ”Include” per inviare il report all’utente oppure ”Exclude” per impedirne l’invio a questo utente. Almeno un utente deve avere lo stato ”Include”. Lo stato ”Exclude” può essere utilizzato per escludere utenti specifici da un gruppo di utenti incluso. Si noti che gli utenti con permessi insufficienti (ossia 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) vedranno ”Inaccessible user” o ”Inaccessible user group” invece dei nomi effettivi nei campi *Recipient* e *Generate report by*; i campi *Status* e *Action* saranno visualizzati come di sola lettura.
Enabled	Stato del report. Deselezionando questa casella di controllo il report verrà disabilitato.
Description	Una 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	Aggiungi 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 se la configurazione del report è corretta inviando un report all’utente corrente.
Delete	Elimina il report.
Cancel	Annulla la modifica delle proprietà del report.

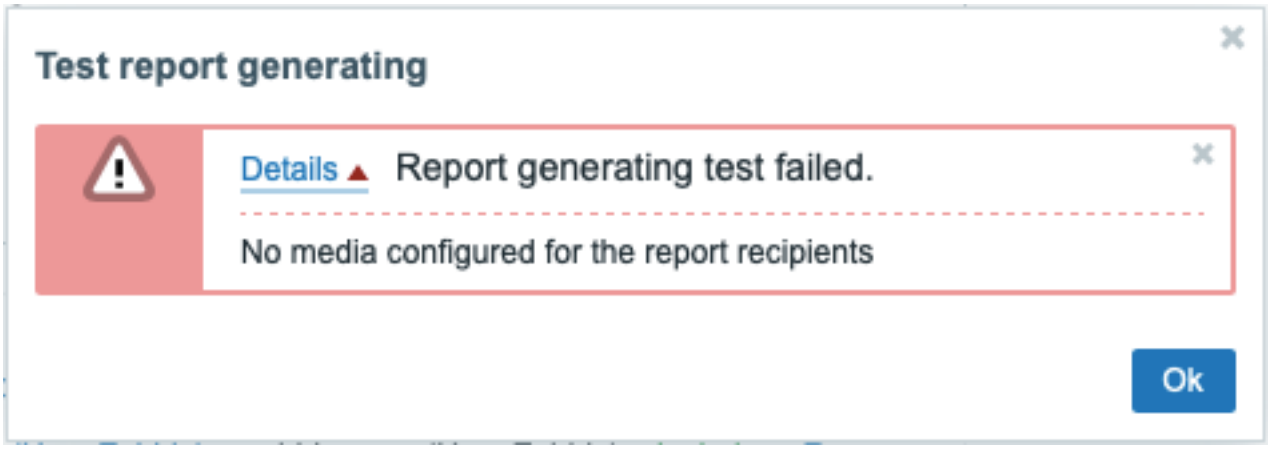
Test

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 azioni della dashboard.

Se la configurazione è corretta, il report di test viene inviato immediatamente all’utente corrente. Per i report di test, gli iscritti e le impostazioni utente *Generate report by* vengono ignorati.

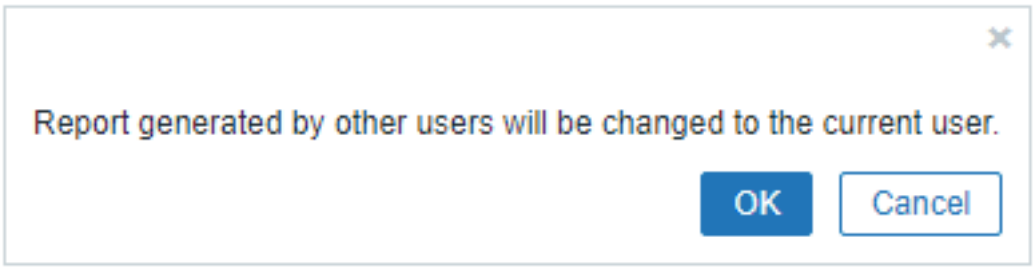
Se la configurazione non è corretta, viene visualizzato un messaggio di errore che descrive la possibile causa.



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*.

15 Esportazione 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)

1 Esporta su file

Panoramica

È possibile configurare l'esportazione in tempo reale degli eventi dei trigger, dei valori degli item e dei trend in formato JSON delimitato da newline.

L'esportazione viene effettuata in file, dove ogni riga del file di esportazione è un oggetto JSON. Le mappature dei valori non vengono applicate.

In caso di errori (se i dati non possono essere scritti nel file di esportazione oppure se il file di esportazione non può essere rinominato o non è possibile crearne uno nuovo dopo la rinomina), l'elemento di dati 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 proprietario. Tutti gli altri permessi sono negati.

Per dettagli precisi su quali informazioni vengono esportate, vedere la pagina del protocollo di esportazione.

Si noti che host/item potrebbero 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 - vedere 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, controlla prima la dimensione del file. Se supera il limite di dimensione configurato, il file viene rinominato aggiungendo .old al suo nome e viene creato un nuovo file con il nome originale.

> **Attention:**
> Verrà creato un file per ciascun processo che scriverà dati (cioè approssimativamente 4-30 file). Poiché la dimensione predefinita per ogni file di esportazione è 1G, mantenere file di esportazione di grandi dimensioni può consumare rapidamente lo spazio su disco.

- `ExportType` consente di specificare quali tipi di entità (eventi, cronologia, trend) verranno esportati.

2 Streaming verso sistemi esterni

Panoramica

È possibile trasmettere in streaming i valori degli item e gli eventi da Zabbix a sistemi esterni tramite HTTP (vedere i dettagli del protocollo).

Il filtro dei tag può essere utilizzato per trasmettere in streaming sottoinsiemi di valori degli item o eventi.

Due processi del server Zabbix sono responsabili dello streaming 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 connettore.

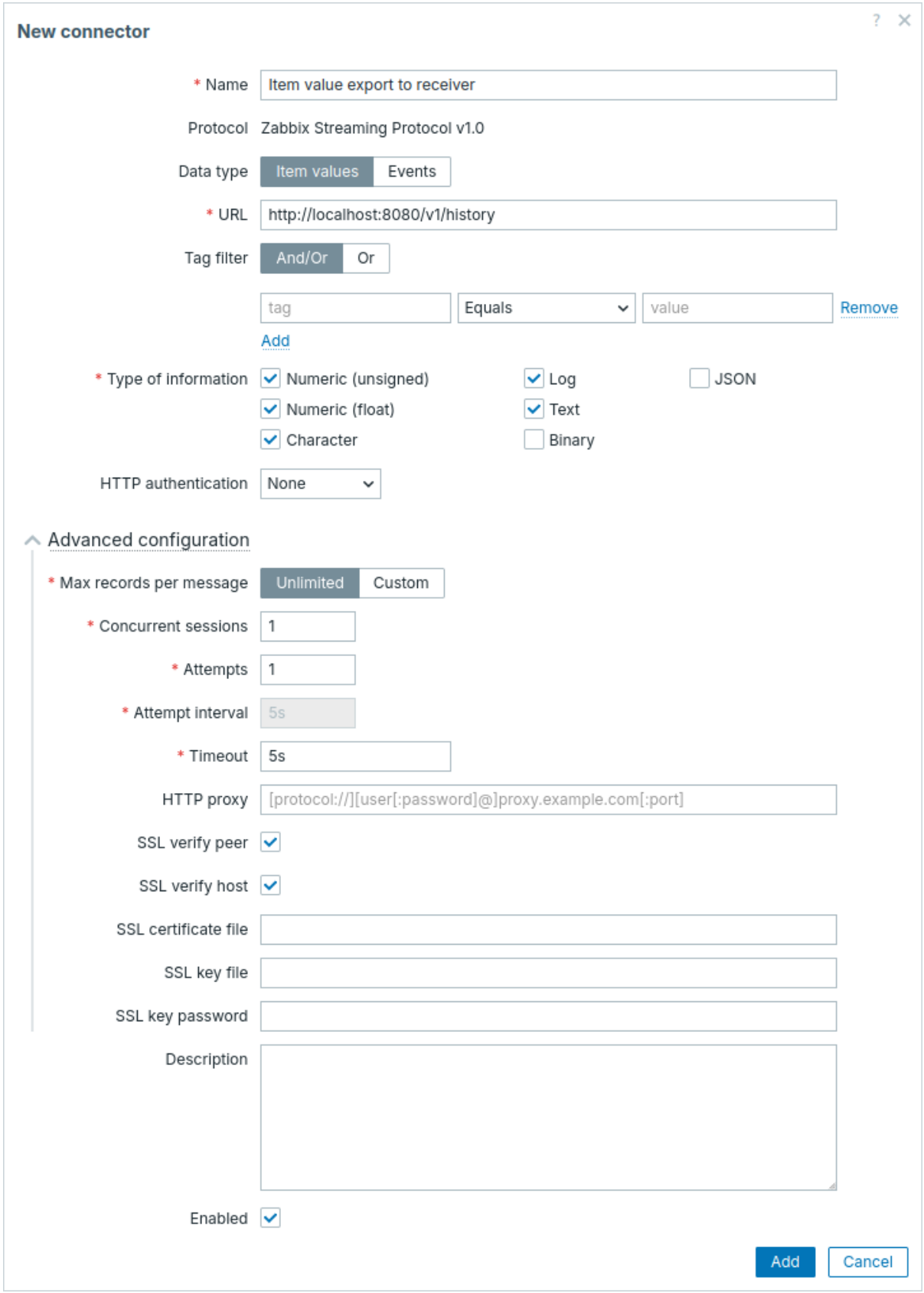
Configurazione

Per configurare lo streaming dei dati verso un sistema esterno sono necessari i seguenti passaggi:

1. Predisporre un sistema remoto per la ricezione dei 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 regolando il parametro `StartConnectors` in `zabbix_server.conf`. Il numero di worker del connector deve corrispondere (o essere superiore, se le sessioni concorrenti sono più di 1) al numero di connector configurati nel frontend di Zabbix. 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.

Parameter	Description
Name	Inserire il nome del connector.
Data type	Selezionare il tipo di dati da trasmettere in streaming: **Item values** - trasmette i valori degli item da Zabbix ai sistemi esterni; **Events** - trasmette gli eventi da Zabbix ai sistemi esterni.

Parameter	Description
URL	Inserire l'URL del receiver. Le macro utente sono supportate.
Tag filter	Esporta solo i valori degli item o gli eventi che corrispondono al filtro tag. Se non impostato, verrà esportato tutto. 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). 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 sola condizione.
Type of information	Selezionare il tipo di informazione (numeric (unsigned), numeric (float), character, ecc.) in base al quale filtrare i valori degli item che il connector deve trasmettere in streaming. 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: Configuring Kerberos with Zabbix); **Digest** - viene utilizzata l'autenticazione Digest; **Bearer** - viene utilizzata l'autenticazione Bearer.
Username	Inserire il nome utente (fino a 255 caratteri). Le macro utente sono supportate. 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). Le macro utente sono supportate. Questo campo è disponibile se *HTTP authentication* è impostato su "Basic", "NTLM", "Kerberos" o "Digest".
Bearer token	Inserire il token Bearer. Le macro utente sono supportate. Questo campo è disponibile ed è obbligatorio se *HTTP authentication* è impostato su "Bearer".
Advanced configuration	Fare clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata (vedere sotto).
Max records per message	Specificare il numero massimo di valori o eventi che possono essere trasmessi in streaming all'interno di un messaggio.
Concurrent sessions	Selezionare il numero di processi mittente da eseguire per questo connector. È possibile specificare fino a 100 sessioni; il valore predefinito è "1".
Attempts	Numero di tentativi per lo streaming 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 in streaming. 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 la connessione non è stata stabilita oppure in cui il codice di risposta HTTP non è 200, 201, 202, 203, 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, 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, predefinito - 5 secondi). I suffissi temporali sono supportati, ad esempio 30s, 1m. Le macro utente sono supportate.

Parameter	Description
HTTP proxy	È possibile specificare un HTTP proxy da utilizzare nel seguente formato: `[protocol://][username[:password]@]proxy.example.com[:port]` Le macro utente sono supportate. Il prefisso facoltativo `protocol://` può essere utilizzato per specificare protocolli proxy alternativi (il supporto del prefisso del protocollo è stato aggiunto in cURL 7.21.7). Se non viene specificato alcun protocollo, il proxy verrà trattato come un HTTP proxy. Per impostazione predefinita, verrà utilizzata la porta 1080. Se *HTTP proxy* è specificato, il proxy sovrascriverà le variabili di ambiente relative al proxy come `http_proxy`, `HTTPS_PROXY`. Se non è specificato, il proxy non sovrascriverà le variabili di ambiente relative al proxy. Il valore inserito viene passato così com'è, senza alcun controllo di validità. È anche possibile inserire un indirizzo proxy SOCKS. Se si specifica il protocollo errato, la connessione non riuscirà e l'item diventerà unsupported. Si noti che con HTTP proxy è supportata solo l'autenticazione semplice.
SSL verify peer	Selezionare la casella per verificare il certificato SSL del web server. Il certificato del server verrà prelevato automaticamente dalla posizione dell'autorità di certificazione (CA) a livello di sistema. È possibile sovrascrivere la posizione dei file CA utilizzando il parametro di configurazione del server o proxy Zabbix `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. Questo 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]. Le macro utente sono supportate. Se il file del certificato contiene anche la chiave privata, lasciare vuoto il campo *SSL key file*. Se la chiave è cifrata, specificare la password nel campo *SSL key password*. La directory che contiene questo file è specificata dal parametro di configurazione del server o proxy Zabbix `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]. Le macro utente sono supportate. La directory che contiene questo file è specificata dal parametro di configurazione del server o proxy Zabbix `SSLKeyLocation`.
SSL key password	Password del file della chiave privata SSL. Le macro utente sono supportate.
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 bootstrap broker (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 relativi metadati del cluster. Se l'elenco contiene indirizzi di cluster Kafka diversi, verrà utilizzato solo il cluster che risponde più rapidamente e gli altri indirizzi verranno registrati come non disponibili; di conseguenza, all'avvio potrebbero comparire avvisi 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 IP possono risolversi in indirizzi di loopback (ad esempio `127.0.0.1/localhost`) oppure essere normalizzati dal client, il che può rendere tali avvisi fuorvianti. 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 gli `advertised.listeners` dei broker, e preferire indirizzi che si risolvono nell'indirizzo pubblicizzato dal broker.

Protocollo

La comunicazione tra il server e il receiver avviene tramite HTTP utilizzando REST API, NDJSON, "Content-Type: application/x-ndjson".

Per maggiori dettagli, vedere Protocollo di esportazione JSON delimitato da newline.

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; per le richieste non riuscite, qualsiasi altro codice.

Esempio di successo:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Esempio con errori:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 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`)

6 Monitoraggio dei servizi

Panoramica Il monitoraggio dei servizi è un monitoraggio a livello business che può essere utilizzato per ottenere una panoramica dell'intero albero dei servizi dell'infrastruttura IT, identificare i punti deboli dell'infrastruttura, calcolare gli SLA di vari servizi IT e consultare 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, l'elevato carico del processore, ecc. 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 apparire così:

```
Servizio
|
|-Postazioni di lavoro
| |
| |-Postazione di lavoro1
| |
| |-Postazione di lavoro2
|
|-Server
```

Ogni nodo della struttura ha l'attributo status. Lo status viene calcolato e propagato ai livelli superiori in base all'algoritmo selezionato. Lo status dei singoli nodi è influenzato dallo status dei problemi mappati. La mappatura dei problemi viene eseguita con il tagging.

Zabbix può inviare notifiche o eseguire automaticamente uno script sullo Zabbix server nel caso venga rilevato un cambiamento dello stato del servizio. È possibile definire regole flessibili per stabilire se un servizio padre debba passare a uno 'stato di problema' in base agli status dei servizi figli. I dati relativi ai problemi dei servizi possono quindi essere utilizzati per calcolare gli SLA e inviare report SLA in base a un insieme flessibile di condizioni.

Il monitoraggio dei servizi è configurato nel menu Servizi, che consiste nelle seguenti sezioni:

- Servizi

La sezione Servizi consente di costruire una gerarchia della propria infrastruttura monitorata aggiungendo servizi padre e quindi servizi figli ai servizi padre.

Oltre alla configurazione dell'albero dei servizi, questa sezione fornisce una panoramica dell'intera infrastruttura e consente di identificare rapidamente i problemi che hanno portato a un cambiamento dello stato del servizio.

- SLA

In questa sezione è possibile definire accordi sul livello di servizio e impostare obiettivi di livello di servizio per servizi specifici.

- Report SLA

In questa sezione è possibile visualizzare i report SLA.

Azioni del servizio

È inoltre possibile configurare le azioni del servizio.

Le azioni del servizio sono facoltative e consentono di:

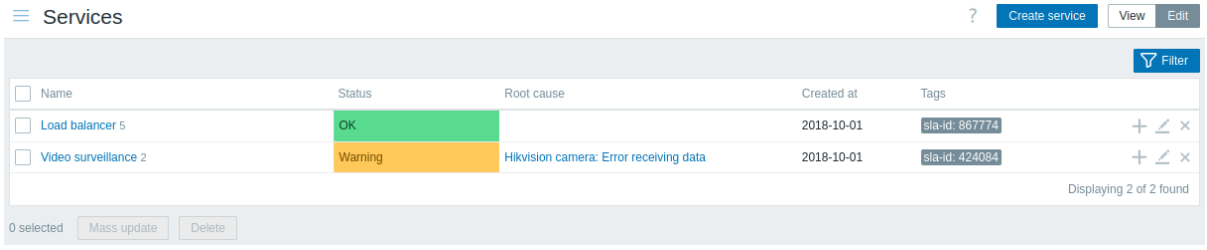
- inviare una notifica che segnala che un servizio non è disponibile
- eseguire un comando remoto sullo Zabbix server in seguito a un cambiamento dello stato del servizio
- inviare una notifica di ripristino quando un servizio torna disponibile.

Vedi anche:

- Esempio di configurazione del monitoraggio SLA
- Note sull'aggiornamento dei servizi da versioni di Zabbix precedenti alla 6.0

1 Albero dei servizi

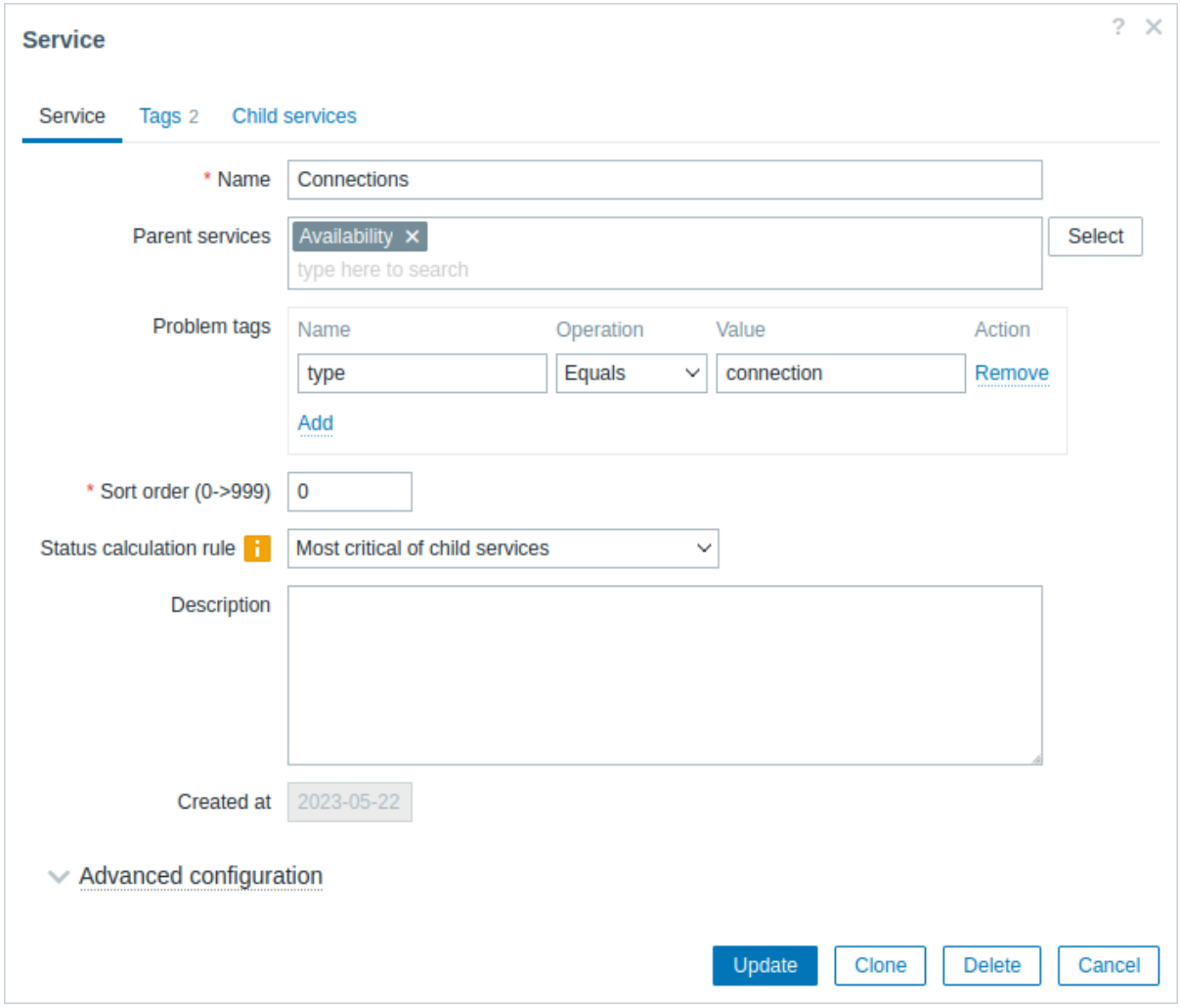
L'albero dei servizi viene configurato nella sezione di menu *Services -> Services*. Nell'angolo in alto a destra, passa dalla visualizzazione alla modalità di modifica.



Per configurare un nuovo servizio, fai clic sul pulsante *Crea servizio* nell'angolo in alto a destra.

Per aggiungere rapidamente un servizio figlio, in alternativa puoi premere l'icona più accanto al servizio padre. Si aprirà lo stesso modulo di configurazione del servizio, ma il parametro *Parent services* sarà già precompilato.

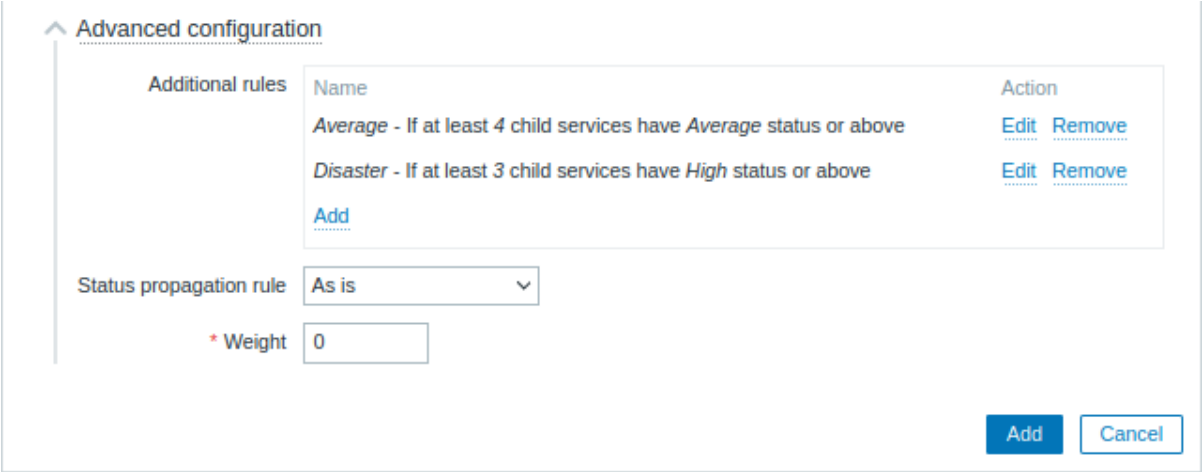
Configurazione del servizio Nella scheda **Service**, specificare i parametri richiesti del servizio:



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

Parametro	Descrizione
Name	Nome del servizio.
Parent services	Servizi padre a cui il servizio appartiene. Lasciare questo campo vuoto se si sta aggiungendo il servizio di livello più alto. Un servizio può avere più servizi padre. In questo caso, verrà visualizzato nell'albero dei servizi sotto ciascuno dei servizi padre.
Problem tags	Specificare i tag per mappare i dati dei problemi al servizio. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Per ogni condizione sono disponibili due operatori: **Equals** - include i nomi e i valori dei tag specificati (distinzione tra maiuscole e minuscole) **Contains** - include i nomi dei tag in cui i valori dei tag contengono la stringa immessa (corrispondenza per sottostringa, distinzione tra maiuscole e minuscole)
Sort order	Ordine di ordinamento per la visualizzazione; il valore più basso viene mostrato per primo.
Status calculation rule	Regola per il calcolo dello stato del servizio: **Most critical if all children have problems** - il problema più critico nei servizi figli viene utilizzato per colorare lo stato del servizio, se tutti i servizi figli hanno problemi **Most critical of child services** - il problema più critico nei servizi figli viene utilizzato per colorare lo stato del servizio **Set status to OK** - non calcolare lo stato del servizio Ulteriori regole di calcolo dello stato possono essere configurate nelle opzioni di advanced configuration.
Description	Descrizione del servizio.
Created at	L'ora in cui il servizio è stato creato; visualizzata durante la modifica di un servizio esistente.
Advanced configuration	Fare clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di advanced configuration.

Configurazione avanzata



Parametro	Descrizione
Regole aggiuntive	Fare clic su *Aggiungi* per configurare regole aggiuntive per il calcolo dello stato.
Imposta stato a	Impostare lo stato del servizio su *OK* (predefinito), *Non classificato*, *Informazione*, *Attenzione*, *Medio*, *Alto* o *Disastro* in caso di corrispondenza della condizione.
Condizione	Selezionare la condizione per i servizi figli diretti: **se almeno (N) servizi figli hanno stato (Status) o superiore** **se almeno il (N%) dei servizi figli ha stato (Status) o superiore** **se meno di (N) servizi figli hanno stato (Status) o inferiore** **se meno del (N%) dei servizi figli ha stato (Status) o inferiore** **se il peso dei servizi figli con stato (Status) o superiore è almeno (W)** **se il peso dei servizi figli con stato (Status) o superiore è almeno il (N%)** **se il peso dei servizi figli con stato (Status) o inferiore è minore di (W)** **se il peso dei servizi figli con stato (Status) o inferiore è minore del (N%)** Se vengono specificate più condizioni e la situazione corrisponde a più di una condizione, verrà impostata la severità più alta.
N (W)	Impostare il valore di N o W (1-100000), oppure N% (1-100) nella condizione.
Status	Selezionare il valore di *Status* nella condizione: *OK* (predefinito), *Non classificato*, *Informazione*, *Attenzione*, *Medio*, *Alto* o *Disastro*.
Regola di propagazione dello stato	Regola per propagare lo stato del servizio al servizio padre: **Così com'è** - lo stato viene propagato senza modifiche **Aumenta di** - è possibile aumentare lo stato propagato da 1 a 5 livelli di severità **Diminuisci di** - è possibile diminuire lo stato propagato da 1 a 5 livelli di severità **Ignora questo servizio** - lo stato non viene propagato affatto al servizio padre **Stato fisso** - lo stato viene propagato staticamente, cioè sempre come lo stesso stato
Peso	Peso del servizio (intero nell'intervallo da 0 (predefinito) a 1000000).

Note:
Le regole aggiuntive per il calcolo dello stato possono essere utilizzate solo per aumentare il livello di severità rispetto al livello calcolato in base al parametro principale *Regola di calcolo dello stato*. Se in base alle regole aggiuntive lo stato dovrebbe essere Attenzione, ma in base alla *Regola di calcolo dello stato* lo stato è Disastro, il servizio avrà stato Disastro.

La scheda **Tag** contiene tag a livello di servizio. I tag a livello di servizio lo sono utilizzato per identificare un servizio. I tag di questo tipo non vengono utilizzati per mappare problemi al servizio (per questo, usa *Tag problema* dalla prima scheda).

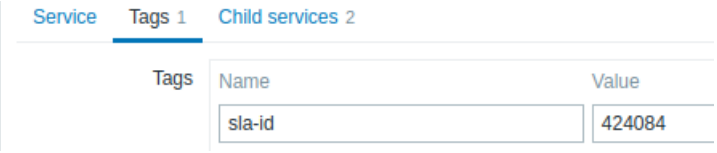
La scheda **Servizi secondari** consente di specificare i servizi dipendenti. Fare clic su *Aggiungi* per aggiungere un servizio dall'elenco dei servizi esistenti. Se desideri aggiungere un nuovo servizio figlio, salva prima questo servizio, quindi fai clic sull'icona più accanto al servizio che hai 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 utilizzati per associare i servizi alle azioni dei servizi e agli SLA. Questi tag vengono specificati nella scheda di configurazione del servizio *Tag*.
Per la mappatura degli SLA viene utilizzata la logica *OR*: un servizio verrà associato a uno SLA se possiede almeno un tag corrispondente. Nelle azioni dei servizi, le regole di mappatura sono configurabili e possono utilizzare la logica *AND*, *OR* oppure *AND/OR*.

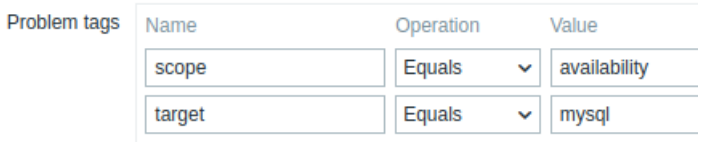


Tag dei problemi

I tag dei problemi vengono utilizzati per associare problemi e servizi. Questi tag vengono specificati nella scheda di configurazione del servizio principale.

Solo i servizi figli del livello più basso della gerarchia possono avere tag dei problemi definiti ed essere correlati direttamente ai problemi. Se i tag dei problemi corrispondono, lo stato del servizio cambierà assumendo lo stesso stato del problema. In caso di più problemi, un servizio assumerà lo stato di quello più grave. Lo stato di un servizio padre viene quindi calcolato in base agli stati dei servizi figli secondo le regole di calcolo dello stato.

Se vengono 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. Ognuno 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 lo stato *Warning*

Il servizio **Web camera 3** ha specificato un solo tag problema: `name:webcam-3`



Lo stato del servizio cambierà da *OK* a *Warning* quando questo problema viene rilevato.

Se il servizio **Web camera 3** avesse i tag problema `name:webcam-3` e `floor:2`, il suo stato non cambierebbe 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, premere l'icona della matita accanto al servizio.

Per clonare un servizio esistente, premere l'icona della matita per aprirne la configurazione e quindi premere il pulsante Clone. Quando un servizio viene clonato, i suoi collegamenti padre vengono mantenuti, mentre i collegamenti figlio no.

Per eliminare un servizio, premere l'icona `x` accanto ad esso. Quando si elimina un servizio padre, i suoi servizi figlio non verranno eliminati e si sposteranno di un livello più in alto nell'albero dei servizi (i figli di 1° livello avranno lo stesso livello del servizio padre eliminato).

I due pulsanti sotto l'elenco dei servizi offrono alcune opzioni di modifica di massa:

- *Mass update* - aggiornamento di massa delle proprietà del servizio
- *Delete* - elimina i servizi

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi servizi, quindi fare clic sul pulsante richiesto.

2 SLA

Panoramica Una volta creati i servizi, puoi iniziare a monitorare se le loro prestazioni sono in linea con il contratto di servizio (SLA).

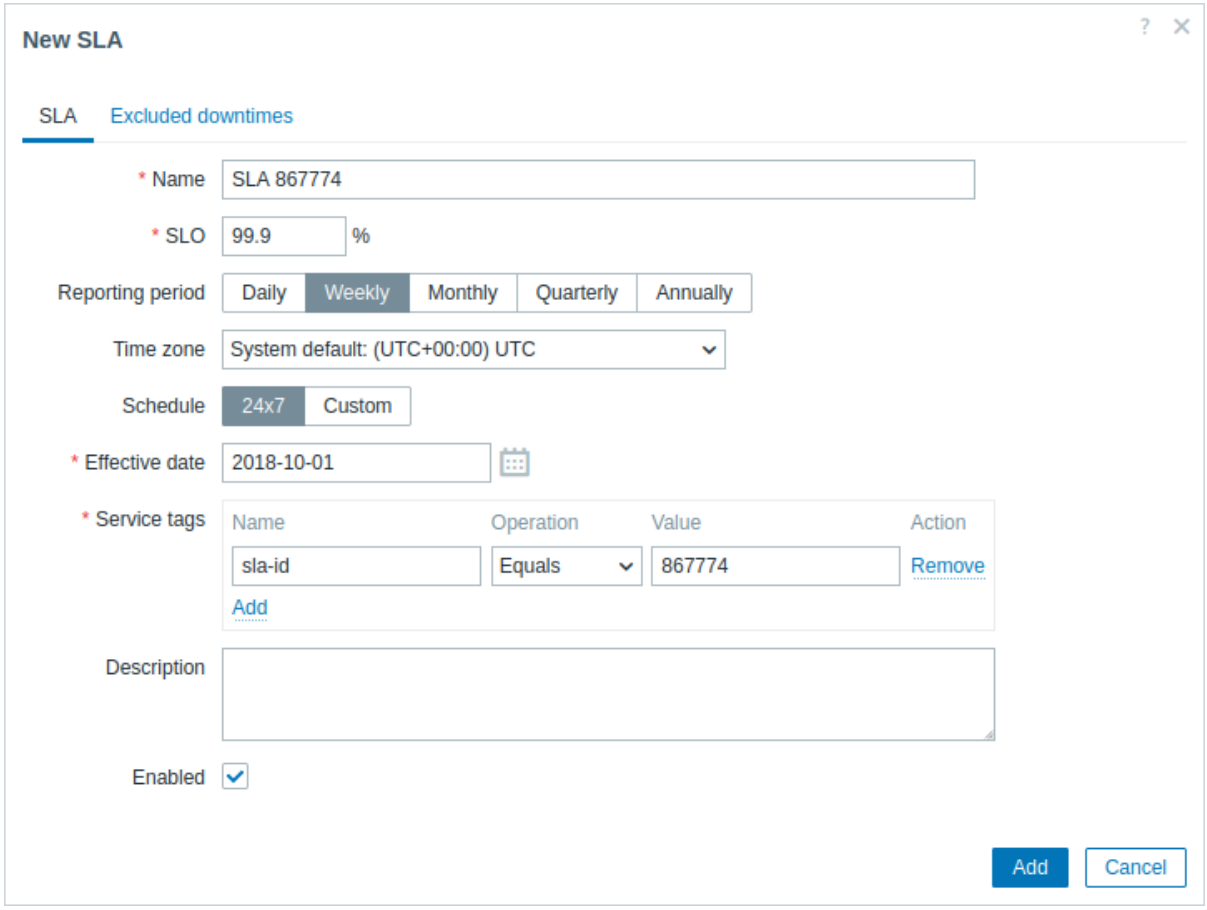
La sezione del menu *Servizi->SLA* consente di configurare gli SLA per vari servizi. Uno SLA in Zabbix definisce l'obiettivo del livello di servizio (SLO), il programma di uptime previsto e i tempi di inattività pianificati.

SLA e servizi sono abbinati a tag di servizio. 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 per ciascuno SLA verranno visualizzati separatamente.

Nei report SLA Zabbix fornisce dati sull'indicatore del livello di servizio (SLI), che misurano la reale disponibilità del servizio. Se un servizio soddisfa gli obiettivi SLA viene determinato confrontando SLO (disponibilità prevista in %) con SLI (disponibilità reale in %).

Configurazione Per creare un nuovo SLA, fare clic sul pulsante *Crea SLA*.

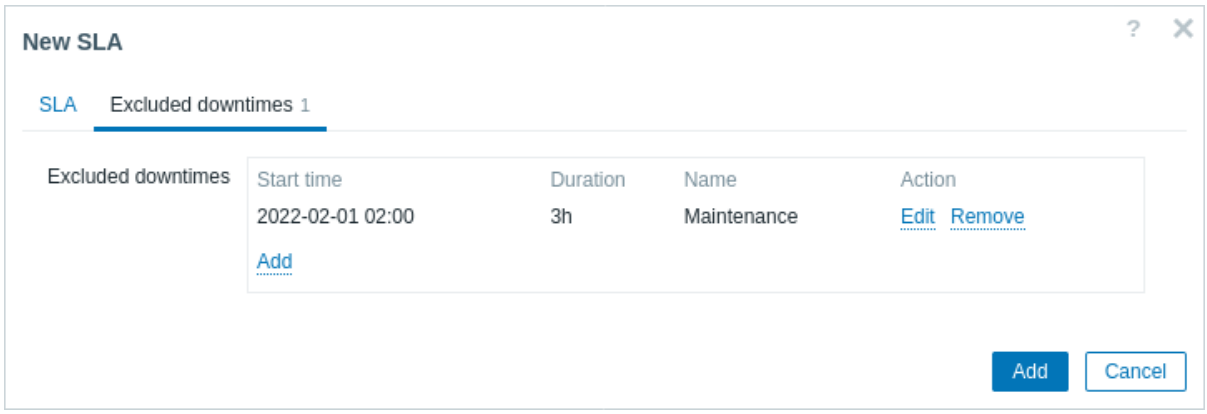
La scheda **SLA** consente di specificare i parametri generali dello SLA.



Parametro	Descrizione
Nome	Inserire il nome dello SLA.
SLO	Inserire l'obiettivo di livello di servizio (SLO) come percentuale.
Periodo di reporting	Selezionare il periodo di reporting dello SLI utilizzato per calcolare i dati SLI nel report SLA: **Giornaliero** - ogni giorno, dalle 00:00:00 alle 23:59:59. **Settimanale** - ogni settimana, dalla domenica 00:00:00 al 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 00:00:00 al 31 dicembre 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.

Parametro	Descrizione
Tag del servizio	Aggiungere tag del servizio per identificare i servizi ai quali deve essere applicato questo SLA. **Nome** - nome del tag del servizio, deve corrispondere esattamente, con distinzione tra maiuscole e minuscole. **Operazione** - selezionare *Uguale* 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 **Tempi di inattività esclusi** consente di specificare i tempi di inattività esclusi dal calcolo dello SLA.



Fare clic su *Aggiungi* per configurare i tempi di inattività esclusi, quindi inserire il nome del periodo, la data di inizio e la durata.

Report SLA Le prestazioni di un servizio rispetto a un SLA sono visibili nel report SLA. I report SLA possono essere visualizzati:

- dalla sezione *SLA* cliccando sul collegamento ipertestuale del report SLA;
- dalla sezione *Servizi* cliccando sul nome dello SLA nella scheda informazioni;
- nel widget *report SLA* della dashboard.

Una volta configurato uno SLA, la scheda *Informazioni* nella sezione servizi mostrerà anche alcune informazioni sulle prestazioni del servizio.

3 Esempio di configurazione

Panoramica Questa sezione descrive una semplice configurazione per il monitoraggio del cluster ad alta affidabilità Zabbix come servizio.

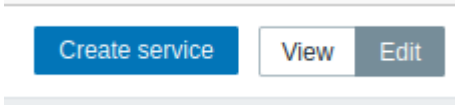
Prerequisiti Prima di configurare il monitoraggio dei servizi, è necessario che gli host siano configurati:

- *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`

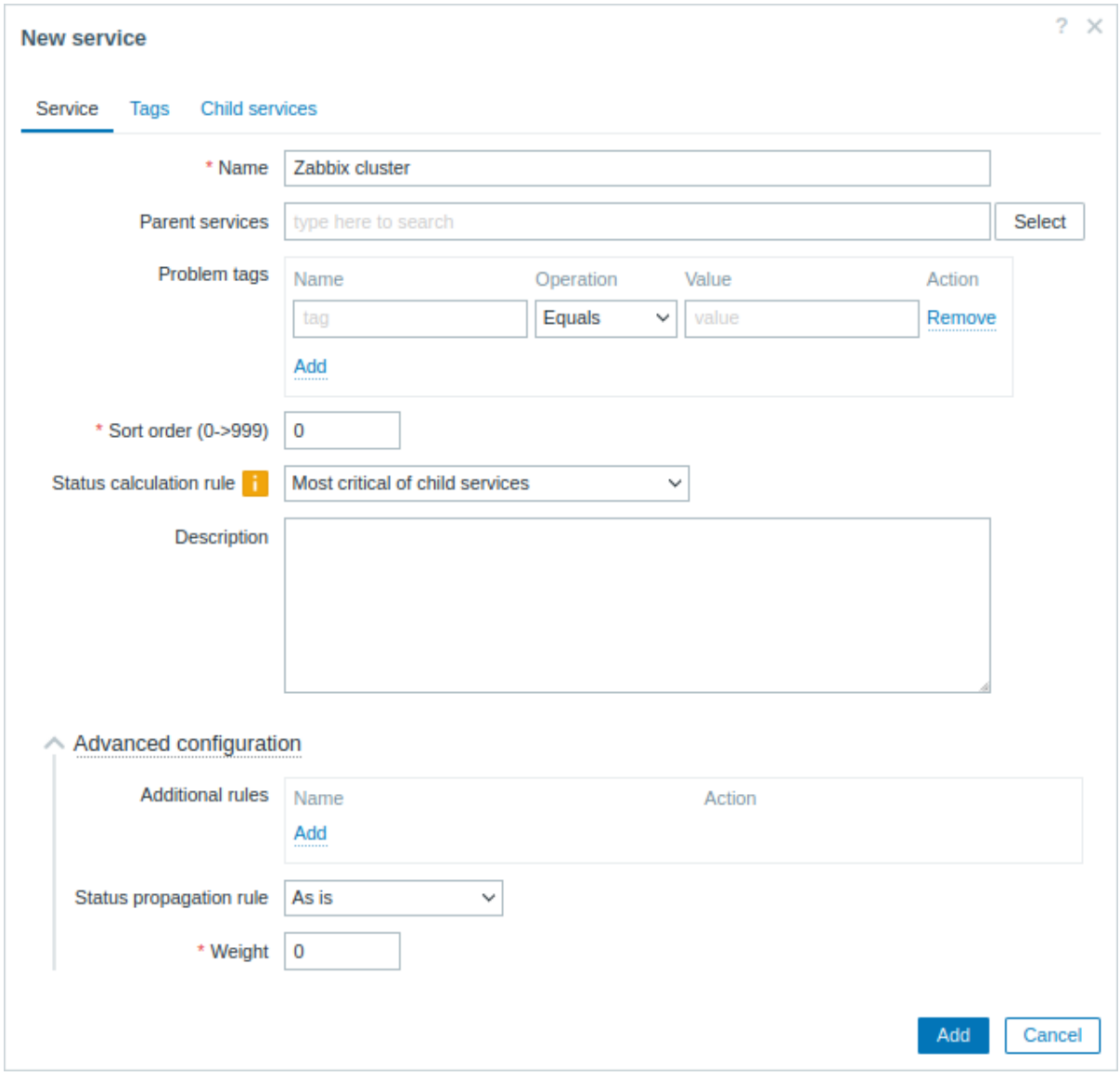
Albero dei servizi Il passaggio successivo consiste nel creare l'albero dei servizi. In questo esempio, l'infrastruttura è molto semplice ed è composta da tre servizi: *Zabbix cluster* (padre) e due servizi figli *Zabbix server node 1* e *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

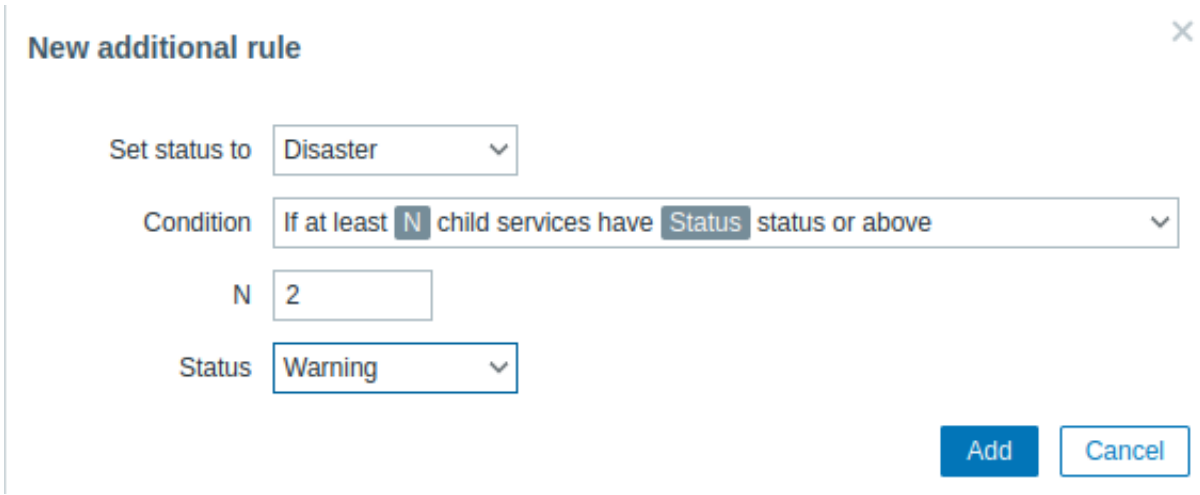
Nella pagina Servizi, attivare la modalità *Edit* e premere Create service:



Nella finestra di configurazione del servizio, inserire il nome *Zabbix cluster* e fare clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata.



Configurare una regola aggiuntiva:

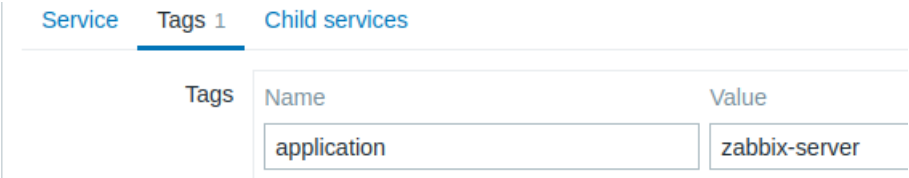


Zabbix cluster avrà due servizi figli, uno per ciascuno dei nodi HA. Se entrambi i nodi HA hanno problemi con stato almeno *Warning*, lo stato del servizio padre deve essere impostato su *Disaster*. Per ottenere questo risultato, la regola aggiuntiva deve essere configurata come segue:

- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2

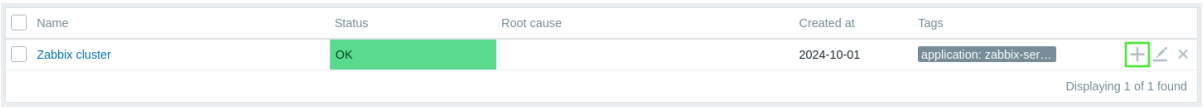
- Status: Warning

Passare alla scheda *Tags* e aggiungere il tag `application:zabbix-server`. Questo tag verrà utilizzato in seguito per le azioni del servizio e i report SLA.



Salvare il nuovo servizio.

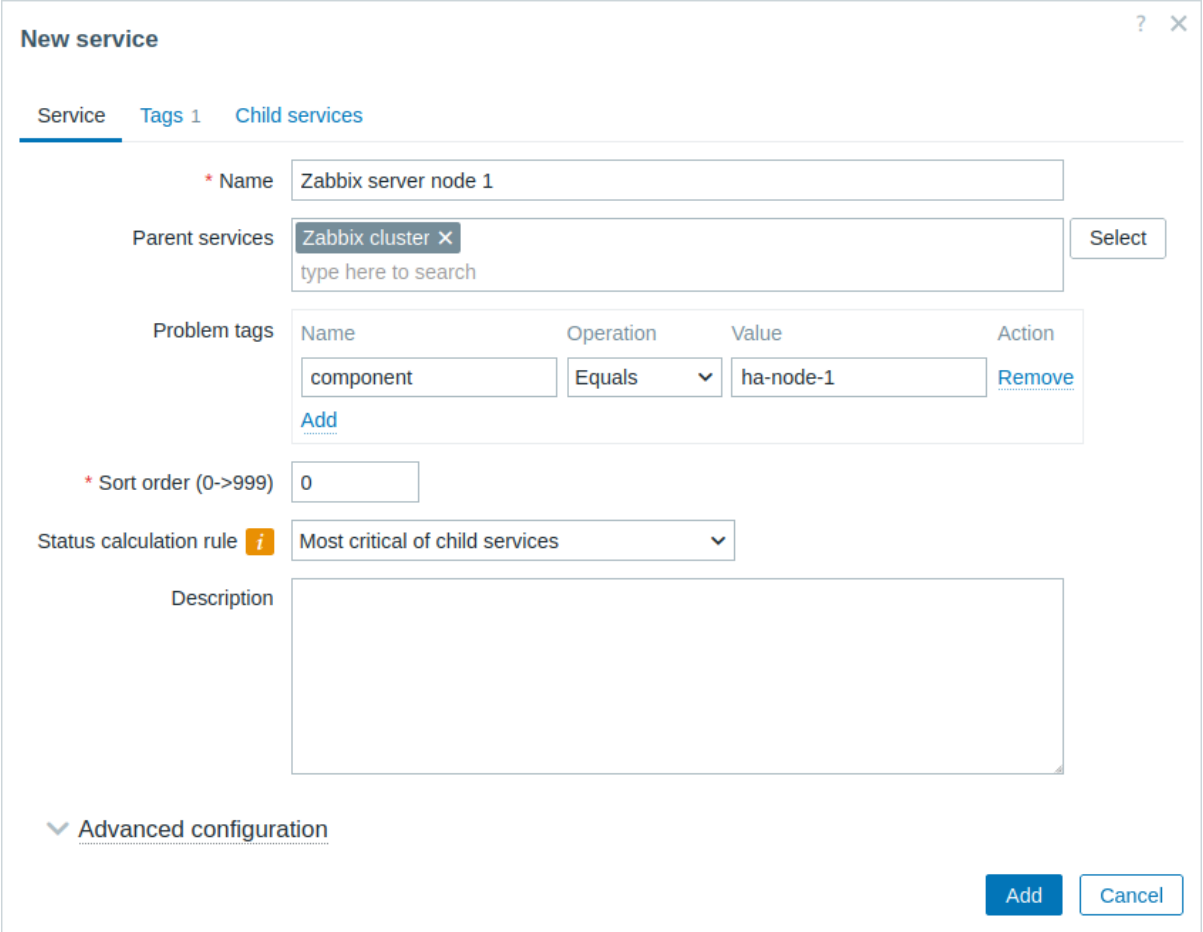
Per aggiungere un servizio figlio, premere sull'icona più accanto al servizio *Zabbix cluster* (l'icona è visibile solo in modalità *Modifica*).



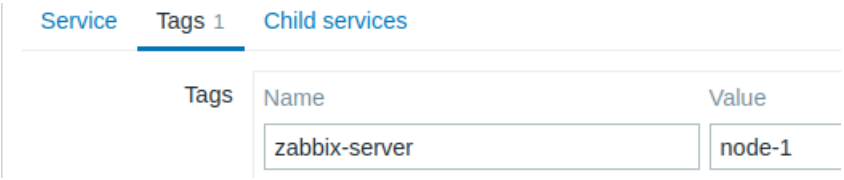
Nella finestra di configurazione del servizio, inserire il nome *Zabbix server node 1*. Si noti che il parametro Parent services è già precompilato con *Zabbix cluster*.

La disponibilità di questo servizio è influenzata dai problemi sul host *HA node 1*, contrassegnati con il tag problema `component:ha-node-1`. Nel parametro Problem tags, inserire:

- Nome: component
- Operazione: Uguale
- Valore: ha-node-1



Passare alla scheda *Tags* e aggiungere un tag di servizio: `zabbix-server:node-1`. Questo tag verrà utilizzato in seguito per le azioni del servizio e i report SLA.



Salvare 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.

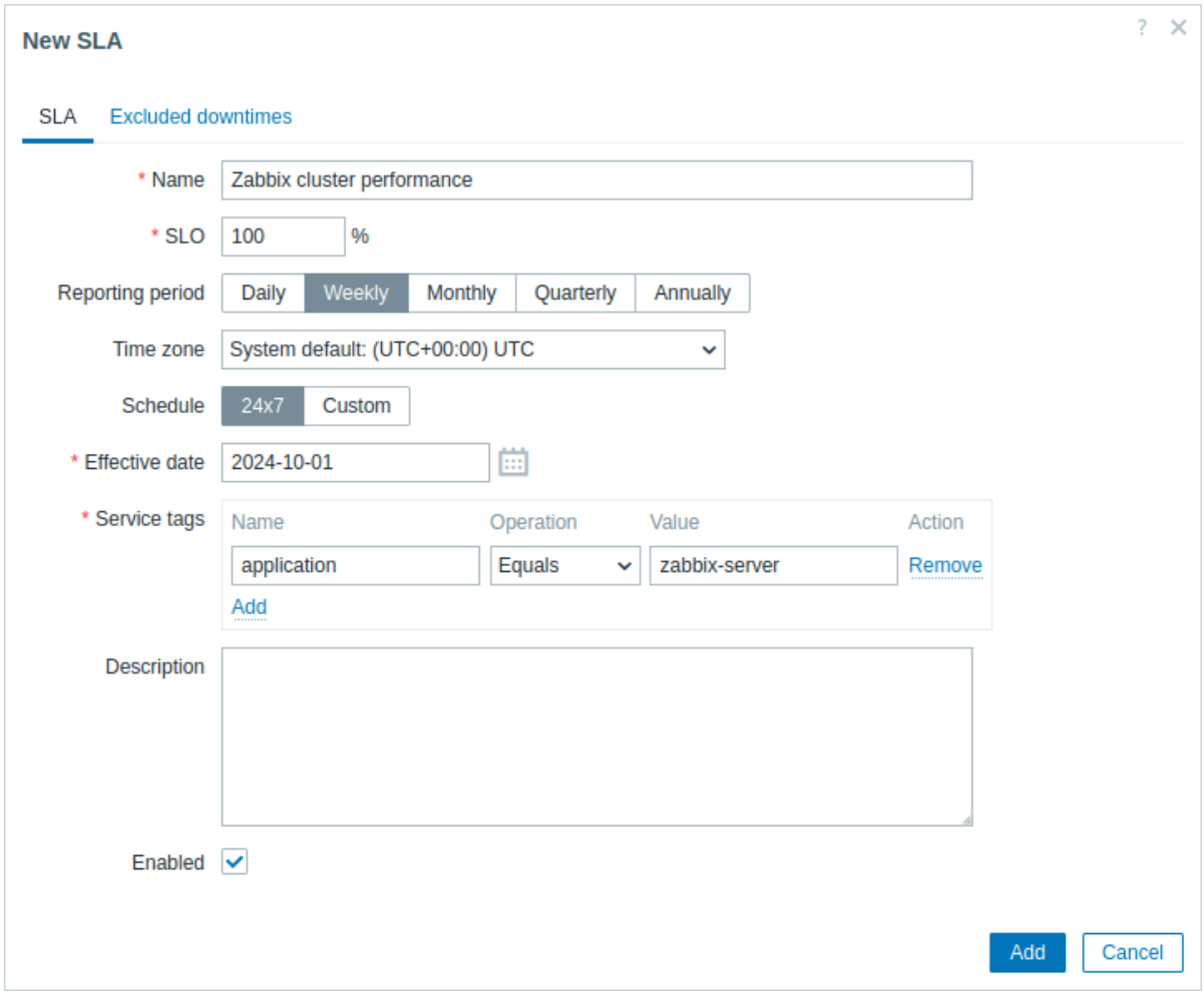
SLA In questo esempio, le prestazioni previste del cluster Zabbix sono del 100%, escludendo il periodo di manutenzione semestrale di un’ora.

Per prima cosa, è necessario aggiungere un nuovo accordo sul livello di servizio.

Andare alla sezione di menu *Services->SLA* e premere Create SLA. Inserire il nome *Zabbix cluster performance* e impostare lo SLO al 100%.

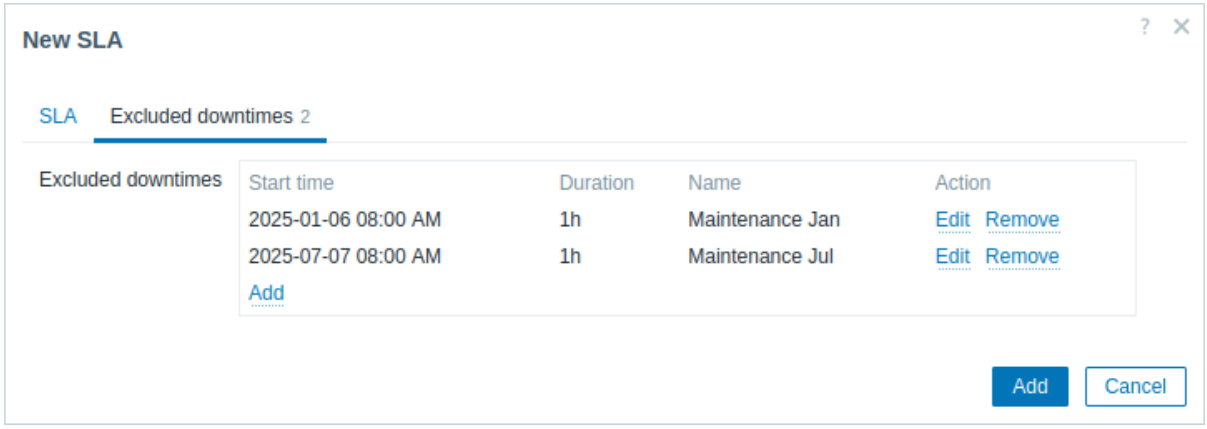
Il servizio Zabbix cluster ha un tag di servizio `application:zabbix-server`. Per utilizzare questo SLA per misurare le prestazioni del cluster Zabbix, nel parametro *Service tags* specificare:

- Nome: application
- Operazione: Equals
- Valore: zabbix-server



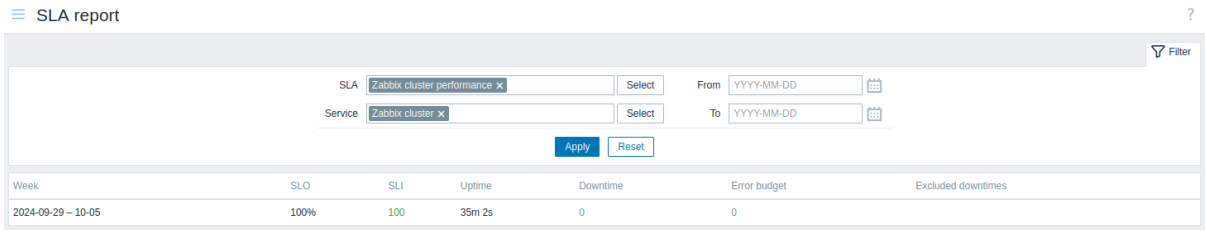
In una configurazione reale, è anche possibile aggiornare il periodo di report desiderato, il fuso orario e la data di inizio oppure modificare la pianificazione da 24/7 a personalizzata. Per questo esempio, le impostazioni predefinite sono sufficienti.

Passare alla scheda *Excluded downtimes* e aggiungere i tempi di inattività per i periodi di manutenzione pianificata, in modo da escludere questi periodi dal calcolo dello SLA. Nella sezione Excluded downtimes premere il collegamento Add, inserire il nome del periodo di inattività, l'ora di inizio pianificata e la durata.

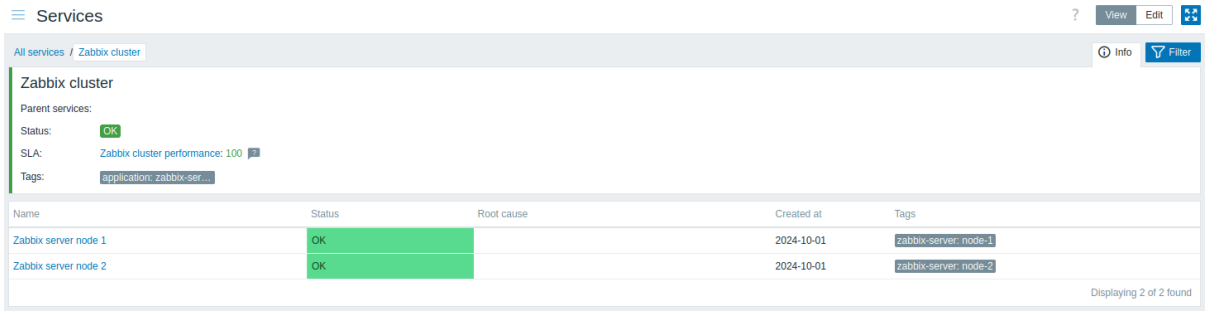


Premere Add per salvare il nuovo SLA.

Passare alla sezione dei report SLA per visualizzare il report SLA per Zabbix cluster.



Le informazioni sullo SLA possono essere controllate anche nella sezione *Services*.



7 Monitoraggio web

Panoramica Con Zabbix è possibile verificare diversi aspetti della disponibilità dei siti web.

> **Attention:**
> Per eseguire il monitoraggio web, Zabbix server deve essere inizialmente configurato con il supporto cURL (libcurl).

Per attivare il monitoraggio web è necessario definire degli scenari web. Uno scenario web consiste in una o più richieste HTTP o "passi". I passi vengono eseguiti periodicamente da Zabbix server in un ordine predefinito. Se un host è monitorato da un proxy, i passi vengono eseguiti dal proxy.

Gli scenari web sono collegati a host/template nello stesso modo di item, trigger, ecc. Ciò significa che gli scenari web possono essere creati anche a livello di template e quindi 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 i passi dell'intero scenario

- numero del passo che non è riuscito
- ultimo messaggio di errore

Le seguenti informazioni vengono raccolte in qualsiasi passo di uno scenario web:

- velocità di download al secondo
- tempo di risposta
- codice di risposta

Per maggiori dettagli, vedere item del monitoraggio web.

I dati raccolti dall'esecuzione degli scenari web vengono conservati nel database. I dati vengono utilizzati automaticamente per grafici, trigger e notifiche.

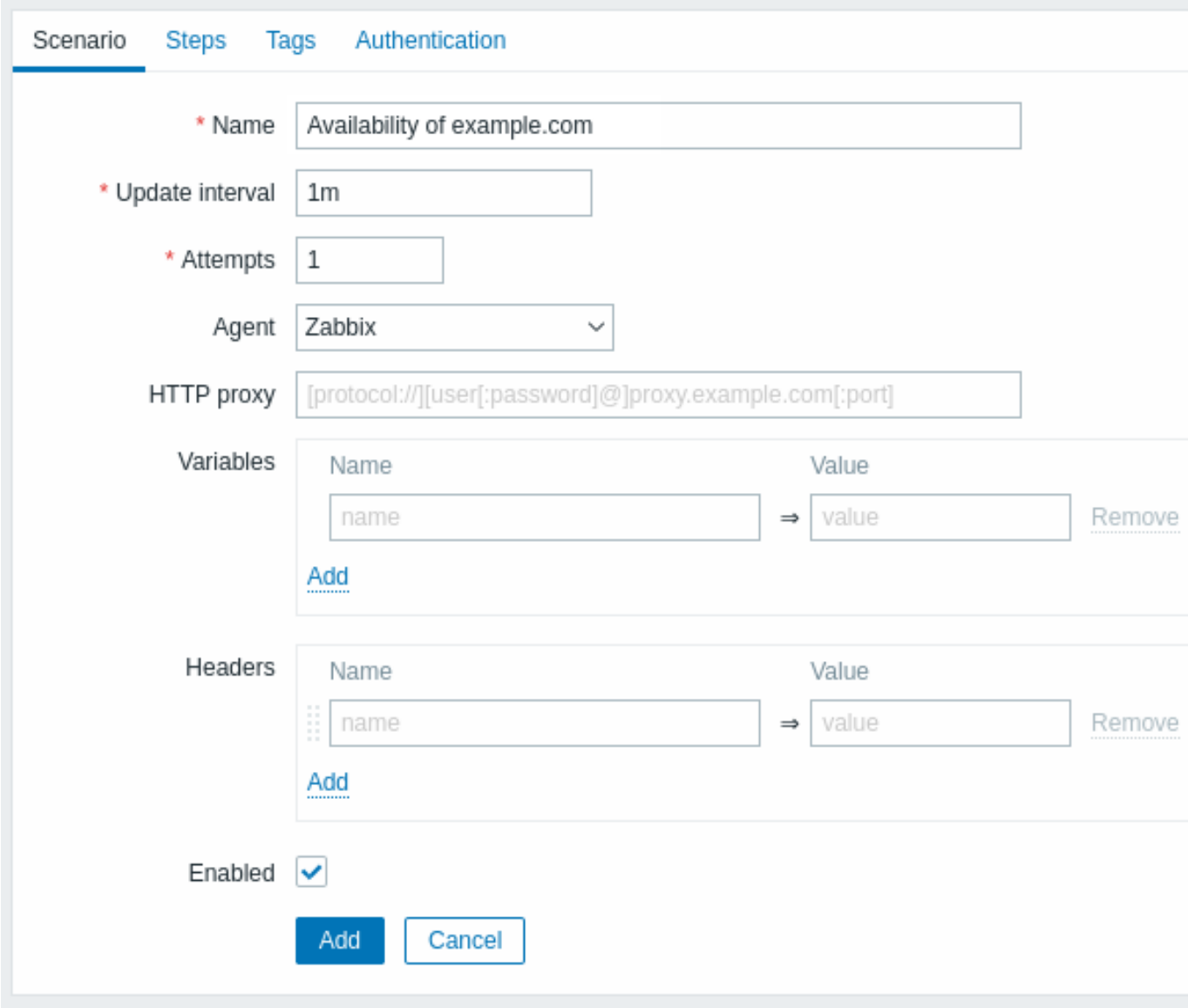
Zabbix può anche verificare se una pagina HTML recuperata contiene una stringa predefinita. Può eseguire un accesso simulato e seguire un percorso di clic del mouse simulati sulla pagina.

Il monitoraggio web di Zabbix supporta sia HTTP che HTTPS. Durante l'esecuzione di uno scenario web, Zabbix seguirà facoltativamente i reindirizzamenti (vedere l'opzione *Follow redirects* sotto). Il numero massimo di reindirizzamenti è fissato nel codice a 10 (utilizzando 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:

- Vai a: *Raccolta dati* → *Host* (oppure *Template*)
- Fai clic su *Web* nella riga dell'host/template
- Fai clic su *Crea scenario web* a destra (oppure sul nome dello scenario per modificare uno scenario esistente)
- 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 con un asterisco rosso.

Parametri dello scenario:

Parametro	Descrizione
Nome	Nome univoco dello scenario. Sono supportate le macro utente. *Nota* che, se vengono utilizzate macro utente, queste macro rimarranno non risolte nei nomi degli item di monitoraggio web.
Intervallo di aggiornamento	Con quale frequenza verrà eseguito lo scenario. Sono supportati i suffissi temporali, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. *Nota* che, 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	Il numero di tentativi per l'esecuzione degli step dello scenario. In caso di problemi di rete (timeout, assenza di connettività, ecc.) Zabbix può ripetere l'esecuzione di uno step più volte. Il valore impostato influirà in ugual misura su ogni step dello scenario. È possibile specificare fino a 10 tentativi, il valore predefinito è 1. *Nota*: Zabbix non ripeterà uno step a causa di un codice di risposta errato o della mancata corrispondenza di una stringa richiesta.
Agent	Seleziona un agent client. Zabbix fingerà di essere il browser selezionato. Questo è utile quando un sito web restituisce contenuti diversi per browser diversi. In questo campo possono essere utilizzate macro utente.
HTTP proxy	Puoi specificare un HTTP proxy da utilizzare, nel formato `[protocol://][username[:password]@]proxy.example.com[:port]`. Questo imposta l'opzione cURL CURLOPT_PROXY. Il prefisso facoltativo `protocol://` può essere utilizzato per specificare protocolli proxy alternativi (il supporto del prefisso di protocollo è stato aggiunto in cURL 7.21.7). Se non viene specificato alcun protocollo, il proxy verrà trattato come un HTTP proxy. Per impostazione predefinita verrà utilizzata la porta 1080. Se specificato, il proxy sovrascriverà le variabili d'ambiente relative al proxy come http_proxy, 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. Puoi anche inserire l'indirizzo di un proxy SOCKS. Se specifichi il protocollo sbagliato, la connessione fallirà e l'item diventerà non supportato. *Nota* che con HTTP proxy è supportata solo l'autenticazione semplice. In questo campo possono essere utilizzate macro utente.

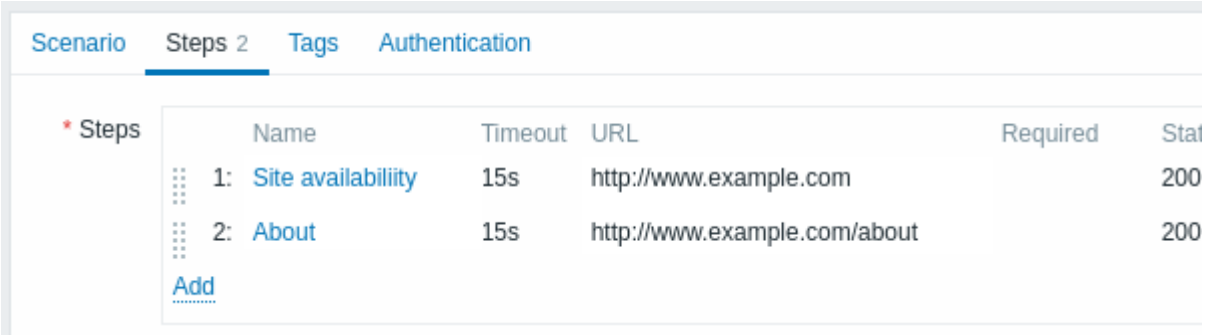
Parametro	Descrizione
Variabili	Variabili che possono essere utilizzate negli step dello scenario (URL, variabili post). Hanno il seguente formato: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (vedi funzioni macro) Per 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 richiamate negli step come {username}, {password}, {hostid}, ecc. Zabbix le sostituirà automaticamente con i valori effettivi. Nota che le variabili con `regex:` necessitano di uno step per ottenere il valore dell'espressione regolare, quindi il valore estratto può essere applicato solo allo step 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. Sono supportate le macro utente e le macro {HOST.*}. Le variabili vengono codificate automaticamente in URL quando utilizzate nei campi di query o nei dati di modulo per le variabili post, ma devono essere codificate manualmente in URL quando utilizzate in post raw o direttamente nell'URL.
Header	Gli header HTTP vengono utilizzati durante l'esecuzione di una richiesta. Possono essere utilizzati header predefiniti e personalizzati. Gli header verranno assegnati utilizzando le impostazioni predefinite in base al tipo di Agent selezionato da un elenco a discesa a livello di scenario e verranno applicati a tutti gli step, a meno che non siano definiti in modo personalizzato a livello di step. **Va notato che definire l'header a livello di step scarta automaticamente tutti gli header definiti in precedenza, ad eccezione di un header predefinito assegnato selezionando 'User-Agent' da un elenco a discesa a livello di scenario.** Tuttavia, anche l'header predefinito 'User-Agent' può essere sovrascritto specificandolo a livello di step. Per annullare l'impostazione dell'header a livello di scenario, l'header deve essere nominato e attribuito senza valore a livello di step. Gli header devono essere elencati utilizzando la stessa sintassi con cui apparirebbero nel protocollo HTTP, facoltativamente usando alcune funzionalità aggiuntive supportate dall'opzione cURL CURLOPT_HTTPHEADER. Per esempio: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Sono supportate le macro utente e le macro {HOST.*}.
Abilitato	Lo scenario è attivo se questa casella è selezionata, altrimenti è disabilitato.

Nota che durante la modifica di uno scenario esistente, nel modulo sono disponibili due pulsanti aggiuntivi:

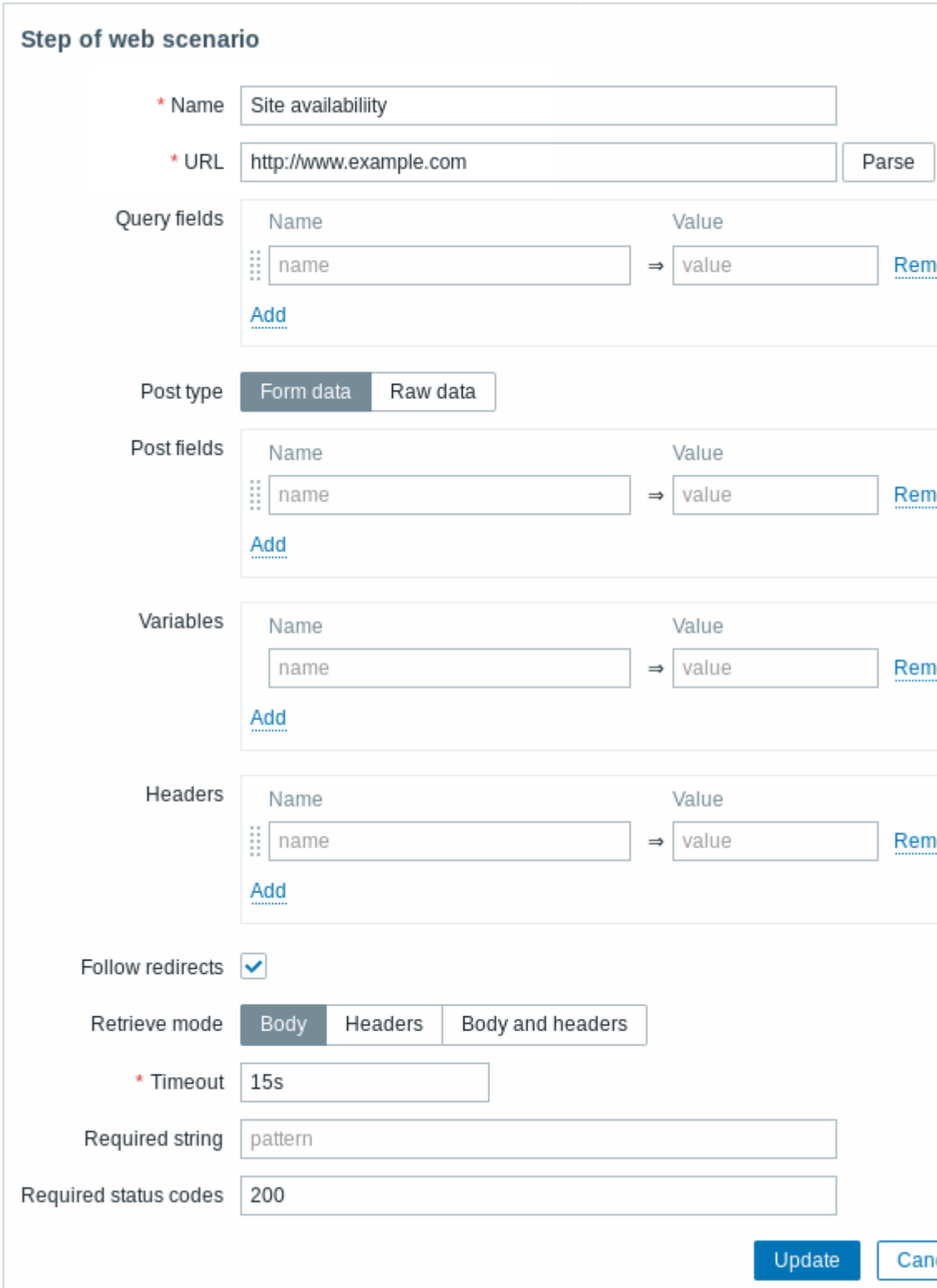
Clone	Crea un altro scenario basato sulle proprietà di quello esistente.
Clear history and trends	Elimina i dati di history e trend dello scenario. Questo farà sì che il server esegua lo scenario immediatamente dopo l'eliminazione dei dati.

Note:
Se il campo *HTTP proxy* viene lasciato vuoto, un altro modo per utilizzare un HTTP proxy è 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. Per esempio, `http_proxy=http://proxy_ip:proxy_port`.
Per i controlli HTTPS, imposta la variabile d'ambiente **HTTPS_PROXY**. Per esempio, `HTTPS_PROXY=http://proxy_ip:proxy_port`. Maggiori dettagli sono disponibili eseguendo un comando della shell: *# man curl*.

La scheda **Step** consente di configurare gli step dello scenario web. Per aggiungere uno step dello scenario web, fai clic su *Aggiungi* nel blocco *Step*.



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* che, se vengono utilizzate macro utente, queste macro rimarranno non risolte nei nomi degli item di monitoraggio web.

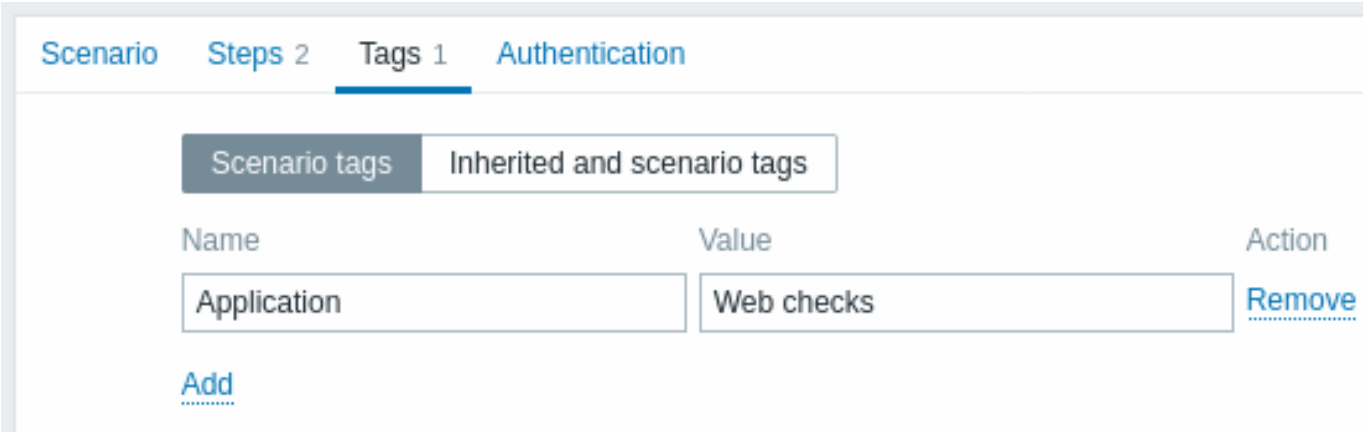
Parametro	Descrizione
URL	URL a cui connettersi e da cui recuperare i dati. Ad esempio: https://www.example.com http://www.example.com/download I nomi di dominio possono essere specificati con caratteri Unicode. Vengono convertiti automaticamente in punycode ASCII durante l'esecuzione del passaggio dello scenario web. Il pulsante *Parse* può essere utilizzato per separare i campi di query opzionali (come ?name=Admin&password=mypassword) dall'URL, spostando attributi e valori in *Campi query* per la codifica URL automatica. Le variabili possono essere utilizzate nell'URL usando la sintassi {macro}. Le variabili possono essere codificate URL manualmente usando la sintassi {{macro}.urlencode()}. Le macro utente e le macro {HOST.*} sono supportate. Limitato a 2048 caratteri.
Campi query	Variabili HTTP GET per l'URL. Specificate come coppie attributo-valore. I valori vengono codificati URL automaticamente. I valori delle variabili dello scenario, delle macro utente o delle macro {HOST.*} vengono risolti e quindi codificati URL automaticamente. L'uso della sintassi {{macro}.urlencode()} comporterà una doppia codifica URL. Le macro utente e le macro {HOST.*} sono supportate.
Post	Variabili HTTP POST. In modalità **Form data**, specificate come coppie attributo-valore. I valori vengono codificati URL automaticamente. I valori delle variabili dello scenario, delle macro utente o delle macro {HOST.*} vengono risolti e quindi codificati URL automaticamente. In modalità **Raw data**, attributi/valori vengono visualizzati su una singola riga e concatenati con il simbolo **&**. I valori grezzi possono essere codificati/decodificati URL manualmente usando la sintassi {{macro}.urlencode()} o {{macro}.urldecode()}. Ad esempio: id=2345&userid={user} Se {user} è definita come variabile dello scenario web, verrà sostituita con il suo valore quando il passaggio viene eseguito. Se si desidera codificare URL la variabile, sostituire {user} con {{user}.urlencode()}. Le macro utente e le macro {HOST.*} sono supportate.
Variabili	Variabili a livello di passaggio che possono essere utilizzate per le funzioni GET e POST. Specificate come coppie attributo-valore. Le variabili a livello di passaggio sovrascrivono le variabili a livello di scenario o le variabili del passaggio precedente. Tuttavia, il valore di una variabile a livello di passaggio influisce solo sul passaggio successivo (e non sul passaggio corrente). Hanno il seguente formato: **{macro}**=value **{macro}**=regex:<espressione regolare> Per ulteriori informazioni vedere la descrizione delle variabili a livello di scenario. Le variabili vengono codificate URL automaticamente quando utilizzate nei campi query o nei dati del modulo per le variabili post, ma devono essere codificate URL manualmente quando utilizzate nel post grezzo o direttamente nell'URL.
Header	Header HTTP personalizzati che verranno inviati durante l'esecuzione di una richiesta. Specificati come coppie attributo-valore. Un header definito a livello di passaggio verrà utilizzato per quel particolare passaggio. **Va notato che definire l'header a livello di passaggio scarta automaticamente tutti gli header definiti in precedenza, ad eccezione di un header predefinito assegnato selezionando 'User-Agent' da un elenco a discesa a livello di scenario.** Tuttavia, anche l'header predefinito 'User-Agent' può essere sovrascritto specificandolo a livello di passaggio. Ad esempio, assegnare il nome a un header senza impostare alcun valore annullerà l'header predefinito a livello di scenario. Le macro utente e le macro {HOST.*} sono supportate. Questo imposta l'opzione cURL CURLOPT_HTTPHEADER.
Segui reindirizzamenti	Selezionare la casella per seguire i reindirizzamenti HTTP. Questo imposta l'opzione cURL CURLOPT_FOLLOWLOCATION.
Modalità di recupero	Selezionare la modalità di recupero: **Body** - recupera solo il body dalla risposta HTTP **Headers** - recupera solo gli header dalla risposta HTTP **Body and headers** - recupera body e header dalla risposta HTTP

Parametro	Descrizione
Timeout	Zabbix non impiegherà più del tempo impostato per elaborare l'URL (da un secondo fino 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 impiegherà più di **2 x Timeout** secondi per il passaggio. I suffissi temporali sono supportati, ad esempio 30s, 1m, 1h. Le macro utente sono supportate.
Stringa richiesta	Pattern di espressione regolare richiesto. A meno che il contenuto recuperato (HTML) non corrisponda al pattern richiesto, il passaggio fallirà. Se vuoto, non viene eseguito alcun controllo sulla stringa richiesta. Ad esempio: Homepage of Zabbix Welcome.*admin *Nota*: il riferimento alle espressioni regolari create nel frontend di Zabbix non è supportato in questo campo. Le macro utente e le macro {HOST.*} sono supportate.
Codici di stato richiesti	Elenco dei codici di stato HTTP previsti. Se Zabbix riceve un codice non presente nell'elenco, il passaggio fallirà. Se vuoto, non viene eseguito alcun controllo sui codici di stato. Ad esempio: 200,201,210-299 Le macro utente sono supportate.

Note:
Qualsiasi modifica nei passaggi dello scenario web verrà salvata solo quando verrà salvato l'intero scenario.

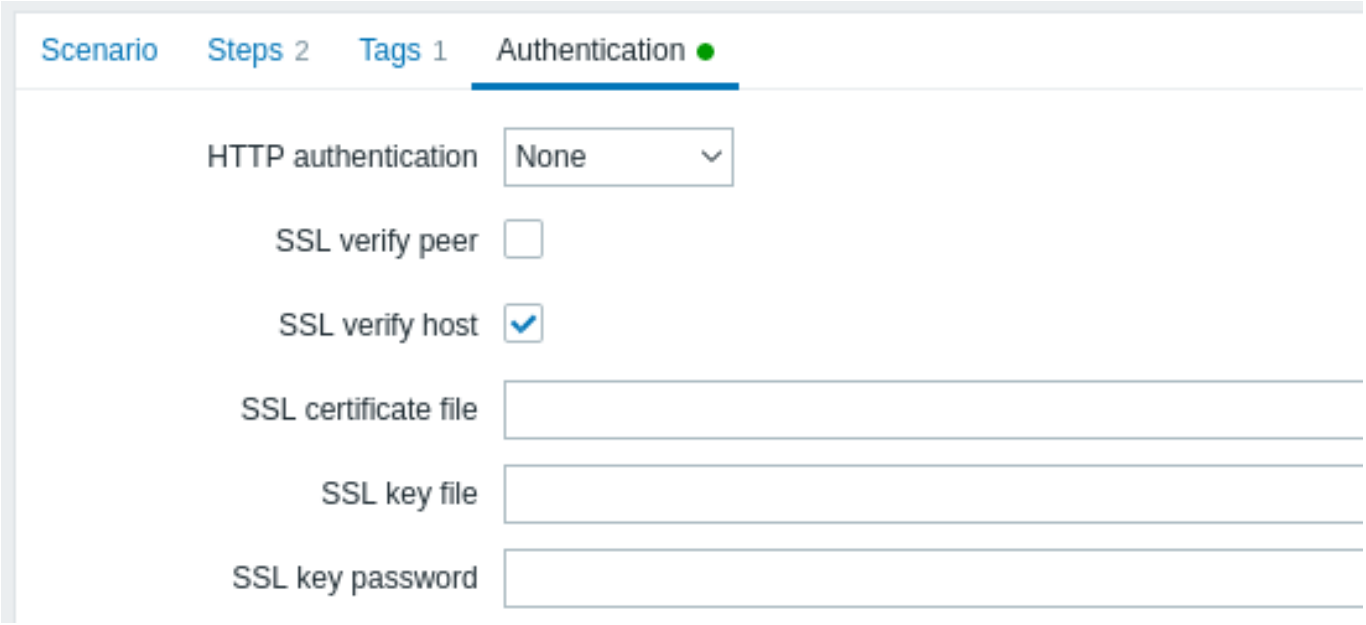
Vedere anche un esempio reale di come i passaggi del monitoraggio web possono essere configurati.

Configurazione dei tag La scheda **Tag** consente di definire il livello di scenario tag.



Il tagging permette di filtrare gli scenari web e il monitoraggio web items.

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 qualche tipo di autenticazione HTTP.



Parametri di autenticazione:

Parameter	Description
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. Le macro utente sono supportate.
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. Le macro utente sono supportate.
Verifica peer SSL	Selezionare la casella per verificare il certificato SSL del web server. Il certificato del server verrà prelevato automaticamente dalla posizione dell'autorità di certificazione (CA) a livello di sistema. È possibile sovrascrivere la posizione dei file CA utilizzando il parametro di configurazione di Zabbix server o proxy SSLCALocation. Questo 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. Questo imposta l'opzione cURL CURLOPT_SSL_VERIFYHOST.
File 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 chiave SSL*. Se la chiave è crittografata, specificare la password nel campo *Password chiave SSL*. La directory che contiene questo file è specificata dal parametro di configurazione di Zabbix server o proxy SSLCertLocation. In questo campo possono essere utilizzate le macro `HOST.*` e le macro utente. Questo imposta l'opzione cURL CURLOPT_SSLCERT.
File 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 che contiene questo file è specificata dal parametro di configurazione di Zabbix server o proxy SSLKeyLocation. In questo campo possono essere utilizzate le macro `HOST.*` e le macro utente. Questo imposta l'opzione cURL CURLOPT_SSLKEY.
Password chiave SSL	Password del file della chiave privata SSL. In questo campo possono essere utilizzate le macro utente. Questo imposta l'opzione cURL CURLOPT_KEYPASSWD.

Attention:
[1] Zabbix supporta solo file di certificato e chiave privata in formato PEM. Se i dati del certificato e della chiave privata sono in un file in formato PKCS #12 (di solito con estensione *.p12 o *.pfx), è possibile generare da esso 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:
Zabbix server recepisce le modifiche ai certificati senza riavvio.

Note:
Se si dispone del certificato client e della chiave privata in un unico file, è sufficiente specificarlo nel campo "File certificato SSL" e lasciare vuoto il campo "File chiave SSL". Il certificato e la chiave devono comunque essere in formato PEM. Combinare certificato e chiave è semplice:

```
cat client.crt client.key > client.pem
```

Visualizzazione Per visualizzare gli scenari web configurati per un host, vai in *Monitoring* → *Hosts*, individua l'host nell'elenco e fai clic sul collegamento *Web* nell'ultima colonna. Fai clic sul nome dello scenario per ottenere informazioni dettagliate.



Una panoramica degli scenari web può essere visualizzata anche in *Dashboards* tramite il widget Web monitoring.

I risultati recenti dell'esecuzione dello scenario web sono disponibili nella sezione *Monitoring* → *Latest data*.

Monitoraggio esteso A volte è necessario registrare il contenuto della pagina HTML ricevuta. Ciò è particolarmente utile se un passaggio dello scenario web non riesce. Il livello di debug 5 (trace) serve a questo scopo. Questo livello può essere impostato nei file di configurazione di server e proxy oppure utilizzando un'opzione di runtime control (`-R log_level_increase="http poller,N"`, dove N è il numero del processo). Gli esempi seguenti mostrano come avviare il monitoraggio esteso, a condizione che il livello di debug 4 sia già impostato:

```
# Aumenta il livello di log di tutti gli http poller:
zabbix_server -R log_level_increase="http poller"

# Aumenta il livello di log del secondo http poller:
zabbix_server -R log_level_increase="http poller,2"
```

Se il monitoraggio web esteso non è necessario, può essere interrotto utilizzando l'opzione `-R log_level_decrease`.

1 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`

2 Scenario reale

Panoramica

Questa sezione presenta un esempio reale passo dopo passo di come il monitoraggio web possa essere utilizzato.

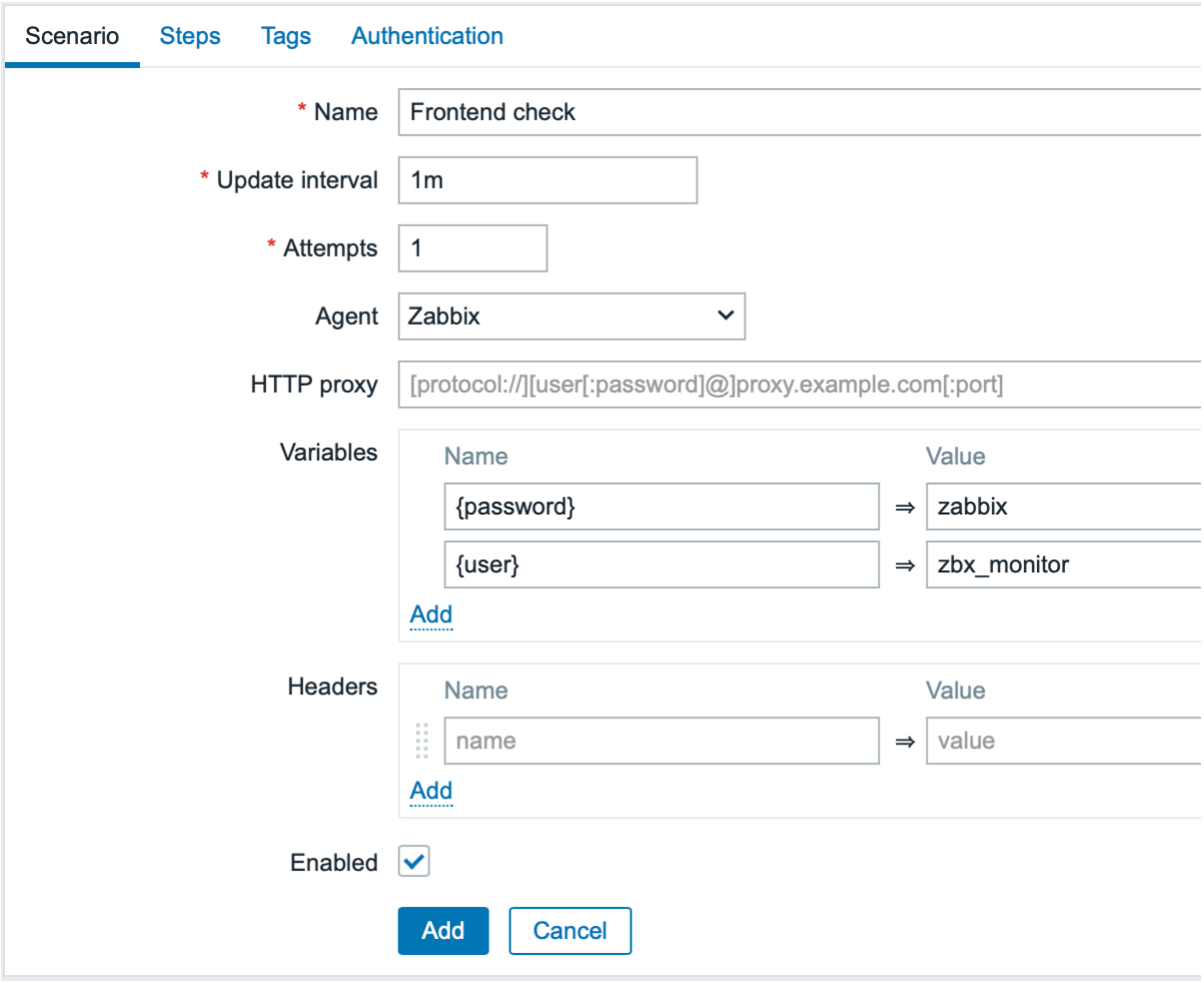
Il monitoraggio web di Zabbix verrà utilizzato per monitorare il frontend di Zabbix. L’obiettivo è determinare se è disponibile, se fornisce il contenuto corretto e con quale rapidità funziona. Per raggiungere questo obiettivo, sono necessari diversi passaggi, tra

cui il controllo della disponibilità della prima pagina, l'accesso con nome utente e password, la verifica dell'avvenuto accesso, la disconnessione e la conferma della disconnessione.

Scenario

Aggiungere un nuovo scenario web

Vai su *Raccolta dati* → *Host*, seleziona un host e fai clic su *Web* nella riga di quell'host.
Quindi fai clic su *Crea scenario web*.



Nel modulo del nuovo scenario, compila i seguenti campi:

- **Nome** - Controllo frontend
- **Intervallo di aggiornamento** - 1m
- **Tentativi** - 1
- **Agent** - Zabbix

Nella sezione *Variabili*, aggiungi due variabili: *{password}* e *{user}*.
Inserisci come valori le credenziali del tuo utente Zabbix esistente.

Per motivi di sicurezza, si consiglia di creare un utente separato con permessi minimi da utilizzare per scopi di monitoraggio.

Facoltativamente, passa alla scheda *Tag* e aggiungi tag dello scenario web.

Note:
Una volta configurato completamente, questo scenario web aggiungerà automaticamente un item Zabbix trapper all'host. Puoi usare i tag dello scenario web per identificare rapidamente item e trigger correlati oppure per cercare tra i dati raccolti. Ad esempio, tag adatti per questa esercitazione sono `component: web-scenario` e/o `target: frontend`.

Configurare i passaggi dello scenario web

Passare alla scheda *Steps* e definire i passaggi per lo scenario. Fare clic sul pulsante *Add* per aggiungere un singolo passaggio.

Campi comuni

Per ogni passaggio descritto di seguito, compilare i seguenti campi oltre ai campi specifici del passaggio:

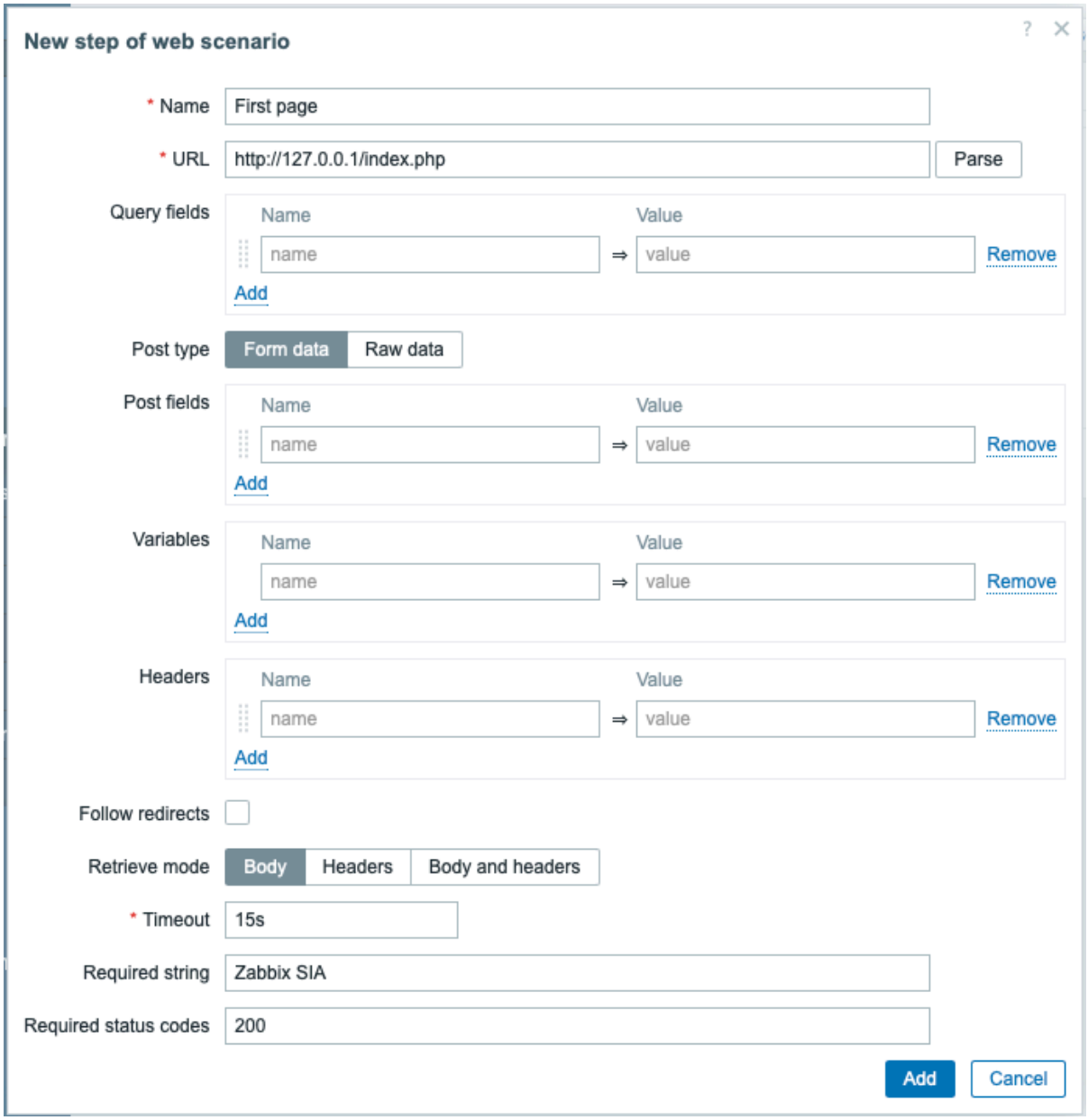
- **URL** - l'URL del frontend di Zabbix
- **Timeout** - 15s
- **Required status codes** - 200

Passo 1 dello scenario web

Verificare che la prima pagina risponda correttamente, restituisca il codice di risposta HTTP 200 e contenga il testo "Zabbix SIA".

- Nel campo **Nome**, inserire *Prima pagina*.
- Nel campo **Stringa richiesta**, inserire *Zabbix SIA*.
- Compilare i campi comuni.

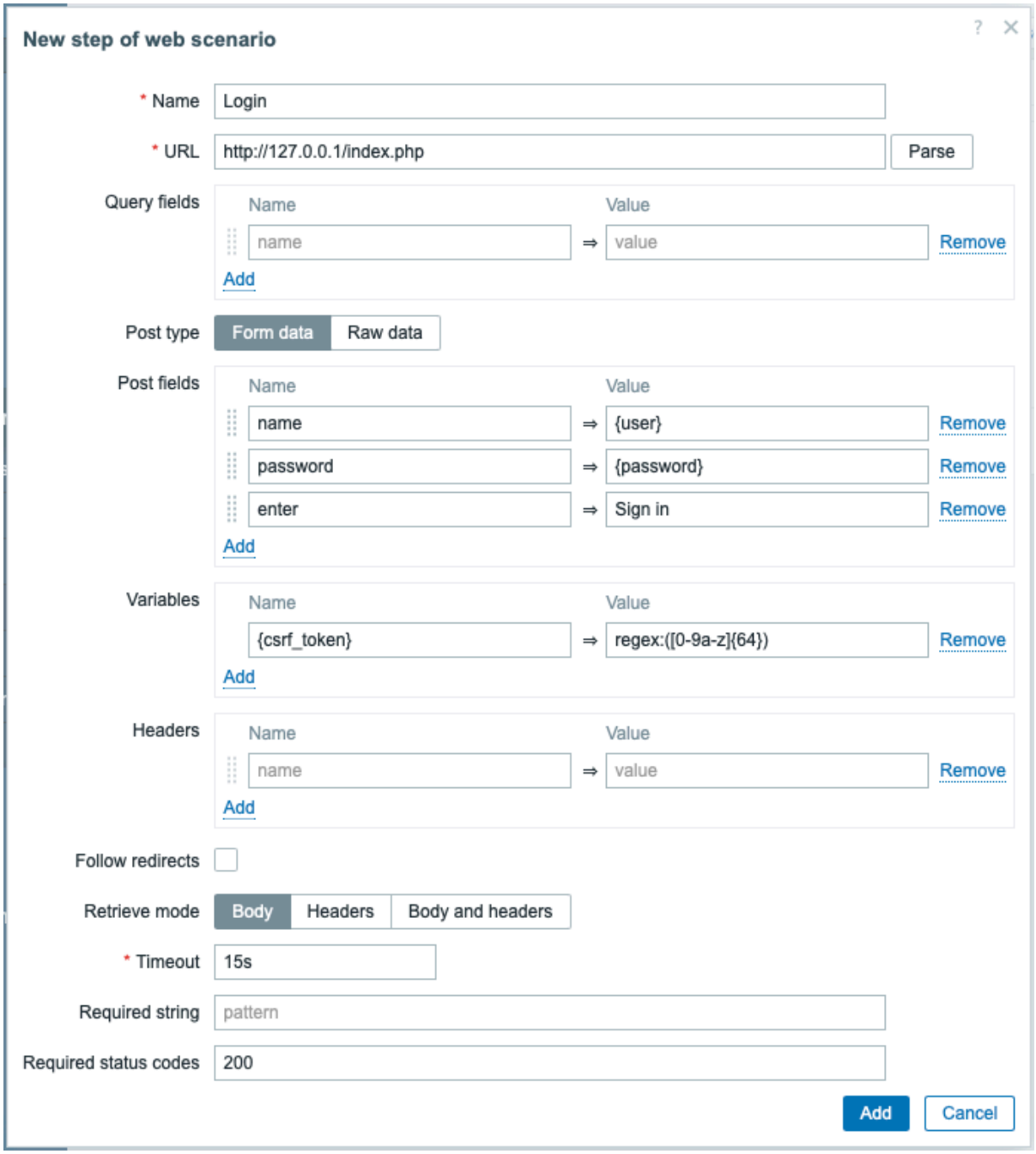
Una volta completata la configurazione del passo, premere il pulsante *Aggiungi*.



Passo 2 dello scenario web

Accedi al frontend di Zabbix utilizzando le macro (variabili) definite a livello di scenario - *{user}* e *{password}*.

- Nel campo **Name**, inserisci *Login*.
- Nella sezione **Post fields**, aggiungi tre campi post:
  - *name* con valore *{user}*
  - *password* con valore *{password}*
  - *enter* con valore *Sign in*
- Nella sezione **Variables**, aggiungi una nuova variabile *{csrf_token}* con valore *regex:([0-9a-z]{64})*. Questa variabile intercetterà il valore del token CSRF assegnato per riutilizzarlo nel passo 4.
- Compila i campi comuni.

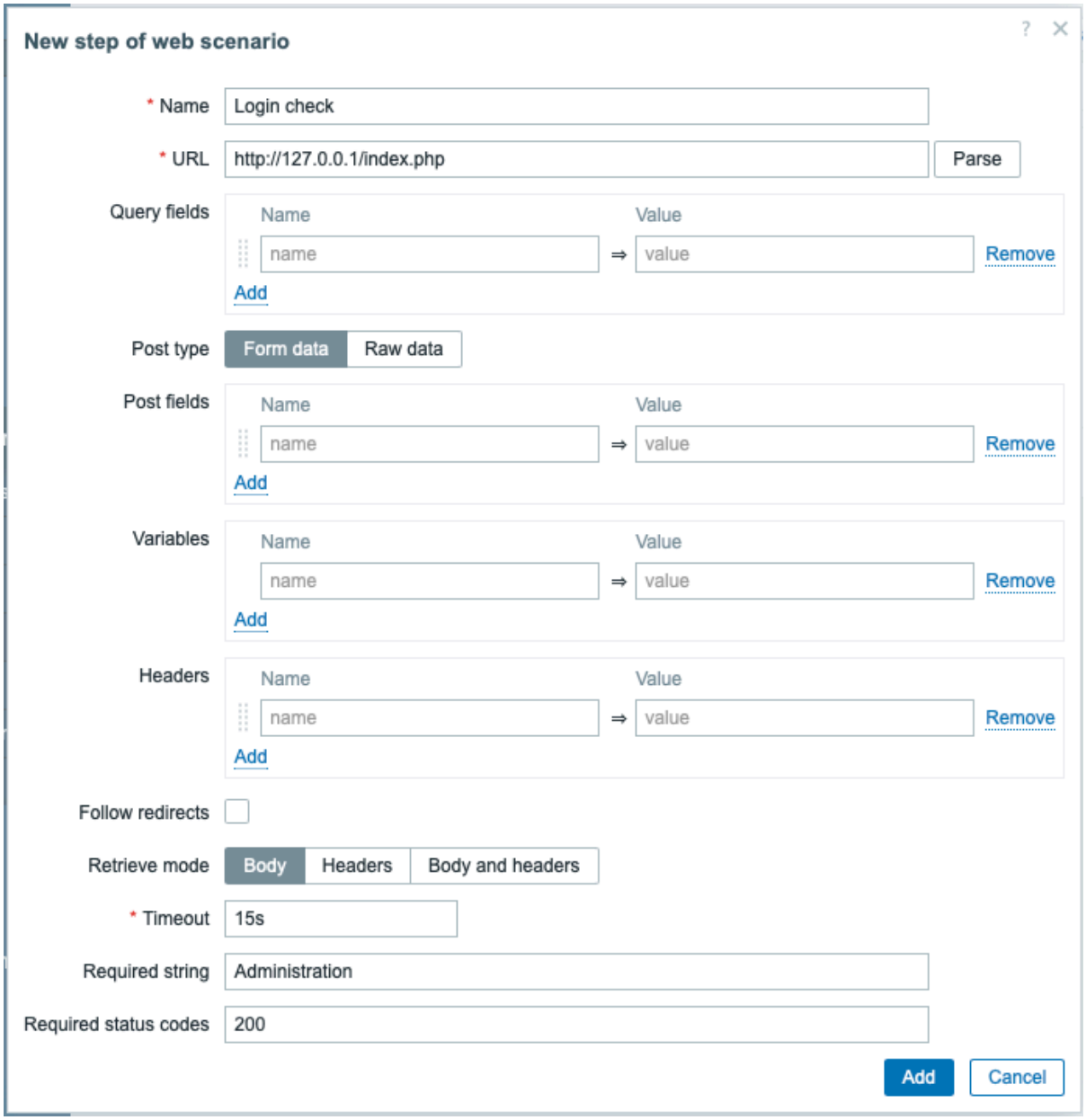


Attention:
Nota che il frontend di Zabbix utilizza un reindirizzamento JavaScript durante l'accesso, quindi l'accesso deve avvenire per primo e le funzionalità disponibili dopo l'accesso possono essere verificate solo nei passaggi successivi. Inoltre, il passaggio di accesso deve utilizzare l'URL completo del file **index.php**.

Passo 3 dello scenario web

Dopo aver effettuato l'accesso, verificare il successo controllando la presenza di una stringa visibile solo quando si è autenticati, ad esempio *Administration*.

- Nel campo **Name**, inserire *Login check*.
- Nel campo **Required string**, inserire *Administration*.
- Compilare i campi comuni.

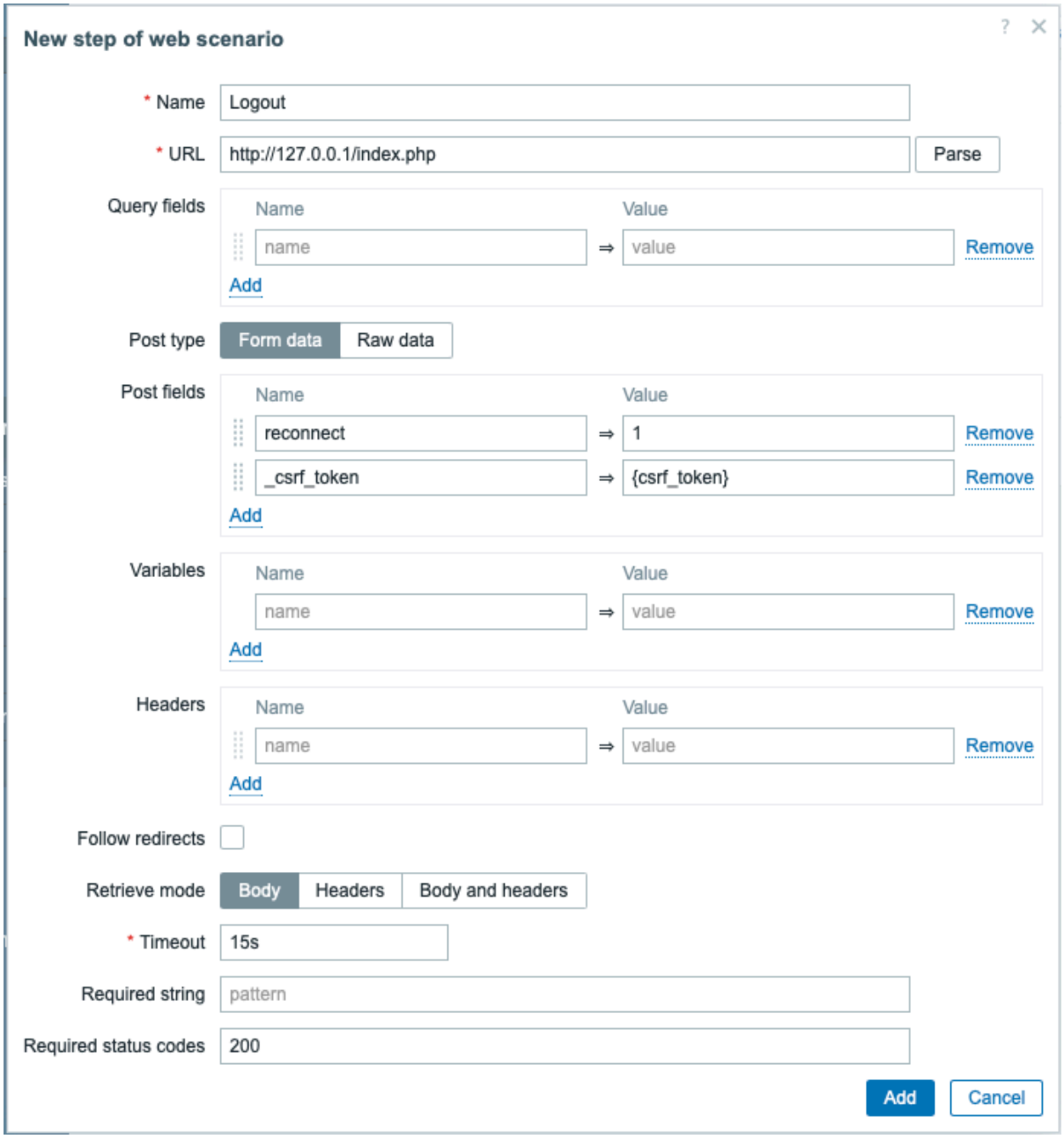


Passo 4 dello scenario web

Una volta verificati l'accessibilità del frontend e il login, aggiungere un passaggio di logout; altrimenti il database di Zabbix si riempirà di molti record di sessioni aperte.

- Nel campo **Name**, inserire *Logout*.
- Nella sezione **Post fields**, aggiungere due campi post:
  - *reconnect* con valore *1*
  - *_csrf_token* con valore *{csrf_token}*.
- Compilare i campi comuni.

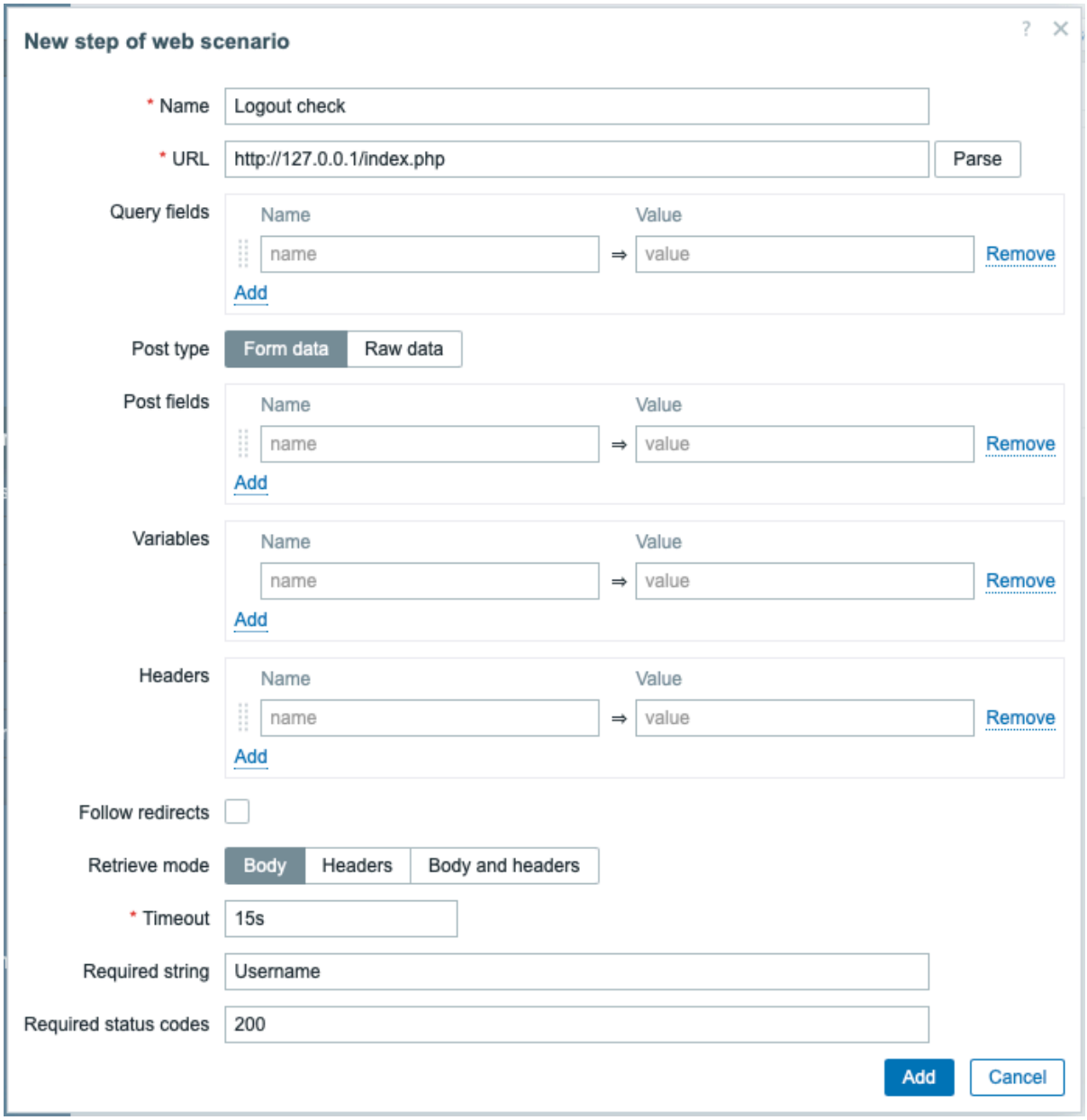
Questo passaggio utilizza la variabile {csrf_token} ottenuta nel passo 2



Passo 5 dello scenario web

Per confermare il logout, verificare la presenza della stringa **Username**.

- Nel campo **Name**, inserire *Logout check*.
- Nel campo **Required string**, inserire *Username*.
- Compilare i campi comuni.



Configurazione completa dei passaggi

Una configurazione completa dei passaggi dello scenario web dovrebbe apparire così:

	Name	Timeout	URL	Required	Status codes	Action
1:	First page	15s	http://127.0.0.1/index.php	Zabbix SIA	200	Remove
2:	Log in	15s	http://127.0.0.1/index.php		200	Remove
3:	Log in Check	15s	http://127.0.0.1/index.php	Administrati…	200	Remove
4:	Log out	15s	http://127.0.0.1/index.php		200	Remove
5:	Log out check	15s	http://127.0.0.1/index.php	Username	200	Remove

Add

Controlla i risultati

Salva lo scenario di monitoraggio web completato.

Lo scenario verrà aggiunto all'host. Per visualizzare le informazioni sullo scenario web, vai in *Monitoring* → *Hosts*, individua l'host nell'elenco e fai clic sul collegamento Web nell'ultima colonna.



Fai clic sul nome dello scenario per visualizzare statistiche più dettagliate:



8 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 Zabbix server è 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 di Zabbix server 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; in caso contrario, il recupero delle statistiche dei contatori di prestazione VMware potrebbe subire ritardi a causa del recupero dei dati di configurazione VMware (che richiede tempo nelle installazioni di grandi dimensioni).

 In generale, il valore di `StartVMwareCollectors` non dovrebbe essere inferiore a 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'estensione 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 almeno su 64. Per ulteriori informazioni, vedere l'articolo della *VMware Knowledge Base*.

Individuazione

Zabbix può utilizzare regole di individuazione di basso livello (ad esempio, *vmware.hv.discovery[{$VMWARE.URL}]*) per individuare automaticamente hypervisor VMware e macchine virtuali. Inoltre, Zabbix può utilizzare prototipi di host per generare automaticamente host reali per le entità individuate. Per ulteriori informazioni, vedere *Prototipi di host*.

Configuration examples

For a basic example of how to set up Zabbix for monitoring VMware using the *VMware FQDN* template, see *Monitor VMware with Zabbix*.

For a more detailed example of how to create a host, a low-level discovery rule, and a host prototype for monitoring VMware, see *Setup example*.

Logging esteso 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 di server e 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 seguente comando:

```
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 seguente comando:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Quando il logging esteso dei dati del VMware collector non è più necessario, 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 rese non disponibili o disattivate per impostazione predefinita nelle versioni recenti di VMware vSphere, oppure che non siano stati applicati limiti alle query del database delle metriche di prestazione. Per ulteriori informazioni, vedere ZBX-12094.
- Se si verifica l'errore `config.vpxd.stats.maxQueryMetrics` non valido o supera il numero massimo di caratteri consentiti, 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 della *VMware Knowledge Base*.
- Se si sospetta che l'installazione di Zabbix stia utilizzando troppa memoria, vedere Profilazione dell'utilizzo eccessivo della memoria con tcmalloc.

1 Elementi chiave di 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 tutti i dettagli.

Item key	Descrizione	Gruppo di item
vmware.eventlog	Il log eventi VMware.	Servizio generale
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	Il rilevamento 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	Il rilevamento 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	Il tempo impiegato per un'operazione di lettura dal datastore.	
vmware.datastore.size	Lo spazio del datastore VMware in byte o in percentuale del totale.	
vmware.datastore.tags.get	L'array dei tag del datastore VMware.	
vmware.datastore.write	Il tempo impiegato per un'operazione di scrittura nel datastore.	
vmware.dc.alarms.get	I dati degli allarmi del datacenter VMware.	Datacenter
vmware.dc.discovery	Il rilevamento dei datacenter VMware.	
vmware.dc.tags.get	L'array dei tag del datacenter VMware.	
vmware.dvswitch.discovery	Il rilevamento degli switch distribuiti VMware vSphere.	vSphere Distributed Switch
vmware.dvswitch.fetchports.get	I dati delle porte dello switch distribuito VMware vSphere.	
vmware.hv.alarms.get	I dati degli allarmi dell'hypervisor VMware.	Hypervisor
vmware.hv.cluster.name	Il nome del cluster dell'hypervisor VMware.	

Item key	Descrizione	Gruppo di item
vmware.hv.connectionstate	Lo stato della 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	Il rilevamento 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	Il tempo medio impiegato per un'operazione di lettura dal datastore.	
vmware.hv.datastore.size	Lo spazio del datastore VMware in byte o in percentuale del totale.	
vmware.hv.datastore.write	Il tempo medio impiegato per un'operazione di scrittura nel datastore.	
vmware.hv.discovery	Il rilevamento degli hypervisor VMware.	
vmware.hv.diskinfo.get	I dati dei dischi 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	Il BIOS UUID 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	Il rilevamento 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	Il consumo energetico dell'hypervisor VMware.	
vmware.hv.sensor.health.state	Il sensore riepilogativo dello stato di salute dell'hypervisor VMware.	
vmware.hv.sensors.get	I sensori di stato del fornitore 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	Il tempo di attività 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 VMware Resource Pool.	Pool di risorse
vmware.rp.memory	Le metriche della memoria del pool di risorse 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.	Macchina virtuale
vmware.vm.attribute	Il valore dell'attributo personalizzato della macchina virtuale VMware.	
vmware.vm.cluster.name	Il nome della macchina virtuale VMware.	
vmware.vm.consolidationneeded	Lo stato della macchina virtuale VMware richiede il consolidamento.	
vmware.vm.cpu.latency	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.	
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 poteva essere pianificata per l'esecuzione sulla CPU fisica.	
vmware.vm.cpu.ready	Il tempo in cui la macchina virtuale era pronta, ma non poteva essere pianificata per l'esecuzione sulla CPU fisica.	
vmware.vm.cpu.swapwait	La percentuale di tempo CPU trascorsa in attesa dello swap-in.	

Item key	Descrizione	Gruppo di item
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	Il rilevamento delle macchine virtuali VMware.	
vmware.vm.guest.memory.swapped	La quantità di memoria fisica guest trasferita nello spazio 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 del host consumata per supportare le 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 in 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 del host che è stata consumata.	
vmware.vm.net.if.discovery	Il rilevamento 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 verso il disco virtuale durante l'intervallo di raccolta.	
vmware.vm.storage.totalreadlatency	Il tempo medio richiesto da una lettura dal disco virtuale.	
vmware.vm.storage.totalwritelatency	Il tempo medio richiesto 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 verso il 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 dei guest tools della macchina virtuale VMware.	
vmware.vm.uptime	Il tempo di attività della macchina virtuale VMware.	
vmware.vm.vfs.dev.discovery	Il rilevamento dei dispositivi disco della macchina virtuale VMware.	
vmware.vm.vfs.dev.read	Le statistiche di lettura dei dispositivi disco della macchina virtuale VMware.	
vmware.vm.vfs.dev.write	Le statistiche di scrittura dei dispositivi disco della macchina virtuale VMware.	
vmware.vm.vfs.fs.discovery	Il rilevamento 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 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 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 item `vmware.eventlog` per URL;
- Vedi 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: oggetto JSON.

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 VMware DVSwitch.
- **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: oggetto JSON.

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 fornitore 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: oggetto JSON.

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 virtual center VMware.
 Valore restituito: oggetto JSON.

Parametri:

- **url** - l'URL del servizio VMware.

vmware.vm.alarms.get[url,uuid]

 I dati degli allarmi 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.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 dei contatori di prestazioni per VMware.

[2] Il valore di questi item viene ottenuto dai contatori di 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.

2 Campi chiave del rilevamento delle macchine virtuali

La tabella seguente elenca i campi restituiti dalle chiavi di rilevamento relative alle macchine virtuali.

Chiave item		
Descrizione	**Campo**	**Contenuto recuperato**
vmware.cluster.discovery		
Esegue il rilevamento dei cluster.	{#CLUSTER.ID}	Identificatore del cluster.
	{#CLUSTER.NAME}	Nome del cluster.

	"resource_pool"	Un array contenente i dati del pool di risorse, inclusi ID del gruppo di risorse, array di tag, percorso del pool di risorse, numero di macchine virtuali.
		
		Struttura dell'array:
		`[{`
		`"rpid":"id gruppo di risorse",`
		`"tags":[{}],`
		`"rpath":"percorso gruppo di risorse",`
		`"vm_count":0`
		`}]`
		
		Per la struttura dell'array `"tags"`, vedere il campo "tags".

Chiave item		

	”tags”	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell’array:
		`[{`
		`"tag":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

vmware.datastore.discovery		
Esegue il rilevamento dei datastore.	{#DATASTORE}	Nome del datastore.

	{#DATASTORE.EXTENT}	Un array contenente l’ID della partizione extent del datastore e il nome dell’istanza.
		
		Struttura dell’array:
		`[{`
		`"partitionid":1,`
		`"instance":"nome"`
		`}]`

	{#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":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

vmware.dc.discovery		
Esegue il rilevamento dei 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":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

vmware.dvswitch.discovery		
Esegue il rilevamento degli switch distribuiti vSphere.	{#DVS.NAME}	Nome dello switch.
	{#DVS.UUID}	Identificatore dello switch.
vmware.hv.discovery		
Esegue il rilevamento degli 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.

Chiave item		
	{#CLUSTER.NAME}	Nome del cluster, potrebbe essere vuoto.
	{#DATACENTER.NAME}	Nome del datacenter.
	{#PARENT.NAME}	Nome del contenitore che memorizza l'hypervisor.
	{#PARENT.TYPE}	Tipo di contenitore in cui è memorizzato 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 pool di risorse, inclusi ID del gruppo di risorse, array di tag, percorso del pool di risorse, numero di macchine virtuali.
		
		Struttura dell'array:
		`[{`
		`"rpid":"id gruppo di risorse",`
		`"tags":[{}],`
		`"rpath":"percorso gruppo di risorse",`
		`"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":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

vmware.hv.datastore.discovery		
Esegue il rilevamento dei datastore dell'hypervisor. Si noti che più hypervisor possono utilizzare 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 extent del datastore e l'ID della partizione.
		
		Struttura dell'array:
		`[{`
		`"partitionid":1,`
		`"instance":"nome"`
		`}]`

	"tags"	Un array contenente tag con nome del tag, descrizione e categoria.
		
		Struttura dell'array:
		`[{`
		`"tag":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

vmware.hv.net.if.discovery		
Esegue il rilevamento delle interfacce di rete dell'hypervisor.	{#IFNAME}	Nome dell'interfaccia.
	{#IFDRIVER}	Driver dell'interfaccia.

Chiave item		
	{#IFDUPLEX}	Impostazioni duplex dell'interfaccia.
	{#IFSPEED}	Velocità dell'interfaccia.
	{#IFMAC}	Indirizzo MAC dell'interfaccia.
vmware.vm.discovery		
Esegue il rilevamento delle macchine virtuali.	{#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 vuota.
	{#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 vuota.
	{#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 pool di risorse.
	{#VM.RPOOL.PATH}	Percorso completo del pool di risorse, escluso 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":"nome tag",`
		`"tag_description":"descrizione tag",`
		`"category":"categoria tag"`
		`}]`

	"vm_customattribute"	Un array di attributi personalizzati della macchina virtuale (se definiti).
		
		Struttura dell'array:
		`[{`
		`"name":"nome campo personalizzato",`
		`"value":"valore campo personalizzato"`
		`}]`

Chiave item		

	”net_if”	Un array di interfacce di rete della macchina virtuale.
		
		Struttura dell’array:
		`[{`
		`"ifname": "nome interfaccia",`
		`"ifdesc": "descrizione interfaccia",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "tipo di interfaccia",`
		`"ifbackingdevice": "dispositivo di backing dell'interfaccia",`
		`"ifdvswitch_uuid": "uuid dello switch dell'interfaccia",`
		`"ifdvswitch_portgroup": "gruppo di porte dello switch dell'interfaccia",`
		`"ifdvswitch_port": "porta dello switch dell'interfaccia",`
		`"ifip": ["indirizzi IP dell'interfaccia"]`
		`}]`
		
		Per la descrizione dei dati restituiti, vedere la chiave item ”vmware.vm.net.if.discovery”.

vmware.vm.net.if.discovery		
Esegue il rilevamento delle interfacce di rete della macchina virtuale.	{#IFNAME}	Nome dell’interfaccia di rete.
	{#IFDESC}	Descrizione dell’interfaccia.
	{#IFMAC}	Indirizzo MAC dell’interfaccia.
	{#IFCONNECTED}	Stato della connessione dell’interfaccia (`false` - disconnessa; `true` - connessa).
	{#IFTYPE}	Tipo di interfaccia.
	{#IFBACKINGDEVICE}	Nome del dispositivo di backing.
	{#IFDVSWITCH.UUID}	Identificatore univoco dello switch distribuito vSphere.
	{#IFDVSWITCH.PORTGROUP}	Gruppo di porte distribuito.
	{#IFDVSWITCH.PORT}	Porta dello switch distribuito vSphere.
	”ifip”	Un array di indirizzi dell’interfaccia.
vmware.vm.vfs.dev.discovery		
Esegue il rilevamento dei dispositivi disco della macchina virtuale.	{#DISKNAME}	Nome del dispositivo disco.
vmware.vm.vfs.fs.discovery		
Esegue il rilevamento del file system della macchina virtuale.	{#FSNAME}	Nome del file system.

3 Esempi JSON per elementi VMware

Panoramica Questa sezione fornisce informazioni aggiuntive sugli oggetti JSON restituiti da vari elementi 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
        },
        {
            "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
        },
        {
            // altri allarmi
        }
    ]
}
```

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
}
```

4 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.

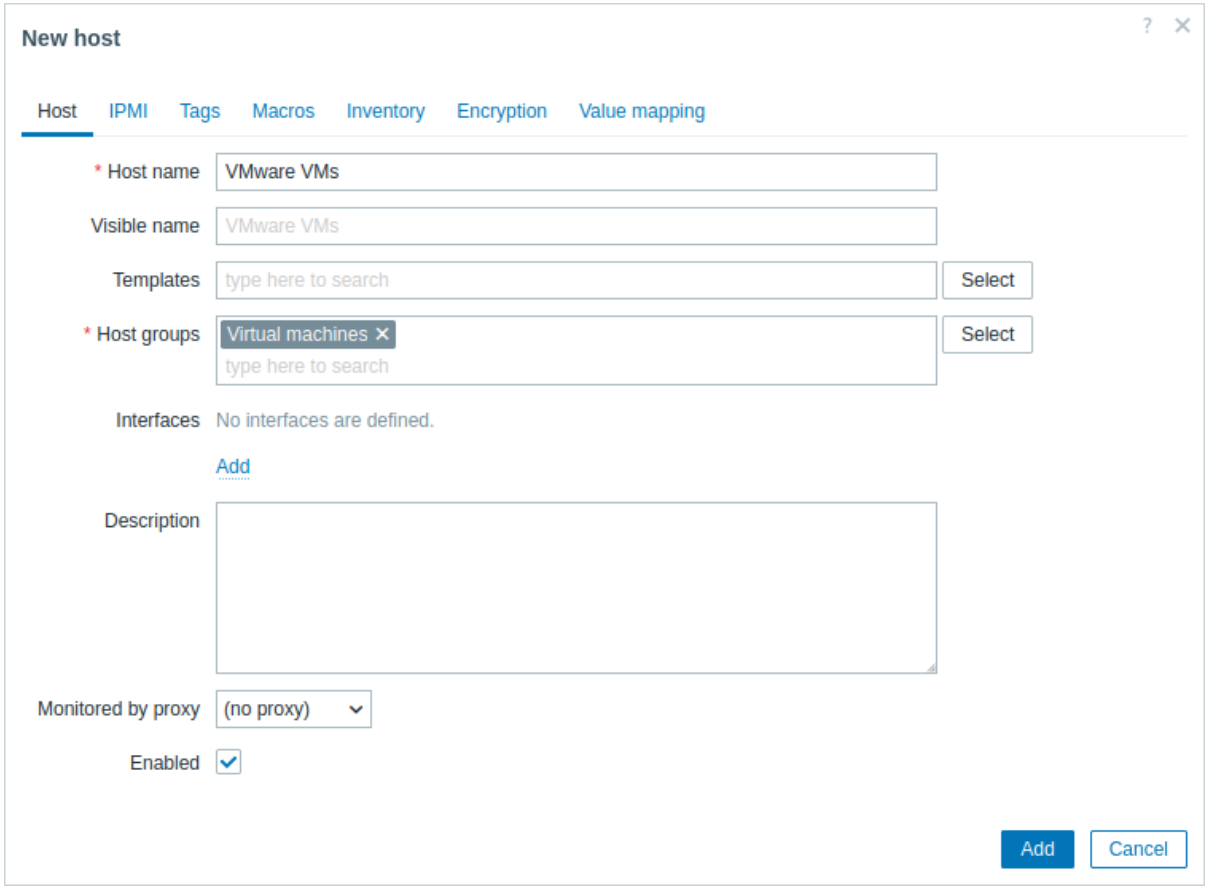
Prima di procedere, impostare il parametro `StartVMwareCollectors` nel file di configurazione del server Zabbix su 2 o più (il valore predefinito è 0).

Creare un host

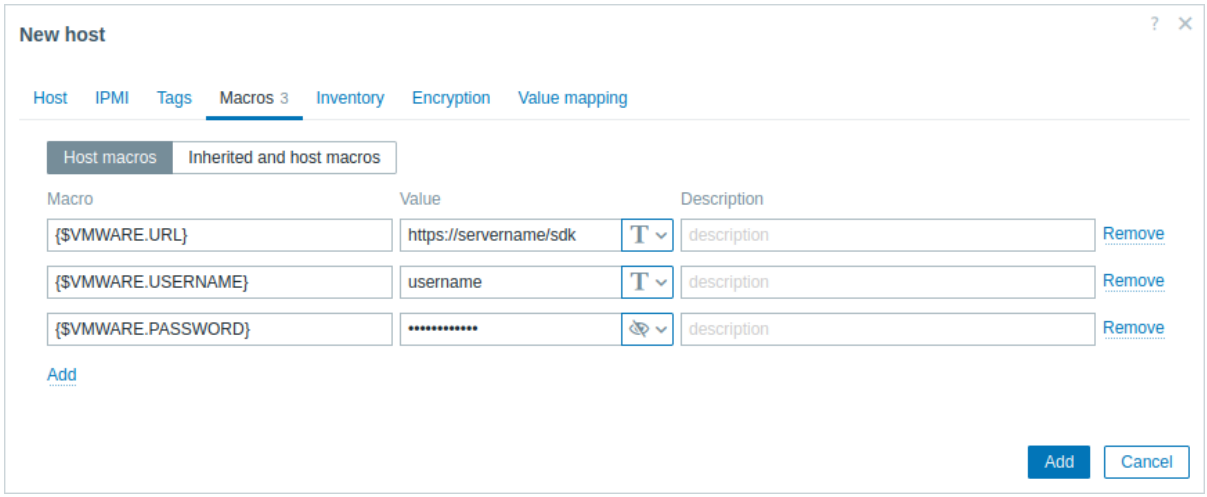
1. Vai a *Raccolta dati* → *Host*.

2. Crea un host:

- Nel campo *Nome host*, inserisci un nome host (ad esempio, "VMware VMs").
- Nel campo *Gruppi host*, digita o seleziona un gruppo host (ad esempio, "Virtual machines").



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



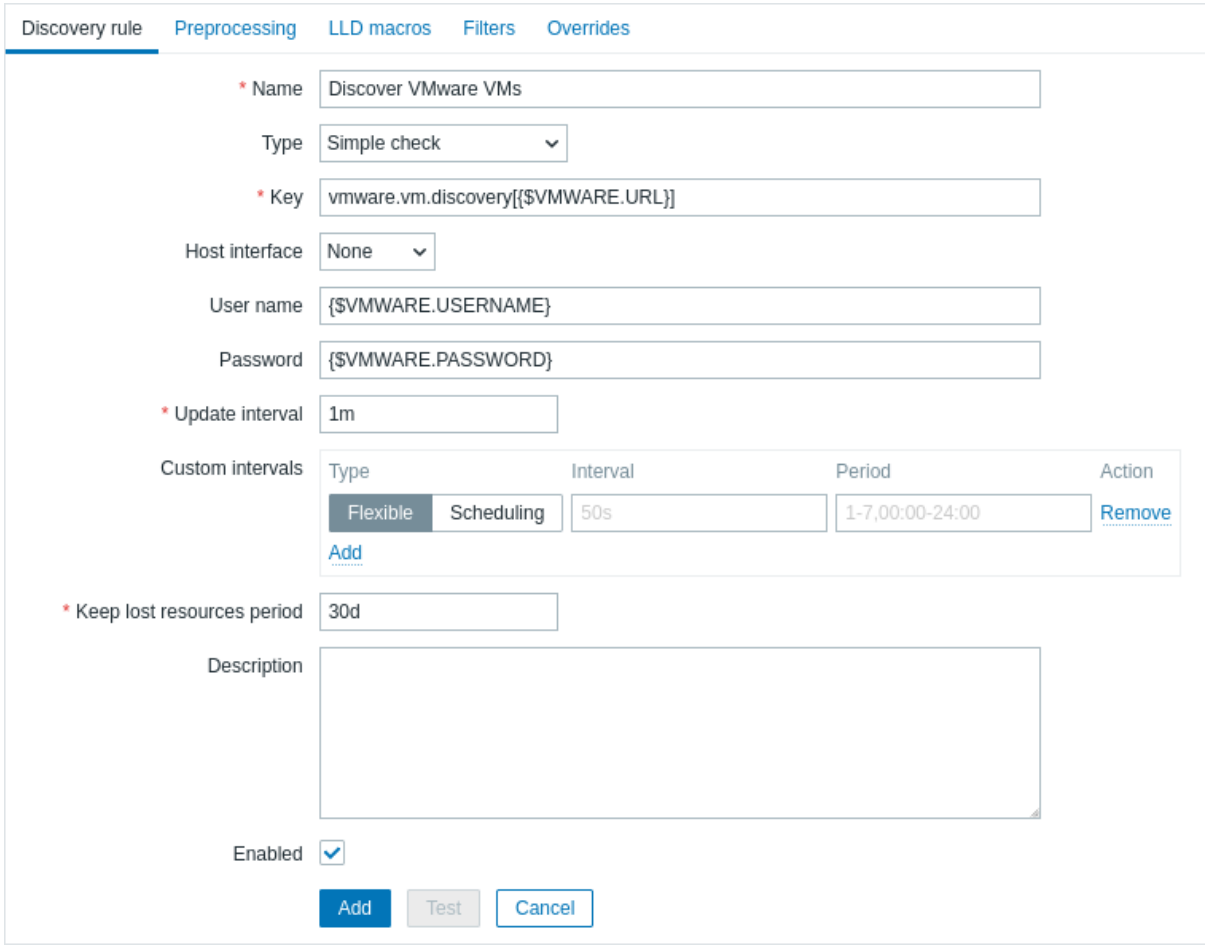
3. Fai clic sul pulsante *Aggiungi* per creare l'host. Questo host rappresenterà il tuo ambiente VMware.

Creare una regola di low-level discovery

1. Fare clic su *Discovery* per il host creato per accedere all'elenco delle regole di low-level discovery per quel host.

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 individuare le macchine virtuali VMware: `vmware.vm.discovery[{$VMWARE.URL`
- Nei campi *User name* e *Password*, inserire le macro corrispondenti precedentemente configurate sul host.



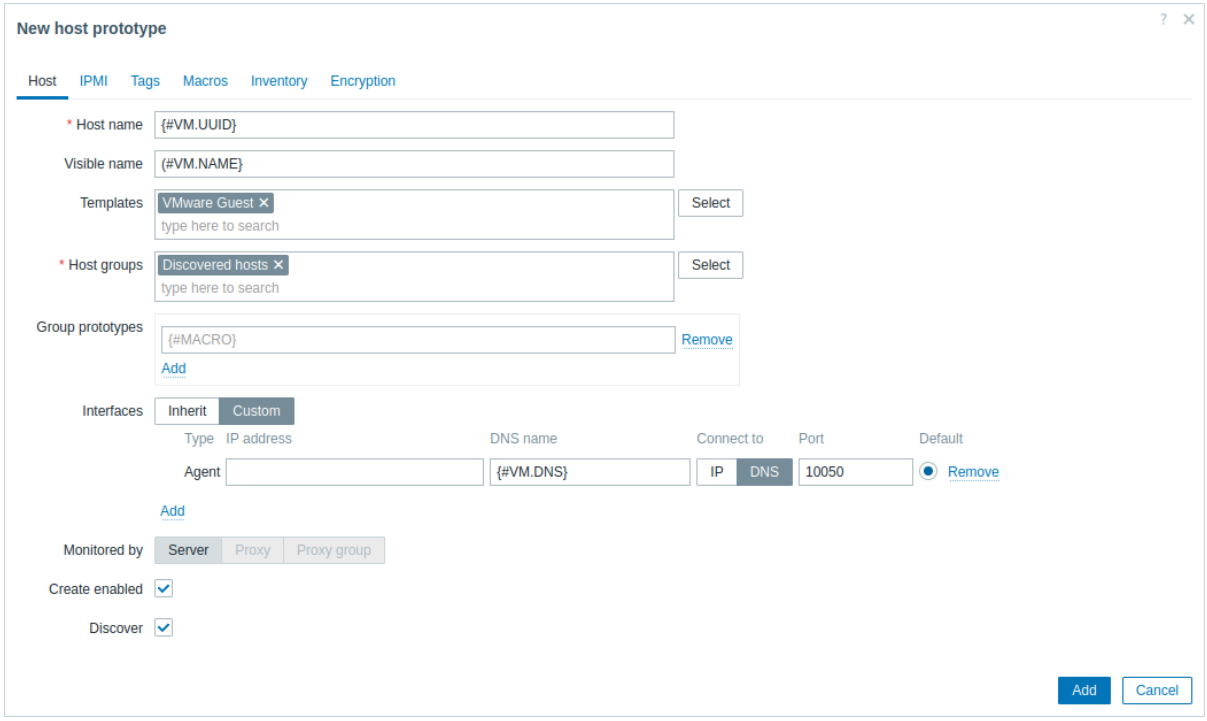
3. Fare clic sul pulsante *Add* per creare la regola di low-level discovery. Questa regola di discovery individuerà le macchine virtuali nel proprio ambiente VMware.

Creare un prototipo di host

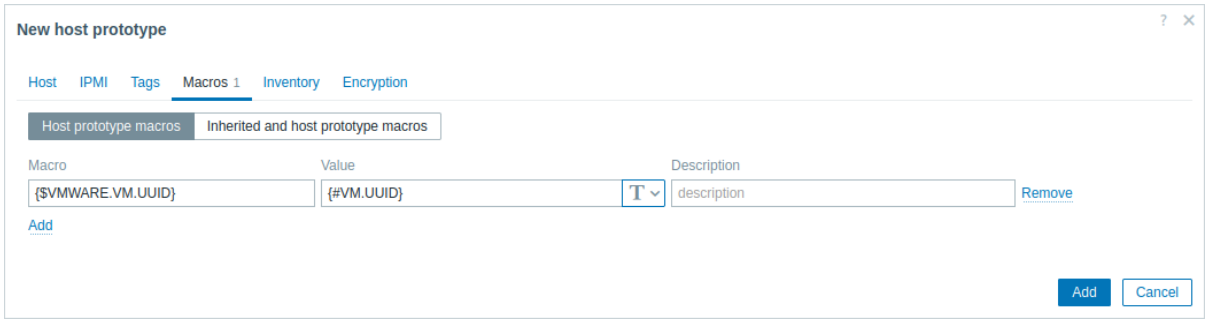
1. Nell'elenco delle regole di low-level discovery, fai clic su *Host prototypes* per la regola di low-level discovery creata in precedenza.

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}]*).

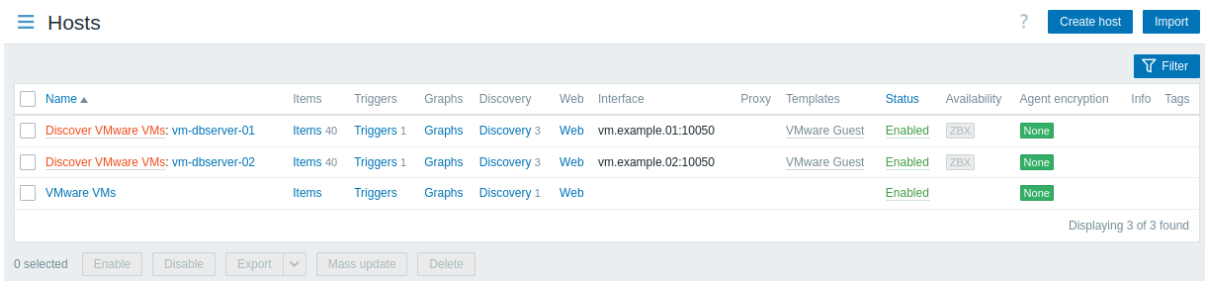


3. Fai clic sul pulsante *Add* per creare il prototipo di host. Questo prototipo di host verrà utilizzato per creare host per le macchine virtuali rilevate dalla regola di low-level discovery creata in precedenza.

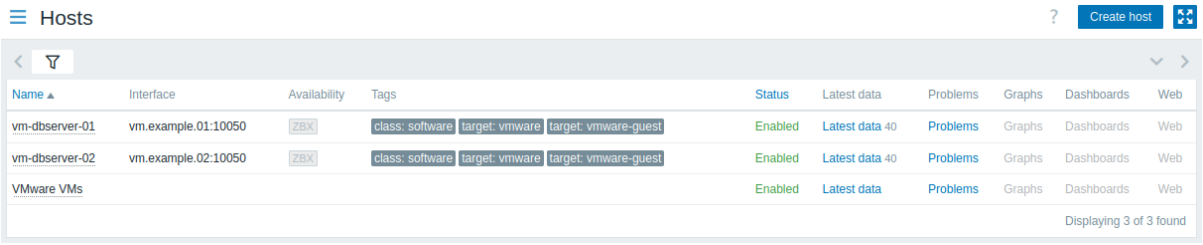
Visualizzare host e metriche

Dopo che il prototipo di host è stato creato, la regola di low-level discovery creerà host per le macchine virtuali VMware individuate e Zabbix inizierà a monitorarle. Si noti che il rilevamento e la creazione degli host possono anche essere eseguiti manualmente, se necessario.

Per visualizzare gli host creati, accedere alla sezione di menu *Data collection → Hosts*.



Per visualizzare le metriche raccolte, accedere alla sezione di menu *Monitoring → Hosts* e fare clic su *Latest data* per uno degli host.



Configurazione avanzata dell'interfaccia host

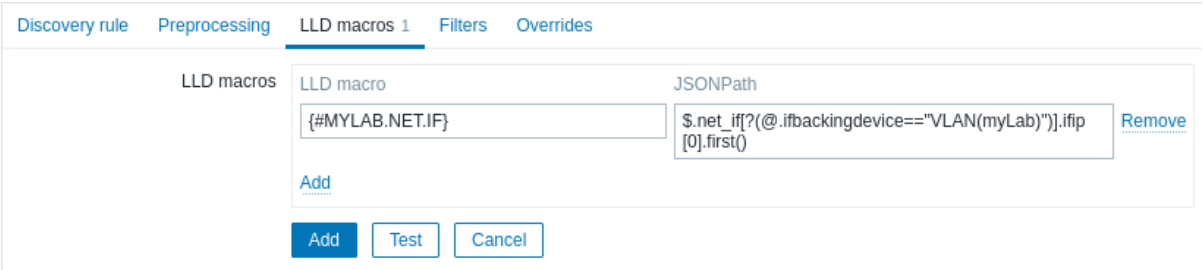
La chiave item `vmware.vm.discovery[{$VMWARE.URL}]`, configurata nella sezione *Creare una regola di low-level discovery*, restituisce i dati delle interfacce di rete nel campo "net_if":

```
"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": []
      }
    ]
```

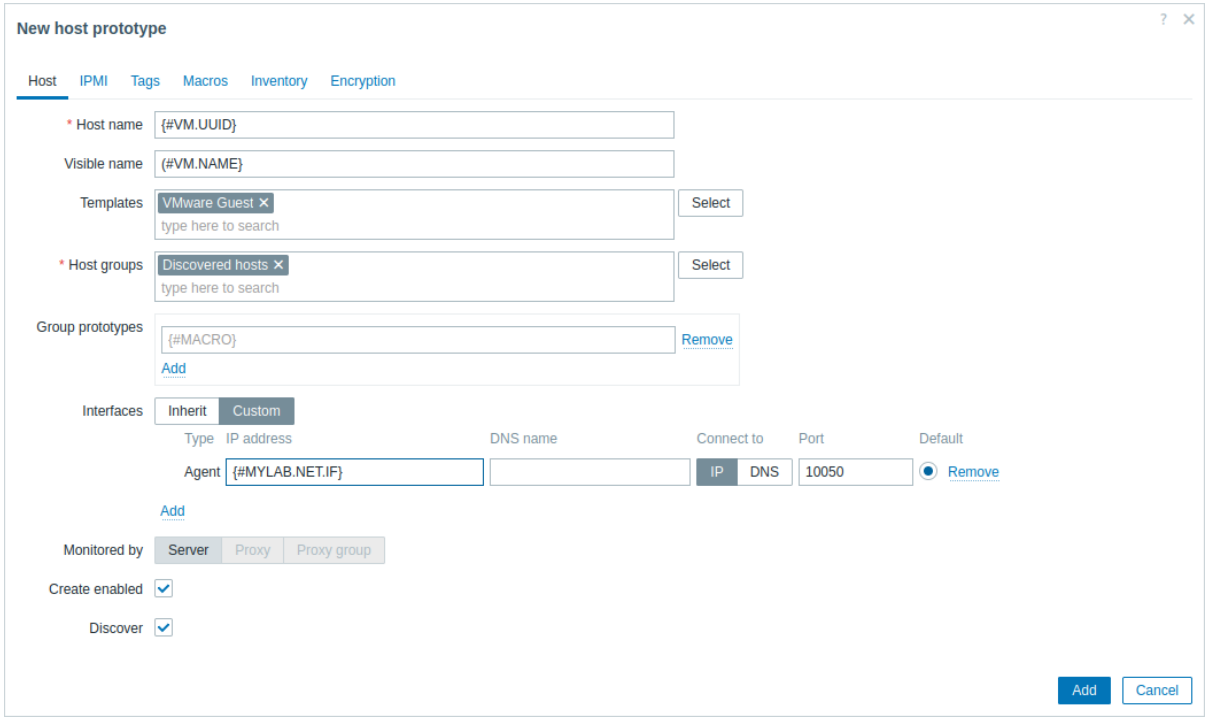
Questi dati possono essere utilizzati per configurare un'interfaccia host personalizzata.

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*.



9 Manutenzione

Panoramica In Zabbix è possibile definire periodi di manutenzione per host e gruppi di host.

Inoltre, è possibile definire la manutenzione solo per un singolo trigger (o un sottoinsieme di trigger) specificando i tag del trigger. In questo caso la manutenzione verrà attivata solo per quei trigger; 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 una manutenzione "con raccolta dati", i trigger vengono elaborati normalmente e gli eventi vengono creati quando necessario. Tuttavia, le escalation dei problemi vengono messe in pausa per host/trigger in manutenzione, se nell'azione è selezionata l'opzione *Pause operations for suppressed problems*. In questo caso, i passaggi di 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.

Ad esempio, se i passaggi di escalation sono pianificati a 0, 30 e 60 minuti dall'inizio di un problema, e c'è una manutenzione di mezz'ora che dura da 10 minuti a 40 minuti dopo il verificarsi di un problema reale, il secondo e il terzo passaggio verranno eseguiti mezz'ora più tardi, cioè a 60 minuti e 90 minuti (a condizione che il problema esista ancora). Allo stesso modo, 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 *Pause operations for suppressed problems* nella configurazione dell'azione.

> **Note:**
> Se almeno un host (utilizzato nell'espressione del trigger) non è in modalità manutenzione, Zabbix invierà una notifica di problema.

Il server Zabbix deve essere in esecuzione durante la manutenzione. Le manutenzioni vengono ricalcolate ogni minuto oppure non appena la cache di configurazione viene ricaricata, se ci sono modifiche al periodo di manutenzione.

I processi timer verificano se lo stato dell'host deve essere cambiato da/in manutenzione allo scoccare di ogni minuto. Inoltre, ogni secondo il processo timer verifica se eventuali manutenzioni devono essere avviate/interrotte in base alla presenza di modifiche nei periodi di manutenzione dopo l'aggiornamento della configurazione. Pertanto, la velocità di avvio/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 *Active since/Active till*. Inoltre, se un host/gruppo di host viene aggiunto 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 avere un impatto sulle prestazioni se ci sono molti problemi aperti. Il server Zabbix leggerà inoltre tutti i problemi aperti all'avvio, anche se in quel momento non sono configurate manutenzioni.

Si noti che il server Zabbix (o proxy) raccoglie sempre i dati indipendentemente dal tipo di manutenzione (inclusa la manutenzione "senza dati"). I dati vengono successivamente ignorati dal server se è impostata l'opzione 'no data collection'.

Quando termina una manutenzione "senza dati", i trigger che utilizzano la funzione nodata() non si attiveranno prima del controllo successivo nel periodo che stanno verificando.

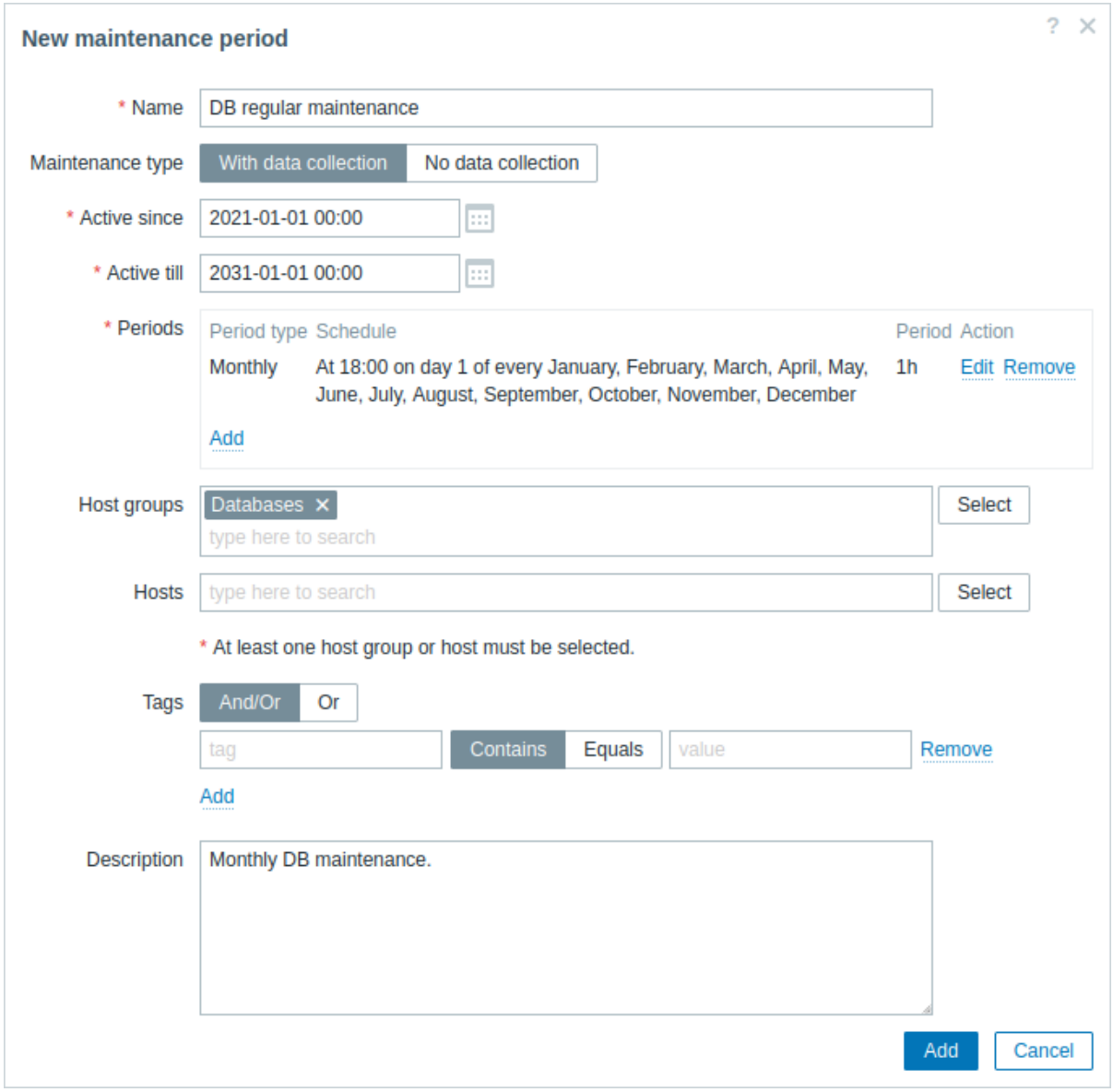
Se un item di log viene aggiunto mentre un host è in manutenzione e la manutenzione termina, verranno raccolte solo le nuove voci del file di log dalla fine della manutenzione.

Se viene inviato un valore con timestamp per un host che si trova in una manutenzione di tipo "senza dati" (ad esempio usando Zabbix sender), tale valore verrà scartato; tuttavia è possibile inviare un valore con timestamp relativo a un periodo di manutenzione scaduto e questo verrà accettato.

Se il periodo di manutenzione, gli host, i gruppi o i tag vengono modificati dall'utente, le modifiche avranno effetto solo dopo la sincronizzazione della cache di configurazione.

Configurazione Per configurare un periodo di manutenzione:

1. Vai a: *Raccolta dati* > *Manutenzione*.
2. Fai clic su *Crea periodo di manutenzione* (oppure sul nome di un periodo di manutenzione esistente).
3. Inserisci i parametri di manutenzione nel modulo.

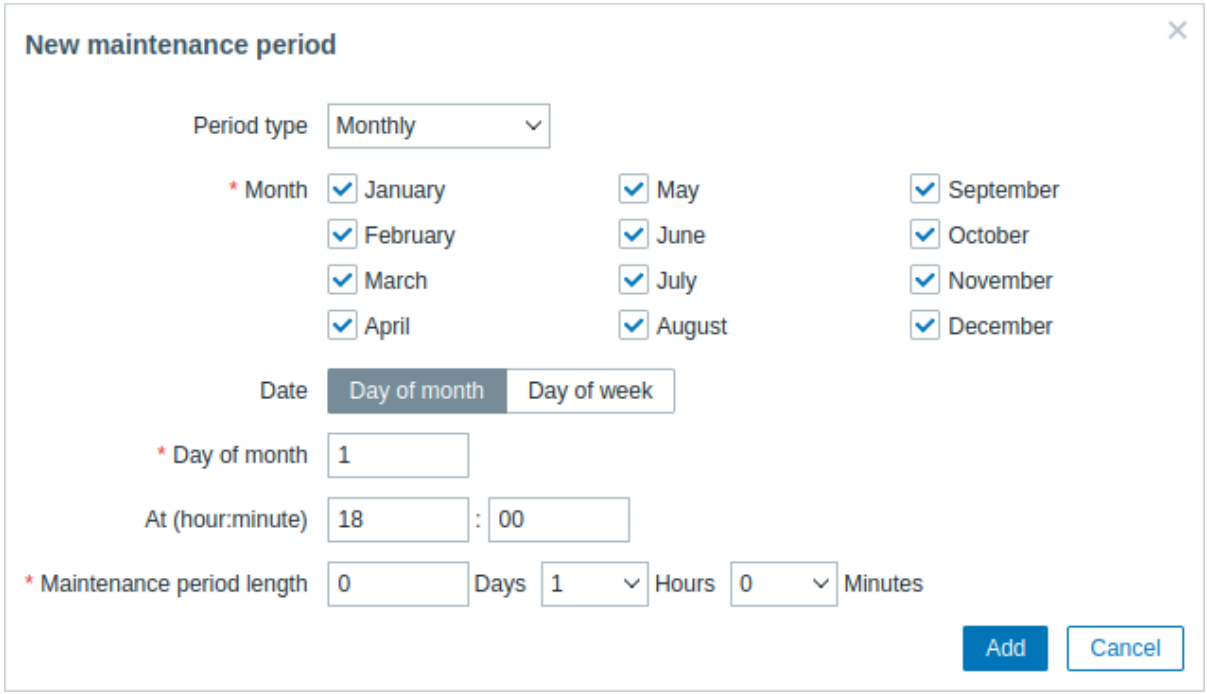


Tutti i campi di input obbligatori sono contrassegnati da un asterisco rosso.

Parametro	Descrizione
Name	Nome del periodo di manutenzione.
Maintenance type	È possibile impostare due tipi di manutenzione: **Con raccolta dati** - i dati verranno raccolti dal server durante la manutenzione, i trigger verranno elaborati; **Senza raccolta dati** - i dati non verranno raccolti dal server durante la manutenzione. Vedi Effetto dei periodi di manutenzione per sapere come ciascun tipo influisce sul report di disponibilità.
Active since	La data e l'ora a partire dalle quali l'esecuzione dei periodi di manutenzione diventa attiva. *Nota:* impostare solo questa ora non attiva un periodo di manutenzione; i periodi di manutenzione devono essere configurati in *Periods* (vedi sotto).
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. Vedi Periodi di manutenzione per una descrizione dettagliata.
Host groups	Seleziona i gruppi di host per i quali verrà attivata la manutenzione. La manutenzione verrà attivata per tutti gli host dei gruppi di host specificati. Questo campo supporta il completamento automatico, quindi iniziando a digitare verrà visualizzato un menu a discesa con tutti i gruppi di host disponibili. Specificare un gruppo di host padre seleziona implicitamente tutti i gruppi di host nidificati. Pertanto, la manutenzione verrà attivata anche sugli host dei gruppi nidificati.
Hosts	Seleziona gli host per i quali verrà attivata la manutenzione. Questo campo supporta il completamento automatico, quindi iniziando a digitare verrà visualizzato un menu a discesa con tutti gli host disponibili.
Tags	Specifica i tag per sopprimere i problemi con tag corrispondenti sugli host in manutenzione. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili due operatori per ciascuna condizione: **Contiene** - include i nomi di tag specificati in cui i valori dei tag contengono la stringa inserita (corrispondenza per sottostringa, distinzione tra maiuscole e minuscole); **Uguale a** - include i nomi e i valori dei tag specificati (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. I tag possono essere specificati solo se è selezionata la modalità *Con raccolta dati*.
Description	Descrizione del periodo di manutenzione.

Periodi di manutenzione

La finestra del periodo di manutenzione serve per pianificare il tempo per 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.
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 iniziale è il giorno in cui ricade l'ora *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 avere luogo 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 iniziale è la settimana in cui ricade l'ora *Attivo da*. Per esempi, vedere la descrizione del parametro *Giornaliero* sopra.
Mensile	Configura un periodo di manutenzione mensile: *Mese* - selezionare tutti i mesi durante i quali viene eseguita la manutenzione regolare; *Data:* ***Giorno del mese*** - selezionare questa opzione se la manutenzione deve avere luogo nella stessa data di ogni mese (ad esempio, ogni 1° giorno del mese), quindi selezionare il giorno richiesto nel campo *Giorno del mese* che appare; *Data:* ***Giorno della settimana*** - selezionare questa opzione se la manutenzione deve avere luogo solo in determinati giorni (ad esempio, ogni primo lunedì del mese), quindi selezionare (nell'elenco a discesa) la settimana del mese richiesta (prima, seconda, terza, quarta o ultima), quindi contrassegnare 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 (*Giornalieri*, *Settimanali*, *Mensili*), 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.

Una volta fatto, premere *Aggiungi* per aggiungere il periodo di manutenzione al blocco *Periodi*.

Si noti che i cambiamenti dell'ora legale (DST) non influiscono sulla durata della manutenzione. Ad esempio, supponiamo di avere una manutenzione di due ore configurata che di solito inizia alle 01:00 e termina alle 03:00:

- se dopo un'ora di manutenzione (alle 02:00) avviene un cambio dell'ora legale e l'ora corrente passa dalle 02:00 alle 03:00, la manutenzione continuerà per un'altra ora (fino alle 04:00);
- se dopo due ore di manutenzione (alle 03:00) avviene un cambio dell'ora legale e l'ora corrente passa dalle 03:00 alle 02:00, la manutenzione si interromperà, perché sono trascorse due ore;
- se un periodo di manutenzione inizia durante l'ora che viene saltata a causa di un cambio dell'ora legale, la manutenzione non verrà avviata.

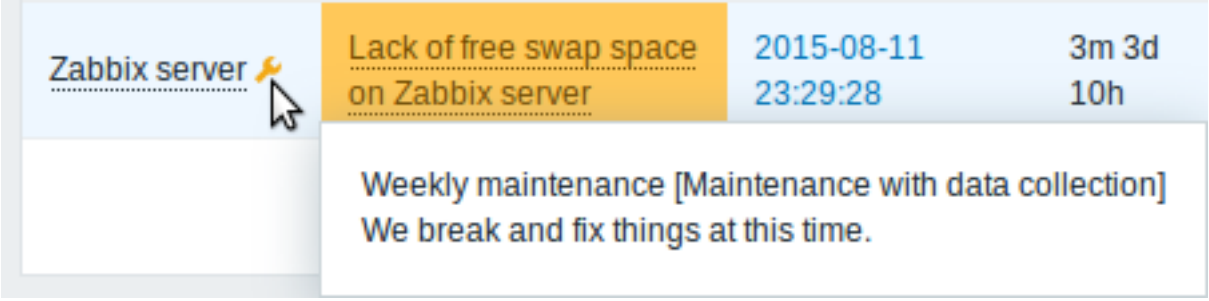
Se un periodo di manutenzione è impostato su "1 giorno" (il periodo effettivo della manutenzione è di 24 ore, poiché Zabbix calcola i giorni in ore), inizia alle 00:00 e termina alle 00:00 del giorno successivo:

- la manutenzione terminerà alle 01:00 del giorno successivo se l'ora corrente viene spostata avanti di un'ora;
- la manutenzione terminerà alle 23:00 di quel giorno se l'ora corrente viene spostata 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:

- *Dashboard*
- *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.

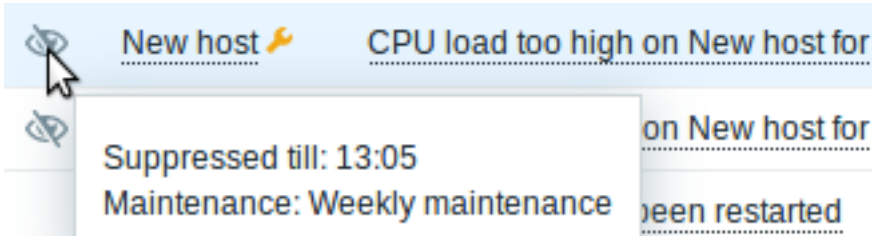
Inoltre, gli host in manutenzione hanno uno sfondo arancione in *Monitoring* > *Maps*.

Visualizzazione dei problemi soppressi

Normalmente i problemi per gli host in manutenzione sono soppressi, cioè non vengono visualizzati nel frontend. Tuttavia, è anche possibile configurare la visualizzazione dei problemi soppressi selezionando l'opzione *Mostra problemi soppressi* nelle seguenti 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: . Passando il mouse sopra l'icona vengono visualizzati ulteriori dettagli.



Calcolo delle code durante la manutenzione Le code visualizzate nel frontend di Zabbix (Administration > Queue) sono calcolate dal server Zabbix. Non includono gli item in manutenzione senza raccolta dati: per questi item la lunghezza della coda è sempre zero, anche quando i loro valori sono in ritardo. Gli item in ritardo in manutenzione con raccolta dati vengono comunque conteggiati nella coda.

Il proxy Zabbix non è a conoscenza dei periodi di manutenzione perché non esiste sincronizzazione della configurazione di manutenzione tra il server Zabbix e il proxy. I controlli interni calcolati sui proxy Zabbix (ad esempio, `zabbix[queue,,]` e `zabbix[stats,,,queue,,]`) segnalano gli item in ritardo indipendentemente dallo stato di manutenzione sul server Zabbix.

Di conseguenza, il frontend di Zabbix e i controlli interni sui proxy Zabbix possono riportare lunghezze della coda diverse per gli stessi item in manutenzione senza raccolta dati.

10 Espressioni regolari

Panoramica Espressioni regolari compatibili con Perl (PCRE, PCRE2) sono supportati in Zabbix.

Esistono due modi per utilizzare le espressioni regolari in Zabbix:

- inserendo manualmente un'espressione regolare
- utilizzando un'espressione regolare globale creata in Zabbix

Espressioni regolari È possibile inserire manualmente un'espressione regolare nei punti in cui è supportata. Si noti che l'espressione non può iniziare con @ perché quel simbolo viene utilizzato in Zabbix per fare riferimento alle espressioni regolari globali.

> **Warning:**
> È possibile esaurire lo stack durante l'uso delle espressioni regolari. Per ulteriori informazioni, vedere la pagina man di pcrestack.

Si noti che nella corrispondenza multilinea, gli ancoraggi ^ e $ corrispondono rispettivamente all'inizio/fine di ogni riga, anziché all'inizio/fine dell'intera stringa.

Vedere anche gli esempi di escaping corretto in vari contesti.

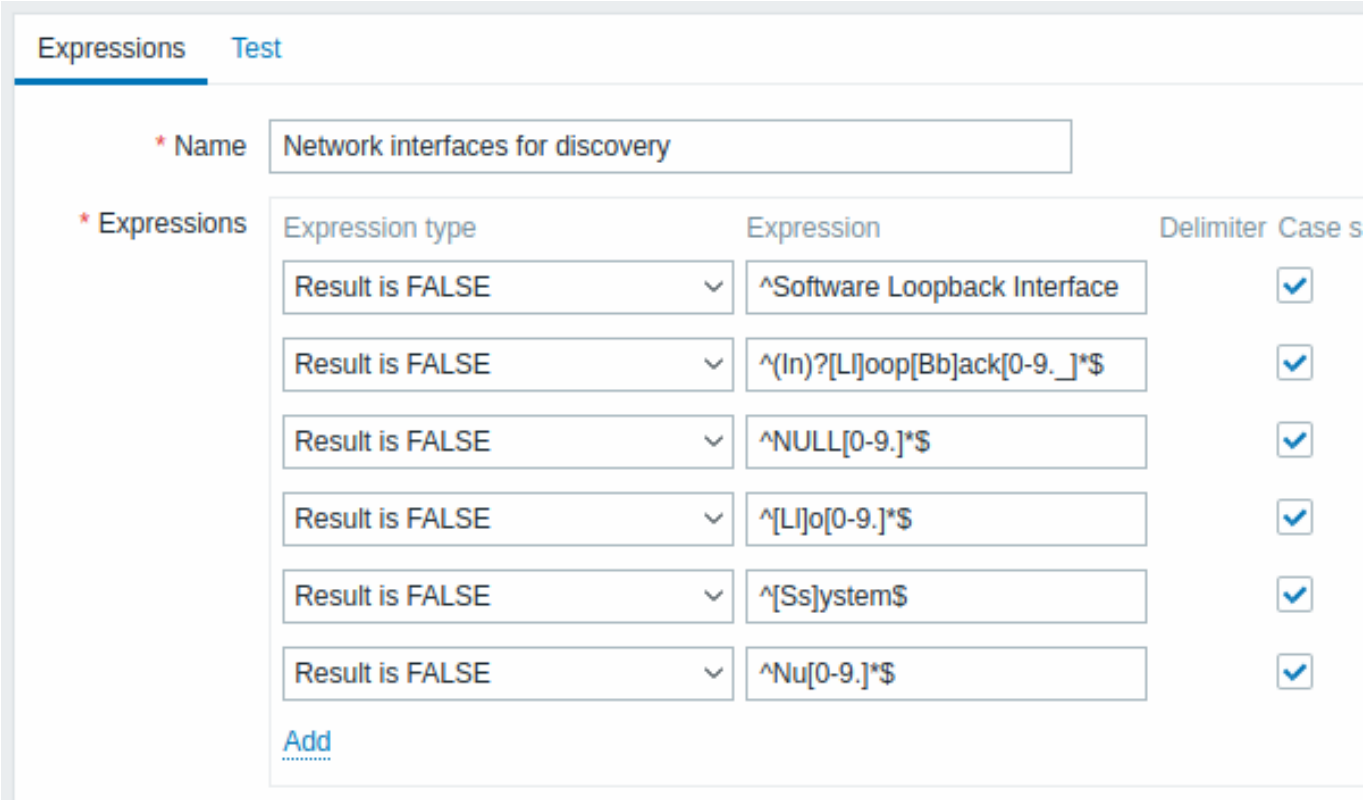
Espressioni regolari globali C'è un editor avanzato per la creazione e il test delle espressioni regolari complesse nel frontend di Zabbix.

Una volta che un'espressione regolare è stata creata in questo modo, può essere utilizzata in diversi posti nel frontend facendo riferimento al suo nome, preceduto da @, ad esempio, *@mycustomregexp*.

Per creare un'espressione regolare globale:

- Vai a: *Amministrazione → Generale*
- Seleziona *Espressioni regolari* dal menu a discesa
- Clicca su *Nuova espressione regolare*

La scheda **Espressioni** consente di impostare il nome dell'espressione regolare e aggiungere sottoespressioni.



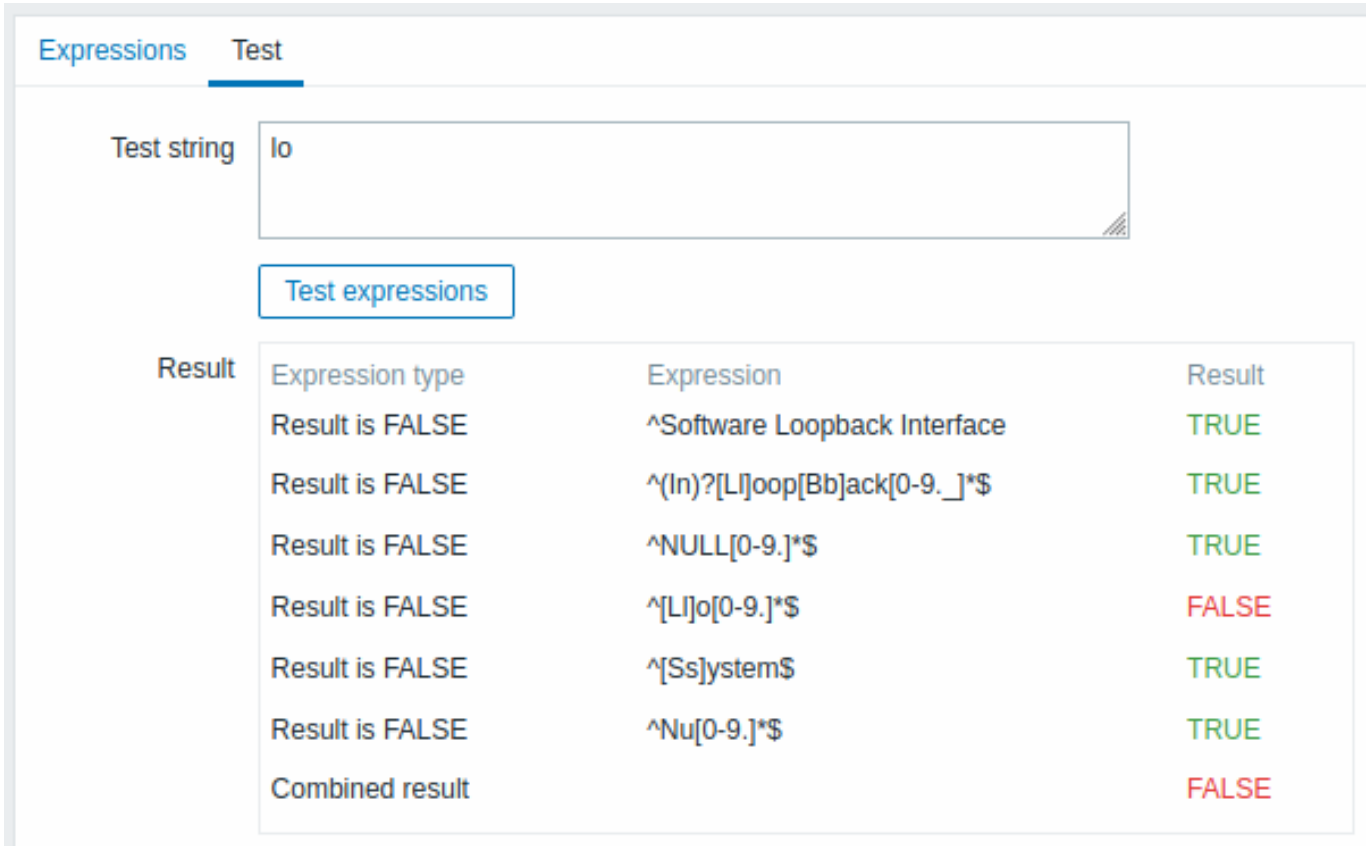
Tutti i campi obbligatori sono contrassegnati da un asterisco rosso.

|Parametro|<|Descrizione| |-|----------|---------- ----| |*Name*|<|Imposta il nome dell'espressione regolare. Sono consentiti tutti i caratteri Unicode.| |*Espressioni*|<|Clicca su *Aggiungi* nel blocco Espressioni per aggiungere una nuova sottoespressione.| | |*Tipo di espressione*|Seleziona il tipo di espressione:
Stringa di caratteri inclusa - corrisponde alla sottostringa
Qualsiasi stringa di caratteri inclusa - corrisponde a qualsiasi sottostringa da un elenco delimitato. L'elenco delimitato include una virgola (,), un punto (.) o una barra (/).
Stringa di caratteri non inclusa: corrisponde a qualsiasi stringa tranne la sottostringa
**Il risultato è VERO* * - corrisponde all'espressione regolare
Il risultato è FALSO - non corrisponde all'espressione regolare| |^|*Espressione*|Inserisci sottostringa/espressione regolare.| |*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 *"Qualsiasi stringa di caratteri inclusa"*.| |*Case sensitive*|<|Una casella di controllo per specificare se un'espressione regolare è sensibile all'uso delle lettere maiuscole.|

Una barra (/) nell'espressione viene trattata letteralmente, piuttosto che un delimitatore. In questo modo è possibile salvare espressioni contenenti a barra, senza errori.

::: nota importante Un nome di espressione regolare personalizzato in Zabbix potrebbe contenere virgole, spazi, ecc. Nei casi in cui ciò può portare a interpretazione errata quando si fa riferimento (ad esempio, una virgola nel file parametro di una chiave di elemento) l'intero riferimento può essere messo tra virgolette come this: *"@My custom regexp for purpose1, purpose2"*.
I nomi delle espressioni regolari non devono essere racchiusi tra virgolette in altre posizioni (ad es esempio, nelle proprietà della regola LLD). :::

Nella scheda **Test** l'espressione regolare e le sue sottoespressioni possono essere testato fornendo una stringa di prova.



I risultati mostrano lo stato di ciascuna sottoespressione e il totale personalizzato stato di espressione.

Lo stato dell'espressione personalizzata totale è definito come *Risultato combinato*. Se diverse sottoespressioni sono definite Zabbix usa l'operatore logico AND to calcolare *Risultato combinato*. Significa che se almeno un Risultato è Anche False *Risultato combinato* ha lo stato False.

Espressioni regolari globali predefinite Zabbix include diverse espressioni regolari globali nel suo set di dati predefinito.

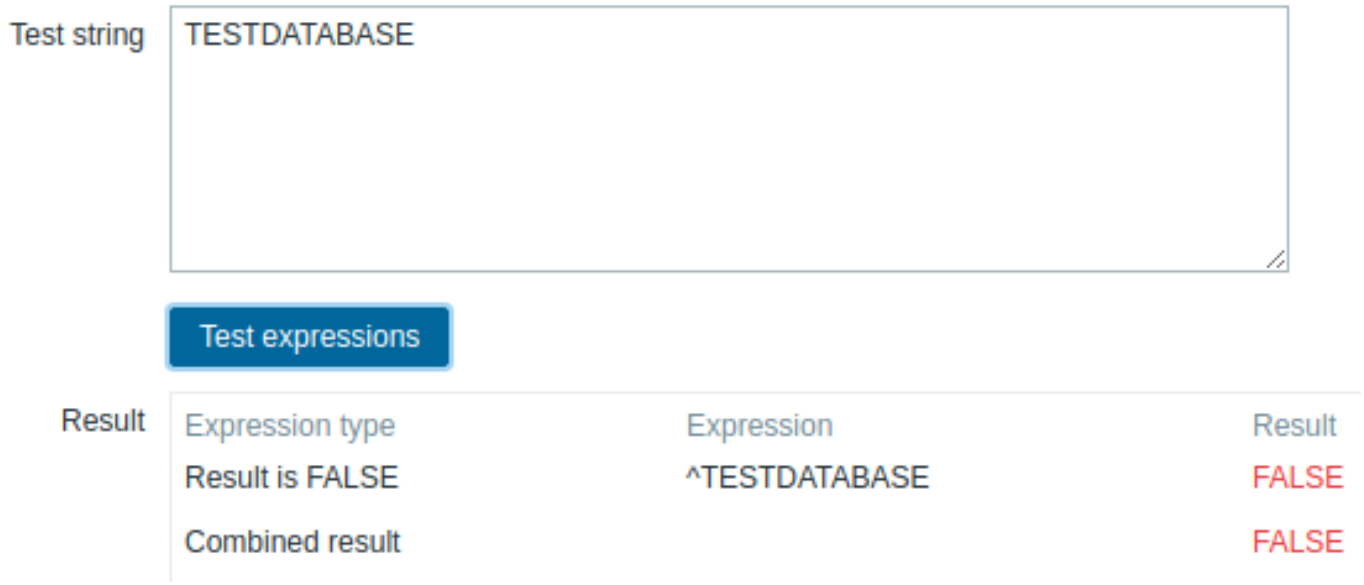
Name	Expression	Matches
File systems for discovery	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)`	"btrfs" oppure "ext2" oppure "ext3" oppure "ext4" oppure "jfs" oppure "reiser" oppure "xfs" oppure "ffs" oppure "ufs" oppure "jfs" oppure "jfs2" oppure "vxfs" oppure "hfs" oppure "refs" oppure "apfs" oppure "ntfs" oppure "fat32" oppure "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", poi contengono "L" oppure "l", quindi "oop", poi "B" oppure "b", quindi "ack", che può essere seguito facoltativamente da un numero qualsiasi di cifre, punti o caratteri di sottolineatura.
	`^NULL[0-9.]*$`	Stringhe che iniziano con "NULL" seguite facoltativamente da un numero qualsiasi di cifre o punti.
	`^[Ll]o[0-9.]*$`	Stringhe che iniziano con "Lo" oppure "lo" e sono seguite facoltativamente da un numero qualsiasi di cifre o punti.
	`^[Ss]ystem$`	"System" oppure "system"
	`^Nu[0-9.]*$`	Stringhe che iniziano con "Nu" seguite facoltativamente da un numero qualsiasi di cifre o punti.

Name	Expression	Matches
Storage devices for SNMP discovery	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory” oppure ”Virtual memory” oppure ”Memory buffers” oppure ”Cached memory” oppure ”Swap space”
Windows service names for discovery	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS” oppure ”gupdate” oppure ”SysmonLog” oppure stringhe come ”clr_optimization_v2.0.50727_32” e ”clr_optimization_v4.0.30319_32” dove al posto dei punti può essere inserito qualsiasi carattere tranne l’a-capo.
Windows service startup states for discovery	`^(automatic|automatic delayed)$`	”automatic” oppure ”automatic delayed”

Esempi Esempio 1

Uso della seguente espressione nella scoperta di basso livello da scoprire database tranne un database con un nome specifico:

`^DATABASEDIPROVA$`

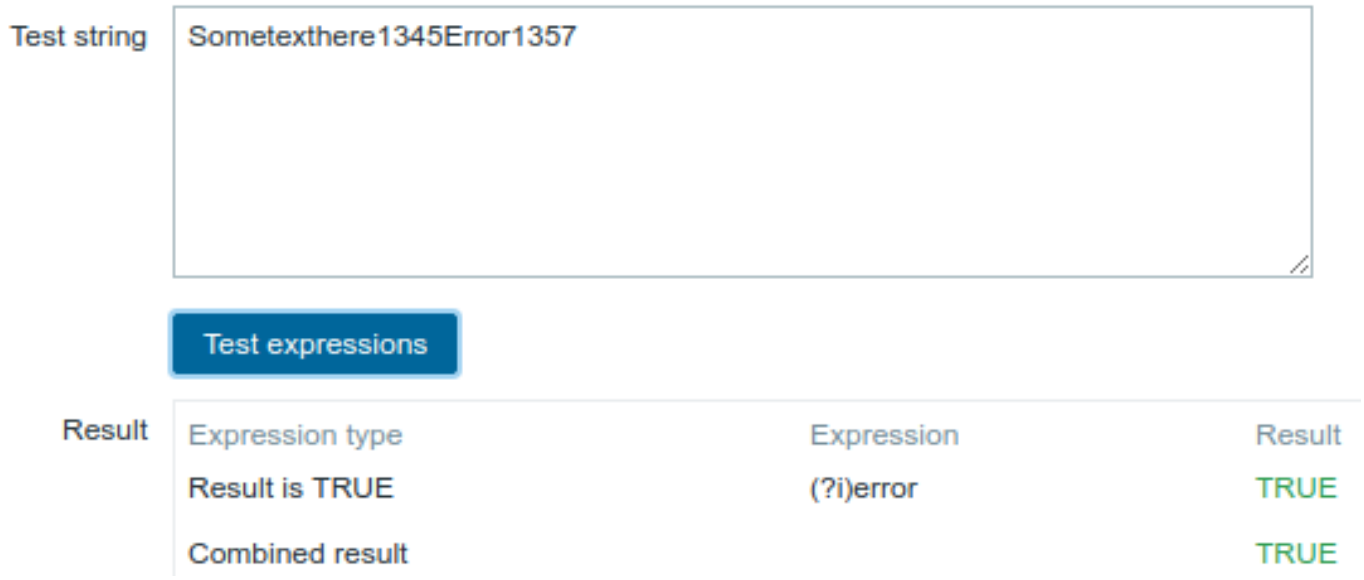


Tipo di espressione scelto: ”Il risultato è FALSO”. non corrisponde al nome, contenente la stringa ”*TESTDATABASE*”.

Esempio con un modificatore regex inline

Uso della seguente espressione regolare incluso un modificatore inline (?i) per abbinare i caratteri ”errore”:

`(?i)errore`



Tipo di espressione scelto: ”Il risultato è VERO”. I caratteri ”errore” sono abbinato.

Un altro esempio con un modificatore regex inline

Uso della seguente espressione regolare incluso multiple inline modificatori per abbinare i caratteri dopo una riga specifica:

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# aggiungiamo il modificatore s per consentire . co
```



Tipo di espressione scelto: ”Il risultato è VERO”. Caratteri dopo uno specifico linea sono abbinati.

::: nota importante Il modificatore **g** non può essere specificato nella riga. Il l'elenco dei modificatori disponibili può essere trovato in pcresyntax man page. Per ulteriori informazioni sulla sintassi PCRE si prega di fare riferimento a PCRE HTML documentazione.

:::

Supporto delle espressioni regolari per posizione

Posizione	Espressione regolare	Espressione regolare globale	Corrispondenza multilinea	Commenti
Item				
agent				
eventlog[]	Sì	Sì	Sì	Parametri `regexp`, `severity`, `source`, `eventid`
eventlog.count[]				Parametri `regexp`, `severity`, `source`, `eventid`
log[]				Parametro `regexp`
log.count[]				
logrt[]		Sì/No		Il parametro `regexp` supporta entrambi, il parametro `file_regexp` supporta solo espressioni non globali
logrt.count[]				
proc.cpu.util[]		No	No	Parametro `cmdline`
proc.get[]				
proc.mem[]				
proc.num[]				
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[]				
Trap				
SNMP				

Posizione	Espressione regolare	Espressione regolare globale	Corrispondenza multilinea	Commenti
snmptrap[]	Sì	Sì	No	Parametro `regexp`
Preelaborazione del valore dell'item	Sì	No	No	Parametro `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 di *Operazione*
Condizioni azione	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 validazione input*
Monitoraggio web	Sì	No	Sì	*Variabili* con prefisso **regex:** Campo *Stringa richiesta*
Contesto macro utente	Sì	No	No	Nel contesto macro con prefisso **regex:**
Funzioni macro				
regsub()	Sì	No	No	Parametro `pattern`
iregsub()				
Indicatori di collegamento nelle mappe	Sì	No	No	Campo *Pattern* (per item di testo)
Mappatura icone	Sì	Sì	No	Campo *Espressione*
Mappatura valori	Sì	No	No	Campo *Valore* se il tipo di mappatura è `regexp`

11 Riconoscimento del problema

Panoramica Gli eventi di problema in Zabbix possono essere confermati dagli utenti.

Se un utente riceve una notifica relativa a un evento di problema, può accedere al frontend di Zabbix, aprire la finestra popup di aggiornamento del problema relativa a quel problema utilizzando uno dei modi elencati di seguito e confermare il problema. Durante la conferma, può inserire un commento, indicando che ci sta lavorando o qualsiasi altra informazione ritenga opportuno aggiungere.

In questo modo, se un altro utente del sistema nota lo stesso problema, vede immediatamente se è già stato confermato e quali commenti sono stati inseriti fino a quel momento.

In questo modo, il flusso di lavoro per la risoluzione dei problemi da parte di più utenti del sistema può svolgersi in modo coordinato.

Lo stato di conferma viene utilizzato anche nella definizione delle operazioni delle action. È possibile definire, ad esempio, che una notifica venga inviata a un responsabile di livello superiore solo se un evento non viene confermato per un certo periodo di tempo.

Per confermare gli eventi e commentarli, un utente deve disporre almeno dei permessi di lettura per i trigger corrispondenti. Per modificare la gravità del problema o chiudere un problema, un utente deve disporre dei permessi di lettura-scrittura per i trigger corrispondenti.

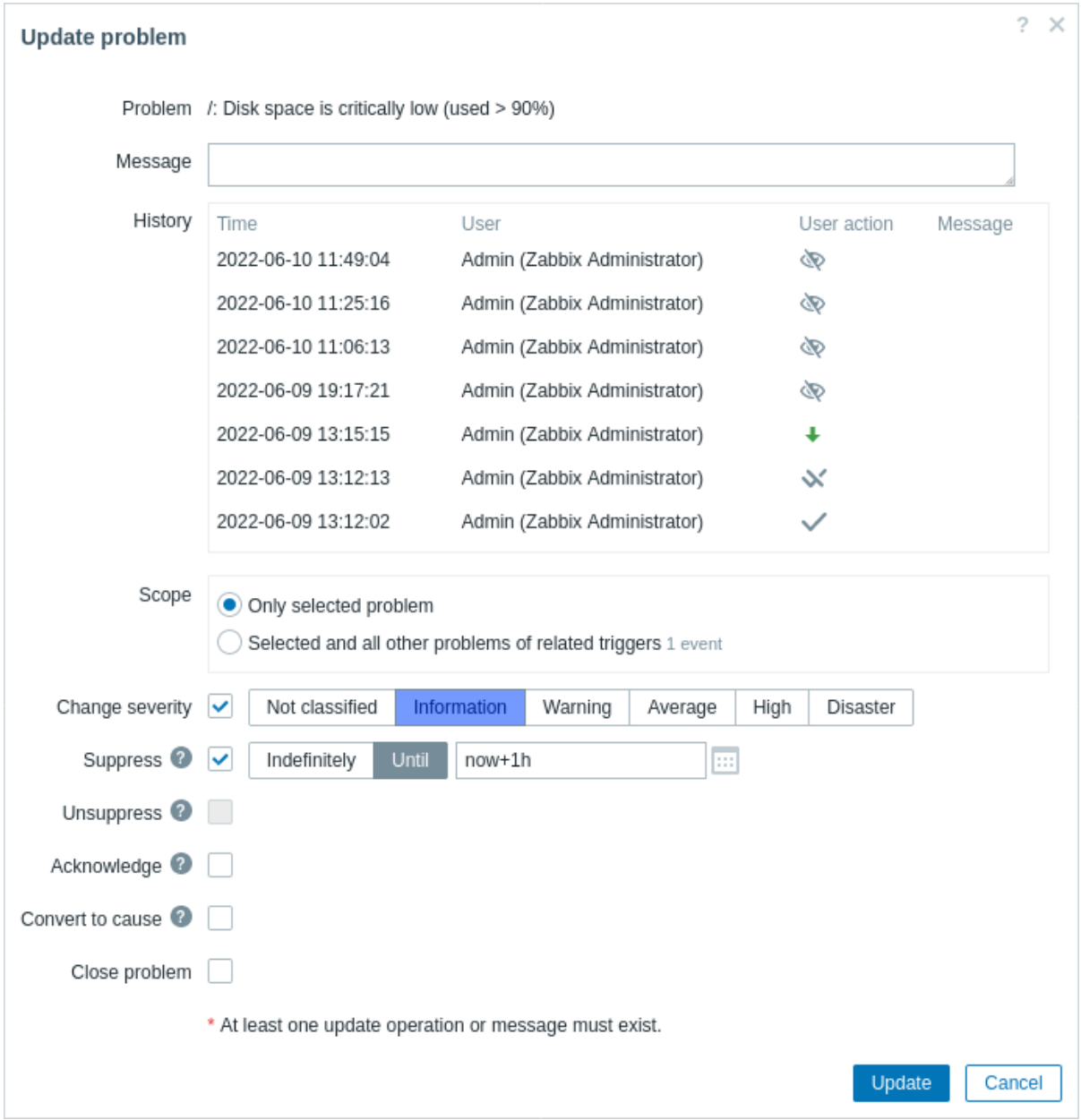
Esistono **diversi** modi per accedere alla finestra popup di aggiornamento del problema, che consente di confermare un problema.

- 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 problema non risolto in:
  - *Dashboards* (widget *Trigger overview*)

Il menu popup contiene un'opzione *Update* che consente di aprire la finestra di aggiornamento del problema.

Aggiornamento dei problemi La finestra popup di aggiornamento del problema consente di:

- commentare il problema
- visualizzare i commenti e le azioni effettuate finora
- modificare la gravità del problema
- sopprimere/rimuovere la soppressione del problema
- riconoscere/rimuovere il riconoscimento del problema
- cambiare un problema sintomo in problema causa
- chiudere manualmente il problema



Tutti i campi di input obbligatori sono contrassegnati con 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 per commentare il problema (massimo 2048 caratteri).
Cronologia	Vengono elencate le attività precedenti e i commenti sul problema, insieme all'ora e ai dettagli dell'utente. Per il significato delle icone utilizzate per indicare le azioni dell'utente, vedere la pagina dettagli evento. Si noti che la cronologia viene visualizzata solo se per l'aggiornamento è selezionato un solo problema.
Ambito	Definire l'ambito di azioni quali la modifica della gravità, il riconoscimento o la chiusura manuale dei problemi: **Solo il problema selezionato** - influirà solo su questo evento **Selezionato e tutti gli altri problemi dei trigger correlati** - in caso di riconoscimento/chiusura del problema, influirà su questo evento e su tutti gli altri problemi che non sono ancora stati riconosciuti/chiusi. Se l'ambito contiene problemi già riconosciuti o chiusi, questi problemi non verranno riconosciuti/chiusi nuovamente. D'altra parte, il numero di operazioni di messaggio e di modifica 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 modificare la gravità è disponibile se esistono permessi di lettura-scrittura per almeno uno dei problemi selezionati. Facendo clic su *Aggiorna* verranno aggiornati solo i problemi per i quali sono disponibili permessi di lettura-scrittura. Se non esistono permessi 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 indefinitamente **Fino a** - sopprime fino a un determinato momento. Sono supportati entrambi i formati di ora assoluta e relativa, ad esempio: `now+1d` - per un giorno a partire da adesso (predefinito) `now/w` - fino alla fine della settimana corrente `2022-05-28 12:00:00` - fino a data/ora assoluta Si noti che un periodo semplice (ad es. `1d`, `1w`) non è supportato. La disponibilità di questa opzione dipende dalle impostazioni del ruolo utente "Suppress problems". 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 "Suppress problems".
Riconosci	Selezionare la casella di controllo per riconoscere il problema. Questa casella di controllo è disponibile se tra quelli selezionati è presente almeno un problema non riconosciuto. Non è possibile aggiungere un altro riconoscimento a un problema già riconosciuto (è comunque possibile aggiungere un altro commento).
Rimuovi riconoscimento	Selezionare la casella di controllo per rimuovere il riconoscimento del problema. Questa casella di controllo è disponibile se tra quelli selezionati è presente almeno un problema riconosciuto.
Converti in causa	Selezionare la casella di controllo per convertire il/i problema/i sintomo in problema/i causa.
Chiudi problema	Selezionare la casella di controllo per chiudere manualmente il/i problema/i selezionato/i. La casella di controllo per la chiusura di un problema è disponibile se l'opzione *Allow manual close* è selezionata nella configurazione del trigger per almeno uno dei problemi selezionati. Facendo clic su *Aggiorna* verranno chiusi solo i problemi per i quali è consentita la chiusura. Se nessun problema può essere chiuso manualmente, la casella di controllo è disabilitata. I problemi già chiusi non verranno chiusi nuovamente.

Visualizzazione In base alle informazioni di conferma è possibile configurare come il conteggio dei problemi viene visualizzato nella dashboard o nelle mappe. Per farlo, è necessario effettuare delle selezioni nell'opzione *Visualizzazione dei problemi*, disponibile sia nella configurazione delle mappe sia nel widget Problemi per gravità della dashboard. È possibile visualizzare il conteggio di tutti i problemi, il conteggio dei problemi non confermati separato dal totale oppure solo il conteggio dei problemi non confermati.

In base alle informazioni di aggiornamento del problema (conferma, ecc.), è possibile configurare operazioni di aggiornamento: inviare un messaggio o eseguire comandi remoti.

1 Soppressione del problema

Panoramica

La soppressione dei problemi offre un modo per nascondere temporaneamente un problema che può essere gestito in seguito. Questo è utile per ripulire l'elenco dei problemi in modo da dare la massima priorità alle questioni più urgenti. Ad esempio, a volte può verificarsi un problema durante il fine settimana che non è abbastanza urgente da essere affrontato immediatamente, quindi può essere "posticipato" fino a lunedì mattina.

La soppressione dei problemi consente di nascondere un *singolo* problema, a differenza della soppressione dei problemi tramite la manutenzione del host, quando tutti i problemi del host in manutenzione vengono nascosti.

Le operazioni per le azioni dei trigger saranno sospese per i problemi soppressi nello stesso modo in cui avviene con la manutenzione del host.

Configurazione

Un problema può essere eliminato tramite **aggiornamento problema** finestra, dove la soppressione è una delle opzioni di aggiornamento del problema insieme a commentare, modificare la gravità, riconoscere, ecc.

Un problema può anche essere non soppresso attraverso lo stesso problema finestra di aggiornamento.

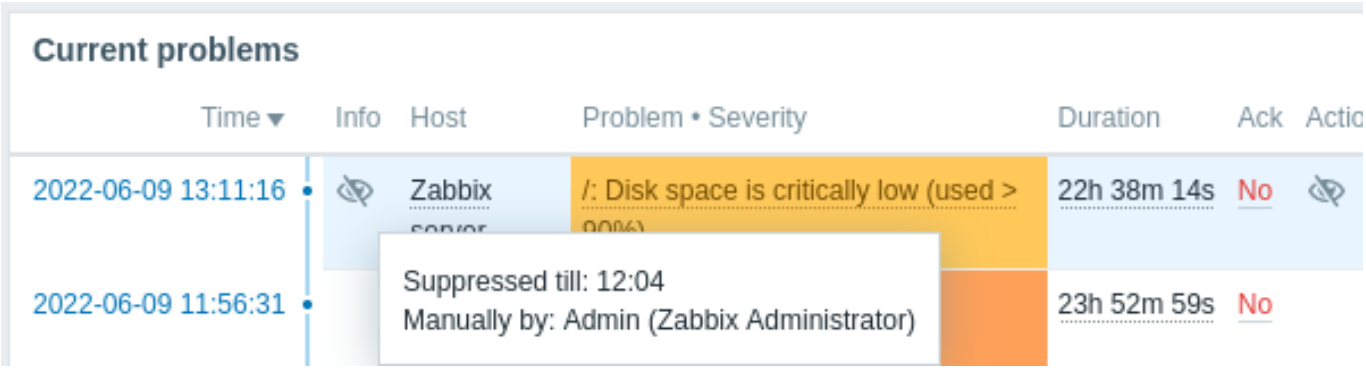
Visualizzazione

Una volta soppressa, il problema viene contrassegnato da un'icona di soppressione lampeggiante nella colonna *Info*, prima di essere nascosto.

L'icona di soppressione lampeggia mentre l'attività di soppressione è nella lista di attesa. Una volta che il gestore delle attività ha soppresso il problema, l'icona smetterà 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 non è attivo e il gestore delle attività non può completare l'operazione. La stessa logica si applica alla rimozione della soppressione. Nel breve periodo successivo all'invio dell'attività, quando il server non l'ha ancora completata, l'icona di rimozione della soppressione lampeggia.

Un problema soppresso può essere sia nascosto che mostrato, a seconda delle impostazioni del filtro problemi/widget.

Quando viene mostrato nell'elenco dei problemi, un problema soppresso è contrassegnato dall'icona di soppressione e i dettagli della soppressione vengono visualizzati al passaggio del mouse:



I dettagli della soppressione vengono visualizzati anche in una finestra popup posizionando il mouse sull'icona di soppressione nella colonna *Actions*.

12 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 in Zabbix Community templates. In seguito, altri utenti possono scaricare il template e importare il file in Zabbix.
- integrazione 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:

- Gruppi di host (*solo tramite Zabbix API*)
- Gruppi di template (*solo tramite Zabbix API*)
- Template
- Host
- Mappe di rete
- Tipi di supporto
- Immagini

Formati di sportazione

I dati possono essere esportati utilizzando il frontend web Zabbix o Zabbix API. i Formati di Esportazione supportati sono YAML, XML and 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 individuazione) ereditate dai template collegati non vengono esportate. Eventuali modifiche apportate a tali entità a livello di host (come l'intervallo dell'item modificato, l'espressione regolare modificata o i prototipi aggiunti alla regola di low-level discovery) 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 low-level discovery 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 per tali entità sono configurati timeout propri.

Dettagli sull'importazione

- L'importazione si interrompe al primo errore.
- Durante l'aggiornamento di 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 utilizzando l'opzione "Delete missing", le macro di host/template non presenti nel file di importazione verranno eliminate da
  host/template dopo l'importazione.
- I tag vuoti per item, trigger, graph, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes non hanno significato, cioè è come se fossero assenti.
- Se le entità dell'host/template importato hanno timeout propri configurati, questi 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 una corretta estensione del file: .yaml e .yml per YAML, .xml per XML e .json per JSON. Vedere le informazioni sulla compatibilità relative alle versioni XML supportate.
- L'importazione supporta file di configurazione solo con codifica UTF-8 (con o senza BOM); altre codifiche (UTF16LE, UTF16BE, UTF32LE, UTF32BE, ecc.) comporteranno un errore di conversione durante l'importazione.

Formato YAML di base Il formato di esportazione YAML contiene i seguenti nodi:

- Nodo radice per l'esportazione YAML di Zabbix
- Versione di esportazione

```yaml
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
<?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 dell'esportazione

```json
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Gli altri oggetti dipendono dagli oggetti esportati.

1 Gruppi di template

Panoramica

Nel frontend, i gruppi di template possono essere esportati solo con l'esportazione di template. Quando un template viene esportato, tutti i gruppi a cui appartiene vengono esportati automaticamente con esso.

L'API consente di esportare i gruppi di template indipendentemente dai template.

Formato di esportazione

```
template_groups:
- uuid: 36bff6c29af64692839d077febfc7079
nome: 'Network devices'
```

Elementi esportati

Elemento	Tipo	Descrizione
uuid	stringa	Identificatore univoco per questo gruppo di template.
name	stringa	Nome del gruppo.

2 Gruppi di host

Panoramica

Nel frontend, i gruppi di host possono essere esportati solo con l'esportazione host. Quando un host viene esportato, tutti i gruppi a cui appartiene vengono esportati automaticamente con esso.

L'API consente di esportare i gruppi di host indipendentemente dagli host.

Formato di esportazione

```
host_groups:
- uuid: 6f6799aa69e844b4b3918f779f2abf08
nome: 'Zabbix servers'
```

Elementi esportati

Elemento	Tipo	Descrizione
uuid	stringa	Identificatore univoco per questo gruppo di host.
name	stringa	Nome del gruppo.

3 Template

Panoramica

I template vengono esportati con molti oggetti correlati e relazioni tra oggetti.

L'esportazione di un template contiene:

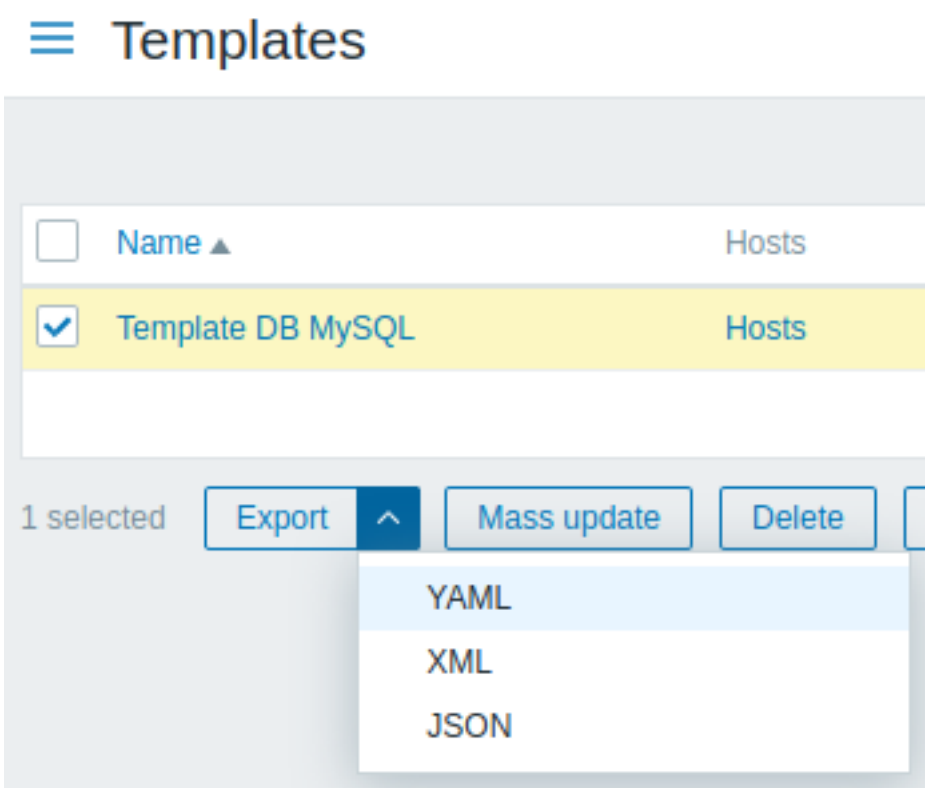
- Gruppi di template correlati
- Gruppi di host correlati (se utilizzati nella configurazione del prototipo host)
- Dati del template
- Collegamento ad altri template
- Collegamento a gruppi di template
- Elementi direttamente correlati
- Trigger direttamente correlati
- Grafici direttamente correlati
- Dashboard direttamente correlate
- Regole di discovery direttamente correlate con tutti i prototipi
- Scenari web direttamente correlati

- Mappe di valori

Esportazione

Per esportare i template, procedi come segue:

1. Vai su *Raccolta dati* → *Template*.
2. Seleziona le caselle di controllo dei template da esportare.
3. Fai clic su *Esporta* sotto l'elenco.



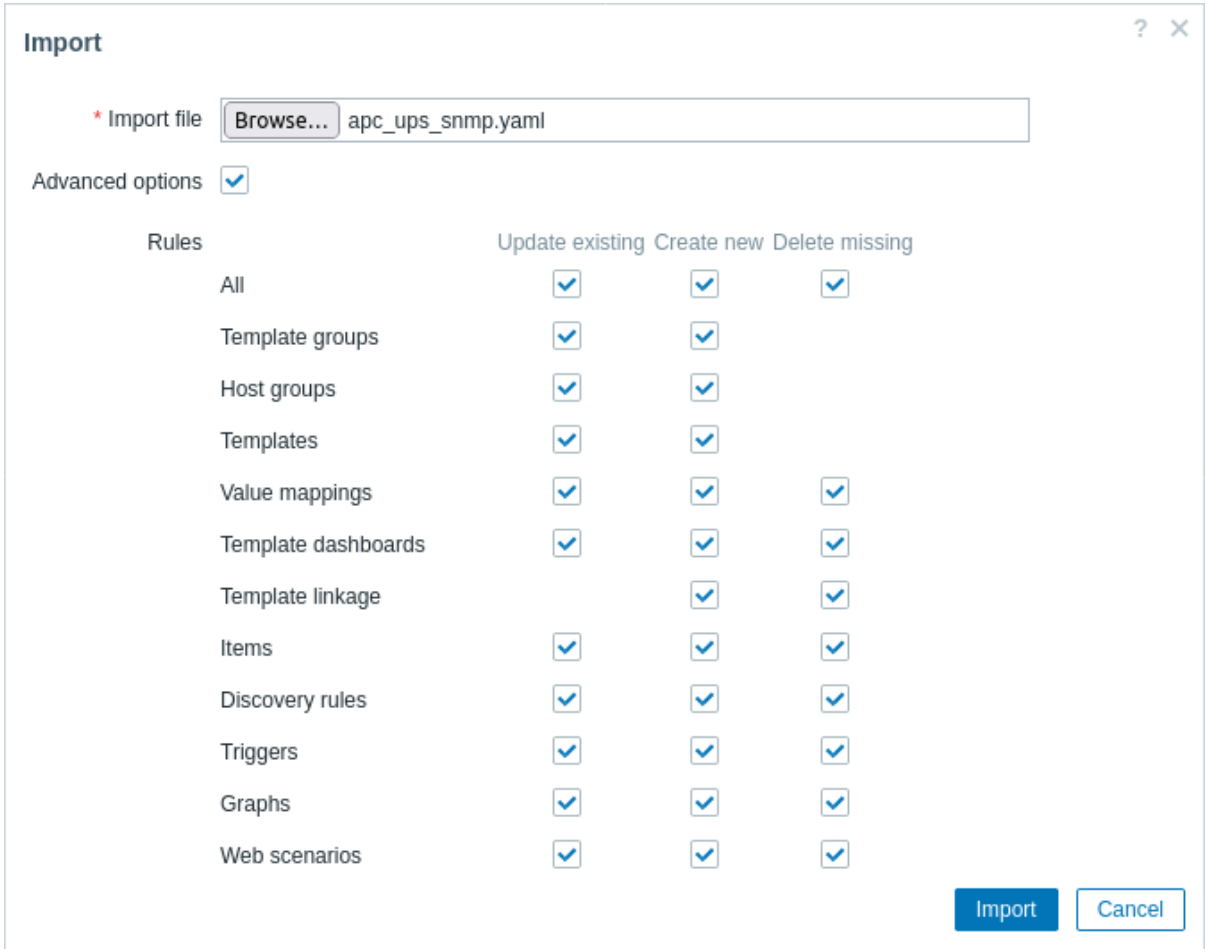
A seconda del formato selezionato, i template vengono esportati in un file locale con un nome predefinito:

- `zabbix_export_templates.yaml` - nell'esportazione YAML (opzione predefinita per l'esportazione);
- `zabbix_export_templates.xml` - nell'esportazione XML;
- `zabbix_export_templates.json` - nell'esportazione JSON.

Importazione

Per importare i template, procedi come segue:

1. Vai su *Raccolta dati* → *Template*.
2. Fai clic su *Importa* nell'angolo in alto a destra.
3. Seleziona il file di importazione.
4. Fai clic su *Importa* nell'angolo in basso a destra del modulo di configurazione.



Se selezioni la casella *Opzioni avanzate*, verrà visualizzato un elenco dettagliato di tutti gli elementi importabili: seleziona o deseleziona ogni regola di importazione secondo necessità.

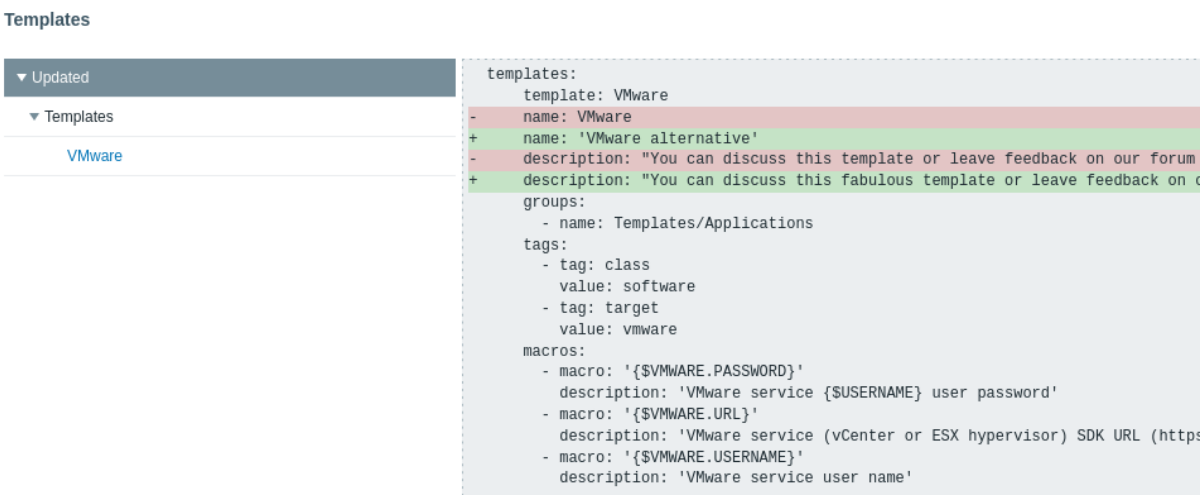
Se fai clic sulla casella nella riga *Tutti*, tutti gli elementi sottostanti verranno selezionati/deselezionati.

Regole di importazione:

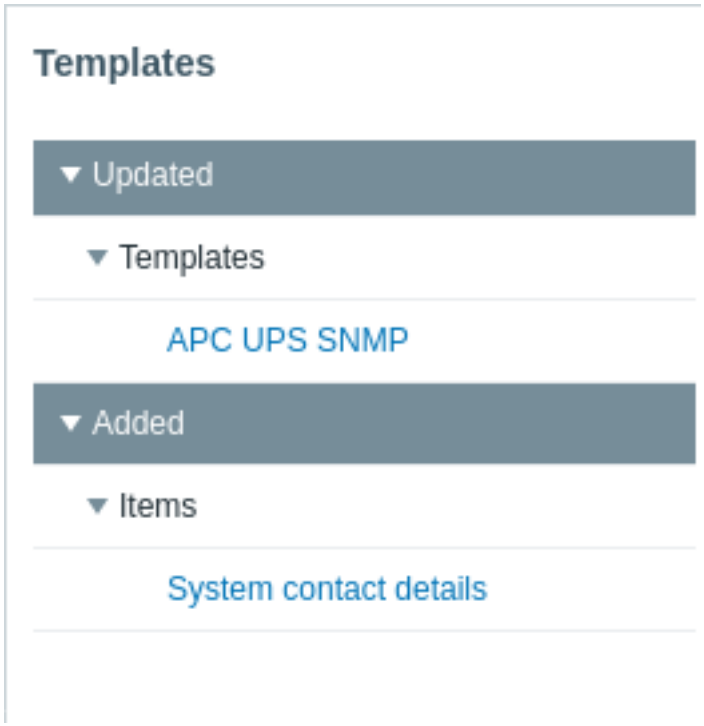
Regola	Descrizione
Aggiorna esistenti	Gli elementi esistenti verranno aggiornati utilizzando i dati del file di importazione. In caso contrario, non verranno aggiornati.
Crea nuovi	I nuovi elementi verranno creati utilizzando i dati del file di importazione. In caso contrario, non verranno creati.
Elimina 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*, il collegamento corrente del template non presente nel file di importazione verrà scollegato. 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, le opzioni *Elimina mancanti* devono essere selezionate durante l'importazione per ottenere un'importazione pulita. In questo modo, i vecchi item che non sono più presenti nel template aggiornato verranno rimossi (nota che ciò comporterà la perdita della cronologia di questi vecchi item).

Nella schermata successiva, potrai visualizzare il contenuto di un template in fase di importazione. Se si tratta di un nuovo template, tutti gli elementi saranno elencati in verde. Se si aggiorna un template esistente, i nuovi elementi del template saranno evidenziati in verde; gli elementi del template 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 navigare 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 in ciascuna sezione sono raggruppati per tipo di elemento; fai clic sulla freccia grigia per espandere o comprimere il gruppo di elementi.



Esamina le modifiche del template e poi fai clic su *Importa* per eseguire l'importazione del template. Nel frontend verrà visualizzato un messaggio di esito positivo o negativo dell'importazione.

Formato di esportazione Quando un template viene esportato, Zabbix genera un formato strutturato YAML, JSON o XML. L'esportazione include elementi del template, come metadati del template, item, macro, trigger, dashboard e altro.

Ogni elemento ha uno scopo specifico e può contenere elementi nidificati.

Le sezioni seguenti descrivono ciascun elemento nel formato di esportazione. Gli esempi utilizzano il template Linux by Zabbix agent.

I puntini di sospensione `(...)` indicano elementi omessi per brevità. La nota `(vedi tabella sotto)` viene utilizzata 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: (vedere la tabella sotto)
```

```
  triggers: (vedere la tabella sotto)
  graphs: (vedere la tabella sotto)
```

Metadati del template

Elemento	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 uguali a quelli dei trigger degli item del template.
graphs		Elemento radice per gli elementi graph del template, che sono uguali a quelli dei graph degli host.

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Questo è un template Linux ufficiale. Richiede Zabbix agent 8.0 o versione successiva. (...)
      wizard_ready: 'YES'
      readme: |
        ## Panoramica

        Questo è un template Linux ufficiale. Richiede Zabbix agent 8.0 o versione successiva. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (vedere la tabella sotto)
      discovery_rules: (vedere la tabella sotto)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Timeout dopo il quale l''agent viene considerato non disponibile. Funziona solo pe
          config: (vedere la tabella sotto)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'Soglia critica dell''utilizzo della CPU espressa in %.'
          config: (vedere la tabella sotto)
        (...)
      dashboards: (vedere la tabella sotto)
      valuemaps: (vedere la tabella sotto)
    (...)
```

Template

Elemento	Type	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). Vedere 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 fornitore del template (presente se il template esportato contiene dati del fornitore).
name	string	(obbligatorio) Nome del fornitore 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 reimpostato 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 low-level discovery 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). Vedere anche: Oggetto macro utente (`type`).
value	string	Valore della macro utente.
description	string	Descrizione della macro utente.
config		Elemento radice per la configurazione delle macro del template, responsabile di come la macro viene visualizzata nella Procedura guidata host.
dashboards		Elemento radice per le dashboard del template.
valuemaps		Elemento radice per le mappe dei valori del template.

Item del template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix agent ping'
          key: agent.ping
          description: 'L agent restituisce sempre "1" per questo item. Può essere utilizzato in combinazi
          valuemap:
            name: 'Stato ping di Zabbix agent'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'Numero di pacchetti installati'
          key: system.sw.packages.get
          delay: 1h
          preprocessing: (vedere la tabella sotto)
```

```
      tags:
        - tag: component
          value: os
      triggers: (vedere la tabella sotto)
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: 'Utilizzo della memoria'
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: 'La percentuale di memoria utilizzata è calcolata come `100-pavailable`.'
      preprocessing: (vedere la tabella sotto)
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: (vedere la tabella sotto)
    (...)
```

Element	Type	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).
snmp_oid	string	(obbligatorio per gli item SNMP_AGENT) ID 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 (usando suffisso temporale, macro utente o macro LLD) per cui i dati storici devono essere conservati. Predefinito: 31d.
trends	string	Periodo di tempo (usando suffisso temporale, macro utente o macro LLD) per cui i dati di trend devono essere conservati. 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 di host, separati da virgole, 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).
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 utilizzato per gli item JMX, deve essere specificato anche `password` (vedere sotto) oppure entrambi gli elementi devono essere lasciati vuoti.

Element	Type	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 utilizzato per gli item JMX, deve essere specificato anche `username` (vedere sopra) oppure entrambi gli elementi devono essere lasciati vuoti.
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 host popolato dall'item. Valori possibili:[1] NONE (0), ALIAS (4), ecc. (vedere Inventario host per i campi supportati).
valuemap		Elemento radice per le mappe di valori dell'item.
name	string	(obbligatorio) Nome della mappa di 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 il preprocessing dei valori dell'item.
step		Elemento radice per i passaggi di preprocessing 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 di tipi di item Timeouts.
url	string	(obbligatorio per gli item HTTP_AGENT) Stringa URL.
query_fields		Elemento radice per i parametri di query. Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome del parametro di query.
value	string	Valore del parametro di 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	Seguire 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 del 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 del 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	Quale parte della risposta deve essere memorizzata. 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	Come elaborare la risposta. Supportato per gli item HTTP_AGENT. Valori possibili:[1] RAW (0, predefinito), JSON (1).

Element	Type	Descrizione
allow_traps	string	Consente di popolare il valore in modo simile 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 per il 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).
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: Item object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Passaggi di preprocessing del valore degli item del template

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Elemento	Type	Descrizione
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)/(ora attuale - ora 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.

Elemento	Type	Descrizione
error_handler	string	Tipo di azione utilizzato 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: Item preprocessing object (fare riferimento alla proprietà pertinente con un nome corrispondente).

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
        (...)
```

Elemento	Type	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo trigger.
expression	string	(obbligatorio) 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 ripristino del trigger.
correlation_mode	string	Modalità di correlazione (nessuna correlazione degli eventi o correlazione degli eventi per tag). Valori possibili:[1] DISABLED (0, predefinito), TAG_VALUE (1).
correlation_tag	string	Nome del tag da utilizzare per la correlazione degli eventi.
name	string	(obbligatorio) Nome del trigger.
event_name	string	Nome dell'evento.
opdata	string	Dati operativi.

Elemento	Type	Descrizione
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 degli eventi (evento singolo di problema o eventi multipli 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	(obbligatorio) Nome del trigger di dipendenza.
expression	string	(obbligatorio) Espressione del trigger di dipendenza.
recovery_expression	string	Espressione di ripristino del trigger di dipendenza.
tags		Elemento radice per i tag del trigger.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Trigger object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Regole di low-level discovery del template

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Rilevamento delle interfacce di rete'
          key: net.if.discovery
          delay: 1h
          filter: (vedere la tabella sotto)
          description: 'Il rilevamento delle interfacce di rete.'
          item_prototypes: (vedere la tabella sotto)
          trigger_prototypes: (vedere la tabella sotto)
          graph_prototypes: (vedere la tabella sotto)
        (...)
```

Attention:
La maggior parte degli elementi delle regole di low-level discovery del template è uguale a quella dei template item. La tabella seguente descrive gli elementi che differiscono dai template item.

Element	Type	Descrizione
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 low-level discovery.
filter		Elemento radice per i filtri delle regole di low-level discovery del template.
lifetime	string	Periodo di tempo (usando secondi, suffisso temporale o macro utente), trascorso il quale le risorse non più rilevate verranno eliminate. Predefinito: 7d.
lifetime_type	string	Scenario per l'eliminazione delle risorse LLD perse. Valori possibili: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.

Element	Type	Descrizione
enabled_lifetime	string	Periodo di tempo (usando secondi, suffisso temporale o macro utente), trascorso il quale le risorse non più rilevate 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 prototipo di item del template, che sono uguali a quelli dei template item.
trigger_prototypes		Elemento radice per gli elementi prototipo di trigger del template, che sono uguali a quelli dei trigger degli item del template.
graph_prototypes		Elemento radice per gli elementi prototipo di grafico del template, che sono uguali a quelli dei grafici del host.
host_prototypes		Elemento radice per gli elementi prototipo di host del template, che sono uguali a quelli dei host.
parent_discovery_rule		Elemento radice per la regola di low-level discovery padre (o prototipo di regola) del prototipo di regola di low-level discovery. Questa proprietà indica che si tratta di un prototipo di regola LLD, figlio diretto della regola/prototipo di regola referenziato.
key	string	(obbligatorio) La chiave della regola di low-level discovery padre (o prototipo di regola).
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 low-level discovery.
lld_macro	string	(obbligatorio) Nome della macro della regola di low-level discovery.
path	string	(obbligatorio) Selettore per il valore, che verrà assegnato alla macro corrispondente.
preprocessing		Elemento radice per il preprocessing del valore della regola di low-level discovery.
step		Elemento radice per gli elementi dei passaggi di preprocessing del valore della regola di low-level discovery, che sono uguali a quelli dei passaggi di preprocessing del valore degli item del template, tranne per il fatto che hanno meno valori possibili. Vedere anche: oggetto di preprocessing della regola LLD.
overrides		Elemento radice per le regole di override della regola di low-level discovery.
name	string	(obbligatorio) Nome univoco dell'override.
step	string	(obbligatorio) Numero d'ordine univoco dell'override.
stop	string	Interrompe l'elaborazione degli override successivi in caso di corrispondenza.
filter		Elemento radice per gli elementi del filtro della regola di override della regola di low-level discovery del template, che sono uguali a quelli dei filtri delle regole di low-level discovery del template.
operations		Elemento radice per le operazioni di override della regola di low-level discovery del template.

Note:
Vedere anche: oggetto regola LLD (fare riferimento alla proprietà pertinente con un nome corrispondente).

Filtri delle regole di low-level discovery 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
          (...)
```

Elemento	Type	Descrizione
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 low-level discovery 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 utilizzato per fare riferimento a una condizione dall'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.

Note:
Vedi anche: oggetto filtro della regola LLD (fare riferimento alla proprietà pertinente con un nome corrispondente).

Operazioni di override della regola di low-level discovery del template

Elemento	Tipo	Descrizione
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 nell'operazione di override.
discover	string	Indica se l'oggetto viene aggiunto come risultato del rilevamento.
delay	string	Intervallo di aggiornamento impostato per il prototipo di item nell'operazione di override.
history	string	Periodo di archiviazione dello storico impostato per il prototipo di item nell'operazione di override.
trends	string	Periodo di archiviazione dei trend impostato per il prototipo di item nell'operazione di override.
severity	string	Gravità del prototipo di trigger impostata nell'operazione di override.
tags		Elemento radice per i tag impostati per l'oggetto nell'operazione di override.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
templates		Elemento radice per i template collegati al prototipo di host nell'operazione di override.
name	string	(obbligatorio) Nome del template.
inventory_mode	string	Modalità inventario del prototipo di host impostata nell'operazione di override.

Note:
Vedi anche: Oggetto operazione di override della regola LLD (fare riferimento alla proprietà pertinente con il nome corrispondente).

Scenari web del template

Elemento	Type	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo scenario web.
name	string	(obbligatorio) Nome dello scenario web.
delay	string	Frequenza di esecuzione dello scenario web (usando secondi, suffisso temporale o macro utente). Predefinito: 1m.
attempts	integer	Numero di tentativi per l'esecuzione dei passaggi dello scenario web. Valori possibili:[1] 1-10 (predefinito: 1).
agent	string	agent client. Zabbix fingerà di essere il browser selezionato. Questo è utile quando un sito web restituisce contenuti diversi per browser diversi. Predefinito: Zabbix.
http_proxy	string	proxy che verrà utilizzato dallo scenario web, specificato come: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento radice per le variabili dello scenario web che possono essere utilizzate nei passaggi dello scenario.
name	string	(obbligatorio) Nome della variabile.
value	text	(obbligatorio) 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	(obbligatorio) Nome dell'intestazione.
value	text	(obbligatorio) 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 utilizzato per l'autenticazione BASIC (HTTP), NTLM, Kerberos o Digest.
http_password	string	Password utilizzata 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 utilizzato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_file	string	Nome del file della chiave privata SSL utilizzato per l'autenticazione del client (deve essere in formato PEM).
ssl_key_password	string	Password del file della chiave privata SSL.
steps		(obbligatorio) Elemento radice per i passaggi dello scenario web del template.
tags		Elemento radice per i tag dello scenario web.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Web scenario object (fare riferimento alla proprietà pertinente con nome corrispondente).

Passi dello scenario web del template

Elemento	Type	Descrizione
name	string	(obbligatorio) Nome del passo dello scenario web.
url	string	(obbligatorio) 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	(obbligatorio) Nome del parametro di query.
value	string	Valore del parametro di query.
posts		Elemento radice per le variabili HTTP POST (una stringa (dati POST grezzi) o un array di campi HTTP (dati dei campi del modulo)).
name	string	(obbligatorio) Nome del campo POST.
value	string	(obbligatorio) Valore del campo POST.

Elemento	Type	Descrizione
variables		Elemento radice delle variabili a livello di passo (macro) che devono essere applicate dopo questo passo. Se il valore della variabile ha il prefisso 'regex:', il suo valore viene estratto dai dati restituiti da questo passo in base al modello di espressione regolare che segue il prefisso 'regex:'
name	string	(obbligatorio) Nome della variabile.
value	text	(obbligatorio) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta.
name	string	(obbligatorio) Nome dell'intestazione.
value	text	(obbligatorio) 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, suffisso temporale o macro utente) dell'esecuzione del passo. Predefinito: 15s.
required	string	Testo che deve essere presente nella risposta (ignorato se vuoto).
status_codes	string	Un elenco di codici di stato HTTP accettati, separati da virgole (ad esempio, 200-201,210-299; ignorato se vuoto).

Note:
Vedi anche: Web scenario step object (fare riferimento alla proprietà pertinente con nome corrispondente).

Configurazione delle macro del template

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Secondi dall'ultima volta in cui è stato visto Zabbix agent'
            description: 'Timeout dopo il quale l'agent è considerato non disponibile.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Soglie
            label: 'Soglia di utilizzo della CPU espressa'
            description: 'Soglia critica di utilizzo della CPU espressa in %. Nell'intervallo da 0 a 100 in
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Controllo delle interfacce'
            description: 'Attiva un trigger quando lo stato operativo dell'interfaccia cambia in "Link down
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Element	Type	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 è raggruppata la macro.
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	Contrassegna la macro come 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: Macro configuration object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Dashboard dei template

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (vedere la tabella sotto)
        (...)
```

Elemento	Type	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.
display period	integer	Periodo di visualizzazione della pagina.
sortorder	integer	Ordine di ordinamento della pagina.
widgets		Elemento radice per i widget del dashboard del template.

Note:
Vedi anche: Template dashboard object (fare riferimento alla proprietà pertinente con un nome corrispondente).

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})]: Grafico di utilizzo dello spazio (relativo al to
                    - 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})]: Grafico di utilizzo dello spazio, in % (relativo
                    - type: STRING
                      name: reference
                      value: AAAEC
        (...)
```

Elemento	Type	Descrizione
type	string	(obbligatorio) 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.
width	integer	Larghezza del widget. Valori possibili:[1] 1-72.
height	integer	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 del template.
type	string	(obbligatorio) 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	(obbligatorio) Nome del campo del widget.

Elemento	Type	Descrizione
value	mixed	(obbligatorio) Valore del campo del widget, a seconda del tipo di campo.

Note:
Vedi anche: Template dashboard widget object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Mappe di 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	Type	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questa mappa di valori.
name	string	(obbligatorio) Nome della mappa di 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: Value map object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Note a piè di pagina [1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono indicati solo come riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ciascuna sezione.

4 Host

Panoramica

Gli host vengono esportati insieme a molti oggetti correlati e alle relative relazioni tra oggetti.

L'esportazione degli host contiene:

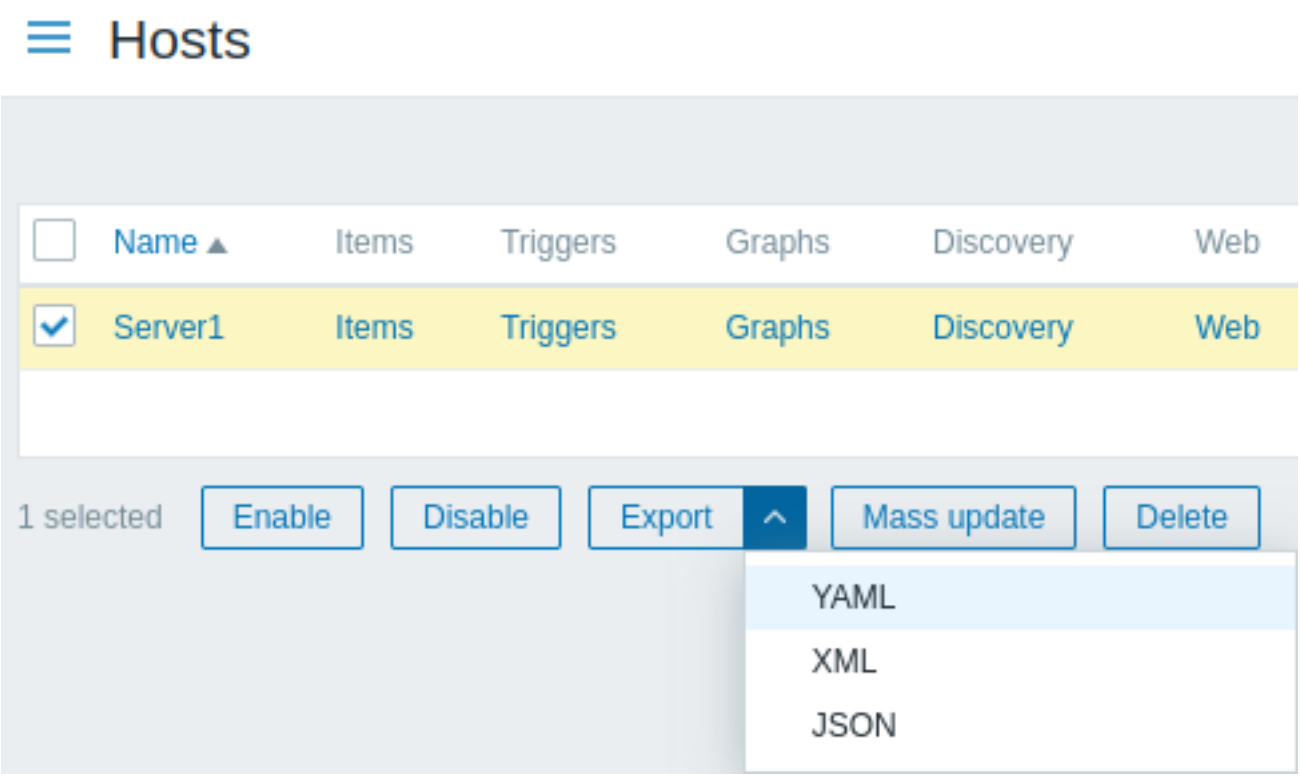
- Gruppi di host collegati
- Dati dell'host

- 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

Esportazione

Per esportare gli host, procedere come segue:

1. Andare su *Raccolta dati* → *Host*.
2. Selezionare le caselle di controllo degli host da esportare.
3. Fare clic su *Esporta* sotto l'elenco.



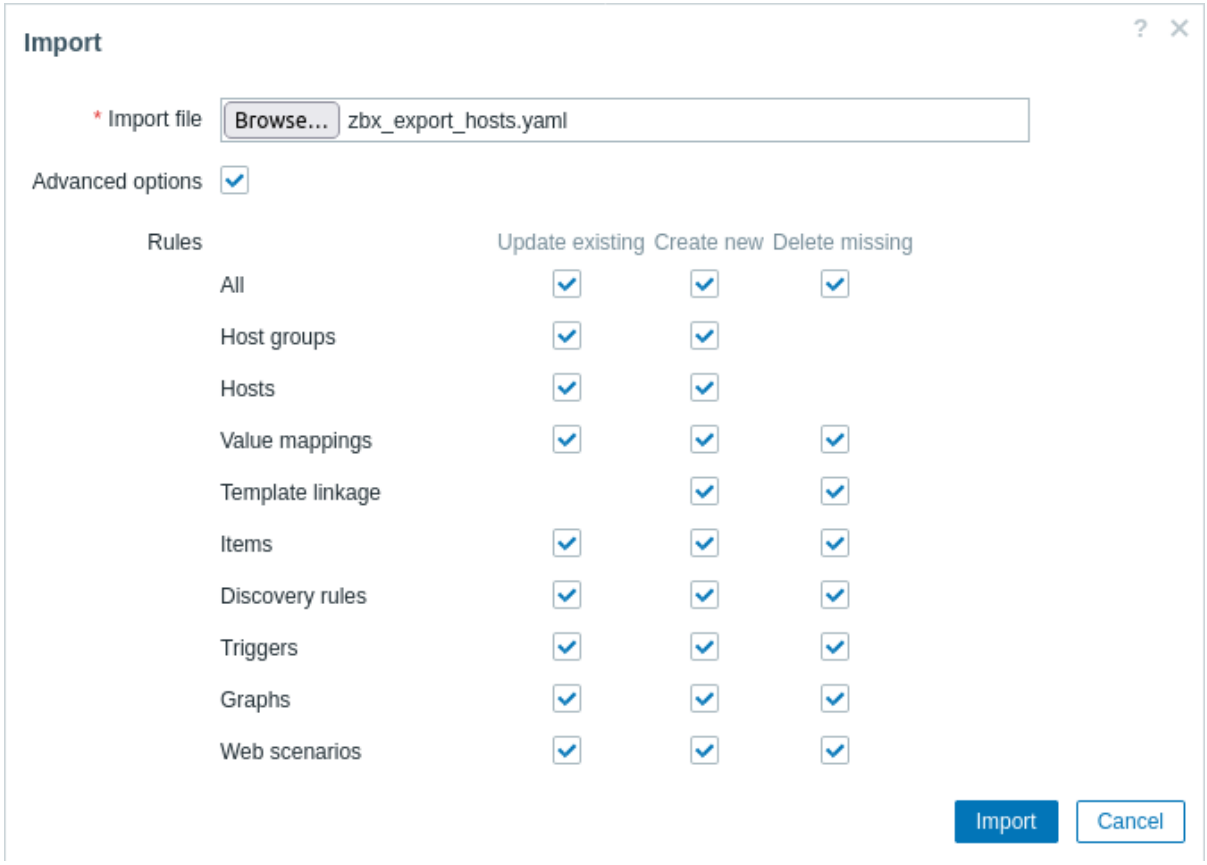
A seconda del formato selezionato, gli host vengono esportati in un file locale con un nome predefinito:

- `zabbix_export_hosts.yaml` - nell'esportazione YAML (opzione predefinita per l'esportazione);
- `zabbix_export_hosts.xml` - nell'esportazione XML;
- `zabbix_export_hosts.json` - nell'esportazione JSON.

Importazione

Per importare host, procedere come segue:

1. Andare su *Raccolta dati* → *Host*.
2. Fare clic su *Importa* nell'angolo in alto a destra.
3. Selezionare il file di importazione.
4. 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 *Tutto*, tutti gli elementi sottostanti verranno selezionati/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*, il collegamento ai template attuale non presente nel file di importazione verrà scollegato. 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).

Nel frontend verrà visualizzato un messaggio di esito positivo o negativo dell'importazione.

Formato di esportazione

Quando un host viene esportato, Zabbix genera un formato strutturato YAML, JSON o XML. L'esportazione include elementi dell'host, come metadati dell'host, item, macro, trigger e altro.

Ogni elemento ha uno scopo specifico e può contenere elementi nidificati.

Le sezioni seguenti descrivono ciascun elemento nel formato di esportazione. Gli esempi utilizzano un host con il template MySQL by Zabbix agent 2. Ai fini della dimostrazione degli elementi, il template viene scollegato dopo la creazione dell'host. Lo scollegamento mantiene tutti gli elementi nella configurazione (a differenza dello scollegamento con rimozione, che li elimina).

I puntini di sospensione (`...`) indicano elementi omessi per brevità. La nota (`see table below`) viene utilizzata quando l'elemento è spiegato più in dettaglio nelle sezioni successive.

```
zabbix_export:
  version: '8.0'
```

```
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (vedere la tabella sotto)
  graphs: (vedere la tabella sotto)
```

Metadati del host

Elemento	Type	Descrizione
version	string	(obbligatorio) Versione di Zabbix.
host_groups		(obbligatorio) Elemento radice per i gruppi di host.
uuid	string	(obbligatorio) Identificatore univoco per questo gruppo di host.
name	string	(obbligatorio) Nome del gruppo di host.
hosts		Elemento radice per gli host.
graphs		Elemento radice per i grafici del host.

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (vedere tabella sotto)
      items: (vedere tabella sotto)
      discovery_rules: (vedere tabella sotto)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'Password utente MySQL.'
        (...)
      valuemaps: (vedere tabella sotto)
  (...)
```

Host

Element	Type	Description
host	string	(obbligatorio) 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 proxy.
name	string	(obbligatorio) Nome del proxy (se presente) che monitora l'host.
proxy_group		Elemento radice per il gruppo di proxy.
name	string	(obbligatorio) 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).
ipmi_privilege	string	Livello di privilegio della sessione IPMI. Valori possibili:[1] CALLBACK (1), USER (2, predefinito), OPERATOR (3), ADMIN (4), OEM (5).

Element	Type	Description
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	(obbligatorio) Nome del template.
groups		Elemento radice per i gruppi host a cui appartiene l'host.
name	string	(obbligatorio) 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 low-level discovery dell'host.
http_tests		Elemento radice per gli scenari web dell'host.
tags		Elemento radice per i tag host.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
macros		Elemento radice per le macro host.
macro	string	(obbligatorio) 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 host.
<inventory_property>	string	Proprietà di inventario. Tutte le proprietà hanno il rispettivo elemento (`type`, `name`, `os`, ecc.; per un esempio, vedere Export format).
inventory_mode	string	Modalità inventario. Valori possibili:[1] DISABLED (-1), MANUAL (0, predefinito), AUTOMATIC (1).
valuemaps		Elemento radice per le mappe di valori host.

Note:
Vedere anche: Host object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Interfacce host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

Element	Type	Descrizione
default	string	Indica se questa è l'interfaccia host primaria. Si noti che su un host può esserci una sola interfaccia primaria per ciascun tipo. 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 utilizzare l'IP come interfaccia per la connessione all'host (altrimenti verrà utilizzato il 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	Utilizzare 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.
max_repetitions	string	Valore massimo di ripetizione per richieste bulk SNMP native (GetBulkRequest-PDUs). Supportato per item SNMPv2 e SNMPv3 (item `discovery[]` e `walk[]`). Predefinito: 10.

Element	Type	Descrizione
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	Utilizzare richieste bulk per SNMP. Valori possibili:[1] NO (0), YES (1, predefinito).
interface_ref	string	Nome di riferimento dell'interfaccia da utilizzare negli item (formato: `if<N>`).

Note:
Vedi anche: Host interface object (fare riferimento alla proprietà pertinente con nome corrispondente).

Item del host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          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: (vedi tabella sotto)
          master_item:
            key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          tags:
            - tag: component
              value: cache
        (...)
        - name: 'Buffer pool utilization'
          type: CALCULATED
          key: mysql.buffer_pool_utilization
          value_type: FLOAT
          units: '%'
          params: |
            ( last(//mysql.innodb_buffer_pool_pages_total) -
            last(//mysql.innodb_buffer_pool_pages_free) ) /
            ( last(//mysql.innodb_buffer_pool_pages_total) +
            ( last(//mysql.innodb_buffer_pool_pages_total) = 0 ) ) * 100 *
            ( last(//mysql.innodb_buffer_pool_pages_total) > 0 )
          description: 'Rapporto tra le pagine utilizzate e il totale delle pagine nel buffer pool.'
          tags:
            - tag: component
```

```
            value: memory
        triggers: (vedi tabella sotto)
      (...)
      - name: Uptime
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: 'Numero di secondi di attività del server.'
        preprocessing: (vedi tabella sotto)
        master_item:
          key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
        tags:
          - tag: component
            value: application
        triggers: (vedi tabella sotto)
      (...)
```

Element	Type	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).
snmp_oid	string	(obbligatorio per gli item SNMP_AGENT) ID 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 suffisso temporale, macro utente o macro LLD) per cui i dati storici devono essere conservati. Predefinito: 31d.
trends	string	Periodo di tempo (utilizzando suffisso temporale, macro utente o macro LLD) per cui i dati di trend devono essere conservati. 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, dei 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).
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 utilizzato per gli item JMX, deve essere specificato anche `password` (vedi sotto) oppure entrambi gli elementi devono essere lasciati vuoti.
password	string	(obbligatorio per gli item SSH e TELNET) Password per l'autenticazione. Supportato per gli item SIMPLE, ODBC, JMX e HTTP_AGENT. Quando utilizzato per gli item JMX, deve essere specificato anche `username` (vedi sopra) oppure entrambi gli elementi devono essere lasciati vuoti.
publickey	string	(obbligatorio per gli item SSH) Nome del file della chiave pubblica.

Element	Type	Descrizione
privatekey	string	(obbligatorio per gli item SSH) Nome del file della chiave privata.
description	text	Descrizione dell'item.
inventory_link	string	Campo dell'inventario del host che viene popolato dall'item. Valori possibili:[1] NONE (0), ALIAS (4), ecc. (vedi Inventario host per i campi supportati).
valuemap		Elemento radice per le mappe di valori dell'item.
name	string	(obbligatorio) Nome della mappa di 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 il preprocessing dei valori dell'item.
step		Elemento radice per i passaggi di preprocessing dei valori degli item del host.
interface_ref	string	Riferimento all'interfaccia del host (formato: `if<N>`).
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 Timeouts dei tipi di item.
url	string	(obbligatorio per gli item HTTP_AGENT) Stringa URL.
query_fields		Elemento radice per i parametri di query. Supportato per gli item HTTP_AGENT.
name	string	(obbligatorio per gli item HTTP_AGENT) Nome del parametro di query.
value	string	Valore del parametro di 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	Segui 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 del 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 del 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	Quale parte della risposta deve essere memorizzata. 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	Come elaborare la 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 simile a 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.

Element	Type	Descrizione
ssl_key_password	string	Password per il file della chiave SSL. Supportato per gli item HTTP_AGENT.
verify_peer	string	Indica se verificare che il certificato del 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 del host per la connessione corrisponda a quello nel certificato del host. Supportato per gli item HTTP_AGENT. Valori possibili:[1] NO (0, predefinito), YES (1).
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 host.

Note:
Vedi anche: Item object (fare riferimento alla proprietà pertinente con nome corrispondente).

Passaggi di preprocessing del valore di item del 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
          (...)
```

Elemento	Type	Descrizione
type	string	(obbligatorio) Il tipo di passaggio di preprocessing del valore di 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)/(ora attuale - ora 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 di item.
parameter	string	Parametro individuale del passaggio di preprocessing del valore di item.
error_handler	string	Tipo di azione utilizzato 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: Item preprocessing object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Trigger degli item del 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 has not received any data for items for the last 30 minutes.'
              dependencies:
                - name: 'MySQL: Service is down'
                  expression: 'last(/MySQL server/mysql.ping["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSW
              tags:
                - tag: scope
                  value: availability
            - expression: 'last(/MySQL server/mysql.uptime)<10m'
              name: 'MySQL: Service has been restarted'
              event_name: 'MySQL: Service has been restarted (uptime < 10m)'
              priority: INFO
              description: 'MySQL uptime is less than 10 minutes.'
              tags:
                - tag: scope
                  value: notice
        (...)
```

Elemento	Type	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo trigger.
expression	string	(obbligatorio) 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 ripristino del trigger.
correlation_mode	string	Modalità di correlazione (nessuna correlazione degli eventi o correlazione degli eventi per tag). Valori possibili:[1] DISABLED (0, predefinito), TAG_VALUE (1).
correlation_tag	string	Nome del tag da utilizzare per la correlazione degli eventi.
name	string	(obbligatorio) 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 degli eventi (evento singolo di problema o eventi multipli 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	(obbligatorio) Nome del trigger di dipendenza.
expression	string	(obbligatorio) Espressione del trigger di dipendenza.
recovery_expression	string	Espressione di ripristino del trigger di dipendenza.

Elemento	Type	Descrizione
tags		Elemento radice per i tag del trigger.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Trigger object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Regole di low-level discovery del host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (vedere la tabella sotto)
          description: 'Scansione dei database nel DBMS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Dimensione del database {#DATABASE}'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Dimensione del database.'
              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 low-level discovery del host è uguale a quella degli item del host. La tabella seguente descrive gli elementi che differiscono dagli item del 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 low-level discovery.
filter		Elemento radice per i filtri delle regole di low-level discovery del host.

Element	Type	Description
lifetime	string	Periodo di tempo (usando secondi, suffisso temporale o macro utente), trascorso il quale le risorse non più rilevate 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 temporale o macro utente), trascorso il quale le risorse non più rilevate 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 prototipo di item del host, che sono uguali a quelli degli item del host.
trigger_prototypes		Elemento radice per gli elementi prototipo di trigger del host, che sono uguali a quelli dei trigger degli item del host.
graph_prototypes		Elemento radice per i prototipi di grafico del host, che sono uguali a quelli dei grafici del host.
host_prototypes		Elemento radice per i prototipi di host, che sono uguali a quelli dei host.
parent_discovery_rule		Elemento radice per la regola di low-level discovery padre (o prototipo di regola) del prototipo di regola di low-level discovery. Questa proprietà indica che si tratta di un prototipo di regola LLD, figlio diretto della regola/prototipo di regola referenziato.
key	string	(obbligatorio) La chiave della regola di low-level discovery padre (o prototipo di regola).
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 low-level discovery.
lld_macro	string	(obbligatorio) Nome della macro della regola di low-level discovery.
path	string	(obbligatorio) Selettore per il valore, che verrà assegnato alla macro corrispondente.
preprocessing		Elemento radice per il preprocessing del valore della regola di low-level discovery.
step		Elemento radice per gli elementi dei passaggi di preprocessing del valore della regola di low-level discovery, che sono uguali a quelli dei passaggi di preprocessing del valore degli item del host, eccetto per un numero inferiore di valori possibili. Vedere anche: oggetto di preprocessing della regola LLD.
overrides		Elemento radice per le regole di override della regola di low-level discovery.
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 low-level discovery, che sono uguali a quelli dei filtri delle regole di low-level discovery del host.
operations		Elemento radice per le operazioni di override della regola di low-level discovery del host.

Note:
Vedere anche: oggetto regola LLD (fare riferimento alla proprietà pertinente con nome corrispondente).

Filtri delle regole di low-level discovery del 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	Type	Descrizione
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 low-level discovery 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 utilizzato per fare riferimento a una condizione dall'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.

Note:
Vedi anche: oggetto filtro della regola LLD (fare riferimento alla proprietà pertinente con un nome corrispondente).

Operazioni di override della regola di low-level discovery del host

Elemento	Type	Descrizione
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 in seguito all'operazione di override.
discover	string	Indica se l'oggetto viene aggiunto come risultato del discovery.
delay	string	Intervallo di aggiornamento impostato per il prototipo di item in seguito all'operazione di override.
history	string	Periodo di archiviazione dello storico impostato per il prototipo di item in seguito all'operazione di override.
trends	string	Periodo di archiviazione dei trend impostato per il prototipo di item in seguito all'operazione di override.
severity	string	Gravità del prototipo di trigger impostata in seguito all'operazione di override.
tags		Elemento radice per i tag impostati per l'oggetto in seguito all'operazione di override.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.
templates		Elemento radice per i template collegati al prototipo di host in seguito all'operazione di override.
name	string	(obbligatorio) Nome del template.
inventory_mode	string	Modalità inventario del prototipo di host impostata in seguito all'operazione di override.

Note:
Vedi anche: Oggetto operazione di override della regola LLD (fare riferimento alla proprietà pertinente con nome corrispondente).

Scenari web del host

Elemento	Tipo	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questo scenario web.
name	string	(obbligatorio) Nome dello scenario web.
delay	string	Frequenza (usando secondi, suffisso temporale o macro utente) di esecuzione dello scenario web. Predefinito: 1m.

Elemento	Tipo	Descrizione
attempts	integer	Numero di tentativi per l'esecuzione dei passaggi dello scenario web. Valori possibili: 1-10 (predefinito: 1).
agent	string	agent client. Zabbix fingerà di essere il browser selezionato. Questo è utile quando un sito web restituisce contenuti diversi per browser diversi. Predefinito: Zabbix.
http_proxy	string	proxy che verrà utilizzato dallo scenario web, indicato come: `http://[username[:password]@]proxy.example.com[:port]`
variables		Elemento radice per le variabili dello scenario web che possono essere utilizzate nei passaggi dello scenario.
name	string	(obbligatorio) Nome della variabile.
value	text	(obbligatorio) 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	(obbligatorio) Nome dell'intestazione.
value	text	(obbligatorio) 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 utilizzato per l'autenticazione BASIC (HTTP), NTLM, Kerberos o Digest.
http_password	string	Password utilizzata 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 utilizzato per l'autenticazione client (deve essere in formato PEM).
ssl_key_file	string	Nome del file della chiave privata SSL utilizzato per l'autenticazione client (deve essere in formato PEM).
ssl_key_password	string	Password del file della chiave privata SSL.
steps		(obbligatorio) Elemento radice per i passaggi dello scenario web del host.
tags		Elemento radice per i tag dello scenario web.
tag	string	(obbligatorio) Nome del tag.
value	string	Valore del tag.

Note:
Vedi anche: Web scenario object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Passi dello scenario web del host

Elemento	Tipo	Descrizione
name	string	(obbligatorio) Nome del passo dello scenario web.
url	string	(obbligatorio) URL per il monitoraggio.
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	(obbligatorio) Nome del parametro di query.
value	string	Valore del parametro di query.
posts		Elemento radice per le variabili HTTP POST (una stringa (dati post grezzi) o un array di campi HTTP (dati dei campi del modulo)).
name	string	(obbligatorio) Nome del campo POST.
value	string	(obbligatorio) 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 il prefisso 'regex:', il suo valore viene estratto dai dati restituiti da questo passo in base al modello di espressione regolare che segue il prefisso 'regex:'

Elemento	Tipo	Descrizione
name	string	(obbligatorio) Nome della variabile.
value	text	(obbligatorio) Valore della variabile.
headers		Elemento radice per le intestazioni HTTP da inviare durante l'esecuzione di una richiesta.
name	string	(obbligatorio) Nome dell'intestazione.
value	text	(obbligatorio) 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 (usando secondi, suffisso temporale o macro utente) dell'esecuzione del passo. Predefinito: 15s.
required	string	Testo che deve essere presente nella risposta (ignorato se vuoto).
status_codes	string	Un elenco di codici di stato HTTP accettati, delimitato da virgole (ad esempio, 200-201,210-299; ignorato se vuoto).

Note:
Vedi anche: Web scenario step object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Grafici host

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (vedere la tabella sotto)
    (...)
```

Elemento	Tipo	Descrizione
uuid	string	Identificatore univoco del grafico.
name	string	(obbligatorio) Nome del grafico.
width	integer	Larghezza del grafico, in pixel. Usata per l'anteprima e per i grafici a torta/esplosi. Valori possibili:[1] 20-65535 (predefinito: 900).
height	integer	Altezza del grafico, in pixel. Usata 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.
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 semplici del trigger 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.

Elemento	Tipo	Descrizione
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 graph item dell'host.

Note:
Vedi anche: Oggetto graph (fare riferimento alla proprietà pertinente con un nome corrispondente).

Item del grafico del 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	Type	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'item 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'item del grafico. Supportato per i grafici NORMAL e STACKED.
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, media e massimo; supportato per grafici semplici), LAST (9, supportato per grafici a torta/esplosi).
type	string	Tipo di item del grafico. Valori possibili:[1] SIMPLE (0, predefinito), GRAPH_SUM (2; il valore dell'item rappresenta l'intera torta; supportato per grafici a torta/esplosi).
item		(obbligatorio) Item individuale.

Elemento	Type	Descrizione
host	string	(obbligatorio) Host dell'item.
key	string	(obbligatorio) Chiave dell'item.

Note:
Vedi anche: Graph item object (fare riferimento alla proprietà pertinente con un nome corrispondente).

Mappe di valori del host

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Example value map
          mappings:
            - value: '1'
              newvalue: Example value
  (...)
```

Elemento	Type	Descrizione
uuid	string	(obbligatorio) Identificatore univoco per questa mappa di valori.
name	string	(obbligatorio) Nome della mappa di 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 mappa di valori (fare riferimento alla proprietà pertinente con un nome corrispondente).

Note a piè di pagina [1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono indicati solo come riferimento. Per ulteriori informazioni, vedere la pagina dell'oggetto API collegata nella voce della tabella o alla fine di ciascuna sezione.

5 Mappe di rete

Panoramica

L'esportazione della mappa di rete contiene:

- Tutte le immagini correlate
- 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)

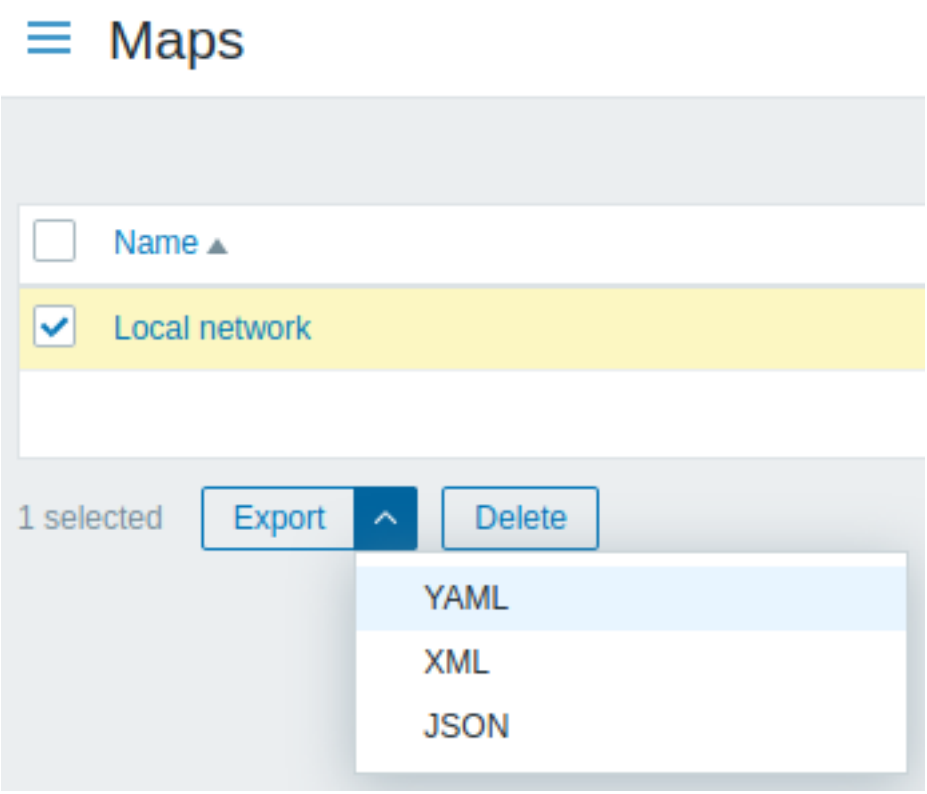
Warning:
Eventuali gruppi di host, host, trigger, altre mappe o altri elementi che possono essere correlati alla mappa esportata non vengono esportati. Pertanto, se manca almeno uno degli elementi a cui la mappa fa riferimento, l'importazione non riuscirà.

Esportazione

Per esportare le mappe di rete, procedi come segue:

1. Vai su *Monitoring* → *Maps*.
2. Seleziona le caselle di controllo delle mappe di rete da esportare.

3. Fai clic su *Export* sotto l'elenco.



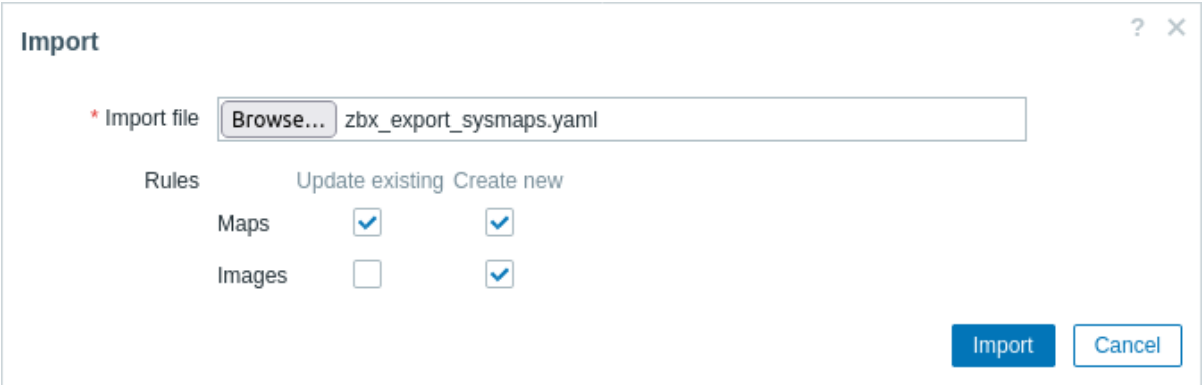
A seconda del formato selezionato, le mappe vengono esportate in un file locale con un nome predefinito:

- `zabbix_export_maps.yaml` - nell'esportazione YAML (opzione predefinita per l'esportazione);
- `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 su *Monitoring* → *Maps*.
2. Fare clic su *Import* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le opzioni richieste nelle regole di importazione.
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	Le nuove mappe verranno create utilizzando i dati del file di importazione. In caso contrario, non verranno create.

Se si deselezionano entrambe le opzioni delle 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*.

Nel frontend verrà visualizzato un messaggio di successo o di errore dell'importazione.

> **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: 'Rete locale'
      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.

Elemento	Tipo	Descrizione
images		Elemento radice per le immagini.
name	string	Nome univoco dell'immagine.
imagetype	integer	Tipo di immagine. Valori possibili: 1 - Immagine; 2 - Sfondo.
encodedImage	string	Immagine codificata in Base64.
maps		Elemento radice per mappe.

Mappe

Elemento	Tipo	Descrizione
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 del host; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Niente.
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.

Elemento	Tipo	Descrizione
highlight	integer	Abilita l'evidenziazione delle icone per i trigger attivi e gli stati del host. Valori possibili: 0 - No; 1 - Sì.
expandproblem	integer	Visualizza il trigger del problema per gli elementi con un singolo problema. Valori possibili: 0 - No; 1 - Sì.
markelements	integer	Evidenzia gli elementi della mappa che hanno cambiato recentemente il loro 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	Visualizza i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione del 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	Visualizza una griglia nella configurazione della mappa. Valori possibili: 0 - Sì; 1 - No.
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	Visualizza l'etichetta come etichetta del host. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 1 - Indirizzo IP del host; 2 - Nome dell'elemento; 3 - Solo stato; 4 - Niente; 5 - Etichetta personalizzata.
label_type_hostgroup	integer	Visualizza 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 - Niente; 5 - Etichetta personalizzata.

Elemento	Tipo	Descrizione
label_type_trigger	integer	Visualizza 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 - Niente; 5 - Etichetta personalizzata.
label_type_map	integer	Visualizza 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 - Niente; 5 - Etichetta personalizzata.
label_type_image	integer	Visualizza l'etichetta come etichetta dell'immagine. Supportato se `label_format` è impostato su 1. Valori possibili: 0 - Etichetta; 2 - Nome dell'elemento; 4 - Niente; 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 le macro nelle etichette nella configurazione della mappa. Valori possibili: 0 - No; 1 - Sì.
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 selement 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.

Elemento		Tipo	Descrizione
	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 - Alto; 2 - Basso.
	border_type	integer	Tipo di bordo della 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.
	background_color	string	Colore di sfondo (riempimento) rappresentato in codice esadecimale.
	zindex	integer	Valore per l’ordinamento di tutte le forme e 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 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).

Elemento	Tipo	Descrizione
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 del collegamento.

Note:
Vedi anche: Oggetto mappa (fare riferimento alla proprietà pertinente con un nome corrispondente).

Elementi selement della mappa

Elemento	Tipo	Descrizione
elementtype	integer	Tipo di elemento della mappa. Valori possibili: 0 - Host; 1 - Mappa; 2 - Trigger; 3 - Gruppo di host; 4 - Immagine.
elements		Elemento radice per le entità Zabbix (host, gruppo di host, mappa, ecc.) rappresentate sulla mappa. Tutte le entità hanno il rispettivo elemento (`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.
elementsubtype	integer	Sottotipo di elemento. Supportato se `elementtype` è impostato su 3. Valori possibili: 0 - Singolo gruppo di host; 1 - Tutti i gruppi di 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 - Distribuisci uniformemente 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`.

Elemento	Tipo	Descrizione
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 di host). Se vengono forniti 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:
Vedere anche: Oggetto elemento della mappa (fare riferimento alla proprietà pertinente con un nome corrispondente).

Indicatori di stato del collegamento della mappa

Elemento	Tipo	Descrizione
drawtype	integer	Stile del collegamento quando il trigger è nello stato 'Problema'. 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 'Problema'.
trigger		Elemento radice per il trigger utilizzato per indicare lo stato del collegamento.
description	string	Nome del trigger.
expression	string	Espressione del trigger.
recovery_expression	string	Espressione di ripristino del trigger.

Note:
Vedi anche: Oggetto trigger del collegamento della mappa (fare riferimento alla proprietà pertinente con un nome corrispondente).

6 Tipi di supporto

Panoramica

I tipi di supporto vengono esportati con tutti gli oggetti correlati e le relative relazioni.

Esportazione

Per esportare i tipi di supporto, procedere come segue:

1. Vai in *Avvisi* → *Tipi di media*.
2. Seleziona le caselle di controllo dei tipi di supporto da esportare.
3. Fai clic su *Esporta* sotto l'elenco.

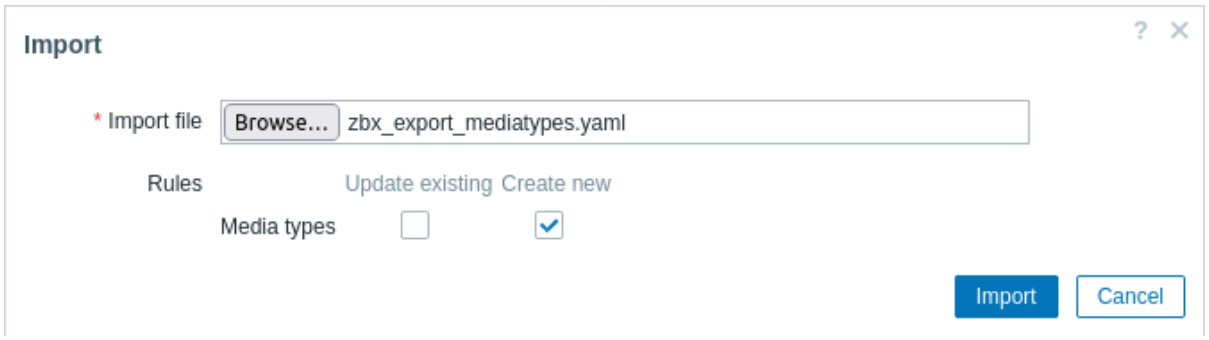
A seconda del formato selezionato, i tipi di supporto vengono esportati in un file locale con un nome predefinito:

- `zabbix_export_mediatypes.yaml` - nell'esportazione YAML (opzione predefinita per l'esportazione);
- `zabbix_export_mediatypes.xml` - nell'esportazione XML;
- `zabbix_export_mediatypes.json` - nell'esportazione JSON.

Importazione

Per importare i tipi di supporto, procedere come segue:

1. Andare su *Avvisi* → *Tipi di supporto*.
2. Fare clic su *Importa* nell'angolo superiore destro.
3. Selezionare il file di importazione.
4. Selezionare le opzioni richieste nelle regole di importazione.
5. Fare clic su *Importa* nell'angolo inferiore destro del modulo di configurazione.



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.

Nel frontend verrà visualizzato un messaggio di esito positivo o negativo dell'importazione.

Formato di esportazione

Esporta in YAML:

```yaml
zabbix_export:
  version: '8.0'
  media_types:
    - name: Pushover
      type: WEBHOOK
      parameters:
        - name: endpoint
          value: 'https://api.pushover.net/1/messages.json'
        - name: eventid
          value: '{EVENT.ID}'
        - name: event_nseverity
          value: '{EVENT.NSEVERITY}'
        - name: event_source
          value: '{EVENT.SOURCE}'
        - name: event_value
          value: '{EVENT.VALUE}'
        - name: expire
          value: '1200'
        - name: message
          value: '{ALERT.MESSAGE}'
        - name: priority_average
          value: '0'
        - name: priority_default
          value: '0'
        - name: priority_disaster
          value: '0'
        - name: priority_high
          value: '0'
        - name: priority_information
          value: '0'
```

```
    - name: priority_not_classified
      value: '0'
    - name: priority_warning
      value: '0'
    - name: retry
      value: '60'
    - name: title
      value: '{ALERT.SUBJECT}'
    - name: token
      value: '<PUSHOVER TOKEN HERE>'
    - name: triggerid
      value: '{TRIGGER.ID}'
    - name: url
      value: '{$ZABBIX.URL}'
    - name: url_title
      value: Zabbix
    - name: user
      value: '{ALERT.SENDTO}'
  status: DISABLED
  max_sessions: '0'
  script: |
    try {
        var params = JSON.parse(value),
            request = new HttpRequest(),
            data,
            response,
            severities = [
                {name: 'not_classified', color: '#97AAB3'},
                {name: 'information', color: '#7499FF'},
                {name: 'warning', color: '#FFC859'},
                {name: 'average', color: '#FFA059'},
                {name: 'high', color: '#E97659'},
                {name: 'disaster', color: '#E45959'},
                {name: 'resolved', color: '#009900'},
                {name: 'default', color: '#000000'}
            ],
            priority;

        if (typeof params.HTTPProxy === 'string' && params.HTTPProxy.trim() !== '') {
            request.setProxy(params.HTTPProxy);
        }

        if ([0, 1, 2, 3].indexOf(parseInt(params.event_source)) === -1) {
            throw 'Parametro "event_source" fornito non corretto: "' + params.event_source + '".\nDeve
        }

        if (params.event_value !== '0' && params.event_value !== '1'
            && (params.event_source === '0' || params.event_source === '3')) {
            throw 'Parametro "event_value" fornito non corretto: ' + params.event_value + '\nDeve esse
        }

        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 di almeno 30 se "priority" è
        }

        if (isNaN(params.expire) || params.expire > 10800) {
            throw 'il campo "expire" deve essere un numero con valore massimo di 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 ] Risposta ricevuta 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. Controllare 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

    Impostare il parametro token con la chiave dell'applicazione Pushover.
    Quando si assegna il tipo di supporto Pushover all'utente Zabbix, aggiungere la chiave utente nel
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problema: {EVENT.NAME}'
      message: |
        Problema iniziato alle {EVENT.TIME} del {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} del {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} ha {EVENT.UPDATE.ACTION} il problema il {EVENT.UPDATE.DATE} alle {EVENT.UPDATE
        {EVENT.UPDATE.MESSAGE}

        Lo stato attuale del problema è {EVENT.STATUS}, l'età è {EVENT.AGE}, riconosciuto: {EVENT.ACK.
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Rilevamento: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        Regola di rilevamento: {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.

Elemento	Tipo	Descrizione
name	string	(obbligatorio) Nome del tipo di supporto.
type	string	(obbligatorio) Trasporto utilizzato dal tipo di supporto. Valori possibili:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Indica se il tipo di supporto è abilitato. Valori possibili:[1] ENABLED (0, predefinito), DISABLED (1)
max_sessions	integer	Il numero massimo di avvisi che possono essere elaborati in parallelo. Valori possibili per SMS:[1] 1 (predefinito). Valori possibili per altri tipi di supporto:[1] 0-100 (dove 0 = illimitato).
attempts	integer	Il numero massimo di tentativi di invio di un avviso. Valori possibili:[1] 1-10 (predefinito: 3).
attempt_interval	string	L'intervallo (utilizzando secondi o suffisso temporale) tra i tentativi di ripetizione. Valori possibili:[1] 0-60s (predefinito: 10s).
description	string	Descrizione del tipo di supporto.
message_templates		Elemento radice per i modelli di messaggio del tipo di supporto.
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 nome corrispondente).

Email

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di supporto *Email*.

Elemento	Type	Descrizione
provider	string	Provider email.
smtp_server	string	Server SMTP.
smtp_port	integer	Porta del server SMTP a cui connettersi. Predefinita: 25.
smtp_helo	string	HELO SMTP.
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 host SSL per SMTP. Valori possibili:[1] NO (0, predefinito), YES (1).
smtp_verify_peer	string	Verifica peer SSL 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 l'autorizzazione OAuth.

Elemento	Type	Descrizione
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.
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 supporto *SMS*.

Elemento	Type	Descrizione
gsm_modem	string	(obbligatorio) Nome del dispositivo seriale del modem GSM.

Note:
Vedi anche: Oggetto tipo di supporto (fare riferimento alla proprietà pertinente con un nome corrispondente).

Script

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di supporto *Script*.

Elemento	Type	Descrizione
nome	string	(obbligatorio) Nome dello script.
script		
parameters		Elemento radice per i parametri dello script.
sortorder	string	(obbligatorio) 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 (fare riferimento alla proprietà pertinente con un nome corrispondente).

Webhook

I seguenti elementi aggiuntivi vengono esportati solo per il tipo di supporto *Webhook*.

Elemento	Type	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 eventi 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 eventi. Supporta la macro {EVENT.TAGS.*}.
event_menu_name	string	Nome della voce del menu eventi. Supporta la macro {EVENT.TAGS.*}.
parameters		Elemento radice per i parametri del tipo di supporto 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 il nome corrispondente).

Note a piè di pagina

[1] I valori interi dell'API tra parentesi, ad esempio ENABLED (0), sono riportati 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.

13 Individuazione

Utilizzare la barra laterale per accedere ai contenuti della sezione Individuazione.

1 Individuazione all'interno della rete

Panoramica

Zabbix offre una funzionalità di network discovery automatica efficace e molto flessibile.

Con la network discovery configurata correttamente è possibile:

- velocizzare il deployment di Zabbix
- semplificare l'amministrazione
- utilizzare Zabbix in ambienti in rapido cambiamento senza un'amministrazione eccessiva

La 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 cifrata)
- informazioni ricevute da SNMP agent

NON fornisce:

- rilevamento della topologia di rete

La network discovery consiste fondamentalmente in due fasi: rilevamento e azioni.

Individuazione

Zabbix esegue periodicamente la scansione degli intervalli IP definiti nelle regole di individuazione di rete. La frequenza del controllo è configurabile singolarmente per ciascuna regola.

Ogni regola dispone di un insieme di controlli dei servizi definiti da eseguire per l'intervallo IP.

Le regole di individuazione vengono elaborate dal discovery manager. Il discovery manager crea un job per ciascuna 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). 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 oppure a circa 4 GB di memoria. Se la coda si riempie, la regola di individuazione verrà saltata e nel log verrà stampato un messaggio di avviso. Puoi usare l'item interno `zabbix[discovery_queue]` per monitorare il numero di controlli di individuazione nella coda.

I controlli di individuazione 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 individuazione viene modificata durante l'esecuzione, l'esecuzione corrente dell'individuazione verrà interrotta.

Ogni controllo di un servizio e di un host (IP) eseguito dal modulo di individuazione di rete genera un evento di individuazione.

Evento	Risultato del controllo del servizio
Servizio individuato	Il servizio è 'up' dopo essere stato 'down' oppure quando viene individuato per la prima volta.
Servizio up	Il servizio è 'up', dopo essere già stato 'up'.
Servizio perso	Il servizio è 'down' dopo essere stato 'up'.
Servizio down	Il servizio è 'down', dopo essere già stato 'down'.

Evento	Risultato del controllo del servizio
Host individuato	Almeno un servizio di un host è 'up' dopo che tutti i servizi di quell'host erano 'down' oppure viene individuato 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 perso	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 essere già stati '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 di host

Un host viene aggiunto se è selezionata l'operazione *Aggiungi host*. Un host viene aggiunto anche se l'operazione *Aggiungi host* manca, se si selezionano operazioni che comportano azioni su un host. Tali operazioni sono:

- abilita host
- disabilita host
- aggiungi host a un gruppo host
- collega template a un host

Gli host creati vengono aggiunti al gruppo *Host rilevati* (per impostazione predefinita, configurabile in *Amministrazione* > *Generale* > *Altro*). Se si desidera che gli host vengano aggiunti a un altro gruppo, aggiungere un'operazione *Rimuovi dai gruppi host* (specificando "Host rilevati") e aggiungere anche un'operazione *Aggiungi ai gruppi host* (specificando un altro gruppo host), perché un host deve appartenere a un gruppo host.

L'indirizzo IP del dispositivo rilevato, insieme alla sorgente di discovery (Zabbix server, Zabbix proxy o gruppo proxy) e al tipo di interfaccia, viene utilizzato 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 la destinazione per l'esecuzione delle operazioni. Se la sorgente di discovery è diversa, l'entità rilevata viene trattata come 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 rilevamento 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 durante l'aggiunta degli host

Quando gli host vengono aggiunti come risultato del rilevamento di rete, le relative interfacce vengono create secondo 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 di Zabbix agent sia a quelle SNMP, verranno creati entrambi i tipi di interfaccia.
- Se i criteri di unicità sono dati restituiti da Zabbix agent o SNMP, la prima interfaccia trovata per un host verrà creata come interfaccia predefinita. Gli altri indirizzi IP verranno aggiunti come interfacce aggiuntive. Le condizioni dell'azione (come Host IP) non influiscono sull'aggiunta delle interfacce. Si noti 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 risponde solo ai controlli agent, verrà creato solo con un'interfaccia agent. Se in seguito iniziasse a rispondere 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 dei criteri di unicità identico, B e C vengono creati come interfacce aggiuntive per A, il primo host. I singoli host B e C rimangono. In *Monitoring* > *Discovery* le interfacce aggiunte verranno visualizzate nella colonna "Discovered device", in nero e rientrate, ma 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 individuati 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, tale 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 comportare 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 individuati e all'appartenenza al gruppo di proxy padre per evitare duplicati.

Ad esempio, i passaggi per sostituire il proxy in una regola di discovery sono:

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 individuati 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

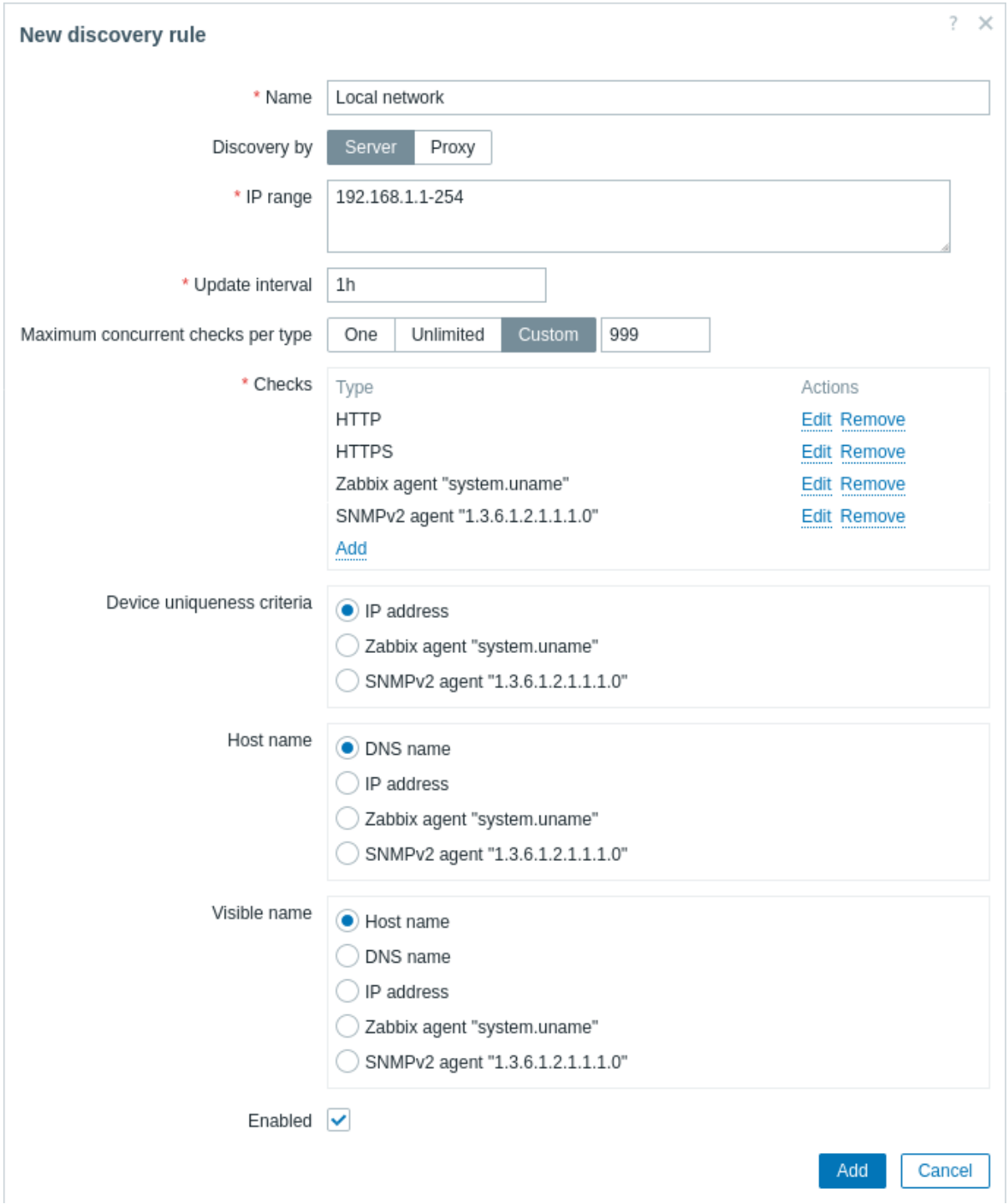
1 Configurazione di una regola di individuazione sulla rete

Panoramica

Per configurare una regola di individuazione della rete utilizzata da Zabbix per individuare host e servizi:

- Vai a *Raccolta dati* → *Individuazione*
- Fai clic su *Crea regola di individuazione* (oppure sul nome della regola per modificarne una esistente)
- Modifica gli attributi della regola di individuazione

Attributi della regola



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

Parametro	Descrizione
Name	Nome univoco della regola. Ad esempio, "Rete locale".
Discovery by	Il rilevamento viene eseguito da: **Server** - da Zabbix server **Proxy** - da Zabbix proxy (selezionato nel campo nome del proxy)
IP range	L'intervallo di indirizzi IP per il rilevamento. 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 (inferiore a 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.

Parametro	Descrizione
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 rilevamento, quindi non vi è sovrapposizione. Sono supportati i suffissi temporali, ad esempio 30s, 1m, 2h, 1d. Sono supportate le macro utente. *Nota*: se viene utilizzata 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 tempo, nell'esempio indicato).
Maximum concurrent checks per type	Imposta il numero massimo di thread di rilevamento (worker) per ciascun controllo di servizio per l'elaborazione parallela dei controlli di rilevamento: **One** - un thread **Unlimited** - numero illimitato di thread (ma non superiore a quanto impostato nel parametro StartDiscoverers) **Custom** - imposta un numero personalizzato di thread (0-999) Si noti che tutte le regole di rilevamento con controlli di servizio asincroni SNMPv3 vengono elaborate da un solo worker a causa delle particolarità dell'implementazione di libsnmp, cioè aumentare il numero di worker non aumenterà la velocità di rilevamento.
Checks	Zabbix utilizzerà questo elenco di controlli per il rilevamento. Fare 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, agent SNMPv1, agent SNMPv2, agent SNMPv3, ping ICMP. Un rilevamento basato 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, vedere item 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 più dispositivi con singolo IP. Se esiste già un dispositivo con lo stesso IP, verrà considerato come già rilevato e non verrà aggiunto un nuovo host. **<discovery check>** - controllo di Zabbix agent oppure controllo agent SNMP. Si noti che il criterio di unicità utilizzato durante il rilevamento non è lo stesso dell'identificazione dell'host nel sistema durante l'esecuzione delle azioni. I criteri di unicità durante il rilevamento definiscono se due o più dispositivi rilevati sono uguali (o diversi), mentre solo l'indirizzo IP è il criterio per l'identificazione dell'host in Zabbix (vedere Creazione host).
Host name	Imposta il nome tecnico dell'host creato utilizzando: **DNS name** - nome DNS (predefinito) **IP address** - indirizzo IP **<discovery check>** - valore stringa ricevuto dal controllo di rilevamento (ad esempio controllo Zabbix agent, controllo agent SNMP) Vedere anche: Denominazione host.
Visible name	Imposta il nome visibile dell'host creato utilizzando: **Host name** - nome tecnico dell'host (predefinito) **DNS name** - nome DNS **IP address** - indirizzo IP **<discovery check>** - valore stringa ricevuto dal controllo di rilevamento (ad esempio controllo Zabbix agent, controllo agent SNMP) Vedere anche: Denominazione host.
Enabled	Con la casella 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 ci si avvicina a questo limite, 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

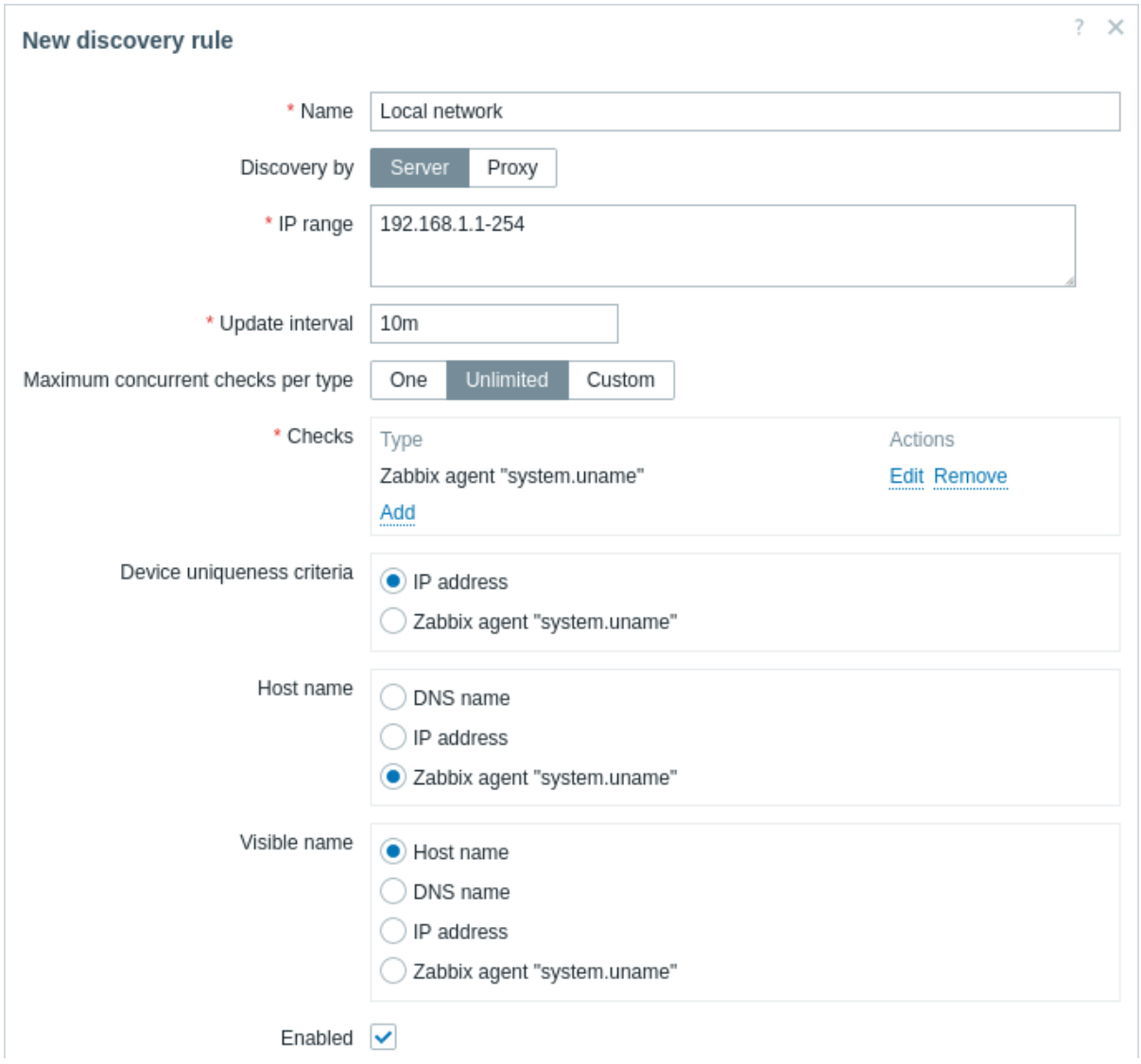
In questo esempio, desideriamo configurare il rilevamento della rete per la rete locale con un intervallo IP di 192.168.1.1-192.168.1.254.

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.



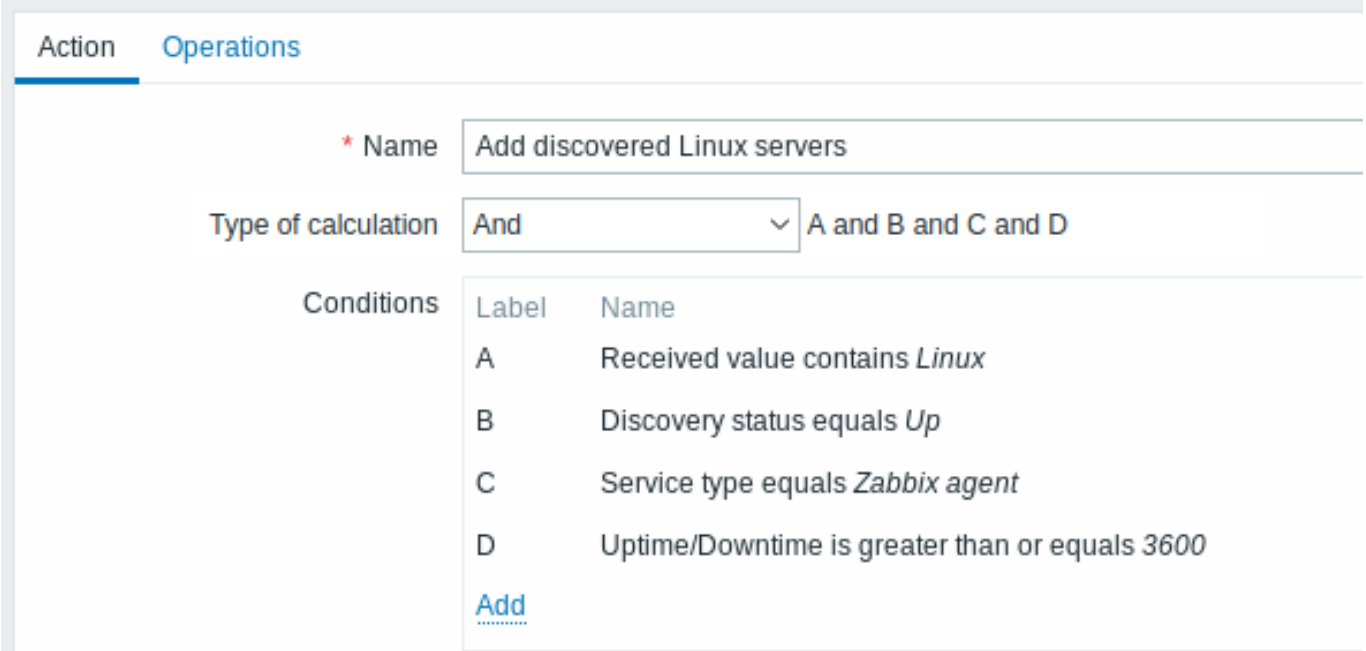
Zabbix proverà a individuare gli host nell'intervallo IP 192.168.1.1-192.168.1.254 collegandosi agli agent Zabbix e ottenendo il valore dalla chiave **system.uname**. Il valore ricevuto dall'agent può essere utilizzato per assegnare un nome agli host e anche per applicare azioni diverse per sistemi operativi 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.

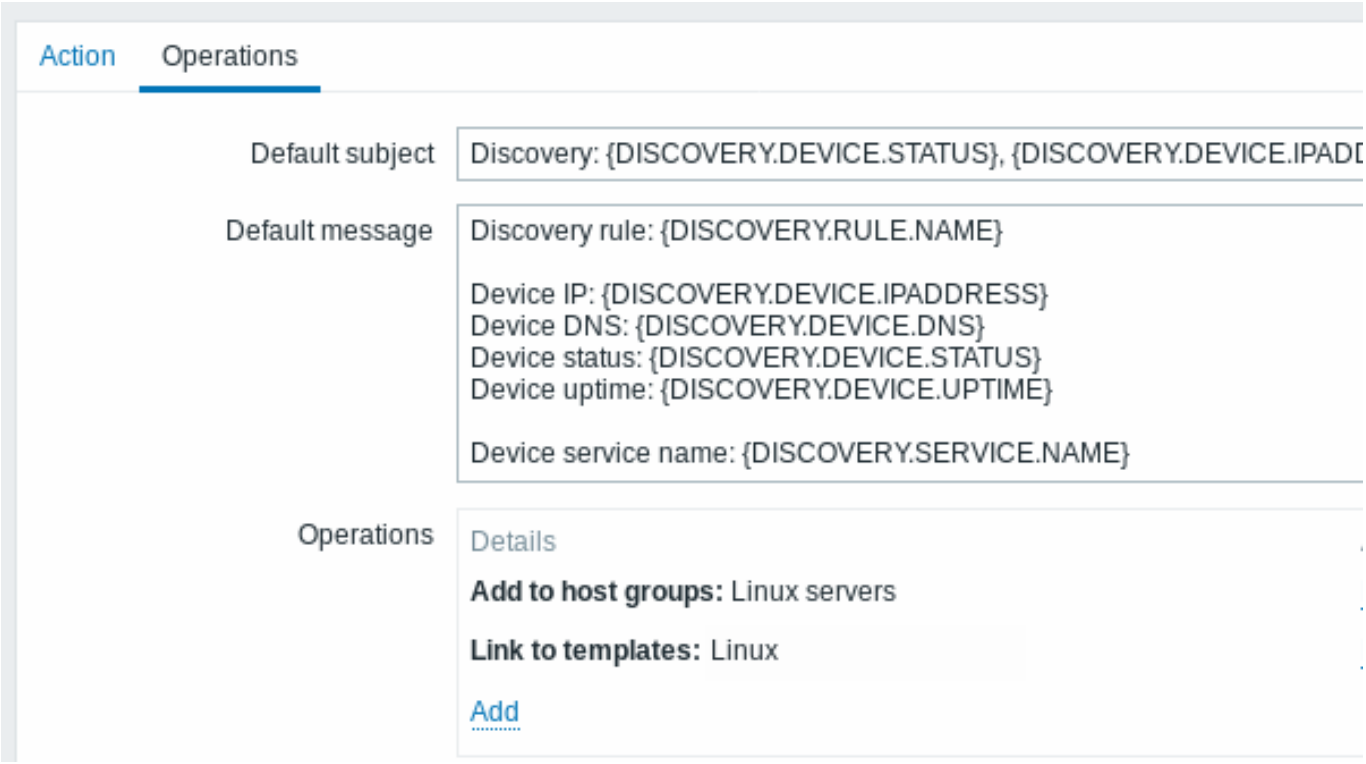
Passaggio 2

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

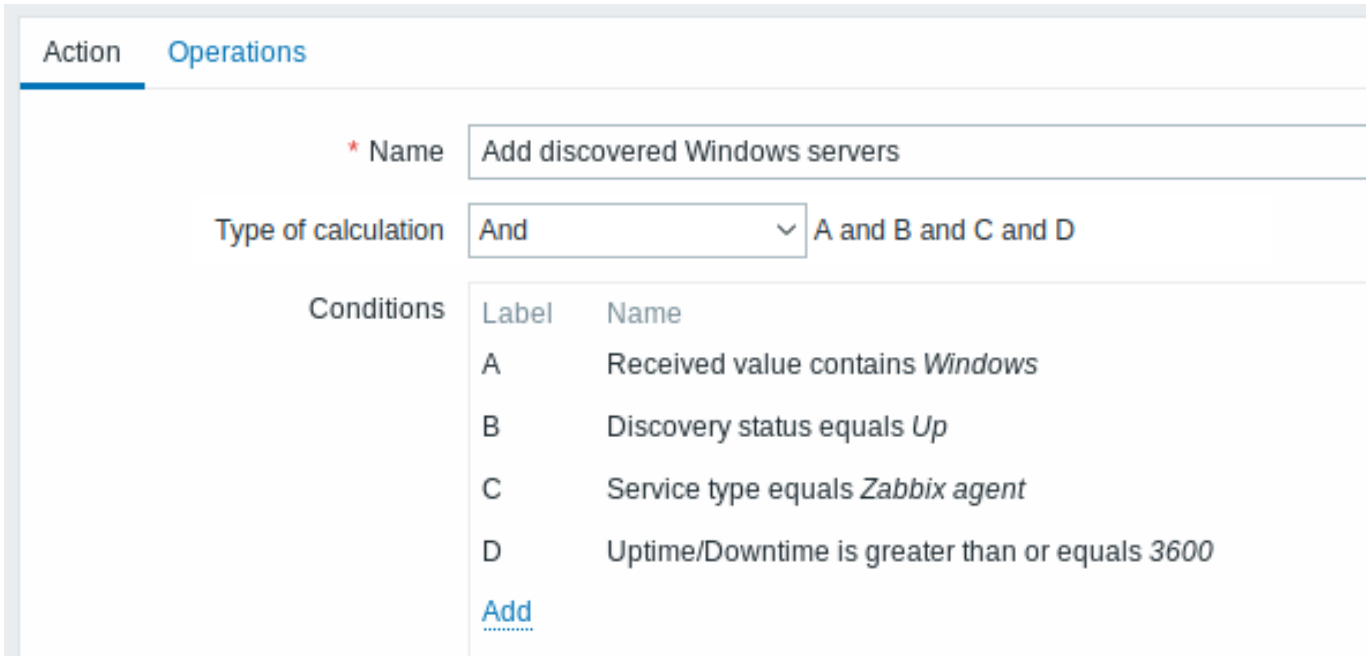


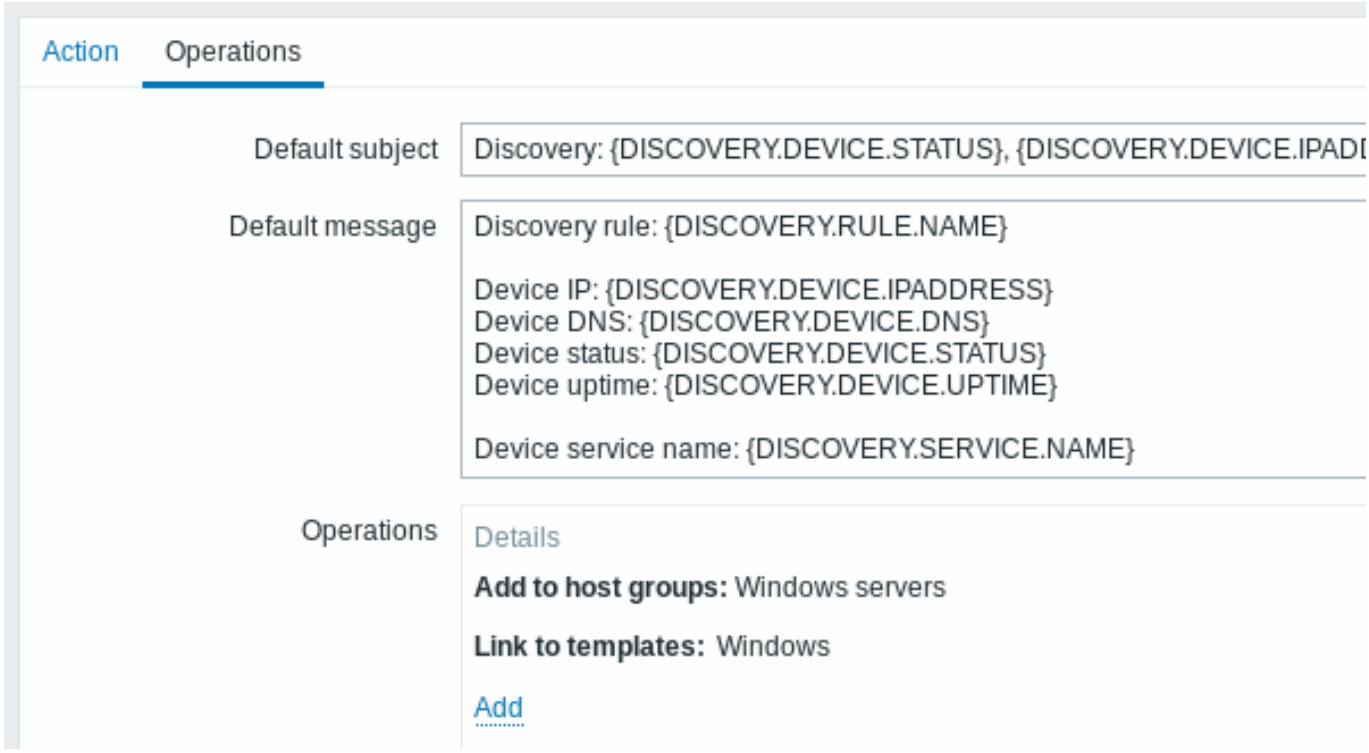
L'azione eseguirà le seguenti operazioni:

- 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

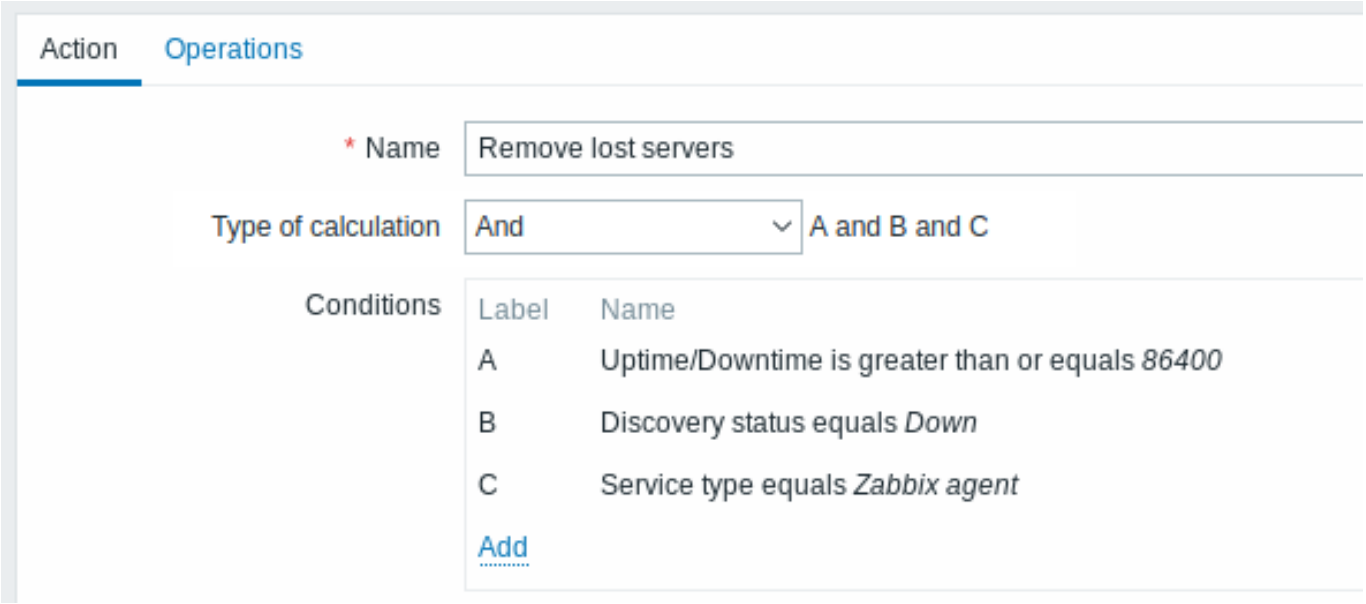
Definizione di un’azione di discovery per aggiungere i server Windows rilevati al rispettivo gruppo/template.

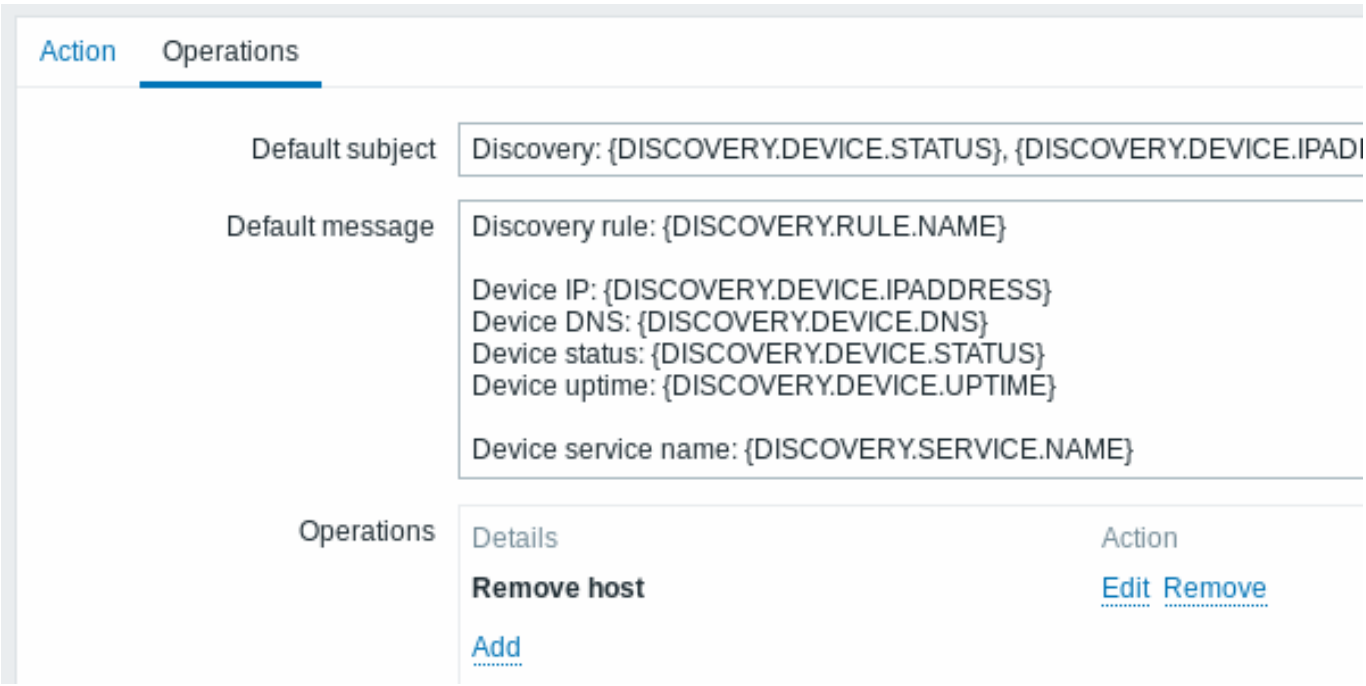




Passaggio 4

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).

2 Registrazione automatica dell’agente attivo

Panoramica

È possibile consentire l’autoregistrazione dell’agent Zabbix attivo, dopo la quale il server può iniziare a monitorarlo. In questo modo è possibile aggiungere nuovi host al monitoraggio senza configurarli manualmente sul server.

L’autoregistrazione può avvenire quando un agent attivo precedentemente sconosciuto richiede i controlli.

Questa funzionalità può essere molto utile per il monitoraggio automatico di nuovi nodi Cloud. Non appena si dispone di un nuovo nodo nel Cloud, Zabbix inizierà automaticamente a raccogliere i dati sulle prestazioni e sulla 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, a condizione che i parametri di configurazione `ListenIP` o `ListenPort` siano definiti nel file di configurazione, questi vengono inviati al server. Se sono specificati più indirizzi IP, il primo viene inviato al server.

Quando aggiunge il nuovo host autoregistrato, il server utilizza l’indirizzo IP e la porta ricevuti per configurare l’agent. Se non viene ricevuto alcun valore di indirizzo IP, viene utilizzato quello usato per la connessione in ingresso. Se non viene ricevuto alcun valore di porta, viene utilizzato 10050.

È possibile specificare che l'host debba essere autoregistrato con un nome DNS come interfaccia agent predefinita.

L'autoregistrazione viene eseguita nuovamente:

- se cambiano le informazioni dei metadati dell'host:
  - perché HostMetadata è stato modificato e l'agent è stato riavviato
  - perché è cambiato il 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'heartbeat di autoregistrazione dell'agent attivo per Zabbix server e Zabbix proxy è di 120 secondi. Quindi, nel caso in cui un host individuato venga eliminato, l'autoregistrazione verrà eseguita nuovamente dopo 120 secondi.

Configurazione

Specificare il server

Assicurarsi di avere il server Zabbix 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 nome host di sistema della posizione dell'agent verrà utilizzato dal server per assegnare il nome al host. Il nome host di sistema in Linux può essere ottenuto eseguendo il comando `hostname`.

Se *Hostname* è definito nella configurazione dell'agent Zabbix come un elenco di host delimitato da virgole, verranno creati host per tutti i nomi host elencati.

Riavviare l'agent dopo aver apportato qualsiasi modifica al file di configurazione.

Azione per l'autoregistrazione attiva dell'agent

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 evento "Autoregistration".

> **Note:**
> La configurazione del rilevamento di rete non è necessaria per consentire agli agent attivi di autoregistrarsi.

Nel frontend di Zabbix, andare su *Avvisi* → *Azioni*, selezionare *Azioni di autoregistrazione* e fare clic su *Crea azione*:

- Nella scheda Azione, assegnare un nome all'azione
- Facoltativamente, specificare le condizioni. È possibile eseguire una corrispondenza per sottostringa o tramite espressione regolare nelle condizioni per nome host/metadati host. Se si intende utilizzare la condizione "Metadati host", vedere la sezione successiva.
- Nella scheda Operazioni, aggiungere le operazioni pertinenti, ad esempio: 'Aggiungi host', 'Aggiungi al gruppo host' (per esempio, *Host rilevati*), 'Collega template', ecc.

> **Note:**
> Se è probabile che gli host che si autoregistreranno siano supportati solo per il monitoraggio attivo (ad esempio host protetti da firewall rispetto al server Zabbix), potrebbe essere utile creare un template specifico come *Template_Linux-active* da collegare.

Gli host creati vengono aggiunti al gruppo *Host rilevati* (per impostazione predefinita, configurabile in *Amministrazione* > *Generale* > *Altro*). Se si desidera che gli host vengano aggiunti a un altro gruppo, aggiungere un'operazione *Rimuovi dal gruppo host* (specificando "Host rilevati") e aggiungere anche un'operazione *Aggiungi al gruppo host* (specificando un altro gruppo host), perché un host deve appartenere a un gruppo host.

Autoregistrazione sicura

Un metodo sicuro di autoregistrazione è possibile configurando l'autenticazione basata su PSK con connessioni cifrate.

Il livello di cifratura viene configurato globalmente in *Administration* > *General* > *Autoregistration*. È possibile selezionare nessuna cifratura, la cifratura TLS con autenticazione PSK oppure entrambe (in modo che alcuni host possano registrarsi senza cifratura mentre altri tramite connessione cifrata).

L'autenticazione tramite PSK viene verificata da Zabbix server prima di aggiungere un host. Se la verifica ha esito positivo, l'host viene aggiunto e *Connections from/to host* viene impostato solo su 'PSK' con identità/chiave precondivisa uguale a quella dell'impostazione globale di autoregistrazione.

Attention:
Per garantire la sicurezza dell'autoregistrazione nelle installazioni che utilizzano proxy, deve essere abilitata la cifratura tra Zabbix server e proxy.

Utilizzo di DNS come interfaccia predefinita

I parametri di configurazione HostInterface e HostInterfaceItem 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 suo indirizzo IP. In tal caso, il nome DNS deve essere specificato o restituito come valore 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 riavvii il processo di autoregistrazione.

Note:
Se i parametri HostInterface o HostInterfaceItem non sono configurati, viene invece 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, ciò può causare un'autoregistrazione errata o non riuscita a causa di un valore dell'interfaccia non valido.

Utilizzo dei metadati del host

Quando l'agent invia una richiesta di autoregistrazione al server, invia il proprio hostname. In alcuni casi (ad esempio, nodi cloud Amazon) un hostname non è sufficiente perché il server Zabbix distingua i host rilevati. I metadati del host possono essere utilizzati facoltativamente per inviare altre informazioni da un agent al server.

I metadati del host si configurano nel file di configurazione dell'agent - `zabbix_agentd.conf`. Esistono 2 modi per specificare i metadati del host nel file di configurazione:

```
HostMetadata
HostMetadataItem
```

Vedere la descrizione delle opzioni nel link sopra.

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 a 3 byte sarà limitato a 21844 caratteri totali, mentre un valore contenente solo caratteri a 4 byte sarà limitato a 16383 simboli.

Attention:
Un tentativo di autoregistrazione avviene ogni volta che un agent attivo invia al server una richiesta di aggiornamento dei 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.

3 Individuazione di basso livello

Panoramica Low-level discovery (LLD) fornisce un modo per creare automaticamente item, trigger e grafici per diverse entità su un host. Ad esempio, Zabbix può iniziare automaticamente a monitorare file system o interfacce di rete sulla tua macchina, senza la necessità di creare manualmente item per ogni file system o interfaccia di rete. LLD può anche creare host, ad esempio per popolare macchine virtuali rilevate su un hypervisor, e regole di discovery annidate, abilitando una discovery multilivello. Inoltre, è possibile configurare Zabbix per rimuovere automaticamente le entità non necessarie in base ai risultati effettivi della discovery eseguita periodicamente.

Un utente può definire i propri tipi di discovery, a condizione che seguano un particolare protocollo JSON.

L'architettura generale del processo di discovery è la seguente.

Per prima cosa, un utente crea una regola di discovery in *Data collection* > *Templates*, nella colonna *Discovery*. Una regola di discovery consiste in (1) un item che rileva le entità necessarie (ad esempio file system o interfacce di rete) e (2) prototipi di item, trigger e grafici che devono essere creati in base al valore di quell'item.

Un item che rileva le entità necessarie è simile a un normale item visibile altrove: il server chiede a uno Zabbix agent (o a qualunque tipo sia impostato per l'item) il valore di quell'item, l'agent risponde con un valore testuale. La differenza è che il valore restituito dall'agent deve contenere un elenco di entità rilevate in formato JSON. Sebbene i dettagli di questo formato siano importanti solo per gli implementatori di controlli di discovery personalizzati, è necessario sapere che il valore restituito contiene un elenco di coppie macro → valore. Ad esempio, l'item "net.if.discovery" potrebbe restituire due coppie: "{#IFNAME}" → "lo" e "{#IFNAME}" → "eth0".

Queste macro vengono utilizzate nei nomi, nelle key e in altri campi dei prototipi, dove vengono poi sostituite con i valori ricevuti per creare item, trigger, grafici o persino host reali per ogni entità rilevata. Vedi l'elenco completo delle opzioni per l'utilizzo delle macro LLD.

Quando il server riceve un valore per un item di discovery, esamina le coppie macro → valore e per ciascuna coppia genera item, trigger e grafici reali, in base ai rispettivi prototipi. Nell'esempio con "net.if.discovery" sopra, il server genererebbe un insieme di item, trigger e grafici per l'interfaccia loopback "lo" e un altro insieme per l'interfaccia "eth0".

Si noti che a partire da **Zabbix 4.2**, il formato del JSON restituito dalle regole di low-level discovery è stato modificato. Non è più previsto che il JSON contenga l'oggetto "data". Low-level discovery ora accetta un normale JSON contenente un array, per supportare nuove funzionalità come il preprocessing del valore dell'item e percorsi personalizzati verso i valori delle macro di low-level discovery in un documento JSON.

Le key di discovery integrate sono state aggiornate per restituire un array di righe LLD alla radice del documento JSON. Zabbix estrarrà automaticamente una macro e un valore se un campo dell'array usa la sintassi {#MACRO} come key. Qualsiasi nuovo controllo di discovery nativo utilizzerà la nuova sintassi senza gli elementi "data". Durante l'elaborazione di un valore di low-level discovery, viene prima individuata la radice (array in `$.` o `$.data`).

Sebbene l'elemento "data" sia stato rimosso da tutti gli item nativi relativi alla discovery, per compatibilità con le versioni precedenti Zabbix continuerà ad accettare la notazione JSON con un elemento "data", anche se il suo utilizzo è sconsigliato. Se il JSON contiene un oggetto con un solo elemento array "data", il contenuto dell'elemento verrà estratto automaticamente usando JSONPath `$.data`. Low-level discovery ora accetta macro LLD opzionali definite dall'utente con un percorso personalizzato specificato nella sintassi JSONPath.

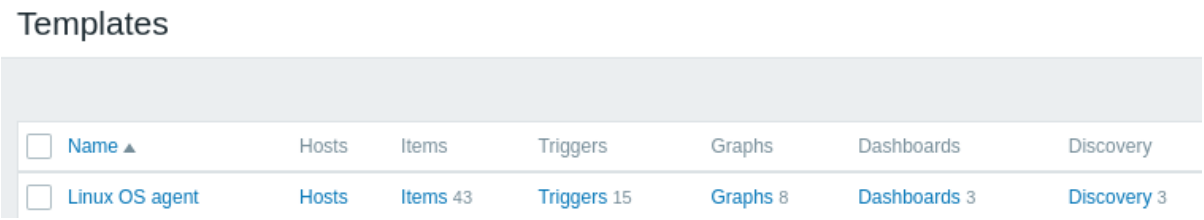
> **Warning:**
> Come risultato delle modifiche sopra indicate, gli agent più recenti non saranno più in grado di funzionare con un server Zabbix meno recente.

Vedi anche: Entità rilevate

Configurazione del low-level discovery Illustreremo il low-level discovery con un esempio di individuazione dei file system.

Per configurare l'individuazione, procedere come segue:

- Andare in: *Data collection* > *Templates* oppure *Hosts*.
- Fare clic su *Discovery* nella riga del template/host appropriato.



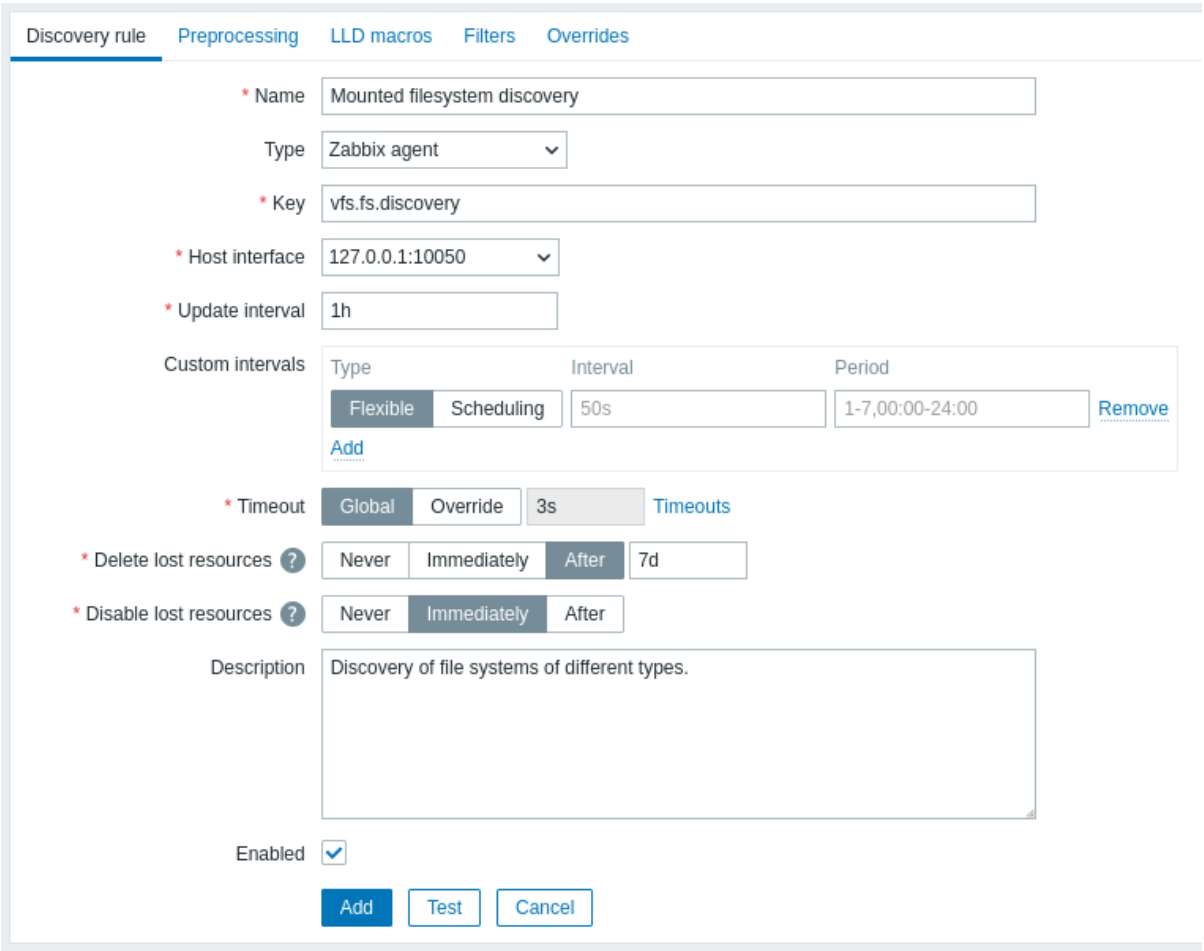
- Fare clic su *Create discovery rule* nell'angolo superiore destro dello schermo.
- Compilare il modulo della regola di individuazione con i dettagli richiesti.

Regola di individuazione

Il modulo della regola di individuazione contiene cinque schede, che rappresentano, da sinistra a destra, il flusso dei dati durante l'individuazione:

- *Regola di individuazione* - specifica, soprattutto, l'item integrato o lo script personalizzato per recuperare i dati di individuazione.
- *Preprocessing* - applica un certo preprocessing ai dati individuati.
- *Macro LLD* - consente di estrarre alcuni valori di macro da utilizzare negli item, trigger, ecc. individuati.
- *Filtri* - consente di filtrare i valori individuati.
- *Override* - consente di modificare item, trigger, grafici o prototipi di host quando vengono applicati a specifici oggetti individuati.

La scheda **Regola di individuazione** contiene la chiave item da utilizzare per l'individuazione (oltre ad alcuni attributi generali della regola di individuazione):



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

Parameter	Description
Name	Nome della regola di individuazione.
Type	Il tipo di controllo per eseguire l'individuazione. In questo esempio stiamo utilizzando un tipo di item *Zabbix agent*. La regola di individuazione può anche essere un item dipendente, in dipendenza da un item normale. Non può dipendere da un'altra regola di individuazione. Per un item dipendente, selezionare il tipo corrispondente (*Item dipendente*) e specificare l'item principale nel campo 'Master item'. L'item principale deve esistere.
Key	Inserire la chiave item di individuazione (fino a 2048 caratteri). Ad esempio, è possibile utilizzare la chiave item integrata "vfs.fs.discovery" per restituire una stringa JSON con l'elenco dei file system presenti nel computer, i relativi tipi e le opzioni di montaggio. Si noti che un'altra opzione per l'individuazione del file system consiste nell'utilizzare i risultati di individuazione tramite la chiave agent "vfs.fs.get" (vedere esempio).
Update interval	Questo campo specifica con quale frequenza Zabbix esegue l'individuazione. All'inizio, quando si sta appena configurando l'individuazione del file system, si potrebbe voler impostare un intervallo ridotto, ma una volta verificato che funziona è possibile impostarlo a 30 minuti o più, poiché i file system di solito 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 durante la durata dell'intervallo personalizzato. Le nuove regole di individuazione verranno controllate entro 60 secondi dalla loro creazione, a meno che non abbiano Scheduling o Flexible update interval e che *Update interval* sia impostato su 0. *Nota* che per una regola di individuazione esistente l'individuazione può essere eseguita immediatamente premendo il pulsante *Execute now*.
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 di polling personalizzata. Per informazioni dettagliate vedere Intervalli personalizzati.
Timeout	Impostare il timeout del controllo di individuazione. Selezionare l'opzione di timeout: **Global** - viene utilizzato il timeout globale/del proxy (visualizzato nel campo *Timeout* disattivato); **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). Si noti che il collegamento *Timeouts* è visibile solo agli utenti di tipo *Super admin* con autorizzazioni per le sezioni frontend *Administration* > *General* o *Administration* > *Proxies*.
Delete lost resources	Specificare dopo quanto tempo l'entità individuata verrà eliminata una volta che il suo stato di individuazione diventa "Non più individuata": *Never* - non verrà eliminata; *Immediately* - verrà eliminata immediatamente; *After* - verrà eliminata dopo il periodo di tempo specificato. Il valore deve essere maggiore del valore di *Disable lost resources*. Sono supportati i suffissi temporali, ad esempio 2h, 1d. Sono supportate le macro utente. *Nota*: l'uso di "Immediately" non è consigliato, poiché una modifica errata del filtro potrebbe comportare l'eliminazione dell'entità insieme a tutti i dati storici. Si noti che le risorse disabilitate manualmente non verranno eliminate dal low-level discovery.

Parameter	Description
Disable lost resources	Specificare dopo quanto tempo l'entità individuata verrà disabilitata una volta che il suo stato di individuazione diventa "Non più individuata": *Never* - non verrà disabilitata; *Immediately* - verrà disabilitata immediatamente; *After* - verrà disabilitata dopo il periodo di tempo specificato. Il valore deve essere maggiore dell'intervallo di aggiornamento della regola di individuazione. Si noti che le risorse disabilitate automaticamente verranno nuovamente abilitate se vengono individuate di nuovo dal low-level discovery. Le risorse disabilitate manualmente non verranno nuovamente abilitate se vengono individuate di nuovo. Questo campo non viene visualizzato se *Delete lost resources* è impostato su "Immediately". Sono supportati i suffissi temporali, ad esempio 2h, 1d. Sono supportate le macro utente.
Description	Inserire una descrizione.
Enabled	Se selezionato, la regola verrà elaborata.

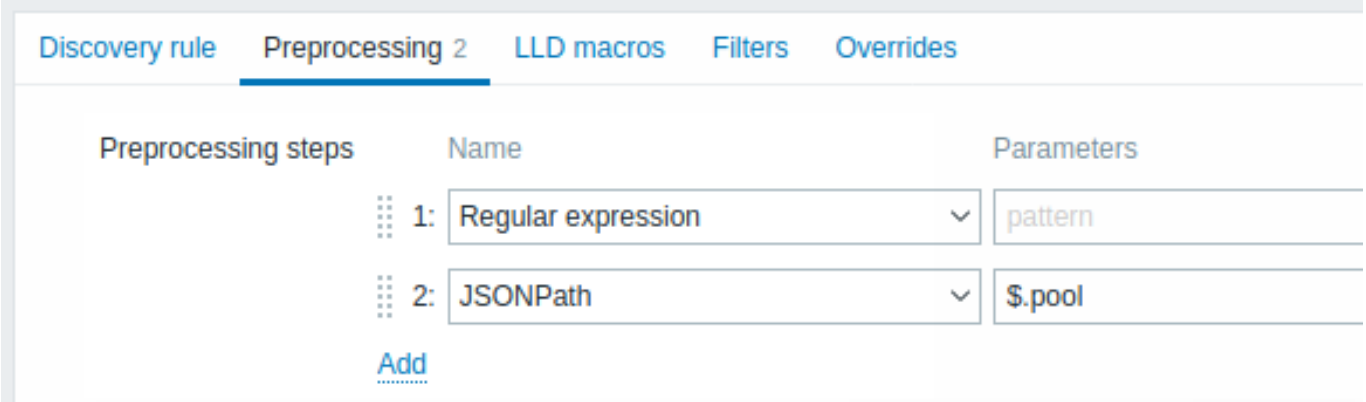
Note:
La cronologia della regola di individuazione non viene conservata.

Pre-elaborazione

La scheda **Pre-elaborazione** consente di definire regole di trasformazione da applicare al risultato del discovery. In questo passaggio è possibile usare una o più trasformazioni. Le trasformazioni vengono eseguite nell'ordine in cui sono definite. Tutta la pre-elaborazione viene eseguita da Zabbix server.

Vedi anche:

- Dettagli sulla pre-elaborazione
- Test della pre-elaborazione



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 catturati con la sequenza \N. Parametri: **pattern** - espressione regolare **output** - modello 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 selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

Tipo	
Sostituisci	Trova la stringa di ricerca e la sostituisce con un'altra stringa (o con nulla). Tutte le occorrenze della stringa di ricerca verranno sostituite. Parametri: **search string** - la stringa da trovare e sostituire, con distinzione tra maiuscole e minuscole (obbligatorio) **replacement** - la 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 usare sequenze di escape per cercare o sostituire interruzioni di riga, ritorni a capo, tabulazioni e spazi "\n \r \t \s"; il backslash può essere sottoposto a escape come "\\" e le sequenze di escape possono essere sottoposte a escape come "\\n". L'escape di interruzioni di riga, ritorni a capo e tabulazioni viene eseguito automaticamente durante il low-level discovery.
Dati strutturati	
JSONPath	Estrae un valore o un frammento da dati JSON usando la funzionalità JSONPath. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

XML XPath	Estrae un valore o un frammento da dati XML usando la funzionalità XPath.
	Affinché questa opzione funzioni, Zabbix server deve essere compilato con supporto 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>`
	Nota che i namespace non sono supportati.
	Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

CSV to JSON	Converte i dati di un file CSV in formato JSON. Per maggiori informazioni, vedi: Pre-elaborazione CSV to JSON.
XML to JSON	Converte dati in formato XML in JSON. Per maggiori informazioni, vedi: Regole di serializzazione. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
SNMP	
SNMP walk value	Estrae il valore tramite il nome OID/MIB specificato e applica opzioni di formattazione: **Unchanged** - restituisce Hex-STRING come stringa esadecimale senza escape (*nota* che gli hint di visualizzazione vengono comunque applicati); **UTF-8 from Hex-STRING** - converte Hex-STRING in una stringa UTF-8; **MAC from Hex-STRING** - converte Hex-STRING in una stringa di indirizzo MAC (in cui `' '` verrà sostituito da `':'`); **Integer from BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad es. "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 selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
SNMP walk to JSON	Converte i valori SNMP in JSON. Specifica un nome di campo nel JSON e il corrispondente percorso OID SNMP. I valori dei campi verranno popolati con i valori presenti nel percorso OID SNMP specificato. Puoi usare questo passaggio di pre-elaborazione per il discovery OID SNMP. Sono disponibili opzioni di formattazione dei valori simili a quelle del passaggio *SNMP walk value*. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.

Tipo	
SNMP get value	Applica opzioni di formattazione al valore ottenuto con SNMP get: **UTF-8 from Hex-STRING** - converte Hex-STRING in una stringa UTF-8; **MAC from Hex-STRING** - converte Hex-STRING in una stringa di indirizzo MAC (in cui `' '` verrà sostituito da `':'`); **Integer from BITS** - converte i primi 8 byte di una stringa di bit espressa come sequenza di caratteri esadecimali (ad es. "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 selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Script personalizzati	
JavaScript	Inserisci il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona della matita accanto ad esso. Nota che la lunghezza disponibile per JavaScript dipende dal database utilizzato. Per maggiori informazioni, vedi: Pre-elaborazione Javascript
Validazione	
Non corrisponde all'espressione regolare	Specifica un'espressione regolare a cui un valore non deve corrispondere. Ad es. `Error:(.*?)\.` Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Controlla errore in JSON	Controlla un messaggio di errore a livello applicativo situato in JSONPath. Interrompe l'elaborazione se il controllo ha esito positivo e il messaggio non è vuoto; altrimenti continua l'elaborazione con il valore presente prima di questo passaggio di pre-elaborazione. Nota che questi errori di servizi esterni vengono riportati all'utente così come sono, senza aggiungere informazioni sul passaggio di pre-elaborazione. Ad es. `$.errors`. Se viene ricevuto un JSON come `{"errors":"e1"}`, il passaggio successivo di pre-elaborazione non verrà eseguito. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Controlla errore in XML	Controlla un messaggio di errore a livello applicativo situato in XPath. Interrompe l'elaborazione se il controllo ha esito positivo e il messaggio non è vuoto; altrimenti continua l'elaborazione con il valore presente prima di questo passaggio di pre-elaborazione. Nota che questi errori di servizi esterni vengono riportati all'utente così come sono, senza aggiungere informazioni sul passaggio di pre-elaborazione. Non verrà segnalato alcun errore in caso di mancata analisi di XML non valido. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Corrisponde all'espressione regolare	Specifica un'espressione regolare a cui un valore deve corrispondere. Se selezioni la casella *Personalizzato in caso di errore*, è possibile specificare opzioni personalizzate di gestione degli errori: scartare il valore, impostare un valore specificato oppure impostare un messaggio di errore specificato.
Limitazione	
Scarta invariato con heartbeat	Scarta un valore se non è cambiato entro il periodo di tempo definito (in secondi). Per specificare i secondi sono supportati valori interi positivi (minimo - 1 secondo). In questo campo possono essere usati suffissi temporali (ad es. 30s, 1m, 2h, 1d). In questo campo possono essere usate macro utente e macro di low-level discovery. Per un item di discovery può essere specificata una sola opzione di limitazione. Ad es. `1m`. Se un testo identico viene passato a questa regola due volte entro 60 secondi, verrà scartato. *Nota*: la modifica dei prototipi di item non reimposta la limitazione. La limitazione viene reimpostata solo quando vengono modificati i passaggi di pre-elaborazione.
Prometheus	

Tipo	
Prometheus to JSON	Converte le metriche Prometheus richieste in JSON. Per maggiori dettagli, vedi Controlli Prometheus.

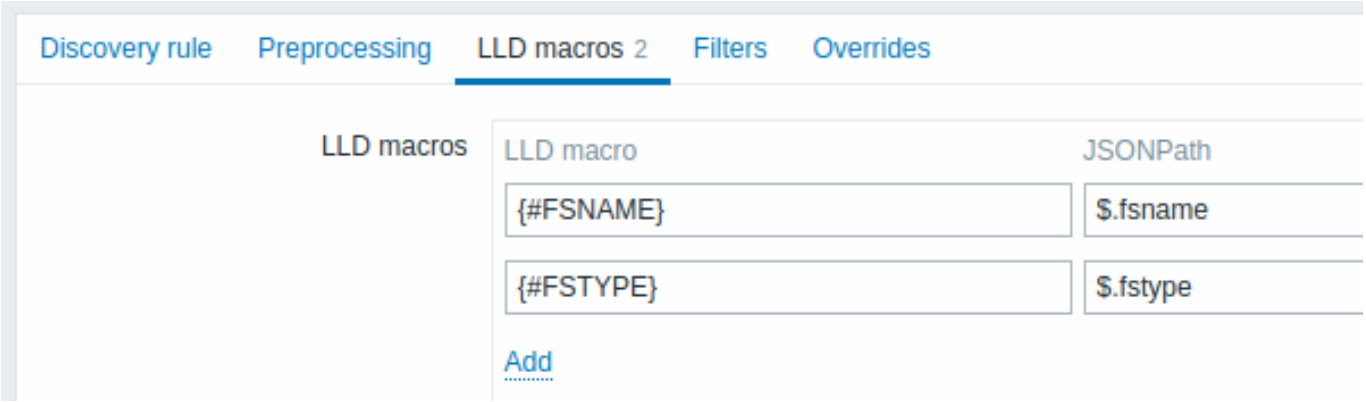
Nota che se la regola di discovery è stata applicata al host tramite template, il contenuto di questa scheda è di sola lettura.

Macro personalizzate

La scheda **Macro LLD** consente di specificare macro personalizzate per il low-level discovery.

Le macro personalizzate sono utili nei casi in cui il JSON restituito non abbia già definite le macro richieste. Quindi, ad esempio:

- La chiave nativa `vfs.fs.discovery` per il rilevamento dei filesystem restituisce un JSON con alcune macro LLD predefinite come {#FSNAME}, {#FSTYPE}. Queste macro possono essere utilizzate direttamente nei prototipi di item e trigger (vedere le sezioni successive della pagina); non è necessario definire macro personalizzate;
- Anche l'item agent `vfs.fs.get` restituisce un JSON con i dati del filesystem, ma senza alcuna macro LLD predefinita. In questo caso è possibile definire personalmente le macro e associarle ai valori nel JSON utilizzando JSONPath:

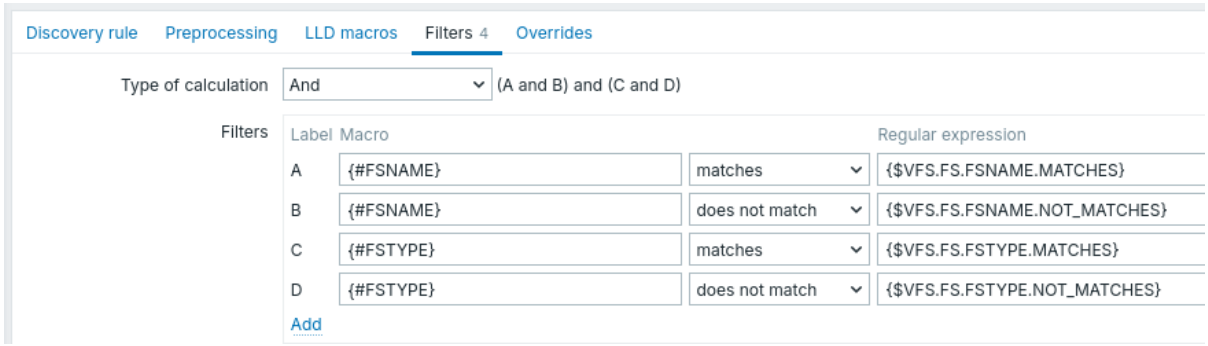


I valori estratti possono essere utilizzati negli item rilevati, nei trigger, ecc. Si noti che i valori verranno estratti dal risultato del rilevamento e da tutti i passaggi di preprocessing eseguiti fino a quel momento.

Parameter	Description
LLD macro	Nome della macro di low-level discovery, utilizzando la seguente sintassi: {#MACRO}.
JSONPath	Percorso utilizzato per estrarre il valore della macro LLD da una riga LLD, usando la sintassi JSONPath. I valori estratti dal JSON restituito vengono utilizzati per sostituire le macro LLD nei campi dei prototipi di item, trigger, ecc. JSONPath può essere specificato utilizzando la notazione con punto oppure la notazione con parentesi. La notazione con parentesi deve essere utilizzata in presenza di caratteri speciali e Unicode, come `$['unicode + special chars #1']['unicode + special chars #2']`. Ad esempio, `$.foo` estrarrà "bar" e "baz" da questo JSON: `[{"foo":"bar"}, {"foo":"baz"}]` Si noti che `$.foo` estrarrà "bar" e "baz" anche da questo JSON: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}` perché un singolo oggetto "data" viene elaborato automaticamente (per retrocompatibilità con l'implementazione del low-level discovery nelle versioni di Zabbix precedenti alla 4.2).

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 della regola di discovery che consentono di filtrare i valori di discovery:



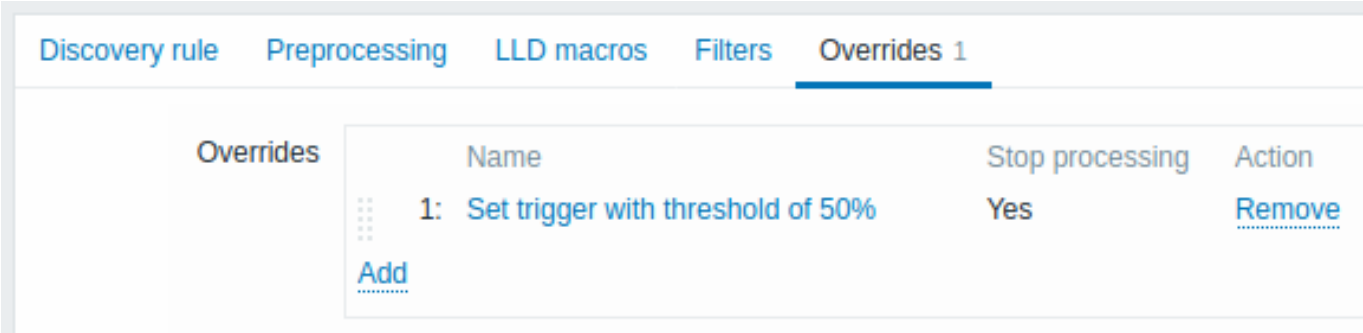
Parametro	Descrizione
Tipo di calcolo	Sono disponibili le seguenti opzioni per il calcolo dei filtri: **And** - tutti i filtri devono essere soddisfatti; **Or** - è sufficiente che un filtro sia soddisfatto; **And/Or** - utilizza *And* con nomi macro diversi e *Or* con lo stesso nome macro; **Espressione personalizzata** - offre la possibilità di definire un calcolo personalizzato dei filtri. La formula deve includere tutti i filtri nell'elenco. Limitata a 255 simboli.
Filtri	Sono disponibili i seguenti operatori di condizione del filtro: *matches*, *does not match*, *exists*, *does not exist*. Gli operatori *matches* e *does not match* richiedono una Perl Compatible Regular Expression (PCRE). Ad esempio, se si è interessati solo ai file system C:, D: ed E:, è possibile inserire {#FSNAME} in "Macro" e l'espressione regolare "^C|^D|^E" nei campi di testo "Regular expression". Il filtraggio è possibile anche in base ai tipi di file system utilizzando la macro {#FSTYPE} (ad esempio, "^ext|^reiserfs") e ai tipi di unità (supportato solo da agent Windows) utilizzando la macro {#FSDRIVETYPE} (ad esempio, "fixed"). È possibile inserire un'espressione regolare oppure fare riferimento a una espressione regolare globale nel campo "Regular expression". Per testare un'espressione regolare è possibile usare "grep -E", ad esempio: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` Gli operatori *exists* e *does not exist* consentono di filtrare le entità in base alla presenza o all'assenza della macro LLD specificata nella risposta. Si noti che se una macro del filtro è assente nella risposta, l'entità trovata verrà ignorata, a meno che per questa macro non sia specificata una condizione "does not exist". Verrà visualizzato un avviso se l'assenza di una macro influisce sul risultato dell'espressione. Ad esempio, se {#B} è assente in: {#A} matches 1 and {#B} matches 2 - verrà mostrato un avviso {#A} matches 1 or {#B} matches 2 - nessun avviso

Warning:
Un errore o un refuso nell'espressione regolare utilizzata nella regola LLD (ad esempio, un'espressione regolare "File systems for discovery" non corretta) può causare l'eliminazione di migliaia di elementi di configurazione, valori storici ed eventi per molti host.

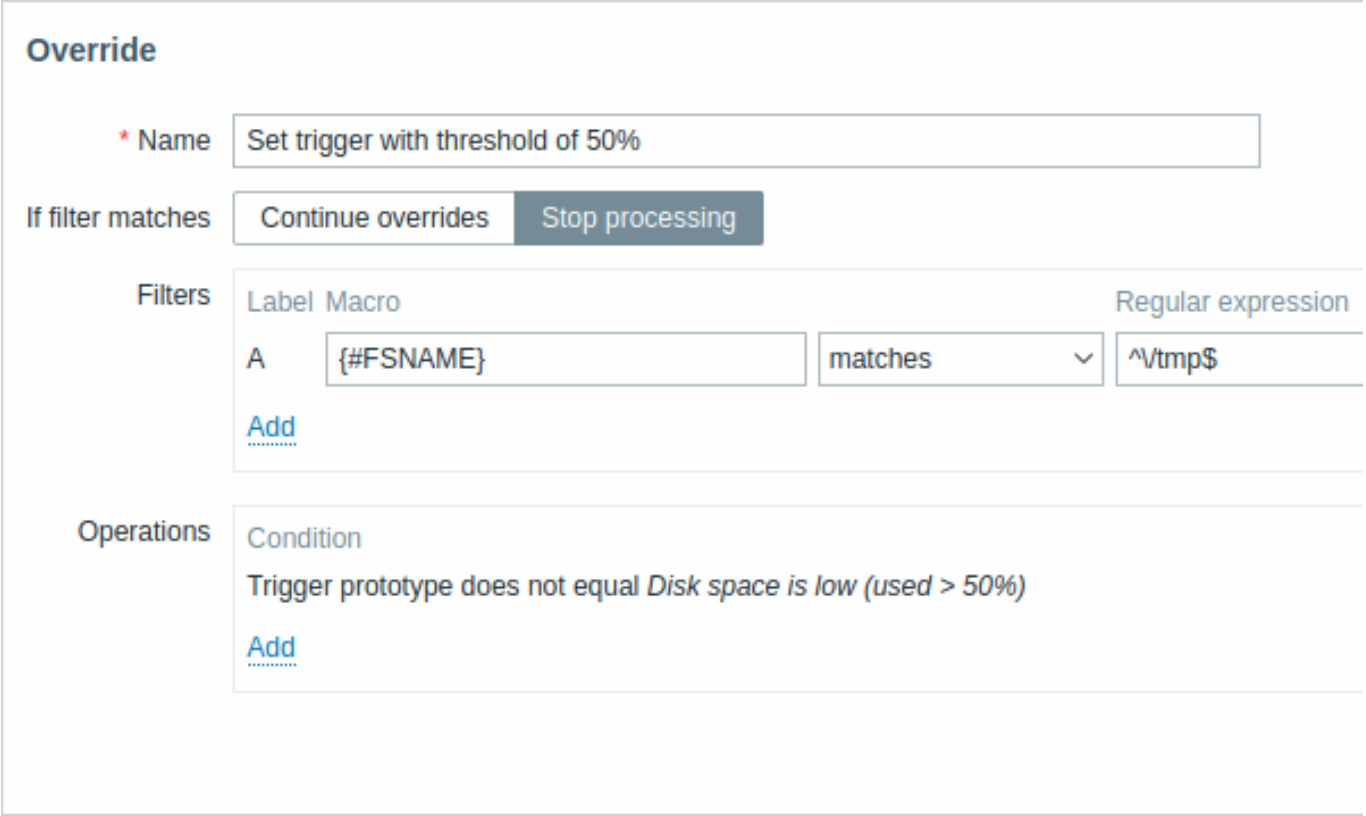
Attention:
Il database Zabbix in MySQL deve essere creato come case-sensitive affinché i nomi dei file system che differiscono solo per maiuscole/minuscole vengano rilevati correttamente.

Override

La scheda **Overrides** consente di impostare regole per modificare l'elenco dei prototipi di item, trigger, grafico, host e discovery oppure i loro attributi per gli oggetti rilevati che soddisfano determinati criteri.



Gli override (se presenti) sono visualizzati in un elenco riordinabile tramite trascinamento e vengono eseguiti nell'ordine in cui sono definiti. Per configurare i dettagli di un nuovo override, fare clic su Add nel blocco *Overrides*. Per modificare un override esistente, fare clic sul nome dell'override. Si aprirà una finestra popup che consentirà di modificare i dettagli della regola di override.

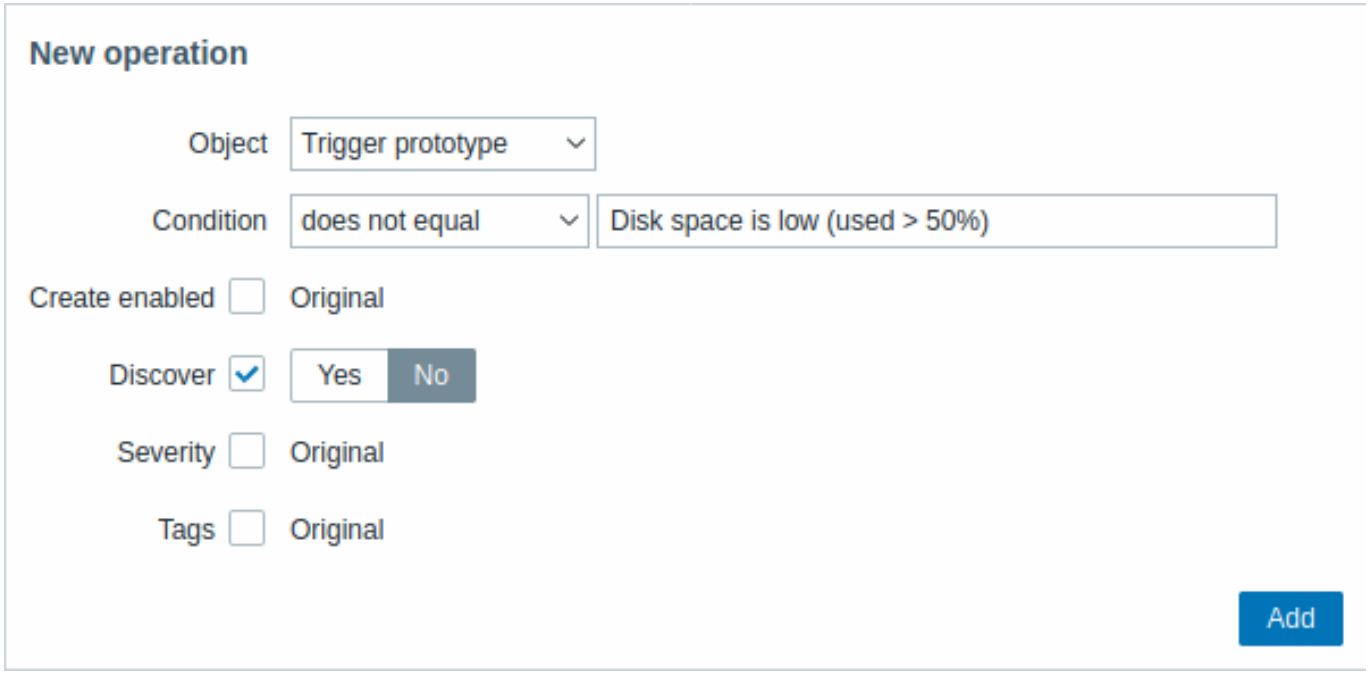


Tutti i parametri obbligatori sono contrassegnati con asterischi rossi.

Parameter	Description
Name	Un nome di override univoco (per regola LLD).
If filter matches	Definisce se gli override successivi devono essere elaborati quando le condizioni del filtro sono soddisfatte: **Continue overrides** - gli override successivi verranno elaborati. **Stop processing** - le operazioni degli override precedenti (se presenti) e di questo override verranno eseguite; gli override successivi verranno ignorati per le righe LLD corrispondenti.
Filters	Determina a quali entità rilevate deve essere applicato l'override. I filtri di override vengono elaborati dopo i filtri della regola di discovery e hanno la stessa funzionalità.
Operations	Le operazioni di override sono visualizzate con questi dettagli: **Condition** - un tipo di oggetto e una condizione che deve essere soddisfatta per il nome dell'oggetto; ad esempio: il prototipo di trigger non è uguale a *Disk space is low (used > 50%)*. **Actions** - 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 Operations. Per modificare un'operazione esistente, fare clic su Edit accanto all'operazione. Si aprirà una finestra popup in cui sarà possibile modificare i dettagli dell'operazione.



Parameter	Description
Object	Sono disponibili cinque tipi di oggetti: Prototipo di item Prototipo di trigger Prototipo di grafico Prototipo di host Prototipo di discovery
Condition	Consente di filtrare le entità a cui deve essere applicata l'operazione.
Operator	Operatori supportati: **equals** - applica a questo prototipo **does not equal** - applica a tutti i prototipi, tranne questo **contains** - applica se il nome del prototipo contiene questa stringa **does not contain** - applica se il nome del prototipo non contiene questa stringa **matches** - applica se il nome del prototipo corrisponde all'espressione regolare **does not match** - applica se il nome del prototipo non corrisponde all'espressione regolare
Pattern	Una espressione regolare o una stringa da cercare.
Object: *Item pro-to-type*	
Create enabled	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di item: *Yes* - l'item verrà aggiunto in stato abilitato. *No* - l'item verrà aggiunto a un'entità rilevata ma in stato disabilitato.
Discover	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di item: *Yes* - l'item verrà aggiunto. *No* - l'item non verrà aggiunto.
Update interval	Quando la casella di controllo è selezionata, verranno visualizzate due opzioni che consentono di impostare un intervallo diverso per l'item: *Delay* - intervallo di aggiornamento dell'item. Sono supportate le macro utente e i suffissi temporali (ad esempio 30s, 1m, 2h, 1d). Deve essere impostato su 0 se viene utilizzato *Custom interval*. *Custom interval* - fare clic su Add per specificare intervalli flessibili/di pianificazione. Per informazioni dettagliate, vedere Intervalli personalizzati.

Parameter		Description
	History	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di impostare per l'item un periodo di archiviazione dello storico diverso: *Do not store* - se selezionato, lo storico non verrà archiviato. *Store up to* - se selezionato, a destra verrà visualizzato un campo di input per specificare il periodo di archiviazione. Sono supportate le macro utente e le macro LLD.
	Trends	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di impostare per l'item un periodo di archiviazione dei trend diverso: *Do not store* - se selezionato, i trend non verranno archiviati. *Store up to* - se selezionato, a destra verrà visualizzato un campo di input per specificare il periodo di archiviazione. Sono supportate le macro utente e le macro LLD.
	Tags	Quando la casella di controllo è selezionata, verrà visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Questi tag verranno aggiunti ai tag specificati nel prototipo di item, anche se i nomi dei tag corrispondono.
Object: *Trigger prototype*		
	Create enabled	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di trigger: *Yes* - il trigger verrà aggiunto in stato abilitato. *No* - il trigger verrà aggiunto a un'entità rilevata, ma in stato disabilitato.
	Discover	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di trigger: *Yes* - il trigger verrà aggiunto. *No* - il trigger non verrà aggiunto.
	Severity	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti di gravità del trigger, che consentono di modificare la gravità del trigger.
	Tags	Quando la casella di controllo è selezionata, verrà visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Questi tag verranno aggiunti ai tag specificati nel prototipo di trigger, anche se i nomi dei tag corrispondono.
Object: *Graph prototype*		
	Discover	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di grafico: *Yes* - il grafico verrà aggiunto. *No* - il grafico non verrà aggiunto.
Object: *Host prototype*		
	Create enabled	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di host: *Yes* - l'host verrà creato in stato abilitato. *No* - l'host verrà creato in stato disabilitato.
	Discover	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di sovrascrivere le impostazioni originali del prototipo di host: *Yes* - l'host verrà rilevato. *No* - l'host non verrà rilevato.
	Link templates	Quando la casella di controllo è selezionata, verrà visualizzato un campo di input per specificare i template. Iniziare a digitare il nome del template oppure fare clic su *Select* accanto al campo e selezionare i template dall'elenco in una finestra popup. I template di questo override vengono aggiunti a tutti i template già collegati al prototipo di host.

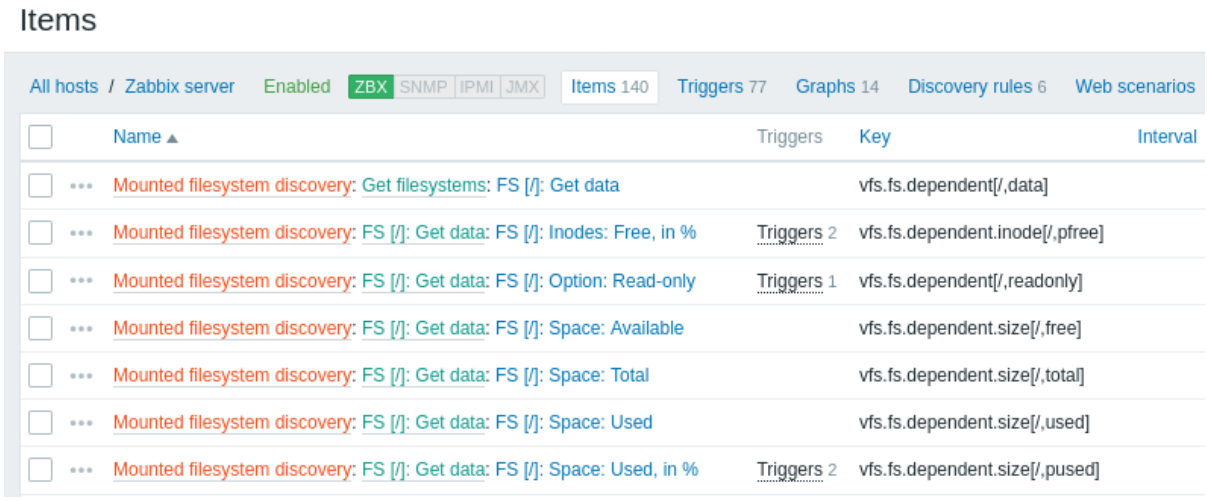
Parameter	Description
Tags	Quando la casella di controllo è selezionata, verrà visualizzato un nuovo blocco che consente di specificare coppie tag-valore. Questi tag verranno aggiunti ai tag specificati nel prototipo di host, anche se i nomi dei tag corrispondono.
Host inventory	Quando la casella di controllo è selezionata, verranno visualizzati i pulsanti che consentono di selezionare una diversa modalità di inventario per il prototipo di host: *Disabled* - non compilare l'inventario dell'host *Manual* - fornire i dettagli manualmente *Automated* - compilare automaticamente i dati dell'inventario dell'host in base alle metriche raccolte.

Pulsanti del modulo

I pulsanti nella parte inferiore del modulo consentono di eseguire diverse operazioni.

Add	Aggiungi una regola di individuazione. Questo pulsante è disponibile solo per le nuove regole di individuazione.
Update	Aggiorna le proprietà di una regola di individuazione. Questo pulsante è disponibile solo per le regole di individuazione esistenti.
Clone	Crea un'altra regola di individuazione basata sulle proprietà della regola di individuazione corrente.
Execute now	Esegui immediatamente l'individuazione in base alla regola di individuazione. La regola di individuazione deve già esistere. Vedi maggiori dettagli. *Nota:* quando si esegue immediatamente l'individuazione, la cache di configurazione non viene aggiornata, pertanto il risultato non rifletterà modifiche molto recenti alla configurazione della regola di individuazione.
Test	Verifica la configurazione della regola di individuazione. Usa questo pulsante per verificare le impostazioni di configurazione (come la connettività e la correttezza dei parametri) senza applicare permanentemente alcuna modifica.
Delete	Elimina la regola di individuazione.
Cancel	Annulla la modifica delle proprietà della regola di individuazione.

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.

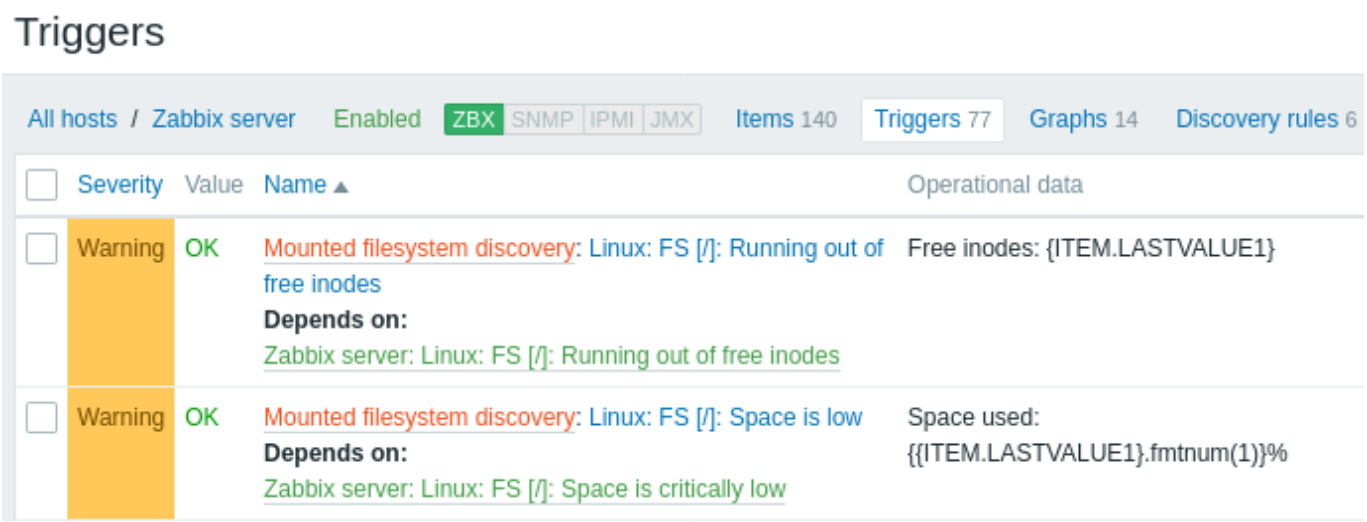
Le risorse perse possono essere automaticamente eliminate in base al valore del parametro *Delete lost resources*. Questo riguarda host, gruppi di host, item, trigger e grafici persi.

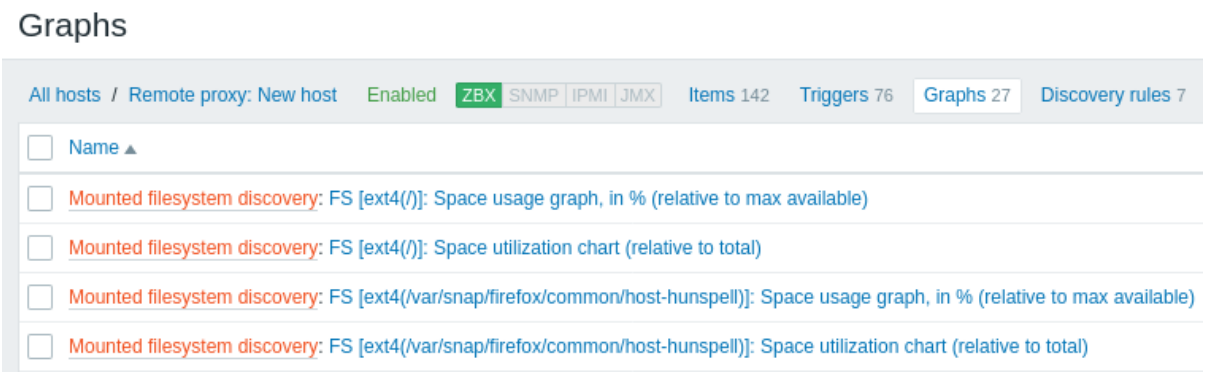
Quando le entità rilevate diventano 'Not discovered anymore', nell'elenco delle entità viene visualizzato un indicatore di durata. Spostando il puntatore del mouse su di esso, verrà visualizzato un messaggio che ne indica i dettagli di stato.



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

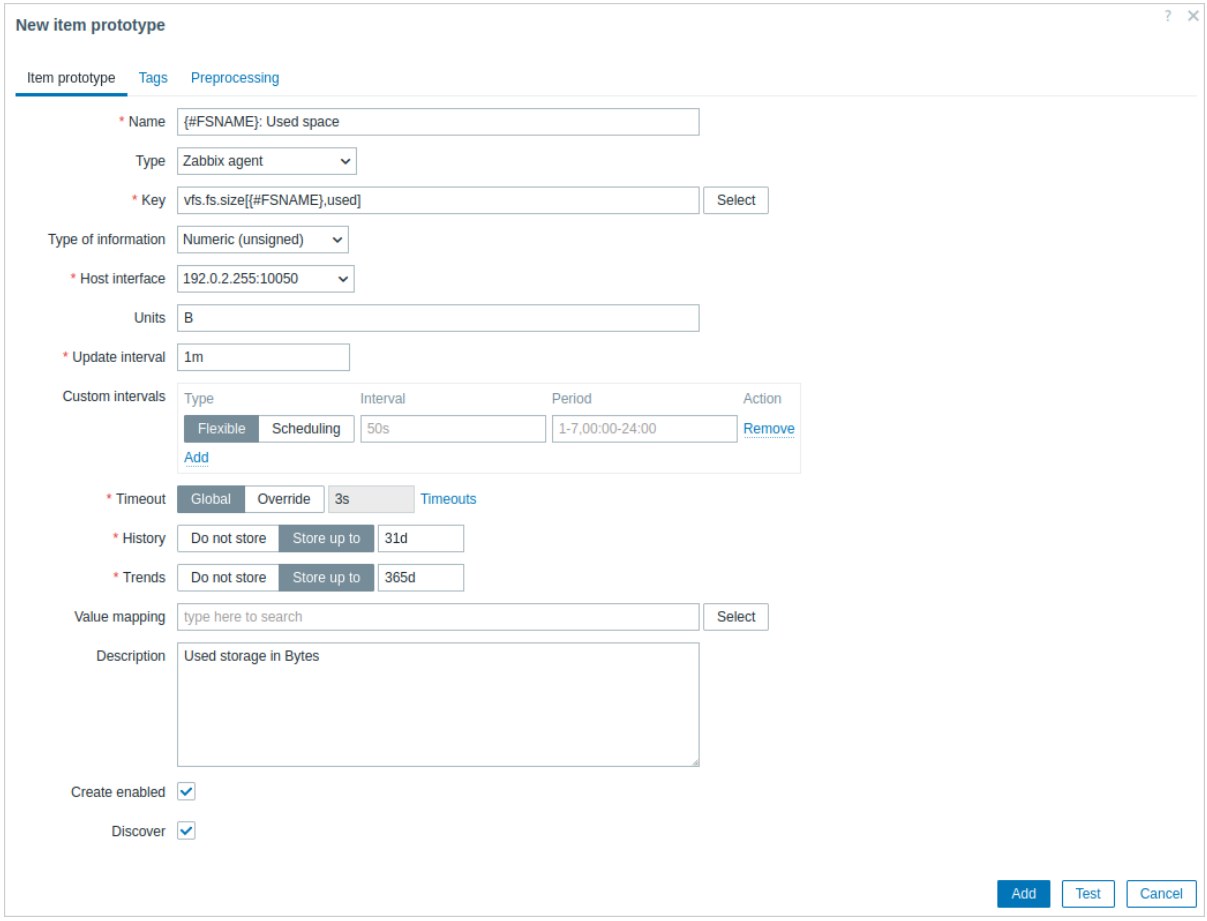
- discovery di interfacce host in Zabbix

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.

1 Prototipi di item

Una volta che una regola è stata creata, vai agli item di quella regola e premi "Crea prototipo di item" per creare un prototipo di item.

Nota come la macro {#FSNAME} venga utilizzata dove è richiesto il nome di un file system. L'uso di una macro di low-level discovery è obbligatorio nella chiave dell'item per assicurarsi che il discovery venga elaborato correttamente. Quando la regola di discovery viene elaborata, questa macro verrà sostituita con il file system individuato.



Note:
Quando i parametri della chiave dell'item non vengono utilizzati, inserisci la macro LLD tra le parentesi della chiave dell'item `[...]` (ad esempio `v_[{#MACRO}]`).

Le macro di low-level discovery e le macro utente sono supportate nella configurazione del prototipo di item e nei parametri di preprocessing del valore dell'item. Nota che, quando vengono utilizzate negli intervalli di aggiornamento, una singola macro deve occupare l'intero campo. Macro multiple in un campo o macro mescolate con testo non sono supportate.

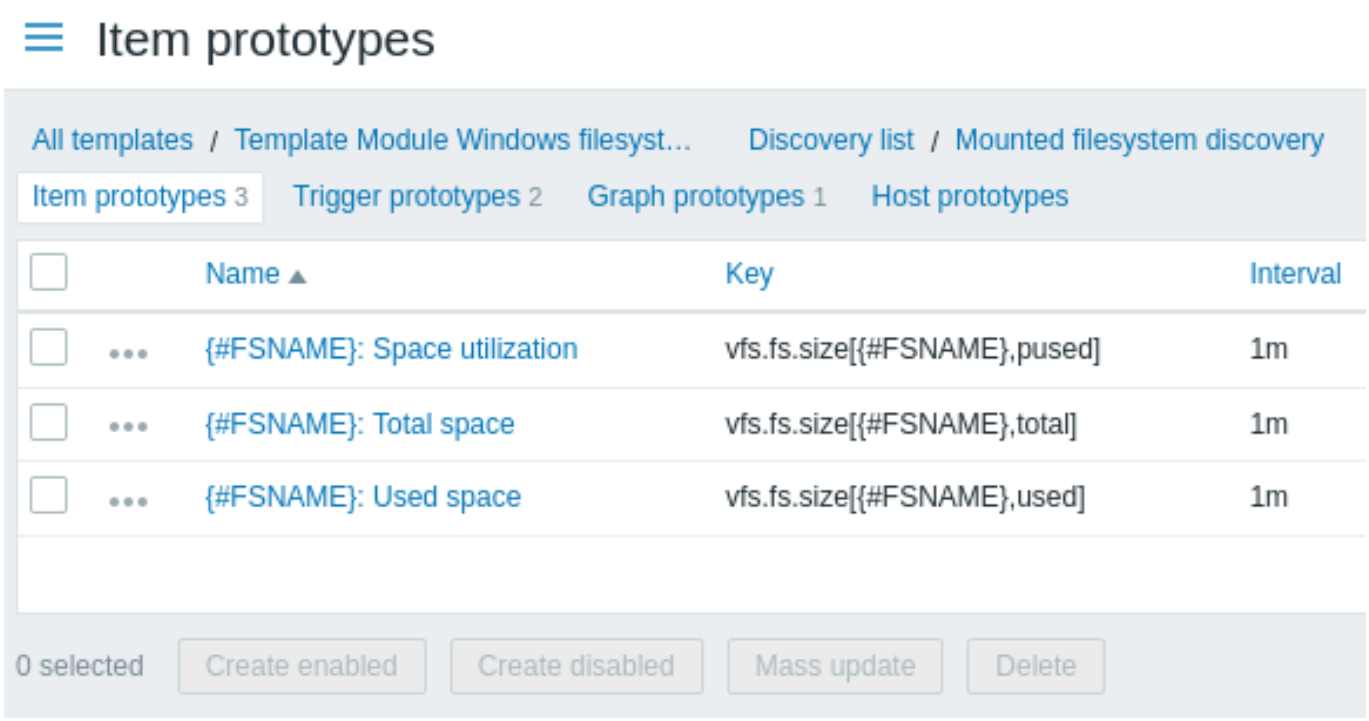
Note:
L'escaping specifico del contesto delle macro di low-level discovery viene eseguito per un utilizzo sicuro nei parametri di preprocessing con espressioni regolari e XPath.

Attributi specifici dei prototipi di item:

Parameter	Description
Create enabled	Se selezionato, l'item verrà aggiunto in stato abilitato. Se non selezionato, l'item verrà aggiunto a un'entità individuata, ma in stato disabilitato.

Parameter	Description
Discover	Se selezionato (predefinito), l'item verrà aggiunto a un'entità individuata. Se non selezionato, l'item non verrà aggiunto a un'entità individuata, a meno che questa impostazione non venga sovrascritta nella regola di discovery.

Possiamo creare diversi prototipi di item per ogni metrica del file system che ci interessa:



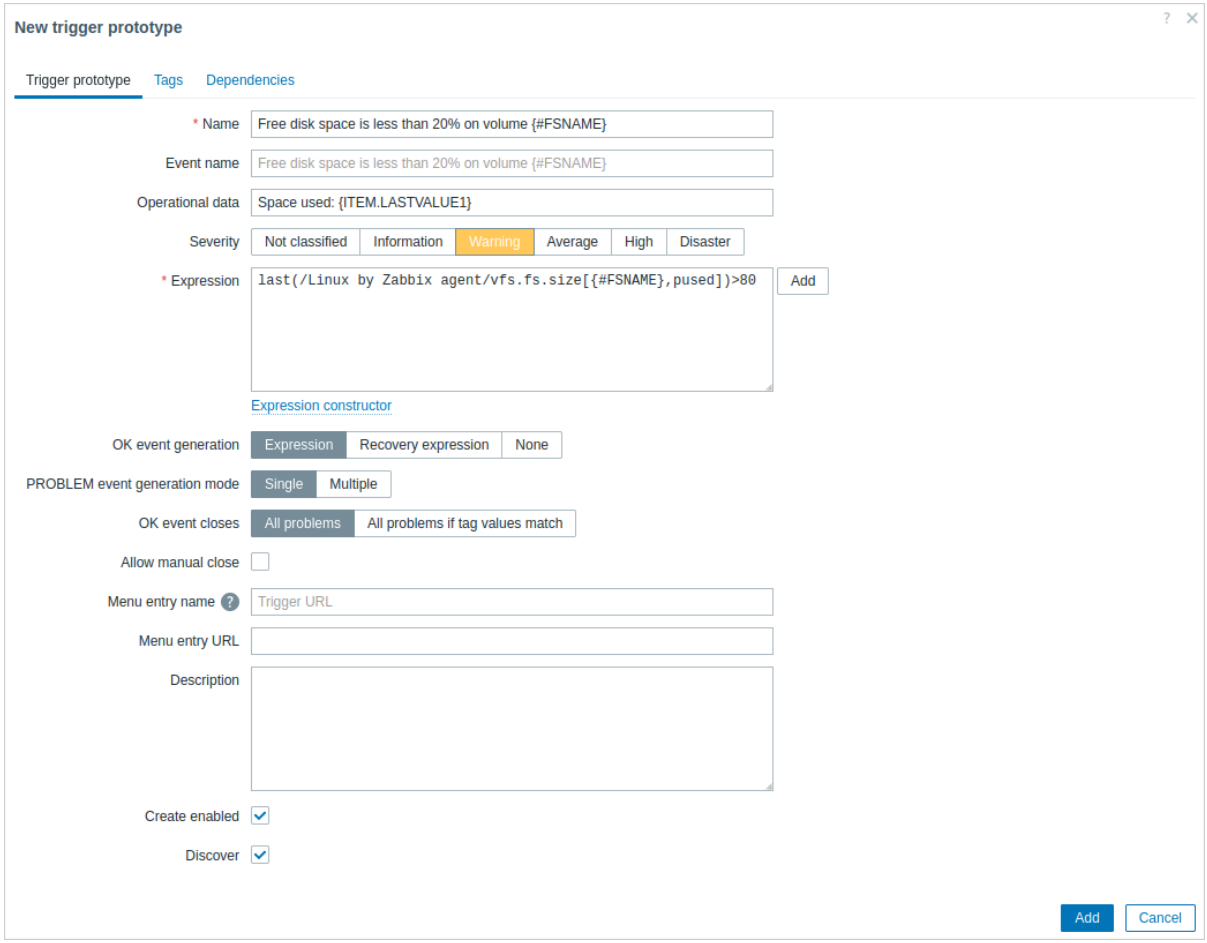
Fai clic sull'icona con i tre puntini per aprire il menu del prototipo di item specifico con queste opzioni:

- *Create trigger prototype* - crea un prototipo di trigger basato su questo prototipo di item
- *Trigger prototypes* - fai clic per vedere un elenco con i collegamenti ai prototipi di trigger già configurati di questo prototipo di item
- *Create dependent item* - crea un item dipendente per questo prototipo di item

L'opzione *Mass update* è disponibile se vuoi aggiornare contemporaneamente le proprietà di diversi prototipi di item.

2 Prototipi di trigger

Creiamo i prototipi di trigger in modo simile ai prototipi di item:

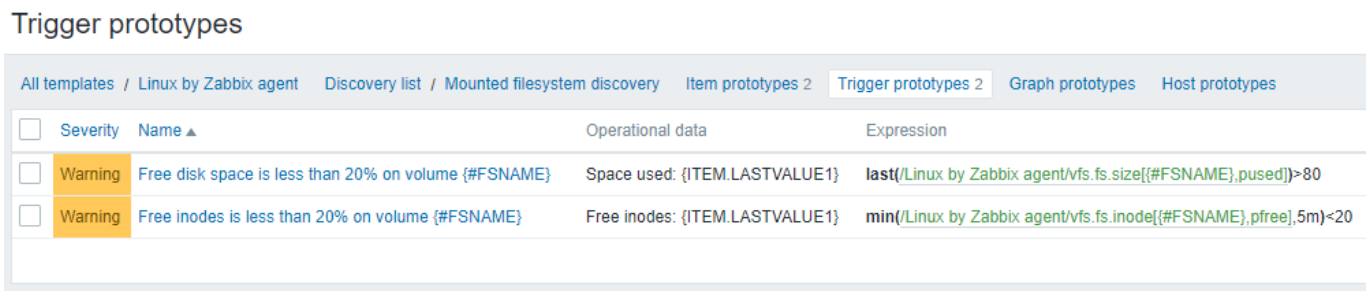


Attributi specifici dei prototipi di trigger:

Parametro	Descrizione
Create enabled	Se selezionato, il trigger verrà aggiunto in stato abilitato. Se non selezionato, il trigger verrà aggiunto a un'entità rilevata, ma in stato disabilitato.
Discover	Se selezionato (predefinito), il trigger verrà aggiunto a un'entità rilevata. Se non selezionato, il trigger non verrà aggiunto a un'entità rilevata, a meno che questa impostazione non venga sovrascritta nella regola di discovery.

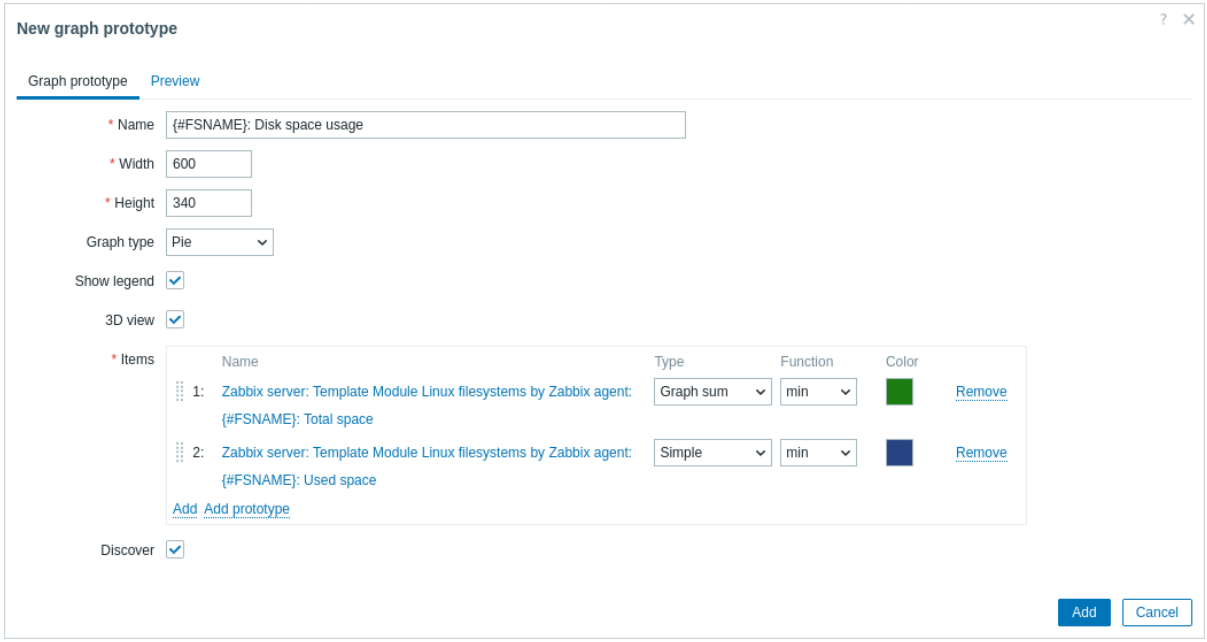
Quando i trigger reali vengono creati dai prototipi, potrebbe essere necessario avere flessibilità riguardo a quale costante ('20' nel nostro esempio) venga usata per il confronto nell'espressione. Vediamo come le macro utente con contesto possano essere utili per ottenere tale flessibilità.

È possibile definire dipendenze tra prototipi di trigger. Per farlo, andare 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 prototipo di trigger di una regola LLD diversa né da un trigger creato da un prototipo di trigger. Un prototipo di trigger di host non può dipendere da un trigger di un template.



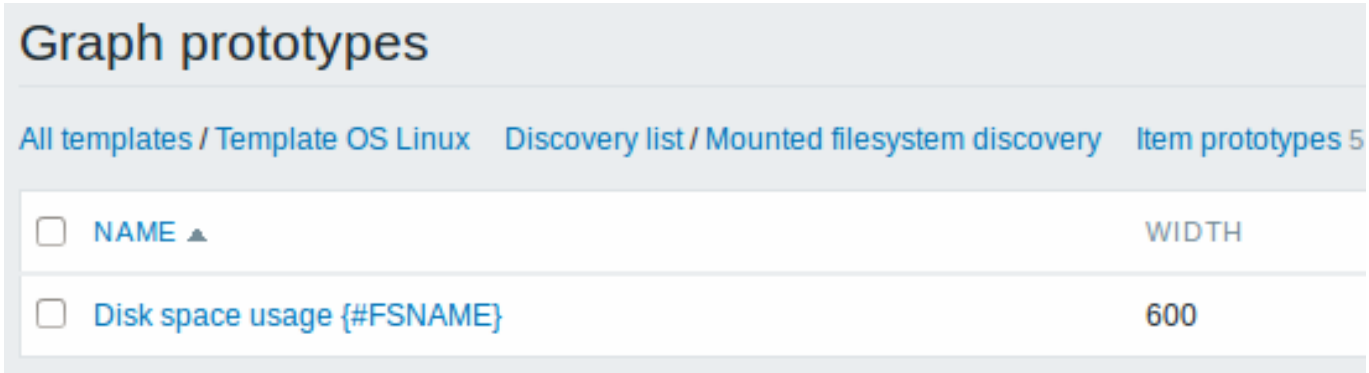
3 Prototipi di grafico

Possiamo creare anche prototipi di grafico:

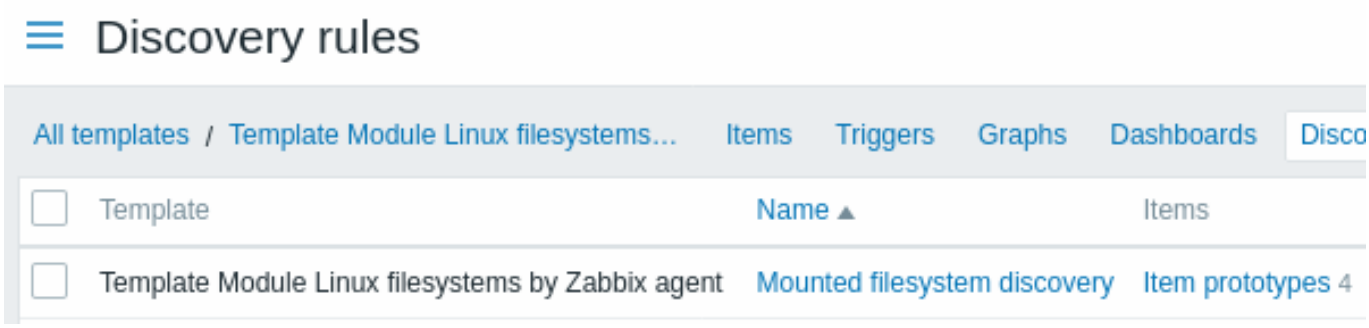


Attributi specifici dei prototipi di grafico:

Parametro	Descrizione
Discover	Se selezionato (predefinito), il grafico verrà aggiunto a un'entità individuata. Se non selezionato, il grafico non verrà aggiunto a un'entità individuata, 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 grafico.



4 Prototipi degli host

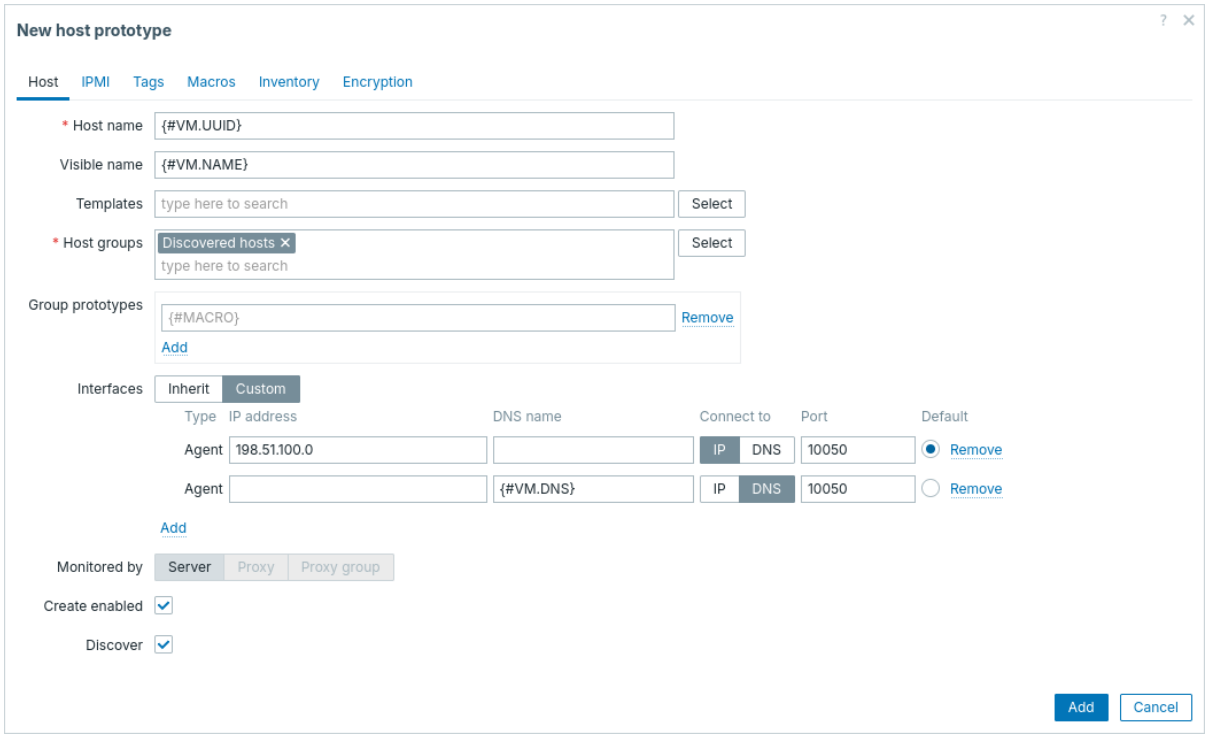
I prototipi di host sono modelli per la creazione di host tramite regole di low-level discovery. Prima di essere rilevati come host, questi prototipi non possono avere item e trigger, ad eccezione di quelli collegati dai template.

Configurazione

I prototipi di host si configurano nelle regole di low-level discovery.

Per creare un prototipo di host:

1. Vai a *Raccolta dati → Host*.
2. Fai clic su *Discovery* per l'host richiesto per passare all'elenco delle regole di low-level discovery configurate per quell'host.
3. Fai clic su *Prototipi di host* per la regola di discovery richiesta.
4. Fai clic sul pulsante *Crea prototipo di host* nell'angolo in alto a destra e completa il modulo di configurazione.



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

Parametro	Descrizione
Nome host	Questo parametro deve contenere almeno una macro di low-level discovery per garantire nomi host univoci per gli host creati.
Nome visibile	Sono supportate le macro di low-level discovery.
Prototipi di gruppi	Consente di specificare prototipi di gruppi di host utilizzando macro di low-level discovery. In base ai prototipi di gruppi specificati, i gruppi di host verranno rilevati, creati e collegati agli host creati; anche i gruppi rilevati che sono già stati creati da altre regole di low-level discovery verranno collegati agli host creati. Tuttavia, i gruppi di host rilevati che corrispondono a gruppi di host creati manualmente non verranno collegati agli host creati.
Interfacce	Imposta se gli host rilevati ereditano l'IP dall'host a cui appartiene la regola di discovery (predefinito) oppure ottengono interfacce personalizzate. Sono supportate le macro di low-level discovery e le macro utente.
Monitorato da	Questo campo viene sincronizzato automaticamente con il rispettivo campo nella configurazione host.
Crea abilitato	Imposta lo stato degli host rilevati; se non selezionato, gli host verranno creati come disabilitati.
Rileva	Imposta se gli host verranno creati dal prototipo di host; se non selezionato, gli host non verranno creati dal prototipo di host (a meno che questa impostazione non venga sovrascritta nella regola di low-level discovery).

Note:
Le macro di low-level discovery sono supportate anche per i valori dei tag e per i valori delle macro utente del prototipo di host.
 Le *mappe dei valori* non sono supportate per i prototipi di host.

Per un esempio di configurazione di un prototipo di host, vedi *Monitoraggio delle macchine virtuali*.

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).

5 Prototipi di discovery

Panoramica

I prototipi di discovery sono regole di **low-level discovery annidate** all'interno di una regola di discovery "padre", che consentono di creare una discovery multilivello di oggetti con i propri item, trigger, ecc. Ad esempio, potresti voler individuare tutte le istanze di database su un server di database, quindi individuare i tablespace per ogni istanza, quindi individuare le tabelle per ogni tablespace.

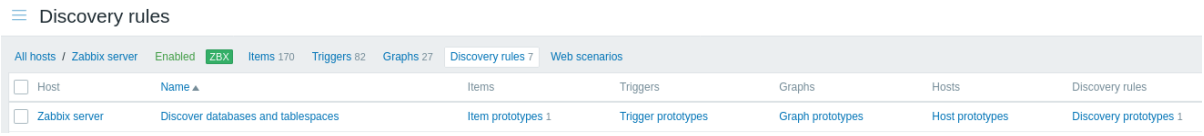
I prototipi di discovery hanno propri prototipi di item, trigger, grafico, host e discovery. Un prototipo di discovery annidato utilizzerà lo stesso valore JSON della regola padre se specifichi un tipo *Nested*.

I livelli di annidamento per i prototipi di discovery sono illimitati.

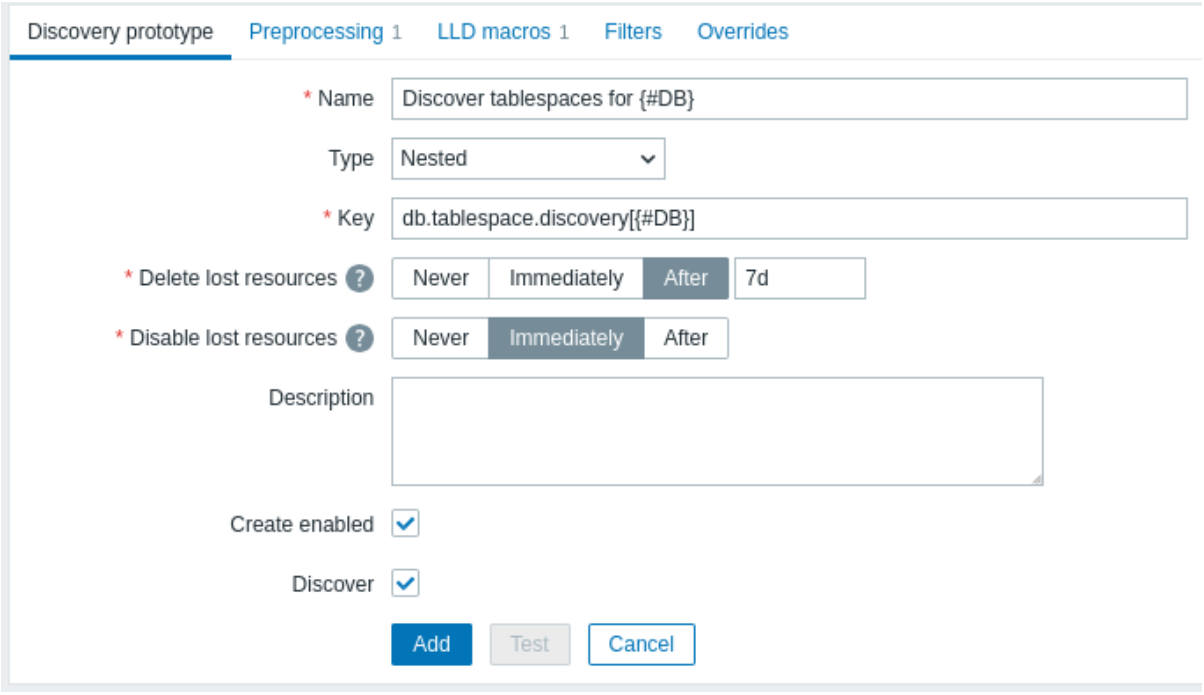
Configurazione

Per creare un prototipo di discovery:

- Fare clic su *Discovery prototypes* nella riga di una regola di discovery 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 prototipo di discovery aperto, le regole di discovery (dal prototipo di discovery) vengono generate in base a un oggetto JSON dallo stesso valore JSON della regola di discovery padre. Ad esempio, se il JSON originale è [<object A>, <object B>] ed è presente un prototipo di regola di discovery annidata, verranno generate due regole di discovery basate rispettivamente sui dati dell’oggetto A e dell’oggetto B.

In questo caso il prototipo di discovery viene attivato contemporaneamente alla regola padre. La regola annidata può quindi usare il preprocessing per lavorare su un’altra ”porzione” degli stessi dati, già acquisiti dalla regola padre.

Le macro LLD della regola LLD padre sono disponibili per le regole di discovery 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 multiliv-ello.

```
[
  {
    "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. Devi avere almeno un host correlato al rilevamento del server di database.
2. Crea una regola LLD per questo host denominata *Rileva database e tablespace*.
3. Passa alla scheda *LLD Macros* di questa regola e aggiungi la macro `{#DB}=$.database`.
4. Aggiungi 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:

```
Connessioni attive a db1, Connessioni attive a db2, Connessioni attive a db3.
```

6. Crea un prototipo di discovery per questa regola denominato *Rileva i tablespace per {#DB}* (Tipo: Nested, Chiave: `db.tablespace.discovery[{#DB}]`).
7. Passa alla scheda *Preprocessing* di questo prototipo di discovery e aggiungi il passaggio `JSONPath=$.tablespaces`.
8. Passa alla scheda *LLD Macros* di questo prototipo di discovery e aggiungi la macro `{#TSNAME}=$.name`.
9. Crea 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:

```
Dimensione del tablespace ts1 per db1, Dimensione del tablespace ts2 per db1, Dimensione del tablespace ts
Dimensione del tablespace ts1 per db2, Dimensione del tablespace ts2 per db2, Dimensione del tablespace ts
Dimensione del tablespace ts1 per db3, Dimensione del tablespace ts2 per db3, Dimensione del tablespace ts
```

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}=$.database`.
10. Aggiungere un prototipo di host per questa regola denominato *Host per database {#DB}*.
11. Passare alla scheda *Macros* di questo prototipo di host, aggiungere la macro `{$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 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 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 database db3	Connessioni attive a db3
	Dimensione del tablespace ts1 per db3
	Dimensione del tablespace ts2 per db3
	Dimensione del tablespace ts3 per db3

6 Note sul rilevamento di 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.

Regole LLD multiple per lo stesso item

È possibile definire più regole di low-level discovery con lo stesso item di discovery.

Per farlo è necessario definire il parametro Alias dell'agent, 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.

7 Regole di individuazione

Utilizza la barra laterale per vedere esempi di configurazione delle regole di individuazione per vari casi.

1 Scoperta dei filesystem montati

Panoramica

È possibile individuare i filesystem montati e le loro proprietà:

- nome del punto di mount
- tipo di filesystem
- dimensione del filesystem
- statistiche degli inode
- opzioni di mount

Per farlo, è possibile utilizzare una combinazione di:

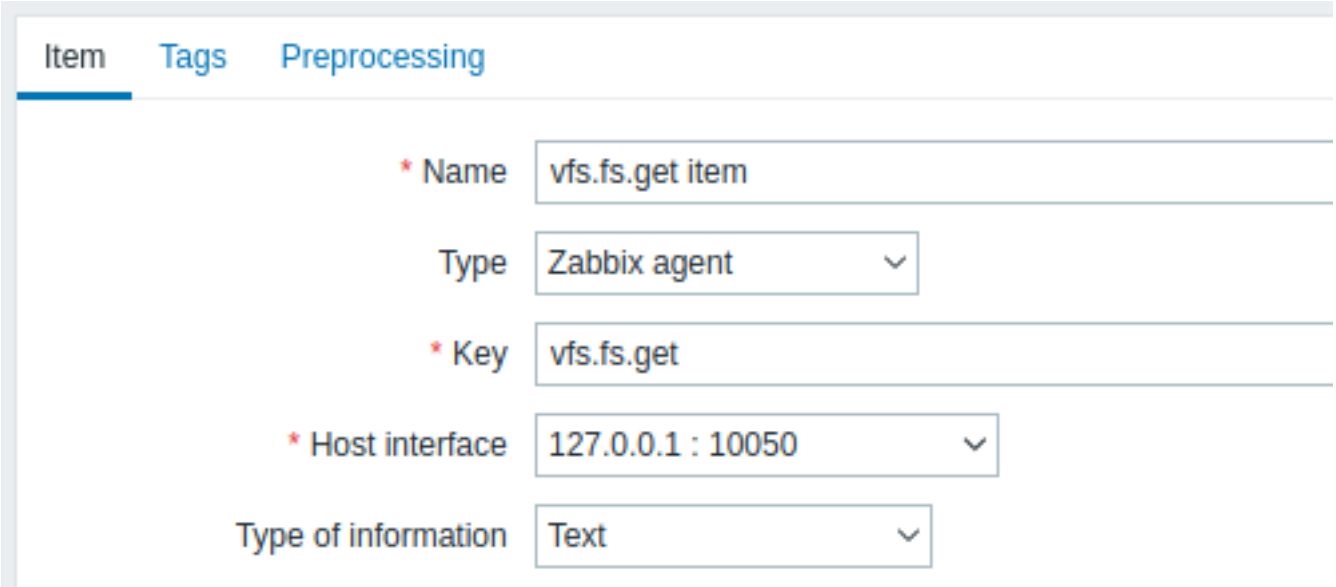
- l'item agent `vfs.fs.get` come item principale
- regola di low-level discovery dipendente e prototipi di item

Configurazione

Item master

Creare un item Zabbix agent utilizzando la seguente chiave:

`vfs.fs.get`



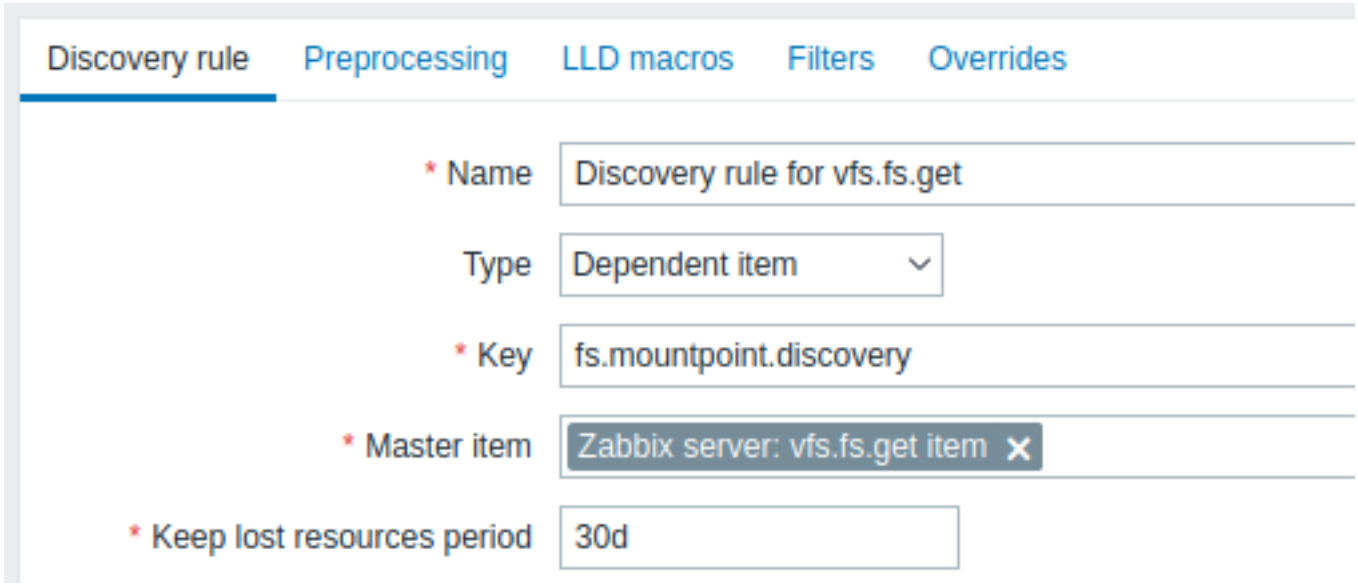
Impostare il tipo di informazioni su "Text" per dati JSON potenzialmente di grandi dimensioni.

I dati restituiti da questo item conterranno qualcosa di simile al seguente per un filesystem montato:

```
[
  {
    "fsname": "/",
    "fstype": "ext4",
    "bytes": {
      "total": 249405239296,
      "free": 24069537792,
      "used": 212595294208,
      "pfree": 10.170306,
      "pused": 89.829694
    },
    "inodes": {
      "total": 15532032,
      "free": 12656665,
      "used": 2875367,
      "pfree": 81.487503,
      "pused": 18.512497
    },
    "options": "rw,noatime,errors=remount-ro"
  }
]
```

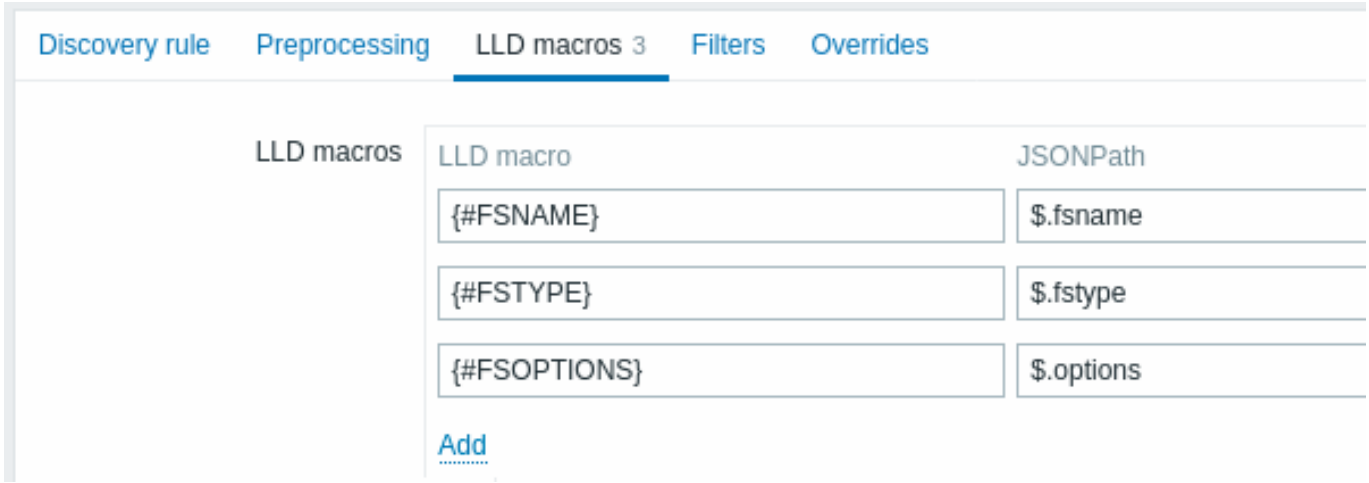
Regola LLD dipendente

Creare una regola di low-level discovery di tipo "Dependent item":

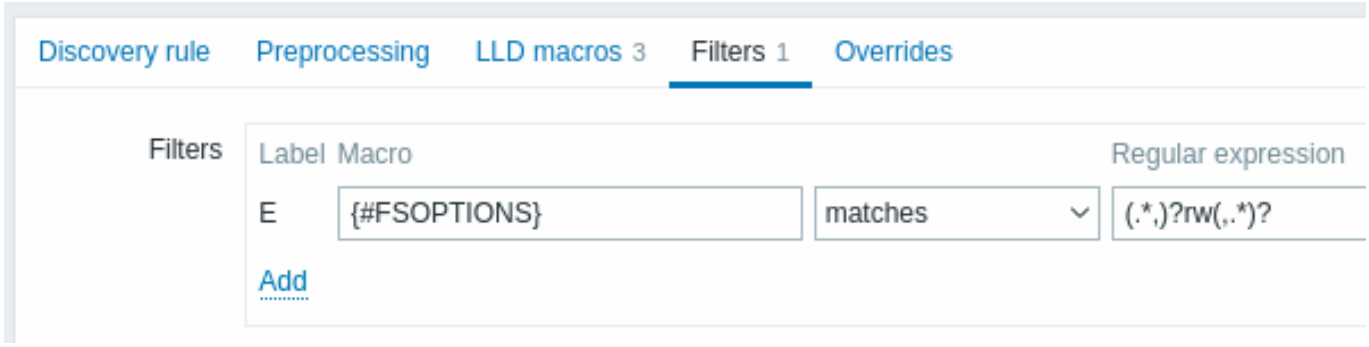


Come item master selezionare l'item `vfs.fs.get` che abbiamo creato.

Nella scheda "LLD macros" definire macro personalizzate con il corrispondente JSONPath:

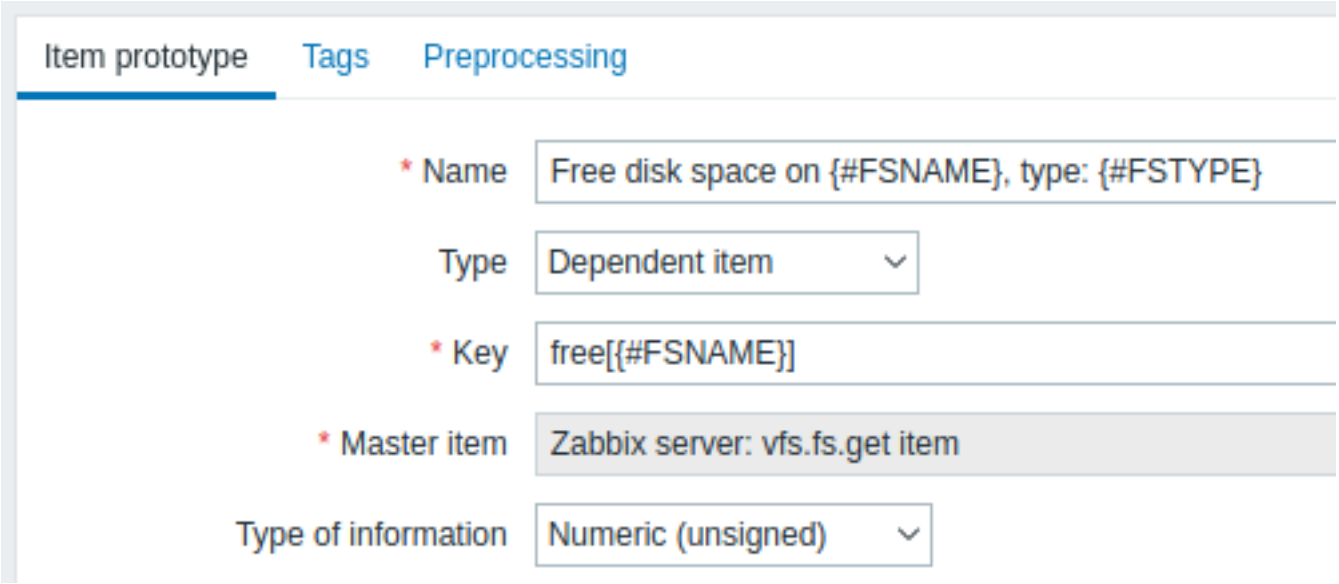


Nella scheda "Filters" è possibile aggiungere un'espressione regolare che filtri solo i filesystem **read-write**:



Prototipo di item dipendente

Creare in questa regola LLD un prototipo di item con tipo "Dependent item". Come item master per questo prototipo selezionare l'item `vfs.fs.get` che abbiamo creato.



Notare l'uso di macro personalizzate nel nome e nella chiave del prototipo di item:

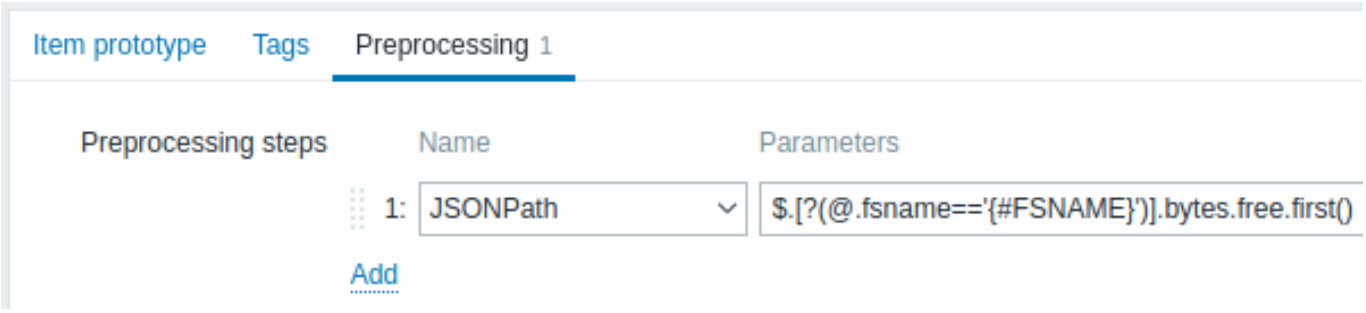
- *Nome*: Spazio libero su disco su {#FSNAME}, tipo: {#FSTYPE}
- *Chiave*: Free[{#FSNAME}]

Come tipo di informazione, utilizzare:

- *Numeric (unsigned)* per metriche come 'free', 'total', 'used'
- *Numeric (float)* per metriche come 'pfree', 'pused' (percentuale)

Nella scheda "Preprocessing" del prototipo di item selezionare JSONPath e utilizzare la seguente espressione JSONPath come parametro:

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Quando la discovery viene avviata, verrà creato un item per ciascun mountpoint. Questo item restituirà il numero di byte liberi per il mountpoint specificato.

2 Individuazione delle interfacce di rete

In modo simile a come vengono individuati i file system, è possibile individuare anche le interfacce di rete.

Chiave item

La chiave item da utilizzare nella regola di discovery è

```
net.if.discovery
```

Macro supportate

È possibile utilizzare la macro {#IFNAME} nel filtro della regola di discovery e nei prototipi di item, trigger e grafici.

Esempi di prototipi di item che si potrebbero voler creare in base a "net.if.discovery":

- "net.if.in[{#IFNAME},bytes]",
- "net.if.out[{#IFNAME},bytes]".

Si noti che in Windows viene restituita anche {#IFGUID}.

3 Individuazione di CPU e core della CPU

In modo simile a come vengono individuati i file system, è possibile individuare anche le CPU e i core della CPU.

Chiave item

La chiave item da utilizzare nella regola di discovery è

`system.cpu.discovery`

Macro supportate

Questa chiave di discovery restituisce due macro - {#CPU.NUMBER} e {#CPU.STATUS} che identificano rispettivamente il numero d'ordine della CPU e il suo stato. Enumera i processori da 0 a N - 1, dove N è il numero totale di processori.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" is only on Windows */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Si noti che non è possibile fare una chiara distinzione tra processori fisici reali, core e hyperthread. {#CPU.STATUS} sui sistemi Linux, UNIX e BSD restituisce lo stato del processore, che può essere "online" oppure "offline". Sui sistemi Windows, questa stessa macro può rappresentare un terzo valore - "unknown" - che indica che un processore è stato rilevato, ma non sono ancora state raccolte informazioni su di esso.

Il discovery della CPU si basa sul processo collector dell'agent per rimanere coerente con i dati forniti dal collector e risparmiare risorse durante l'ottenimento dei dati. Questo comporta che questa chiave item non funzioni con il flag da riga di comando di test (-t) del binario dell'agent, che restituirà uno stato NOT_SUPPORTED e un messaggio associato che indica che il processo collector non è stato avviato.

I prototipi di item che possono essere creati sulla base del discovery della CPU includono, ad esempio:

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Per una descrizione dettagliata delle chiavi item, vedere chiavi item di Zabbix agent.

Note sugli item dei contatori delle prestazioni individuati sui sistemi NUMA

Alcuni contatori delle prestazioni di Windows potrebbero non essere disponibili per alcuni processori logici sui sistemi NUMA.

Ad esempio, gli item individuati con un prototipo di item con la chiave seguente potrebbero funzionare solo per il primo nodo NUMA. Gli item per gli altri nodi NUMA potrebbero trovarsi in stato non supportato.

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

Inoltre, una regola di discovery con chiave "system.cpu.discovery" non fornisce alcun valore di macro di low-level discovery da sostituire in contatori delle prestazioni come:

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

Mentre gli item system.cpu.util si basano su contatori delle prestazioni diversi per monitorare la percentuale di utilizzo della CPU a seconda del numero di gruppi di processori e del numero di processori. Inoltre, esiste un raro bug relativo ai gruppi di processori su Windows.

4 Scoperta degli OID SNMP

Panoramica

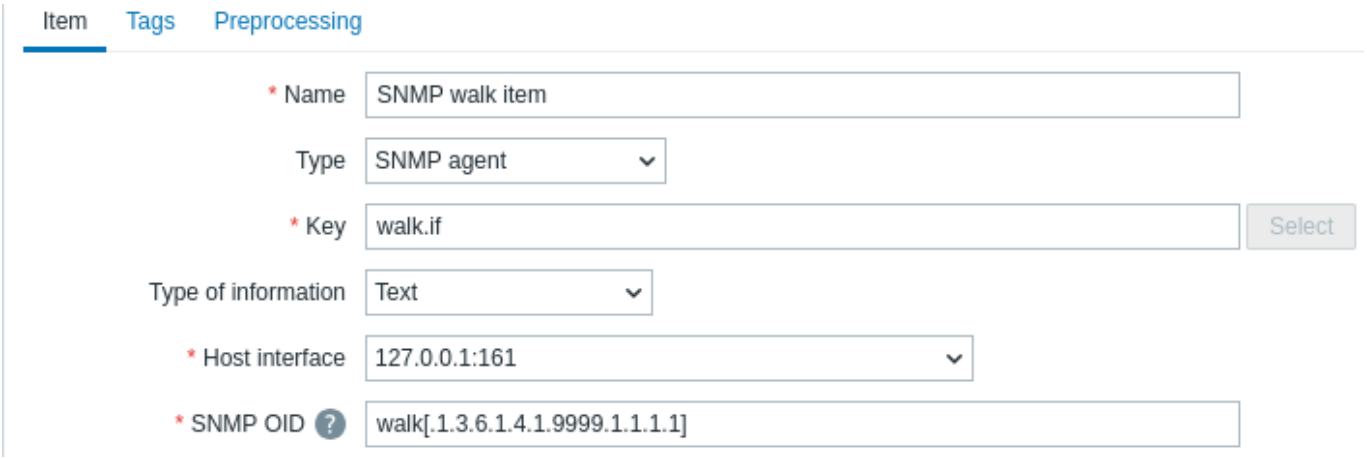
In questa sezione eseguiremo una low-level discovery su un dispositivo SNMP.

Questo metodo di discovery degli OID SNMP è supportato da Zabbix server/proxy 6.4.

Configurazione di esempio

1. Creare un item agent SNMP con una chiave come:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



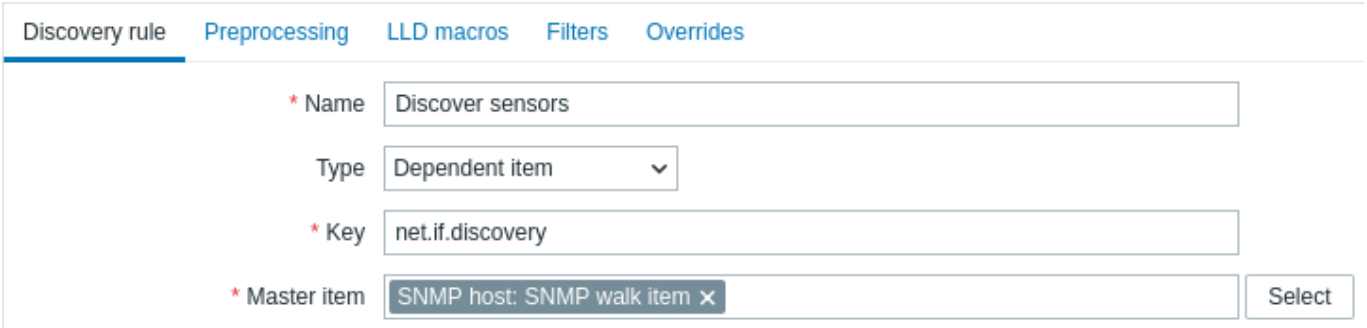
Questo item esegue una singola scansione della tabella SNMP e restituisce tutte le voci della tabella in un'unica richiesta, in un formato corrispondente all'output dell'utilità `snmpwalk` con le opzioni di formattazione `-Oe -Ot -On`.

Restituirà il seguente valore di testo multilinea:

```
.1.3.6.1.4.1.9999.1.1.1.1.1.1 = STRING: "Temperature Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.1 = STRING: "temp"
.1.3.6.1.4.1.9999.1.1.1.1.3.1 = 100
.1.3.6.1.4.1.9999.1.1.1.1.1.2 = STRING: "Humidity Sensor"
.1.3.6.1.4.1.9999.1.1.1.1.2.2 = STRING: "humidity"
.1.3.6.1.4.1.9999.1.1.1.1.3.2 = 200
```

2. Creare una regola di discovery:

- Nel campo *Name*, inserire un nome descrittivo per la regola di discovery (ad esempio, "Discover sensors").
- Nel campo *Type*, selezionare "Dependent item".
- Nel campo *Key*, inserire una chiave descrittiva (ad esempio, "net.if.discovery").
- Nel campo *Master item*, selezionare "SNMP walk item".



3. Nella scheda *Preprocessing*, aggiungere un passaggio di preprocessing con "SNMP walk to JSON" nel menu a discesa *Name* con 3 parametri:

- *Field name*: "{#SENSORNAME}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.1": *Format*: "Unchanged".
- *Field name*: "{#SENSORTYPE}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.2": *Format*: "Unchanged".
- *Field name*: "{#SENSORVALUE}"; *OID prefix*: ".1.3.6.1.4.1.9999.1.1.1.1.3": *Format*: "Unchanged".

Dopo il preprocessing, la regola di discovery restituisce un array JSON di insiemi di macro.

Per esempio:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

Ogni oggetto rappresenta un sensore individuato e fornisce macro come {#SNMPINDEX}, {#SENSORNAME}, {#SENSORTYPE} e {#SENSORVALUE}.

Sono raggruppati in base all'indice SNMP, che è il suffisso numerico alla fine di ogni OID (ad esempio, .1, .2) — questo indice identifica in modo univoco ogni riga nella tabella SNMP ed è estratto automaticamente come {#SNMPINDEX}.

4. Sotto la regola di discovery, creare uno o più prototipi di item (con la regola di discovery come item master).

Per esempio, item dipendente del valore del sensore:

- Nel campo *Name*, inserire "Sensor {#SNMPINDEX}: {#SENSORNAME}".
- Nel campo *Type*, selezionare "Dependent item".
- Nel campo *Key*, inserire "sensor.value[{#SNMPINDEX}]".
- Nel campo *Master item*, selezionare "SNMP walk item".



Nella scheda *Preprocessing*, aggiungere un passaggio di preprocessing con nome "SNMP walk value" con l'OID ".1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}" nel campo *Parameter*. *Format*: "Unchanged".

Saranno individuati i seguenti item:

Name	Key	OID da cui viene estratto il valore	Valore item
Sensor 1: Temperature Sensor	sensor.value[1]	.1.3.6.1.4.1.9999.1.1.1.1.3.1	100
Sensor 2: Humidity Sensor	sensor.value[2]	.1.3.6.1.4.1.9999.1.1.1.1.3.2	200

Quando la regola di discovery viene eseguita, vengono creati item come `sensor.value[1]`, `sensor.value[2]`.

Ogni item dipendente estrae il proprio valore dal risultato SNMP walk dell'item master usando il preprocessing, senza eseguire richieste SNMP separate.

5. Fare riferimento ai prototipi di item dipendenti nei prototipi di trigger usando le stesse macro della regola di discovery. Esempio:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Questo produce un trigger per ogni sensore individuato (per esempio, sensor.value[1], sensor.value[2]) e si attiva se l'ultimo valore (temperatura o umidità) supera 75.

6. Includere item dipendenti per ogni entità individuata. Esempio di chiave dell'elemento del grafico:

```
sensor.value[{#SNMPINDEX}]
```

Viene creato un grafico per ogni {#SNMPINDEX}, che traccia temperatura e umidità nel tempo.

Questa configurazione esegue una sola richiesta SNMP walk per ciclo di polling, indipendentemente dal numero di item individuati. Tutti gli item dipendenti estraggono i propri valori dal risultato SNMP walk master usando il preprocessing, riducendo significativamente il traffico SNMP e il carico.

Indici dinamici con walk[]

Gli indici dinamici (ad esempio, gli indici delle interfacce) possono cambiare quando l'hardware viene riconfigurato. Per gestire questo comportamento, viene creata una regola di discovery master SNMP walk con una chiave come:

```
walk[1.3.6.1.2.1.2.2.1.10]
```

Dopo la pre-elaborazione da SNMP walk a JSON, il risultato potrebbe essere simile a:

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Un prototipo di item dipendente utilizza la macro {#SNMPINDEX} per costruire la chiave:

```
net.if.in[{#SNMPINDEX}]
```

La pre-elaborazione per questo prototipo include il nome "SNMP walk value" con l'OID "1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}" nel campo *Parameter*. *Format*: "Unchanged".

In fase di esecuzione, vengono creati item effettivi come `net.if.in[2]` e `net.if.in[3]`. Se un determinato indice di interfaccia cambia (ad esempio, se l'indice 2 viene sostituito da 5 nella tabella SNMP), alla successiva esecuzione della regola di discovery:

- Il vecchio item dipendente net.if.in[2] viene contrassegnato come "lost" oppure rimosso, e non vengono più raccolti nuovi dati per quell'item.
- Viene creato un nuovo item dipendente net.if.in[5], che inizia con una cronologia vuota.
- I dati storici di net.if.in[2] non vengono spostati automaticamente in net.if.in[5].

Esempio di prototipo di trigger:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

L'esempio di prototipo di grafico include gli item:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Questa configurazione garantisce un monitoraggio affidabile delle tabelle con indici dinamici, riducendo al minimo il traffico SNMP: è richiesto un solo SNMP walk per ciclo di polling, mentre i prototipi di item dipendenti estraggono i valori necessari.

Entità rilevate

Quando il server è in esecuzione, creerà item dipendenti reali, trigger e grafici in base ai valori restituiti dalla regola di individuazione SNMP.

5 Scoperta degli OID SNMP (legacy)

Panoramica

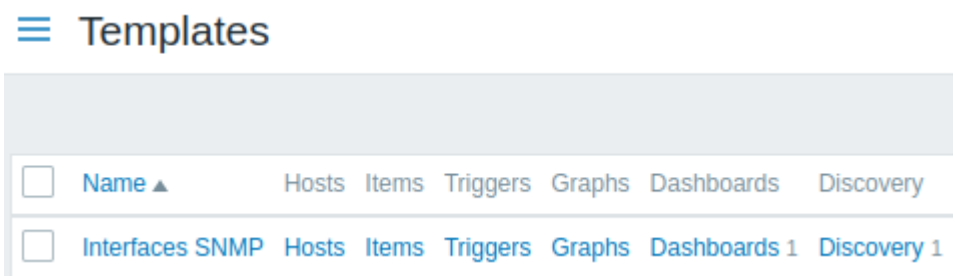
In questa sezione eseguiremo una discovery SNMP su uno switch.

Chiave item

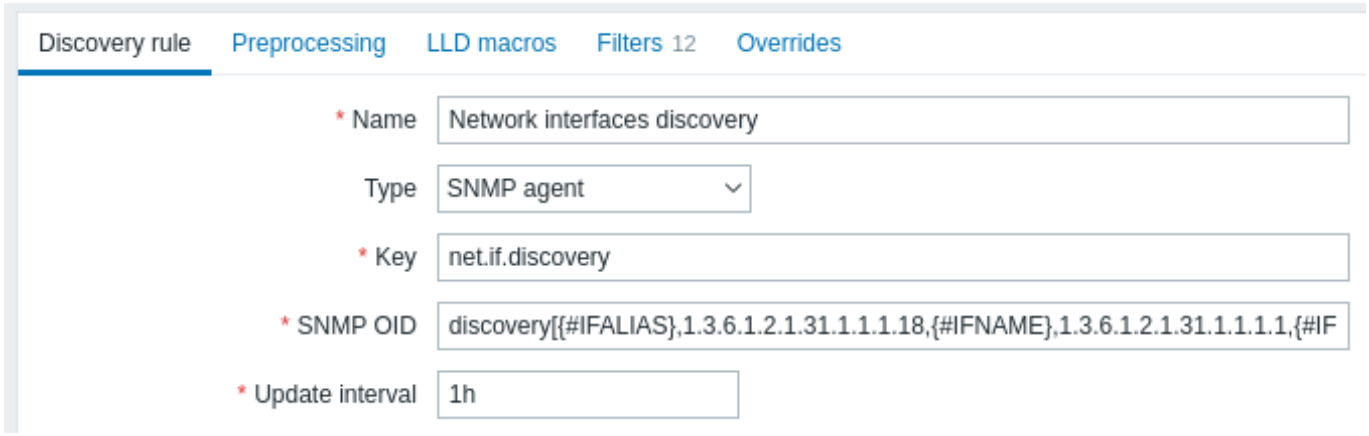
A differenza del rilevamento del file system e dell'interfaccia di rete, l'item non deve necessariamente avere una chiave "snmp.discovery": è sufficiente il tipo di item SNMP agent.

Per configurare la regola di rilevamento, procedere come segue:

- Andare in: *Data collection* → *Templates*
- Fare clic su *Discovery* nella riga del template appropriato



- Fare clic su *Create discovery rule* nell'angolo superiore destro della schermata
- Compilare il modulo della regola di rilevamento con i dettagli richiesti come mostrato nello screenshot seguente



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

Gli OID da rilevare sono definiti nel campo SNMP OID nel seguente formato: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

dove *{#MACRO1}*, *{#MACRO2}* ... sono nomi validi di macro lld e *oid1*, *oid2*... sono OID in grado di generare valori significativi per queste macro. Una macro integrata *{#SNMPINDEX}* contenente l'indice dell'OID rilevato viene applicata alle entità rilevate. Le entità rilevate sono raggruppate in base al valore della macro *{#SNMPINDEX}*.

Ecco un esempio in cui vengono eseguiti alcuni snmpwalk sullo switch:

```
$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: WAN
IF-MIB::ifDescr.2 = STRING: LAN1
IF-MIB::ifDescr.3 = STRING: LAN2

$ snmpwalk -v 2c -c public 192.168.1.1 IF-MIB::ifPhysAddress
IF-MIB::ifPhysAddress.1 = STRING: 8:0:27:90:7a:75
IF-MIB::ifPhysAddress.2 = STRING: 8:0:27:90:7a:76
IF-MIB::ifPhysAddress.3 = STRING: 8:0:27:2b:af:9e
```

E lo SNMP OID è impostato su: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Ora questa regola rileverà entità con macro {#IFDESCR} impostate su **WAN**, **LAN1** e **LAN2**, macro {#IFPHYSADDRESS} impostate su **8:0:27:90:7a:75**, **8:0:27:90:7a:76** e **8:0:27:2b:af:9e**, macro {#SNMPINDEX} impostate sugli indici degli OID rilevati **1**, **2** e **3**:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#IFDESCR}": "WAN",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:75"
    },
    {
```

```
        "{#SNMPINDEX}": "2",
        "{#IFDESCR}": "LAN1",
        "{#IFPHYSADDRESS}": "8:0:27:90:7a:76"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#IFDESCR}": "LAN2",
        "{#IFPHYSADDRESS}": "8:0:27:2b:af:9e"
    }
]
```

Se un'entità non ha l'OID specificato, la macro corrispondente verrà omessa per tale entità. Ad esempio, se abbiamo i seguenti dati:

```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

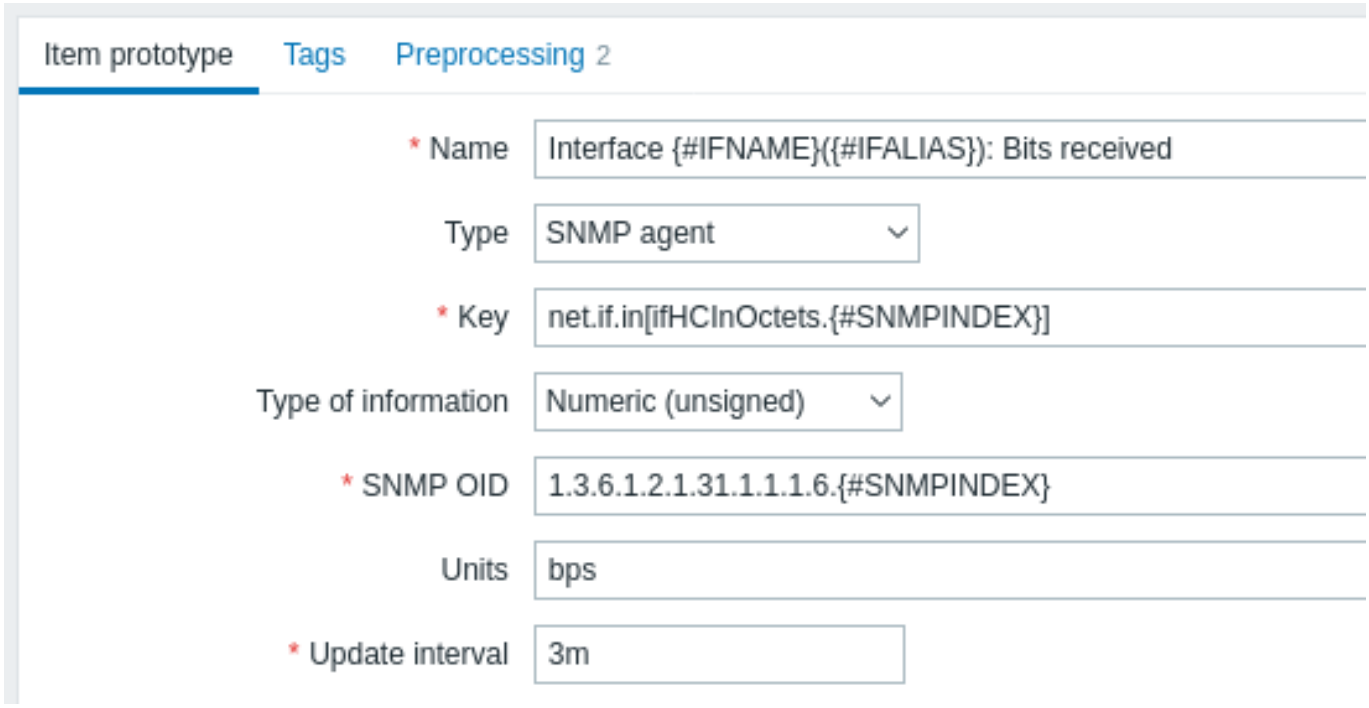
ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

In questo caso, il rilevamento SNMP `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` restituirà la seguente struttura:

```
[
    {
        "{#SNMPINDEX}": 1,
        "{#IFDESCR}": "Interface #1",
        "{#IFALIAS}": "eth0"
    },
    {
        "{#SNMPINDEX}": 2,
        "{#IFDESCR}": "Interface #2",
        "{#IFALIAS}": "eth1"
    },
    {
        "{#SNMPINDEX}": 3,
        "{#IFALIAS}": "eth2"
    },
    {
        "{#SNMPINDEX}": 4,
        "{#IFDESCR}": "Interface #4"
    },
    {
        "{#SNMPINDEX}": 5,
        "{#IFALIAS}": "eth4"
    }
]
```

Prototipi di item

La seguente schermata illustra come possiamo utilizzare queste macro nei prototipi di item:



È possibile creare tutti i prototipi di item necessari:

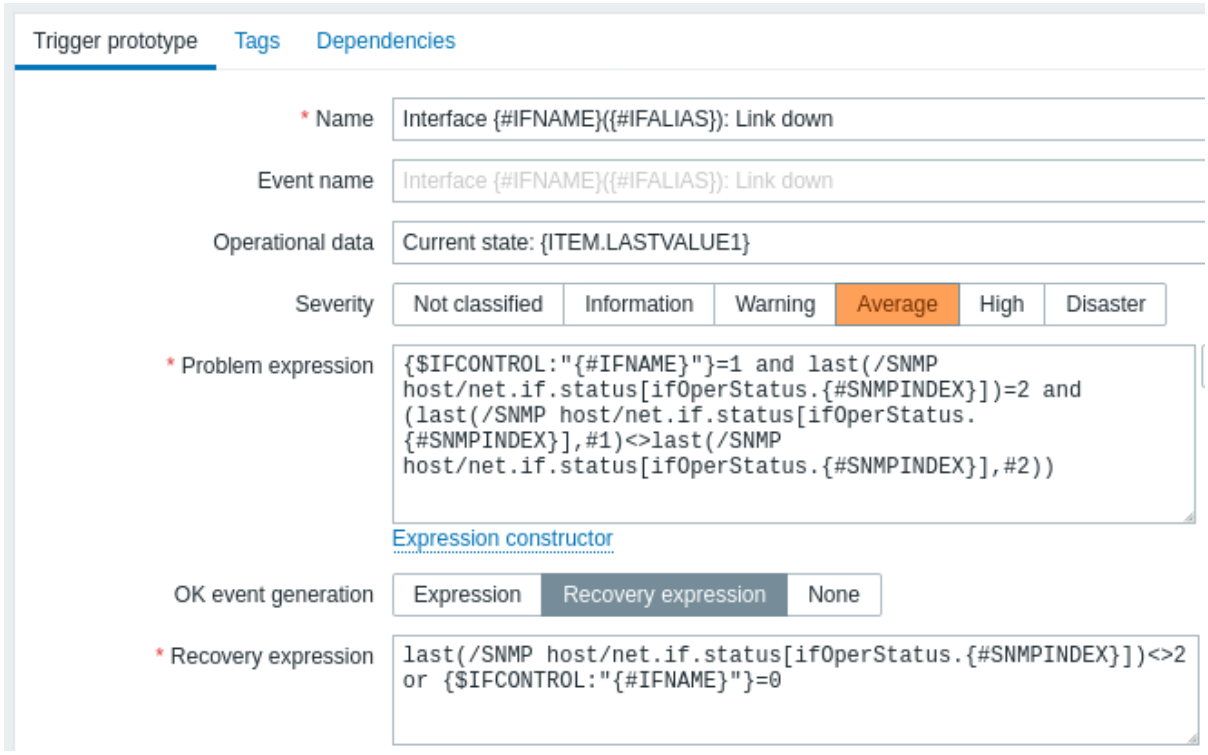
Item prototypes

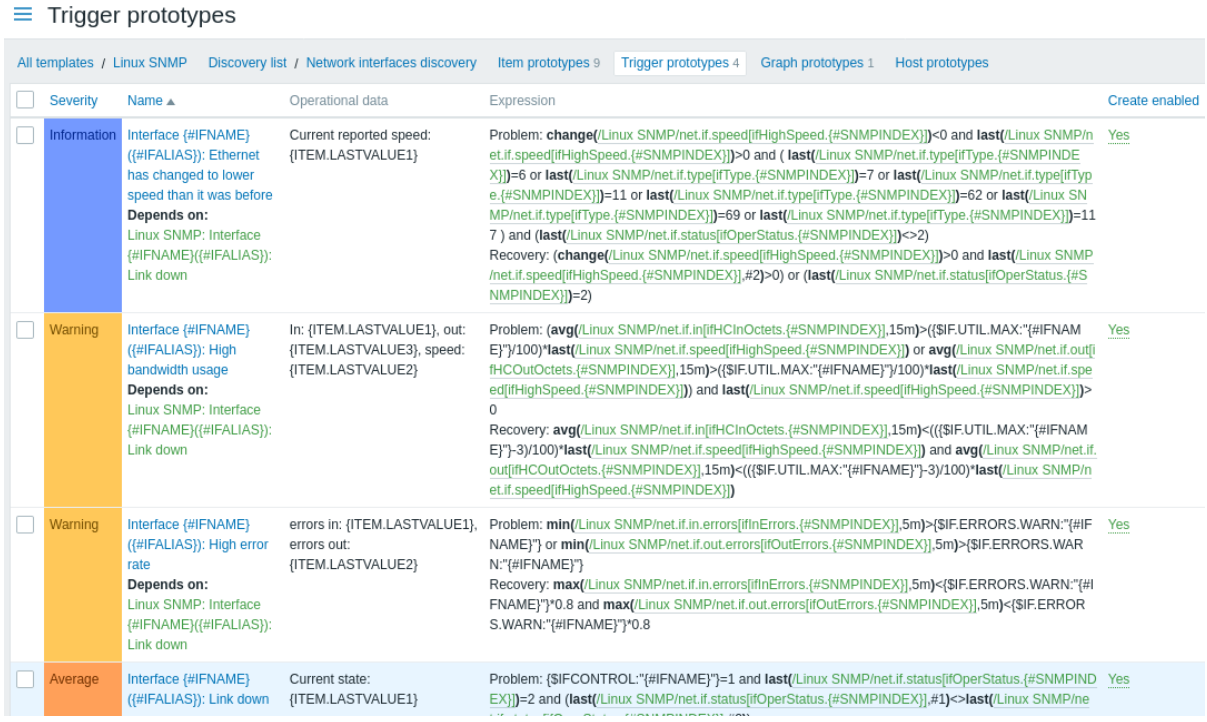
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

Name ▲	Key	Interval	History	Trends	Type	Create enabled
Interface {#IFNAME}({#IFALIAS}): Bits received	net.if.in[ifHCInOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Bits sent	net.if.out[ifHCOutOctets.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets discarded	net.if.in.discards[ifInDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Inbound packets with errors	net.if.in.errors[ifInErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Interface type	net.if.type[ifType.{#SNMPINDEX}]	1h	7d	0d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Operational status	net.if.status[ifOperStatus.{#SNMPINDEX}]	1m	7d	0	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets discarded	net.if.out.discards[ifOutDiscards.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Outbound packets with errors	net.if.out.errors[ifOutErrors.{#SNMPINDEX}]	3m	7d	365d	SNMP agent	Yes
Interface {#IFNAME}({#IFALIAS}): Speed	net.if.speed[ifHighSpeed.{#SNMPINDEX}]	5m	7d	0d	SNMP agent	Yes

Prototipi di trigger

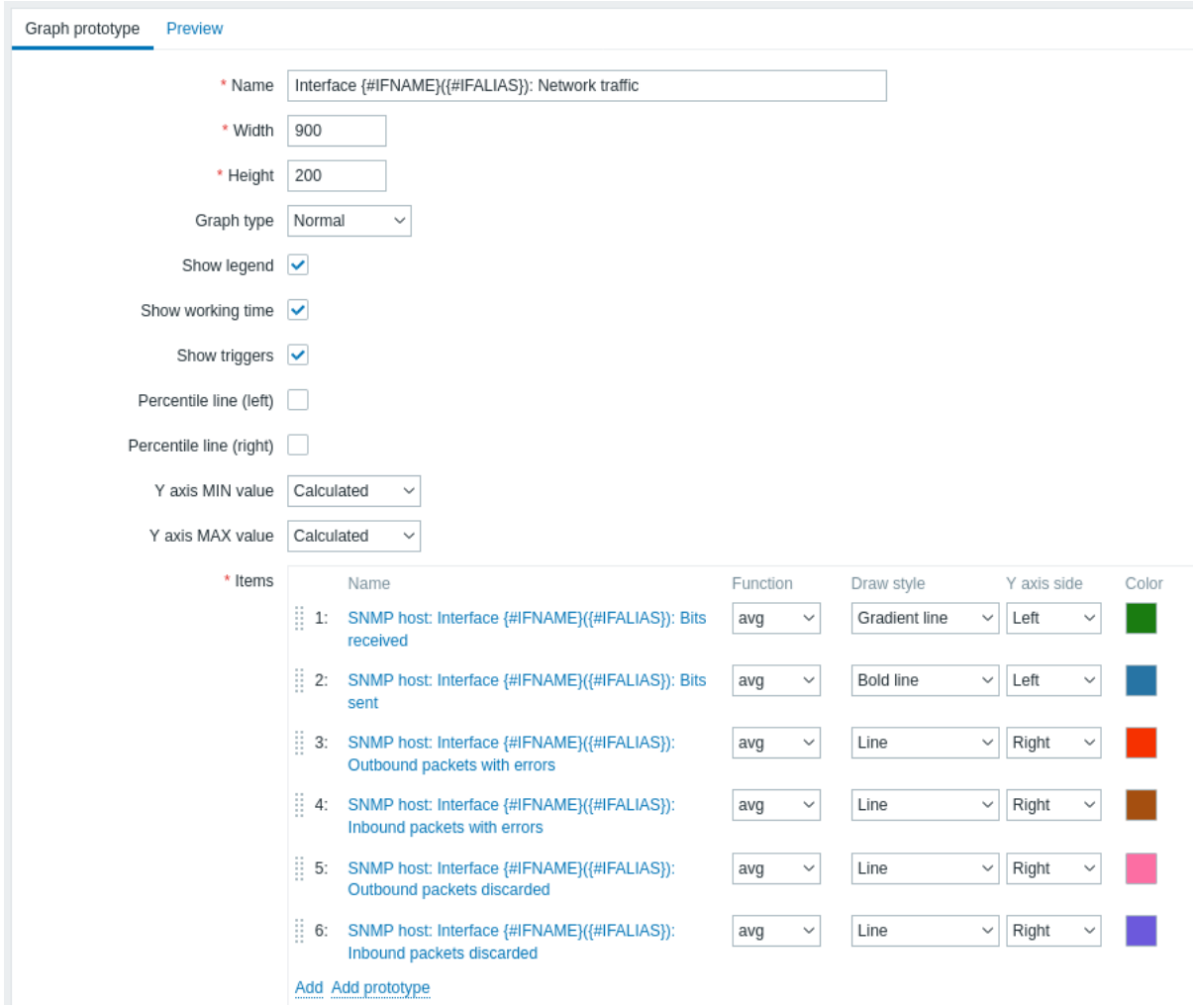
La seguente schermata illustra come possiamo utilizzare queste macro nei prototipi di trigger:

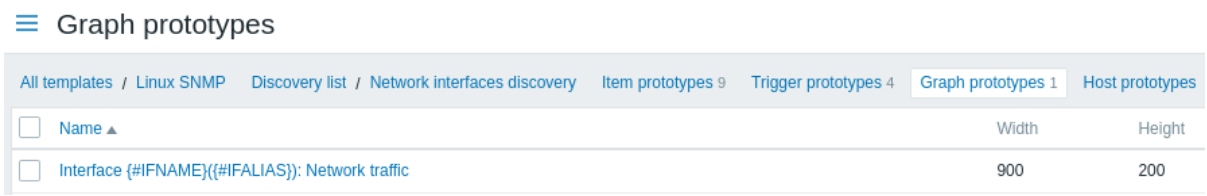




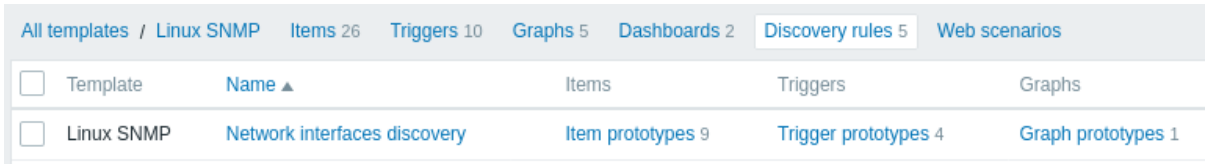
Prototipi di grafico

La seguente schermata illustra come possiamo utilizzare queste macro nei prototipi di grafico:





Un riepilogo della nostra regola di discovery:



Entità rilevate

Quando il server è in esecuzione, creerà item, trigger e grafici reali in base ai valori restituiti dalla regola di individuazione SNMP. Nella configurazione del host sono preceduti da un collegamento arancione alla regola di individuazione da cui provengono.

Items

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲	Triggers	Key	Interval	History	Trends	Type	Status
Network interfaces discovery: Interface enp4s0(): Bits received	Triggers 1	net.if.in[ifHCInOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Bits sent	Triggers 1	net.if.out[ifHCOutOctets.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets discarded		net.if.in.discards[ifInDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Inbound packets with errors	Triggers 1	net.if.in.errors[ifInErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Interface type	Triggers 1	net.if.type[ifType.2]	1h	7d	0d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Operational status	Triggers 2	net.if.status[ifOperStatus.2]	1m	7d	0	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets discarded		net.if.out.discards[ifOutDiscards.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Outbound packets with errors	Triggers 1	net.if.out.errors[ifOutErrors.2]	3m	7d	365d	SNMP agent	Enabled
Network interfaces discovery: Interface enp4s0(): Speed	Triggers 2	net.if.speed[ifHighSpeed.2]	5m	7d	0d	SNMP agent	Enabled

Triggers

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Severity	Value	Name ▲	Operational data	Expression
Information	OK	Network interfaces discovery: Interface enp4s0(): Ethernet has changed to lower speed than it was before **Depends on:** SNMP host: Interface enp4s0(): Link down	Current reported speed: {ITEM.LASTVALUE1}	Problem: **change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**<0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and (**last(**/SNMP host/net.if.type[ifType.2]**)**=6 or **last(**/SNMP host/net.if.type[ifType.2]**)**=7 or **last(**/SNMP host/net.if.type[ifType.2]**)**=11 or **last(**/SNMP host/net.if.type[ifType.2]**)**=62 or **last(**/SNMP host/net.if.type[ifType.2]**)**=69 or **last(**/SNMP host/net.if.type[ifType.2]**)**=117) and (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2) Recovery: (**change(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 and **last(**/SNMP host/net.if.speed[ifHighSpeed.2],#2**)**>0) or (**last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2)
Warning	OK	Network interfaces discovery: Interface enp4s0(): High bandwidth usage **Depends on:** SNMP host: Interface enp4s0(): Link down	In: {ITEM.LASTVALUE1}, out: {ITEM.LASTVALUE3}, speed: {ITEM.LASTVALUE2}	Problem: (**avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) or **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**>({$IF.UTIL.MAX:"enp4s0"}/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**) and **last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**>0 Recovery: **avg(**/SNMP host/net.if.in[ifHCInOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)** and **avg(**/SNMP host/net.if.out[ifHCOutOctets.2],15m**)**<(({$IF.UTIL.MAX:"enp4s0"}-3)/100)***last(**/SNMP host/net.if.speed[ifHighSpeed.2]**)**
Warning	OK	Network interfaces discovery: Interface enp4s0(): High error rate **Depends on:** SNMP host: Interface enp4s0(): Link down	errors in: {ITEM.LASTVALUE1}, errors out: {ITEM.LASTVALUE2}	Problem: **min(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} or **min(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**>{$IF.ERRORS.WARN:"enp4s0"} Recovery: **max(**/SNMP host/net.if.in.errors[ifInErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8 and **max(**/SNMP host/net.if.out.errors[ifOutErrors.2],5m**)**<{$IF.ERRORS.WARN:"enp4s0"}*0.8
Average	OK	Network interfaces discovery: Interface enp4s0(): Link down	Current state: {ITEM.LASTVALUE1}	Problem: {$IFCONTROL:"enp4s0"}=1 and **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**=2 and (**last(**/SNMP host/net.if.status[ifOperStatus.2],#1**)**<>**last(**/SNMP host/net.if.status[ifOperStatus.2],#2**)**) Recovery: **last(**/SNMP host/net.if.status[ifOperStatus.2]**)**<>2 or {$IFCONTROL:"enp4s0"}=0

Graphs

All hosts / SNMP host Enabled SNMP Items 81 Triggers 23 Graphs 14 Discovery rules 6 Web scenarios

Name ▲
Mounted filesystem discovery: /: Disk space usage
Linux SNMP: CPU jumps
CPU discovery: CPU usage
CPU discovery: CPU utilization
Network interfaces discovery: Interface enp4s0(): Network traffic

6 Scoperta di oggetti JMX

Panoramica

È possibile individuare tutti gli MBean JMX o gli attributi MBean, oppure specificare un pattern per l'individuazione di questi oggetti.

È obbligatorio comprendere la differenza tra un MBean e gli attributi MBean per la configurazione della regola di individuazione. Un MBean è un oggetto che può rappresentare un dispositivo, un'applicazione o qualsiasi risorsa che deve essere gestita.

Ad esempio, esiste un MBean che rappresenta un server web. I suoi attributi sono il numero di connessioni, il numero di thread, il timeout delle richieste, la cache dei file HTTP, l'utilizzo della memoria, ecc. Esprimendo questo concetto in un linguaggio comprensibile alle persone, possiamo definire una macchina da caffè come un MBean che ha i seguenti attributi da monitorare: quantità d'acqua per tazza, consumo medio di acqua in un determinato periodo di tempo, numero di chicchi di caffè necessari per tazza, tempo di ricarica dei chicchi di caffè e dell'acqua, ecc.

Chiave item

Nella configurazione della regola di discovery, selezionare **JMX agent** nel campo *Tipo*.

Per la discovery degli oggetti JMX sono supportate due chiavi item: jmx.discovery[] e jmx.get[]:

Chiave item	Valore restituito	Parametri	Commento
jmx.discovery[<modalità di discovery>,<nome oggetto>,<breve descrizione univoca>]	Questo item restituisce un array JSON con macro LLD che descrivono gli oggetti MBean o i loro attributi.	**modalità di discovery** - una delle seguenti: *attributes* (recupera gli attributi JMX MBean, predefinito) oppure *beans* (recupera gli MBean JMX) **nome oggetto** - pattern del nome oggetto (vedere la documentazione) che identifica i nomi MBean da recuperare (vuoto per impostazione predefinita, recupera tutti i bean registrati) **breve descrizione univoca** - una descrizione univoca che consente più item JMX con la stessa modalità di discovery e nome oggetto sull'host (opzionale)	Esempi: → jmx.discovery - recupera tutti gli attributi JMX MBean → jmx.discovery[beans] - recupera tutti gli MBean JMX → jmx.discovery[attributes,"*:type=GarbageCollector,name - recupera tutti gli attributi del garbage collector → jmx.discovery[beans,"*:type=GarbageCollector,name=* - recupera tutti i garbage collector Esistono alcune limitazioni relative alle proprietà MBean che questo item può restituire, a causa del numero limitato di caratteri supportati nella generazione dei nomi delle macro (i caratteri supportati possono essere descritti dalla seguente espressione regolare: `A-Z0-9_\.`). Quindi, ad esempio, per individuare proprietà MBean con parole contenenti trattini o caratteri non ASCII, è necessario usare `jmx.get[]`.
jmx.get[<modalità di discovery>,<nome oggetto>,<breve descrizione univoca>]			

Chiave item			
	Questo item restituisce un array JSON con oggetti MBean o i loro attributi. Rispetto a `jmx.discovery[]`, non definisce macro LLD.	**modalità di discovery** - una delle seguenti: *attributes* (recupera gli attributi JMX MBean, predefinito) oppure *beans* (recupera gli MBean JMX) **nome oggetto** - pattern del nome oggetto (vedere la documentazione) che identifica i nomi MBean da recuperare (vuoto per impostazione predefinita, recupera tutti i bean registrati) **breve descrizione univoca** - una descrizione univoca che consente più item JMX con la stessa modalità di discovery e nome oggetto sull'host (opzionale)	Quando si utilizza questo item, è necessario definire macro di low-level discovery personalizzate, che puntino ai valori estratti dal JSON restituito usando JSONPath.

Attention:
Se non viene passato alcun parametro, vengono richiesti tutti gli attributi MBean da JMX. Non specificare parametri per la discovery JMX oppure tentare di ricevere tutti gli attributi per un intervallo ampio come `*:type=*,name=*` può causare potenziali problemi di prestazioni.

Utilizzo di jmx.discovery

Questo item restituisce un oggetto JSON con macro di low-level discovery che descrivono oggetti o attributi MBean. Ad esempio, nel rilevamento degli attributi MBean (riformattato per maggiore chiarezza):


```
[
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionCount",
        "{#JMXATTR}":"CollectionCount"
    },
    {
        "{#JMXVALUE}":"0",
        "{#JMXTYPE}":"java.lang.Long",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,CollectionTime",
        "{#JMXATTR}":"CollectionTime"
    },
    {
        "{#JMXVALUE}":"true",
        "{#JMXTYPE}":"java.lang.Boolean",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Valid",
        "{#JMXATTR}":"Valid"
    },
    {
        "{#JMXVALUE}":"PS Scavenge",
        "{#JMXTYPE}":"java.lang.String",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,Name",
        "{#JMXATTR}":"Name"
    },
    {
        "{#JMXVALUE}":"java.lang:type=GarbageCollector,name=PS Scavenge",
```

```
        "{#JMXTYPE}":"javax.management.ObjectName",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXDESC}":"java.lang:type=GarbageCollector,name=PS Scavenge,ObjectName",
        "{#JMXATTR}":"ObjectName"
    }
]
```

Nel rilevamento degli MBean (riformattato per maggiore chiarezza):

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Macro supportate

Le seguenti macro sono supportate per l'uso nel filtro della regola di discovery e nei prototipi di item, trigger e grafici:

Macro	Descrizione
Discovery degli attributi MBean	
{#JMXVALUE}	Valore dell'attributo.
{#JMXTYPE}	Tipo di attributo.
{#JMXOBJ}	Nome dell'oggetto.
{#JMXDESC}	Nome dell'oggetto incluso il nome dell'attributo.
{#JMXATTR}	Nome dell'attributo.
Discovery degli MBean	
{#JMXDOMAIN}	Dominio MBean. (*nome riservato di Zabbix*)
{#JMXOBJ}	Nome dell'oggetto. (*nome riservato di Zabbix*)
{#JMX<key property>}	Proprietà MBean (come {#JMXTYPE}, {#JMXNAME}) (vedere Limitazioni sotto).

Limitazioni

Esistono alcune limitazioni associate all'algoritmo di creazione dei nomi delle macro LLD a partire dai nomi delle proprietà MBean:

- i nomi degli attributi vengono convertiti in maiuscolo
- i nomi degli attributi vengono ignorati (non vengono generate macro LLD) se sono costituiti da caratteri non supportati per i nomi delle macro LLD. I caratteri supportati possono essere descritti dalla seguente espressione regolare: `A-Z0-9_\.`
- se un attributo si chiama "obj" o "domain", verrà ignorato a causa della sovrapposizione con i valori delle proprietà Zabbix riservate {#JMXOBJ} e {#JMXDOMAIN}

Si consideri questo esempio di jmx.discovery (con modalità "beans"). MBean ha definite le seguenti proprietà (alcune delle quali verranno ignorate; vedi sotto):

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

Come risultato della discovery JMX, verranno generate le seguenti macro LLD:

- {#JMXDOMAIN} - interna di Zabbix, descrive il dominio di MBean
- {#JMXOBJ} - interna di Zabbix, descrive l'oggetto MBean
- {#JMXNAME} - creata dalla proprietà "name"

Le proprietà ignorate sono:

- тип : il suo nome contiene caratteri non supportati (non ASCII)
- attributes[] : il suo nome contiene caratteri non supportati (le parentesi quadre non sono supportate)
- Name : è già definita (name=test)
- domAin : è un nome riservato di Zabbix

Esempi

Esaminiamo altri due esempi pratici di creazione di una regola LLD con l'uso di MBean. Per comprendere meglio la differenza tra una regola LLD che raccoglie MBean e una regola LLD che raccoglie attributi MBean, consultare la tabella seguente:

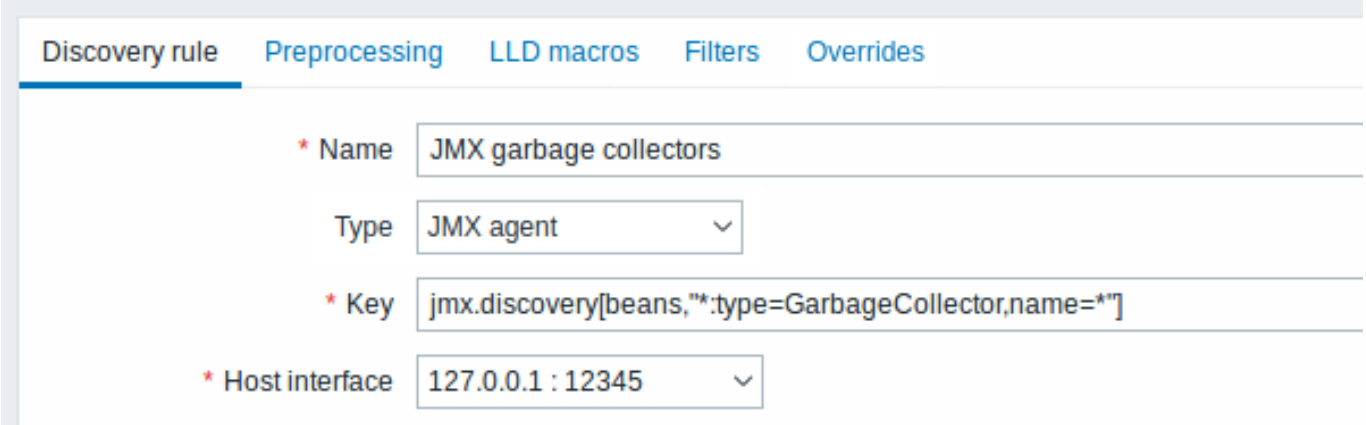
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Esempio 1: individuazione degli MBean

Questa regola restituirà 3 oggetti: la riga superiore della colonna: MBean1, MBean2, MBean3.

Per ulteriori informazioni sugli oggetti, fare riferimento alla tabella delle macro supportate, sezione *Individuazione degli MBean*.

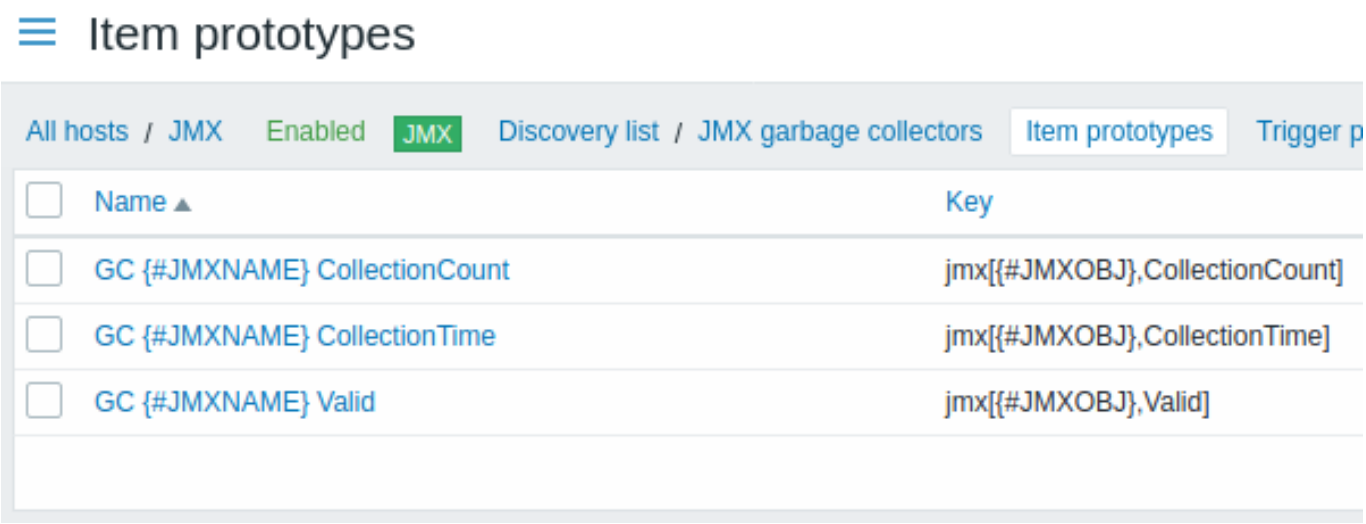
La configurazione della regola di discovery che raccoglie gli MBean (senza gli attributi) appare come segue:



La chiave utilizzata qui:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

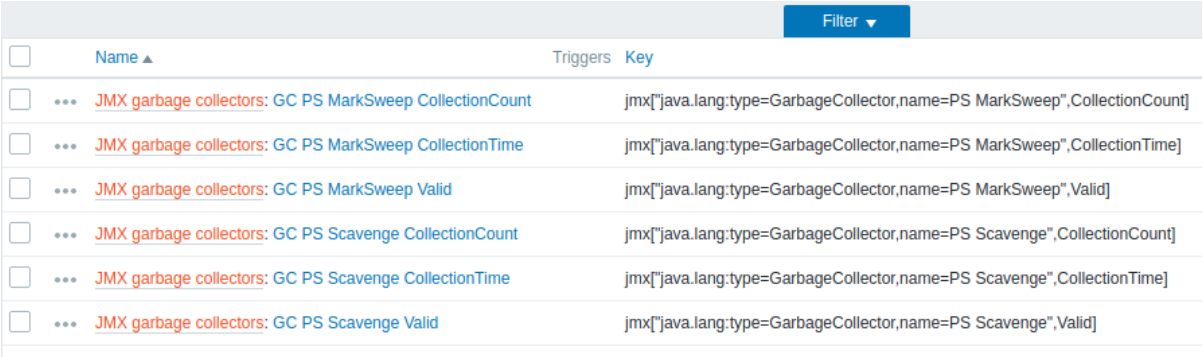
Verranno individuati tutti i garbage collector senza attributi. Poiché i garbage collector hanno lo stesso insieme di attributi, possiamo usare gli attributi desiderati nei prototipi di item nel modo seguente:



Le chiavi utilizzate qui:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

La regola di discovery LLD produrrà un risultato simile a questo (gli item vengono individuati per due garbage collector):

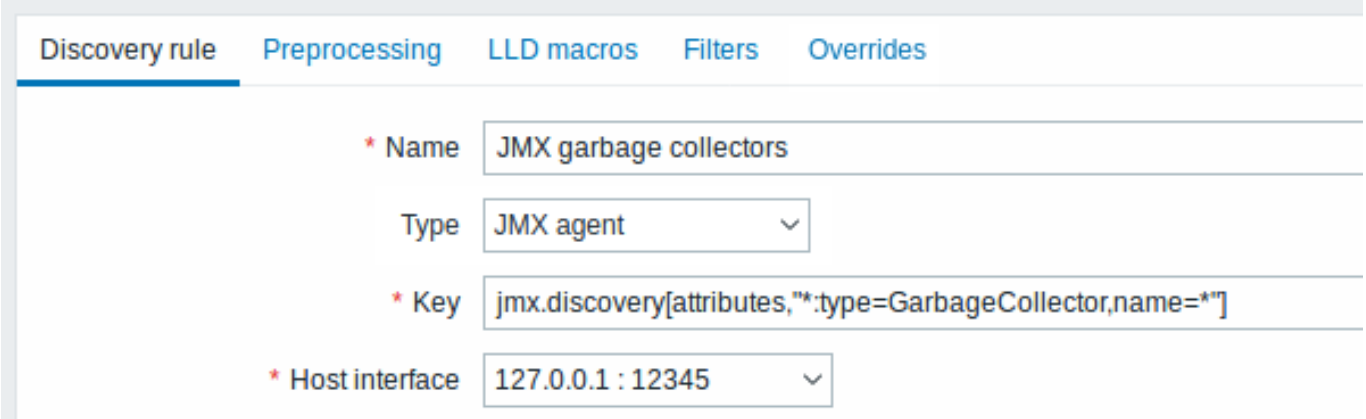


Esempio 2: individuazione degli attributi MBean

Questa regola restituirà 9 oggetti con i seguenti campi: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBean2At MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Per ulteriori informazioni sugli oggetti, fare riferimento alla tabella delle macro supportate, sezione *Individuazione degli attributi MBean*.

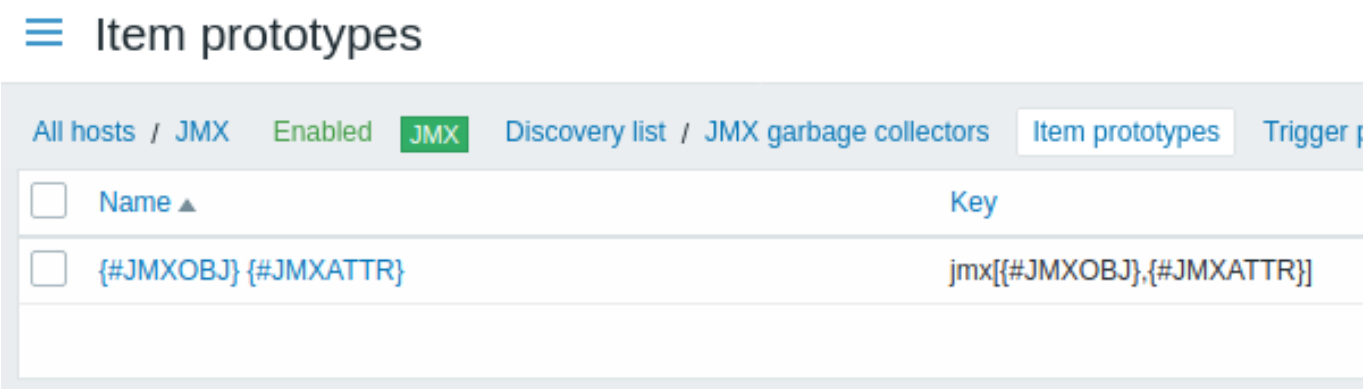
La configurazione della regola di individuazione che raccoglie gli attributi MBean appare come segue:



La chiave utilizzata qui:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

Verranno individuati tutti i garbage collector con un singolo attributo item.



In questo caso particolare, verrà creato un item dal prototipo per ogni attributo MBean. Il principale svantaggio di questa configurazione è che la creazione dei trigger dai prototipi di trigger è impossibile, poiché esiste un solo prototipo di item per tutti gli attributi. Pertanto, questa configurazione può essere utilizzata per la raccolta dei dati, ma non è consigliata per il monitoraggio automatico.

Utilizzo di jmx.get

`jmx.get[]` è simile all'item `jmx.discovery[]`, ma non trasforma le proprietà degli oggetti Java in nomi di macro di low-level discovery e pertanto può restituire valori senza le limitazioni che sono associate alla generazione dei nomi delle macro LLD, come trattini o caratteri non ASCII.

Quando si utilizza `jmx.get[]` per il discovery, le macro di low-level discovery possono essere definite separatamente nella scheda personalizzata macro LLD della configurazione della regola di discovery, utilizzando JSONPath per puntare ai valori richiesti.

Individuazione degli MBean

Item di discovery: `jmx.get[beans,"com.example:type=*,*"]`

Risposta:

```
[
    {
        "object": "com.example:type=Hello,data-src=data-base,  =    ",
        "domain": "com.example",
        "properties": {
            "data-src": "data-base",
            "  ": "    ",
            "type": "Hello"
        }
    },
    {
        "object": "com.example:type=Atomic",
        "domain": "com.example",
        "properties": {
            "type": "Atomic"
        }
    }
]
```

Individuazione degli attributi MBean

Item di discovery: `jmx.get[attributes,"com.example:type=*,*"]`

Risposta:

```
[
    {
        "object": "com.example:type=*",
        "domain": "com.example",
        "properties": {
            "type": "Simple"
        }
    },
    {
        "object": "com.zabbix:type=yes,domain=zabbix.com,data-source=/dev/rand,  =    ,obj=true",
        "domain": "com.zabbix",
        "properties": {
            "type": "Hello",
            "domain": "com.example",
            "data-source": "/dev/rand",
            "  ": "    ",
            "obj": true
        }
    }
]
```

7 Scoperta dei sensori IPMI

Panoramica

È possibile individuare automaticamente i sensori IPMI.

Per farlo, è possibile utilizzare una combinazione di:

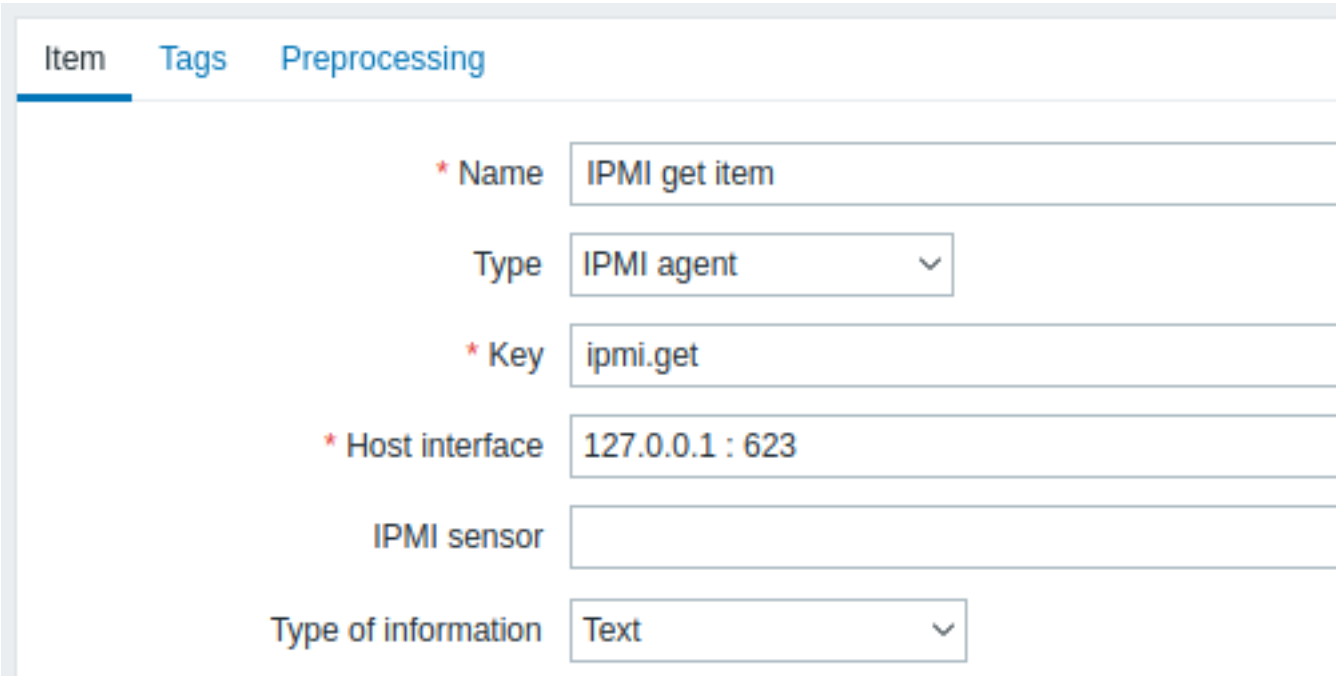
- l'item IPMI `ipmi.get` come item principale
- regola di low-level discovery dipendente e prototipi di item

Configurazione

Item master

Creare un item IPMI utilizzando la seguente chiave:

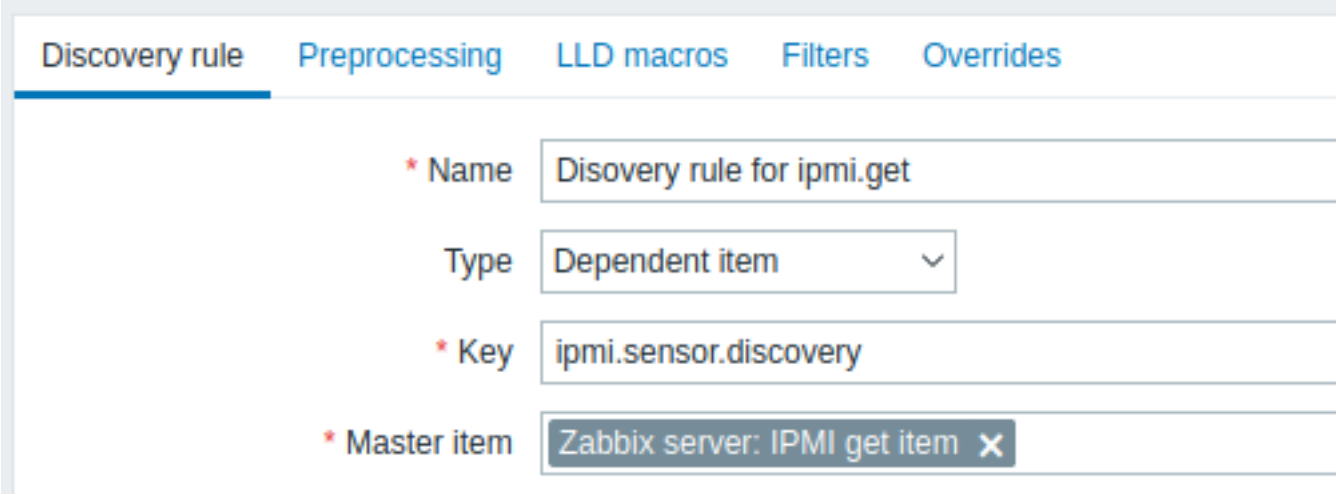
`ipmi.get`



Impostare il tipo di informazioni su ”Text” per dati JSON potenzialmente di grandi dimensioni.

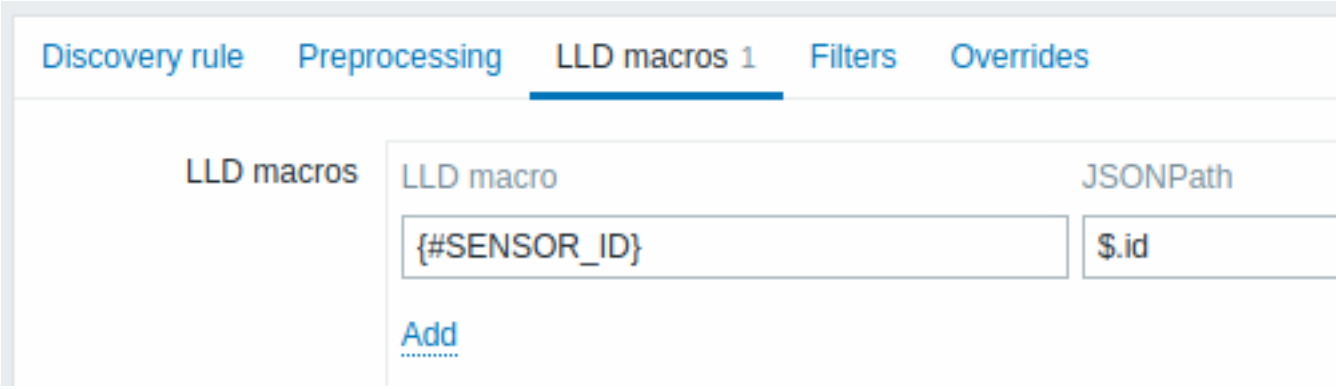
Regola LLD dipendente

Creare una regola di low-level discovery di tipo ”Dependent item”:



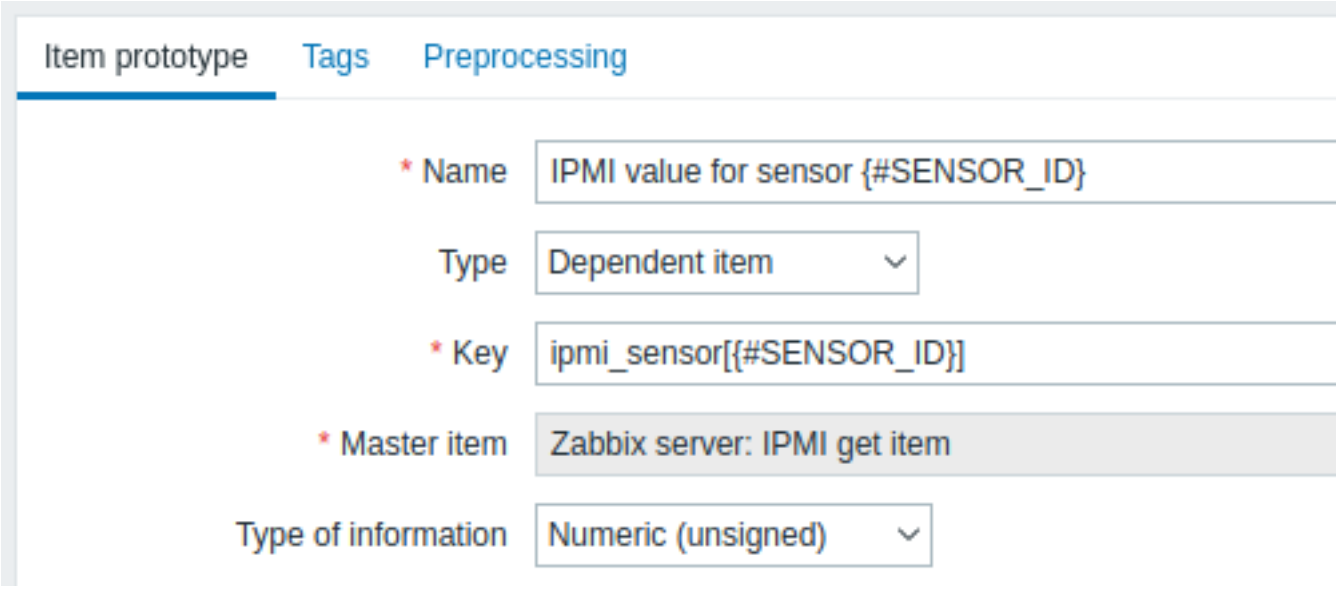
Come item principale selezionare l’item `ipmi.get` creato.

Nella scheda ”LLD macros” definire una macro personalizzata con il corrispondente JSONPath:



Prototipo di item dipendente

Creare un prototipo di item con tipo ”Item dipendente” in questa regola LLD. Come item master per questo prototipo selezionare l’item `ipmi.get` che abbiamo creato.



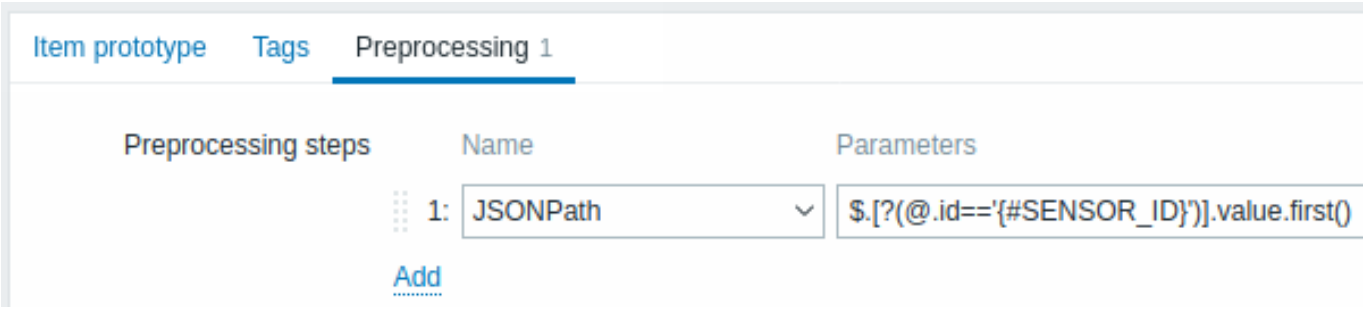
Notare l'uso della macro {#SENSOR_ID} nel nome e nella chiave del prototipo di item:

- *Nome*: Valore IPMI per il sensore {#SENSOR_ID}
- *Chiave*: ipmi_sensor[{#SENSOR_ID}]

Come tipo di informazione, *Numerico (senza segno)*.

Nella scheda "Preprocessing" del prototipo di item selezionare JSONPath e utilizzare la seguente espressione JSONPath come parametro:

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



Quando il discovery si avvia, verrà creato un item per ciascun sensore IPMI. Questo item restituirà il valore intero del sensore specificato.

8 Individuazione dei servizi systemd

Panoramica

È possibile individuare le unità systemd (i servizi, per impostazione predefinita) con Zabbix.

Chiave item

L'item da utilizzare nella regola di discovery è

```
systemd.unit.discovery
```

> **Attention:**
> Questa chiave item è supportata solo in Zabbix agent 2.

Questo item restituisce un JSON con informazioni sulle unità systemd, ad esempio:

```
[{
    "{#UNIT.NAME}": "mysqld.service",
    "{#UNIT.DESCRIPTION}": "MySQL Server",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
```

```
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/mysqld_2eservice",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
    "{#UNIT.SERVICETYPE}": "simple"
}, {
    "{#UNIT.NAME}": "systemd-journald.socket",
    "{#UNIT.DESCRIPTION}": "Journal Socket",
    "{#UNIT.LOADSTATE}": "loaded",
    "{#UNIT.ACTIVESTATE}": "active",
    "{#UNIT.SUBSTATE}": "running",
    "{#UNIT.FOLLOWED}": "",
    "{#UNIT.PATH}": "/org/freedesktop/systemd1/unit/systemd_2djournald_2esocket",
    "{#UNIT.JOBID}": 0,
    "{#UNIT.JOBTYPE}": "",
    "{#UNIT.JOBPATH}": "/",
    "{#UNIT.UNITFILESTATE}": "enabled"
}]
```

Individuazione delle unità systemd disabilitate

È anche possibile individuare le unità systemd **disabilitate**. In questo caso, nel JSON risultante vengono restituite tre macro:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
Per fare in modo che item e trigger vengano creati dai prototipi per le unità systemd disabilitate, assicurati di modificare (o rimuovere) i filtri LLD che ne impediscono la creazione per {#UNIT.ACTIVESTATE} e {#UNIT.UNITFILESTATE}.

Macro supportate

Le seguenti macro sono supportate per l'uso nel filtro della regola di discovery e nei prototipi di item, trigger e grafici:

Macro	Descrizione
{#UNIT.NAME}	Nome dell'unità principale.
{#UNIT.DESCRIPTION}	Descrizione leggibile dall'utente.
{#UNIT.LOADSTATE}	Stato di caricamento (cioè se il file dell'unità è stato caricato correttamente)
{#UNIT.ACTIVESTATE}	Stato attivo (cioè se l'unità è attualmente avviata o meno)
{#UNIT.SUBSTATE}	Sottostato (una versione più dettagliata dello stato attivo, specifica per il tipo di unità, a differenza dello stato attivo)
{#UNIT.FOLLOWED}	Unità il cui stato viene seguito da questa unità, se presente; altrimenti una stringa vuota.
{#UNIT.PATH}	Percorso dell'oggetto unità.
{#UNIT.JOBID}	ID numerico del job se è presente un job in coda per l'unità del job; 0 altrimenti.
{#UNIT.JOBTYPE}	Tipo di job.
{#UNIT.JOBPATH}	Percorso dell'oggetto job.
{#UNIT.UNITFILESTATE}	Stato di installazione del file dell'unità.
{#UNIT.SERVICETYPE}	Tipo dell'unità di servizio (ad esempio `simple`, `forking`, `oneshot`, `idle`, ecc.). Questa macro viene restituita solo se l'unità è un servizio.

Prototipi di item

I prototipi di item che possono essere creati in base al rilevamento dei servizi systemd includono, ad esempio:

- Nome item: `{#UNIT.DESCRIPTION} informazioni sullo stato attivo`; chiave item: `systemd.unit.info["{#UNIT.NAME`
- Nome item: `{#UNIT.DESCRIPTION} informazioni sullo stato di caricamento`; chiave item: `systemd.unit.info["{#U`

9 Individuazione dei servizi Windows

Panoramica

In modo simile a come vengono individuati i file system, è possibile individuare anche i servizi Windows.

Chiave item

L'item da utilizzare nella regola di discovery è

`service.discovery`

Macro supportate

Le seguenti macro sono supportate per l'uso nel filtro della regola di discovery e nei prototipi di item, trigger e grafici:

Macro	Descrizione
{#SERVICE.NAME}	Nome del servizio.
{#SERVICE.DISPLAYNAME}	Nome visualizzato del servizio.
{#SERVICE.DESCRIPTION}	Descrizione del servizio.
{#SERVICE.STATE}	Valore numerico dello stato del servizio. Per i dettagli, vedere l'item service.info.
{#SERVICE.STATENAME}	Nome dello stato del servizio. Per i dettagli, vedere l'item service.info.
{#SERVICE.PATH}	Percorso del servizio.
{#SERVICE.USER}	Utente del servizio.
{#SERVICE.STARTUP}	Valore numerico del tipo di avvio del servizio. Per i dettagli, vedere l'item service.info.
{#SERVICE.STARTUPNAME}	Nome del tipo di avvio del servizio. Per i dettagli, vedere l'item service.info.
{#SERVICE.STARTUPTRIGGER}	Valore numerico che indica se il tipo di avvio del servizio ha: 0 - nessun trigger di avvio 1 - trigger di avvio presenti È utile per individuare tipi di avvio del servizio come *Automatico (avvio tramite trigger)*, *Automatico ritardato (avvio tramite trigger)* e *Manuale (avvio tramite trigger).*

In base alla discovery dei servizi Windows è possibile creare un prototipo di item come

`service.info[{#SERVICE.NAME},<param>]`

dove `param` accetta i seguenti valori: *state*, *displayname*, *path*, *user*, *startup* oppure *description*.

Ad esempio, per ottenere il nome visualizzato di un servizio è possibile usare un item "service.info[{#SERVICE.NAME},displayname]". Se il valore di `param` non è specificato ("service.info[{#SERVICE.NAME}]"), viene utilizzato il parametro predefinito *state*.

10 Individuazione di istanze di contatori delle prestazioni di Windows

Panoramica

È possibile individuare le istanze degli oggetti dei contatori delle prestazioni di Windows. Questo è utile per i contatori delle prestazioni con più istanze.

Chiave item

Per configurare la regola di individuazione, utilizzare il seguente item:

- `perf_instance.discovery[object]`

Si noti che il nome dell'oggetto può essere localizzato. Ad esempio:

```
perf_instance.discovery[Processor] # Il nome dell'oggetto è in inglese.
perf_instance.discovery[Processador] # Il nome dell'oggetto è in portoghese.
```

In alternativa, per garantire che il nome dell'oggetto venga fornito in inglese, indipendentemente dalla localizzazione del sistema operativo, utilizzare il seguente item:

- `perf_instance_en.discovery[object]`

Ad esempio:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Macro supportate

Il processo di discovery restituirà tutte le istanze dell'oggetto specificato nella macro `{#INSTANCE}`:

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Questa macro può essere utilizzata nei prototipi di item `perf_counter[]` e `perf_counter_en[]`.

Ad esempio, se la chiave item utilizzata nella regola di discovery è `perf_instance.discovery[Processor]`, puoi creare il seguente prototipo di item:

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Nota:

- Se l'oggetto specificato non viene trovato o non supporta istanze variabili, l'item di discovery diventerà NOTSUPPORTED.
- Se l'oggetto specificato supporta istanze variabili ma al momento non ha alcuna istanza, verrà restituito un array JSON vuoto.
- Le istanze duplicate verranno ignorate.

11 Individuazione tramite query WMI

Panoramica

WMI è una potente interfaccia di Windows che può essere utilizzata per recuperare varie informazioni sui componenti di Windows, i servizi, lo stato e il software installato.

Può essere utilizzata per l'individuazione dei dischi fisici e la raccolta dei relativi dati sulle prestazioni, l'individuazione delle interfacce di rete, l'individuazione dei guest Hyper-V, il monitoraggio dei servizi Windows e molte altre attività nel sistema operativo Windows.

Questo tipo di discovery di basso livello viene eseguito utilizzando query WQL i cui risultati vengono trasformati automaticamente in un oggetto JSON adatto alla discovery di basso livello.

Chiave item

L'item da utilizzare nella regola di discovery è

`wmi.getall[<namespace>,<query>]`

Questo item trasforma il risultato della query in un array JSON. Per esempio:

`select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'`

può restituire qualcosa di simile:

```
[
    {
        "DeviceID" : "\\.\PHYSICALDRIVE0",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 0,
        "InterfaceType" : "IDE"
    },
```

```
    {
        "DeviceID" : "\\.\PHYSICALDRIVE1",
        "BytesPerSector" : 512,
        "Capabilities" : [
            3,
            4
        ],
        "CapabilityDescriptions" : [
            "Random Access",
            "Supports Writing"
        ],
        "Caption" : "VBOX HARDDISK ATA Device",
        "ConfigManagerErrorCode" : 0,
        "ConfigManagerUserConfig" : "False",
        "CreationClassName" : "Win32_DiskDrive",
        "Description" : "Disk drive",
        "FirmwareRevision" : "1.0",
        "Index" : 1,
        "InterfaceType" : "IDE"
    }
]
```

Macro di low-level discovery

Anche se nel JSON restituito non vengono create macro di low-level discovery, queste macro possono essere definite dall'utente come passaggio aggiuntivo, utilizzando la funzionalità macro LLD personalizzata con JSONPath che punta ai valori individuati nel JSON restituito.

Le macro possono quindi essere utilizzate per creare prototipi di item, trigger, ecc.

12 Individuazione tramite query SQL ODBC

Panoramica

Questo tipo di discovery di basso livello viene eseguito utilizzando query SQL, i cui risultati vengono automaticamente trasformati in un oggetto JSON adatto alla discovery di basso livello.

Chiave item

Le query SQL vengono eseguite utilizzando un tipo di item "Database monitor". Pertanto, la maggior parte delle istruzioni nella pagina monitoraggio ODBC si applica per ottenere una regola di discovery "Database monitor" funzionante.

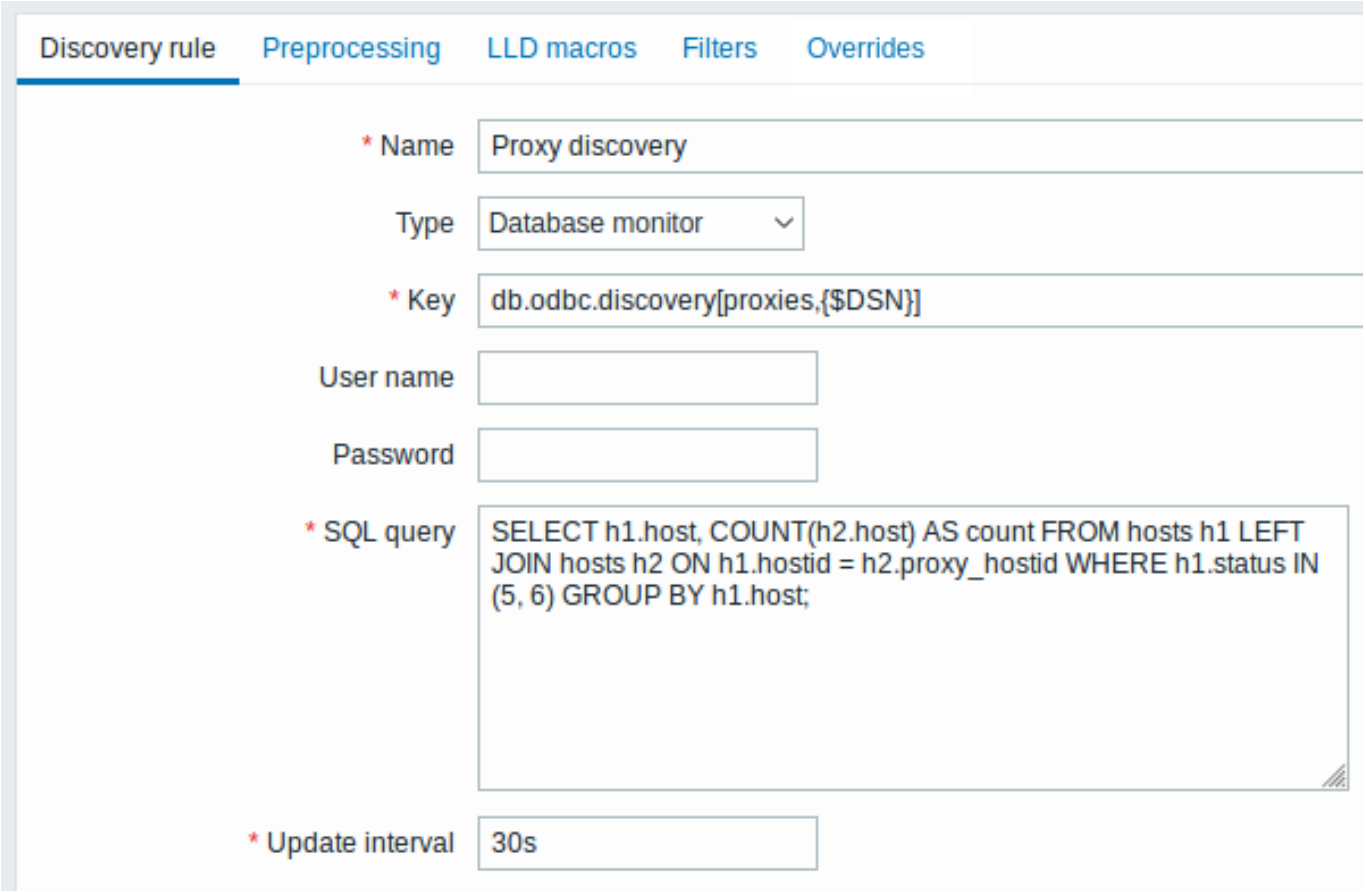
Nelle regole di discovery "Database monitor" possono essere utilizzate due chiavi item:

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] - questo item trasforma il risultato della query SQL in un array JSON, convertendo i nomi delle colonne del risultato della query 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. Vedi anche: Uso di db.odbc.discovery.
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] - questo item trasforma il risultato della query SQL in un array JSON, mantenendo i nomi originali delle colonne del risultato della query come nomi di campo nel JSON associati ai valori rilevati. Rispetto a `db.odbc.discovery[]`, questo item non crea macro di low-level discovery nel JSON restituito, pertanto non è necessario verificare se i nomi delle colonne possano essere nomi di macro validi. Le macro di low-level discovery possono essere definite come passaggio aggiuntivo secondo necessità, utilizzando la funzionalità macro LLD personalizzata con JSONPath che punta ai valori rilevati nel JSON restituito. Vedi anche: Uso di db.odbc.get.

Utilizzo di db.odbc.discovery

L'esempio seguente mostra come una query SQL venga trasformata in JSON utilizzando la low-level discovery dei proxy Zabbix, sulla base di una query ODBC sul database Zabbix. Questo è utile per la creazione automatica di internal items "zabbix[proxy,<name>,lastaccess]" per monitorare quali proxy sono attivi.

Iniziare con la configurazione della regola di discovery:



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

Qui viene utilizzata la seguente query diretta sul database Zabbix per selezionare tutti i proxy Zabbix, insieme al numero di host che stanno monitorando. Il numero di host può essere utilizzato, ad esempio, per escludere i proxy vuoti:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

Per il funzionamento interno dell'item "db.odbc.discovery[,{$DSN}]", il risultato di questa query viene trasformato automaticamente nel seguente JSON:

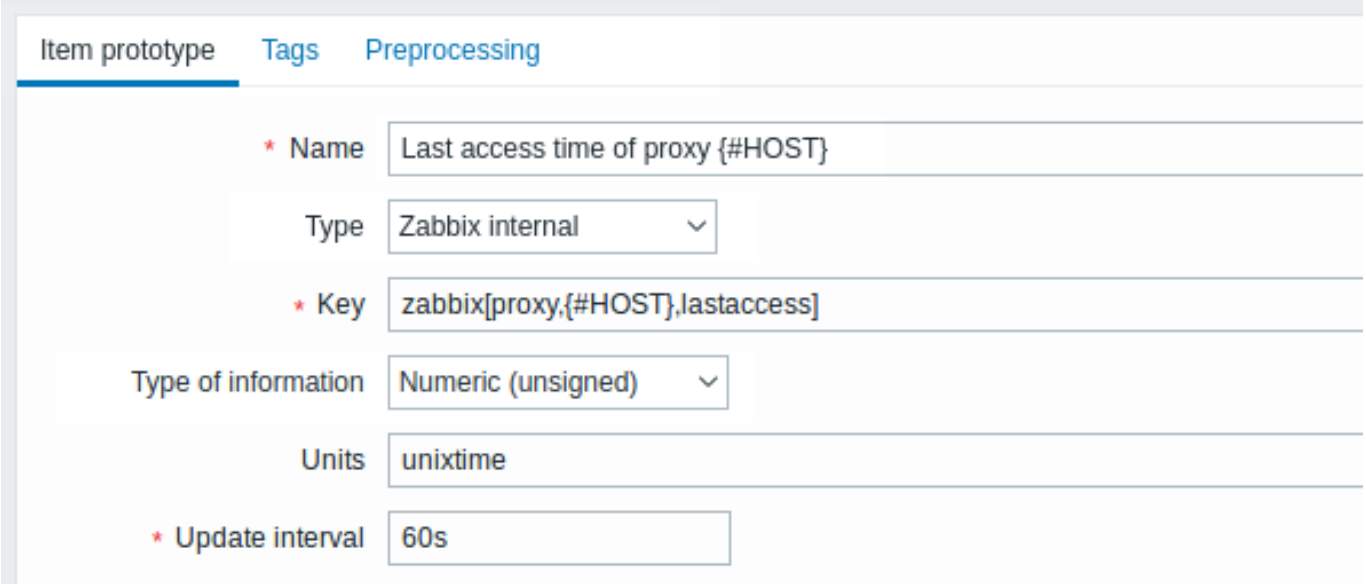
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

Si può notare che i nomi delle colonne diventano nomi di macro e le righe selezionate diventano i valori di queste macro.

Note:
Se non è evidente come un nome di colonna verrebbe trasformato in un nome di macro, si consiglia di utilizzare alias di colonna come "COUNT(h2.host) AS count" nell'esempio sopra.
Nel caso in cui un nome di colonna non possa essere convertito in un nome di macro valido, la regola di discovery diventa non supportata, con il messaggio di errore che riporta il numero della colonna problematica. Se si desidera ulteriore aiuto, i nomi delle colonne ottenuti sono disponibili con DebugLevel=4 nel file di log del server Zabbix:

```
$ grep db.odbc.discovery /tmp/zabbix_server.log
 ...
 23876:20150114:153410.856 In db_odbc_discovery() query:'SELECT h1.host, COUNT(h2.host) FROM hosts h1 L
 23876:20150114:153410.860 db_odbc_discovery() column[1]:'host'
 23876:20150114:153410.860 db_odbc_discovery() column[2]:'COUNT(h2.host)'
 23876:20150114:153410.860 End of db_odbc_discovery():NOTSUPPORTED
 23876:20150114:153410.860 Item [Zabbix server:db.odbc.discovery[proxies,{$DSN}]] error: Cannot convert
```

Ora che abbiamo compreso come una query SQL venga trasformata in un oggetto JSON, possiamo utilizzare la macro {#HOST} nei prototipi di item:



Una volta eseguita la discovery, verrà creato un item per ciascun proxy:



Utilizzo di db.odbc.get

Utilizzando `db.odbc.get[,{$DSN}]` e il seguente esempio SQL:

```
mysql> SELECT h1.host, COUNT(h2.host) AS count FROM hosts h1 LEFT JOIN hosts h2 ON h1.hostid = h2.proxyid
+---------+-------+
| host    | count |
+---------+-------+
| Japan 1 |     5 |
| Japan 2 |    12 |
| Latvia  |     3 |
+---------+-------+
3 rows in set (0.01 sec)
```

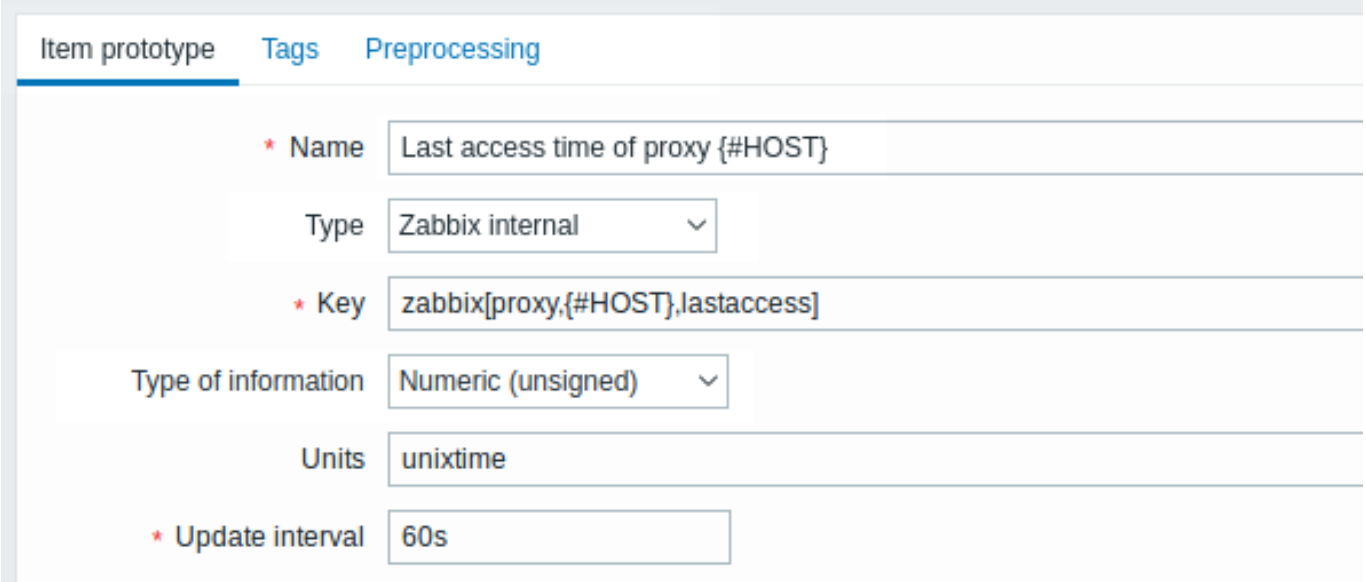
verrà restituito questo JSON:

```
[
    {
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```


Come puoi vedere, non sono presenti macro di low-level discovery. Tuttavia, è possibile creare macro personalizzate di low-level discovery nella scheda LLD macros di una regola di discovery utilizzando JSONPath, ad esempio:

```
{#HOST} → $.host
```

Ora questa macro {#HOST} può essere utilizzata nei prototipi di item:



13 Scoperta utilizzando i dati di Prometheus

Panoramica

I dati forniti nel formato di riga Prometheus possono essere utilizzati per il rilevamento di basso livello.

Per i dettagli su come l'interrogazione dei dati Prometheus è implementata in Zabbix, vedere Prometheus checks.

Configurazione

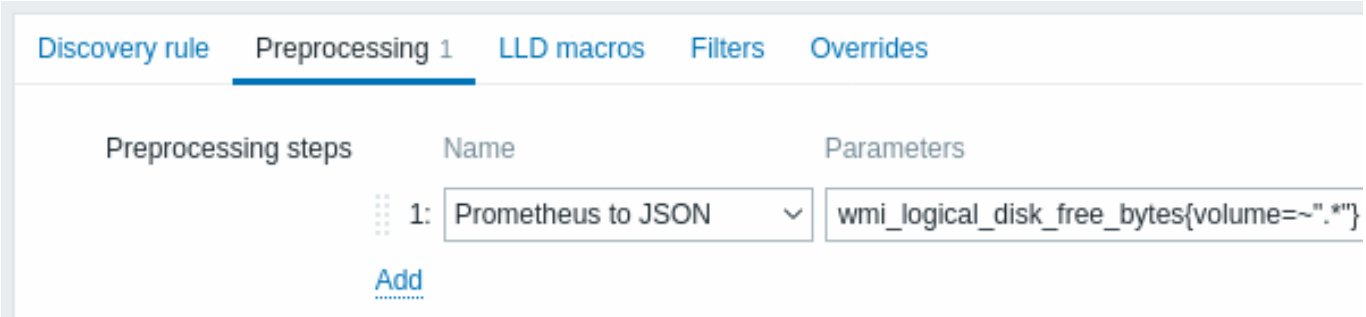
La regola di low-level discovery deve essere creata come item dipendente dell'item master HTTP che raccoglie i dati di Prometheus.

Da Prometheus a JSON

Nella regola di discovery, vai alla scheda Preprocessing e seleziona l'opzione di preprocessing *Prometheus to JSON*. I dati in formato JSON sono necessari per la discovery e l'opzione di preprocessing *Prometheus to JSON* restituirà esattamente questo, con i seguenti attributi:

- nome della metrica
- valore della metrica
- help (se presente)
- tipo (se presente)
- etichette (se presenti)
- riga grezza

Ad esempio, interrogando `wmi_logical_disk_free_bytes`:



da queste righe Prometheus:

```
# HELP wmi_logical_disk_free_bytes Free space in bytes (LogicalDisk.PercentFreeSpace)
# TYPE wmi_logical_disk_free_bytes gauge
wmi_logical_disk_free_bytes{volume="C:"} 3.5180249088e+11
wmi_logical_disk_free_bytes{volume="D:"} 2.627731456e+09
wmi_logical_disk_free_bytes{volume="HarddiskVolume4"} 4.59276288e+08
```

restituirà:

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

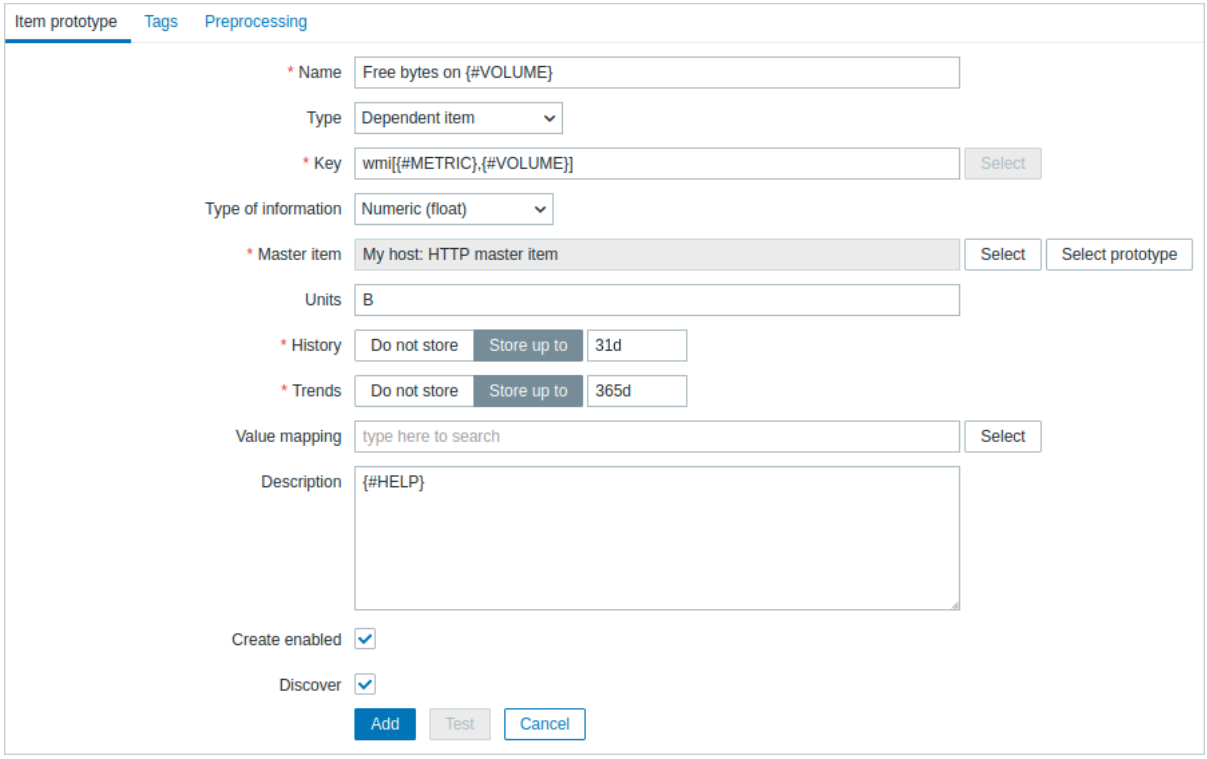
Mappatura delle macro LLD

Successivamente è necessario andare alla scheda delle macro LLD ed effettuare le seguenti mappature:

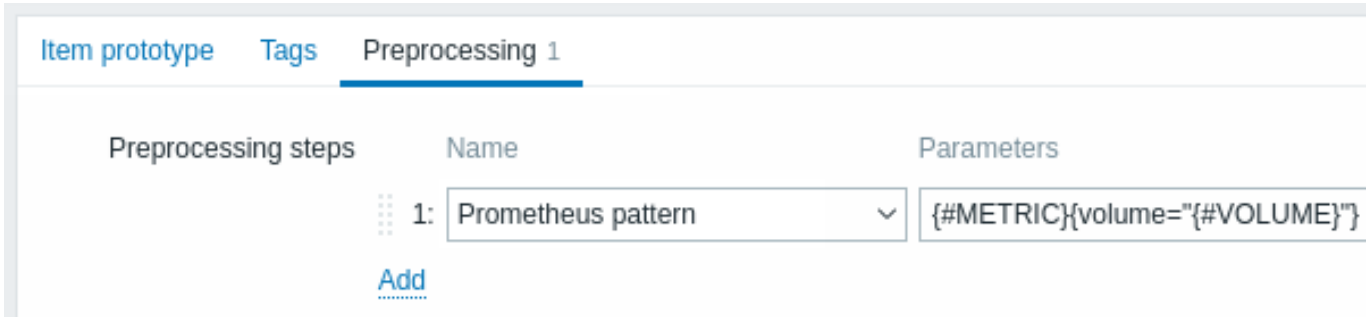
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

Prototipo di item

Potresti voler creare un prototipo di item come questo:



con opzioni di preprocessing:



14 Individuazione dei dispositivi a blocchi

In modo simile a come vengono individuati i file system, è possibile individuare anche i dispositivi a blocchi e il loro tipo.

Chiave item

La chiave item da usare nella regola di discovery è

`vfs.dev.discovery`

Questo item è supportato solo sulle piattaforme Linux.

Puoi creare regole di discovery usando questo item di discovery e:

- filtro: **{#DEVNAME} corrisponde a** `sd[\D]$` - per individuare dispositivi denominati "sd0", "sd1", "sd2", ...
- filtro: **{#DEVTYPE} corrisponde a** `disk` **AND {#DEVNAME} non corrisponde a** `^loop.*` - per individuare dispositivi di tipo disco il cui nome non inizia con "loop"

Macro supportate

Questa chiave di discovery restituisce due macro - {#DEVNAME} e {#DEVTYPE} che identificano rispettivamente il nome e il tipo del dispositivo a blocchi, ad esempio:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
```

```
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

Il discovery dei dispositivi a blocchi consente di usare gli item `vfs.dev.read[]` e `vfs.dev.write[]` per creare prototipi di item utilizzando la macro {#DEVNAME}, per esempio:

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} è destinata al filtraggio dei dispositivi.

15 Scoperta delle interfacce host in Zabbix

Panoramica

È possibile rilevare tutte le interfacce configurate nel frontend di Zabbix per un host.

Chiave item

L'item da utilizzare nella regola di discovery è l'item interno di Zabbix:

`zabbix[host,discovery,interfaces]`

Questo item restituisce un JSON con la descrizione delle interfacce, inclusi:

- Indirizzo IP/nome host DNS (a seconda dell'impostazione host "Connect to")
- Numero di porta
- Tipo di interfaccia (Zabbix agent, SNMP, JMX, IPMI)
- Se è o meno l'interfaccia predefinita
- Se la funzionalità di richieste bulk è abilitata - solo per le interfacce SNMP.

Per esempio:

```
[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG
```

Con più interfacce, i relativi record nel JSON sono ordinati per:

- Tipo di interfaccia,
- Predefinita - l'interfaccia predefinita viene posizionata prima delle interfacce non predefinite,
- ID dell'interfaccia (in ordine crescente).

Macro supportate

Le seguenti macro sono supportate per l'uso nel filtro della regola di discovery e nei prototipi di item, trigger e grafici:

Macro	Descrizione
{#IF.CONN}	Indirizzo IP dell'interfaccia o nome host DNS.
{#IF.IP}	Indirizzo IP dell'interfaccia.
{#IF.DNS}	Nome host DNS dell'interfaccia.
{#IF.PORT}	Numero di porta dell'interfaccia.
{#IF.TYPE}	Tipo di interfaccia ("AGENT", "SNMP", "JMX" o "IPMI").
{#IF.DEFAULT}	Stato predefinito dell'interfaccia: 0 - interfaccia non predefinita 1 - interfaccia predefinita
{#IF.SNMP.BULK}	Stato dell'elaborazione bulk SNMP per l'interfaccia: 0 - disabilitato 1 - abilitato Questa macro viene restituita solo se il tipo di interfaccia è "SNMP".

8 Regole LLD personalizzate

Panoramica

È anche possibile creare una regola LLD completamente personalizzata, in grado di rilevare qualsiasi tipo di entità, ad esempio 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 stiate eseguendo un vecchio agent Zabbix 1.8 (che non supporta la chiave vfs.fs.discovery), ma che abbiate comunque bisogno di individuare i file system. Ecco un semplice script Perl per Linux che rileva i file system montati e produce un output in JSON, che include sia il nome che il tipo del 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 suo 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 utilizzati 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 suo 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, potete 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; siete liberi di usare qualsiasi nome preferiate. Nel caso in cui venga usato JSONPath, allora la riga LLD sarà un elemento dell'array che può essere un oggetto, ma può anche essere un altro array o un valore.

Si noti che, se si utilizza un parametro utente, il valore restituito è limitato a 16MB. Per maggiori dettagli, vedere limiti dei dati per i valori restituiti LLD.

14 Monitoraggio distribuito

Panoramica Zabbix offre un modo efficace e affidabile per monitorare un'infrastruttura IT distribuita utilizzando i proxy di Zabbix.

I proxy possono essere utilizzati per raccogliere dati localmente per conto di un server Zabbix centralizzato e quindi inviare i dati al server.

Funzionalità del proxy

Quando si sceglie se utilizzare o meno un proxy, è necessario tenere in considerazione diversi aspetti.

	Proxy
Leggero	**Sì**
GUI	No

	Proxy
Funziona in modo indipendente	**Sì**
Manutenzione semplice	**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. Gli altri database supportati richiedono una configurazione manuale.

Attention:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; per i dettagli, vedere Calcolo delle code durante la manutenzione.

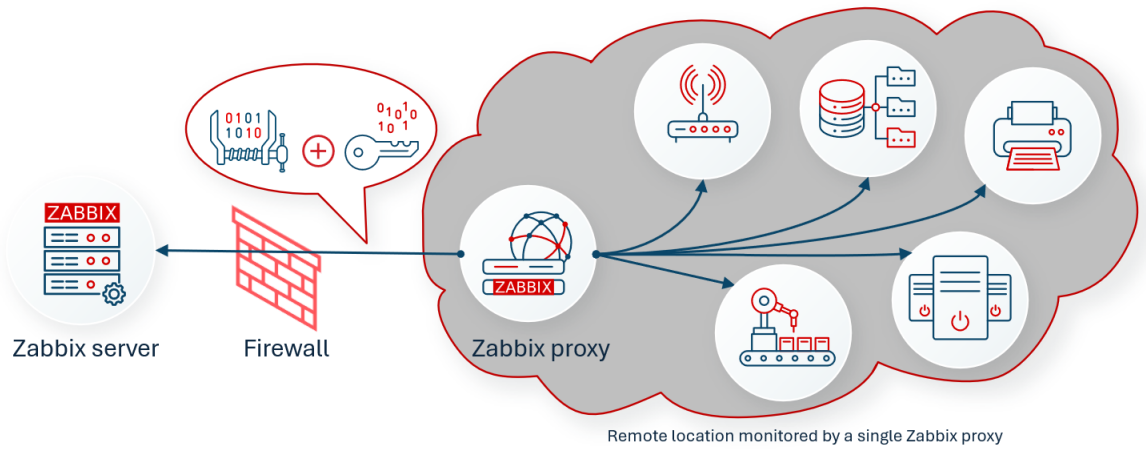
1 Proxy

Panoramica Un proxy Zabbix può raccogliere dati sulle prestazioni e sulla disponibilità per conto del server Zabbix. In questo modo, un proxy può assumersi parte del carico della raccolta dei dati e alleggerire il server Zabbix.

Inoltre, l'utilizzo di un proxy è il modo più semplice per implementare un monitoraggio centralizzato e distribuito, quando tutti gli agent e i proxy riportano a un server Zabbix e tutti i dati vengono raccolti centralmente.

Un proxy Zabbix può essere utilizzato per:

- Monitorare sedi remote
- Monitorare sedi con comunicazioni inaffidabili
- Alleggerire il server Zabbix quando si monitorano migliaia di dispositivi
- Semplificare la manutenzione del monitoraggio distribuito



Il proxy richiede una sola connessione TCP al server Zabbix. In questo modo è più facile superare un firewall, poiché è necessario configurare una sola regola del firewall.

Attention:
Zabbix proxy deve utilizzare un database separato. Puntarlo al database del server Zabbix comprometterà la configurazione.

Tutti i dati raccolti dal proxy vengono memorizzati localmente prima di essere trasmessi al server. In questo modo non si perde alcun dato a causa di eventuali 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 le modifiche più recenti della configurazione direttamente dal database del server Zabbix, 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 può iniziare a raccogliere dati e inviarli al server Zabbix, che ignora tali dati.

Zabbix proxy è un raccoglitore di dati. Non calcola i trigger, non elabora eventi e non invia avvisi. Per una panoramica delle funzionalità del proxy, consultare la tabella seguente:

Funzione		Supportata dal proxy
Item		
	Controlli Zabbix agent	**Sì**
	Controlli Zabbix agent (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**
	Controlli SSH	**Sì**
	Controlli Telnet	**Sì**
	Controlli esterni	**Sì**
	Item dipendenti	**Sì**
	Item script	**Sì**
	Item browser	**Sì**
Monitoraggio web integrato		**Sì**
Pre-elaborazione dei valori degli item		**Sì**
Rilevamento di rete		**Sì**
Autoregistrazione attiva dell'agent		**Sì**
Low-level discovery		**Sì** [2]
Comandi remoti		**Sì**
Calcolo dei trigger		*No*
Elaborazione degli eventi		*No*
Correlazione degli eventi		*No*
Invio di 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 LLD, Zabbix proxy raccoglie e pre-elabora solo i dati, quindi li invia al server Zabbix per un'ulteriore elaborazione.

Protezione dal sovraccarico

Se il server Zabbix è rimasto inattivo per un certo periodo di tempo e i proxy hanno raccolto molti dati, quando poi il server si avvia potrebbe sovraccaricarsi (l'utilizzo della cache della cronologia rimane al 95-100% per un certo tempo). Questo sovraccarico può causare un calo delle prestazioni, per cui i controlli vengono elaborati più lentamente del dovuto. È stata implementata una protezione da questo scenario per evitare i problemi che derivano dal sovraccarico della cache della cronologia.

Quando la cache della cronologia del server Zabbix è piena, l'accesso in scrittura alla cache della cronologia viene limitato, bloccando i processi del server che raccolgono i dati. Il caso più comune di sovraccarico della cache della cronologia si verifica dopo un periodo di inattività del server, quando i proxy stanno caricando i dati raccolti. Per evitarlo è stata aggiunta una limitazione dei proxy (attualmente non può essere disabilitata).

Quando l'utilizzo della cache della cronologia raggiunge l'80%, il server Zabbix entra in modalità di limitazione. In modalità di limitazione, il server accetta dati dai proxy solo quando l'utilizzo della cache della cronologia è inferiore al 60%, alternando i proxy accettati. Una volta che l'utilizzo della cache della cronologia 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 (oltre 10.000 record) se l'utilizzo della cache della cronologia supera il 60%. Questa decisione viene applicata nell'istante in cui il server valuta un caricamento da un proxy e pertanto potrebbe non riflettersi sempre immediatamente nei grafici di utilizzo della cache della cronologia (l'item interno `zabbix[wcache,history,pused]` e il suo intervallo di aggiornamento potrebbero non rilevare brevi picchi).

Questa modalità di limitazione continuerà finché l'utilizzo della cache non raggiungerà di nuovo l'80%, oppure non scenderà al 20%, oppure finché l'elenco di limitazione non sarà vuoto. Nel primo caso il server smetterà nuovamente di accettare dati dai proxy. Negli altri due casi il server riprenderà a funzionare normalmente, accettando dati da tutti i proxy.

Le informazioni sopra riportate possono essere illustrate nella seguente tabella:

Utilizzo della cache di scrittura della cronologia	Modalità del server Zabbix	Azione del server Zabbix
Raggiunge l'80%	Attesa	Smette di accettare dati dai proxy, ma mantiene un *elenco di limitazione* (elenco prioritario di proxy da contattare successivamente).
Raggiunge il 60%	Normale, ma pronto per essere limitato	Può rifiutare caricamenti molto grandi dai proxy (più di 10k record) quando decide se accettare i dati; continua ad accettare gli altri dati dai proxy.
Scende al 20%	Normale	Elimina l'elenco di limitazione e inizia ad accettare normalmente i dati dai proxy.

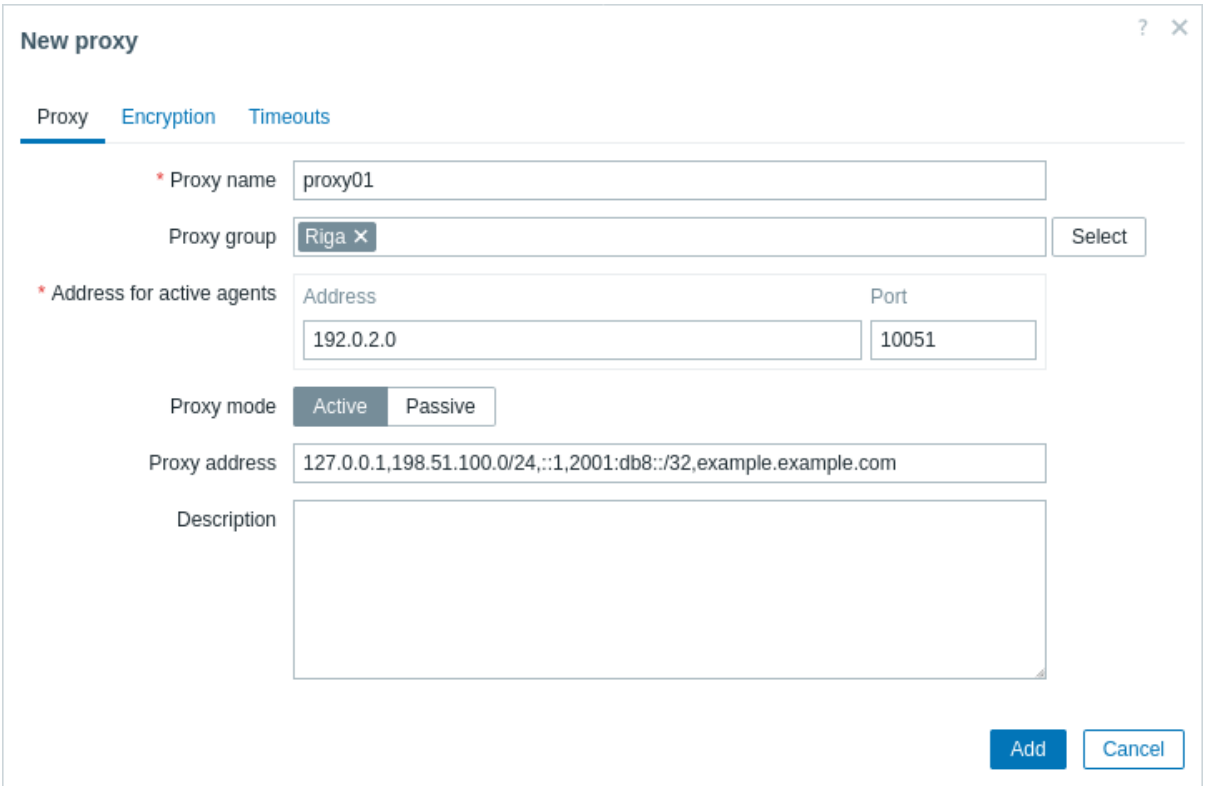
Puoi usare l'item interno zabbix[wcache,history,pused] per correlare questo comportamento del server Zabbix con una metrica.

Configurazione Una volta installato e configurato un proxy, è il momento di configurarlo nel frontend di Zabbix.

Aggiunta di proxy

Per configurare un proxy nel frontend di Zabbix:

- Andare in: *Administration* > *Proxies*
- Fare clic su *Create proxy*



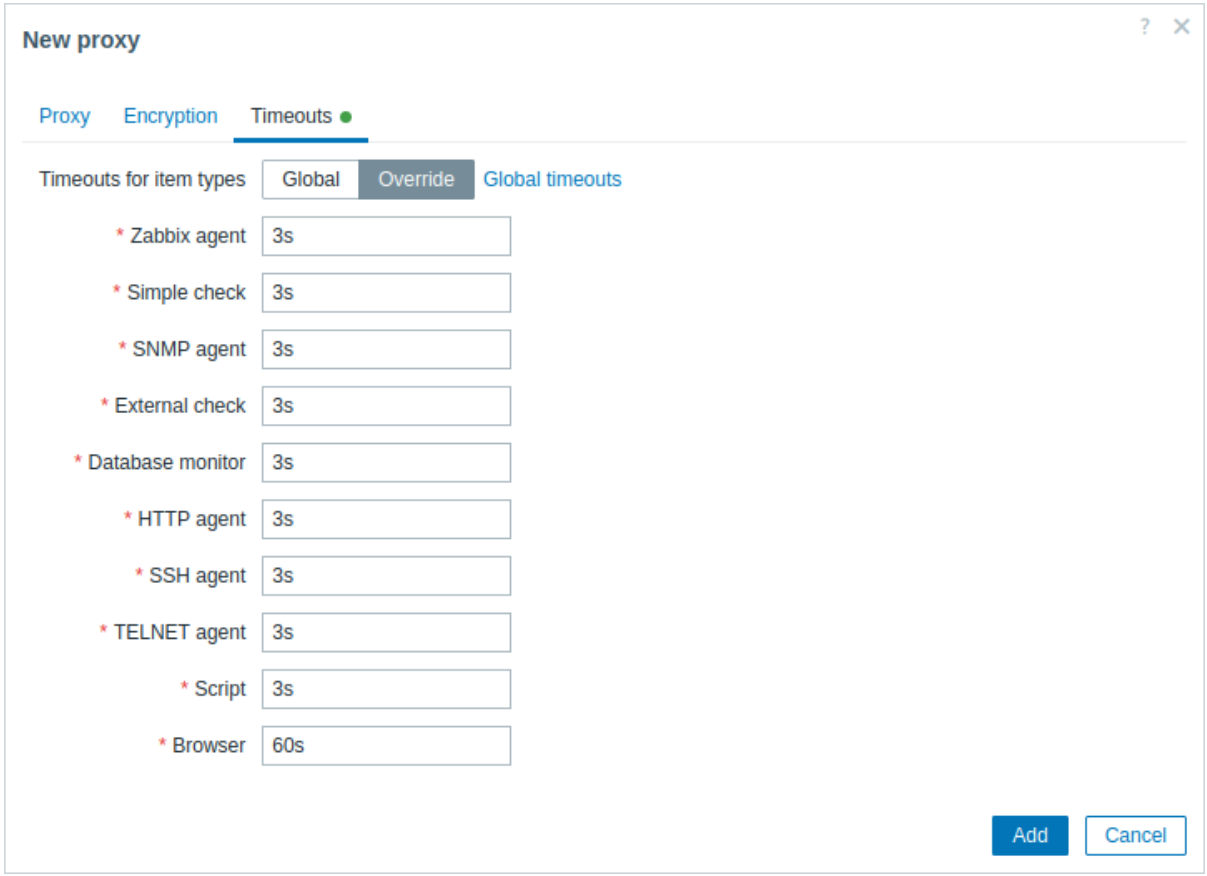
Parametro	Descrizione
Proxy name	Inserire il nome del proxy. Deve essere lo stesso nome specificato nel parametro *Hostname* del file di configurazione del proxy.
Proxy group	Selezionare un gruppo di proxy per il bilanciamento del carico/alta disponibilità del proxy.
Address for active agents	Inserire l'indirizzo a cui devono connettersi gli agent attivi monitorati o i sender. Supportato **solo** per agent Zabbix 7.0 o successivi. Questo indirizzo viene utilizzato per connettersi sia ai proxy attivi che a quelli passivi. Questo campo è disponibile solo se è selezionato un gruppo di proxy nel campo *Proxy group*.

Parametro	Descrizione
Address	Indirizzo IP/nome DNS a cui connettersi.
Port	Numero di porta TCP (10051 per impostazione predefinita) a cui connettersi. Le macro utente sono supportate.
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 *Si noti* che senza comunicazioni cifrate, 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 dati di configurazione se non avviene l'autenticazione o se gli indirizzi del proxy non sono limitati nel campo *Proxy address*.
Proxy address	Se specificato, le richieste del proxy attivo vengono accettate solo da questo elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o da 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 di porta TCP del proxy passivo (10051 per impostazione predefinita). Le macro utente sono supportate.
Description	Inserire la descrizione del proxy.

La scheda **Encryption** consente di richiedere connessioni cifrate con il proxy.

Parametro	Descrizione
Connections to proxy	Come il server si connette al proxy passivo: nessuna cifratura (predefinito), tramite PSK (chiave precondivisa) o certificato.
Connections from proxy	Selezionare quali tipi di connessione sono consentiti dal proxy attivo. È possibile selezionare contemporaneamente più tipi di connessione (utile per test e per passare a un altro tipo di connessione). Il valore predefinito 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 restringere ulteriormente le CA consentite. Questo campo è facoltativo ed è pensato per l'uso se l'installazione 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 identificativa della chiave precondivisa. Non inserire informazioni sensibili nell'identità PSK, poiché viene trasmessa in chiaro 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 **Timeouts** consente di sovrascrivere 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 suo tipo): **Global** - utilizza il timeout globale (visualizzato nel campo *Timeout* in grigio per ciascun tipo di item); **Override** - imposta un timeout personalizzato (nel campo *Timeout* per ciascun tipo di item). Intervallo consentito: 1 - 600s (predefinito: ereditato dai timeout globali). Sono supportati i suffissi temporali, ad esempio 30s, 1m, e le macro utente. Facendo clic sul collegamento *Global timeouts* è possibile configurare i timeout globali. Si noti che 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 sovrascrivano quelli globali, essi saranno sovrascritti dai timeout dei singoli item, se configurati.

Note:
Se la versione major del proxy non corrisponde alla versione major del server, accanto a *Timeouts for item types* verrà visualizzata l'icona , con il messaggio al passaggio del mouse "Timeouts disabled because the proxy and server versions do not match". In tali casi, il proxy utilizzerà il parametro `Timeout` del 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 del host

È possibile specificare che un singolo host debba essere monitorato da un proxy o da un gruppo di proxy nel modulo di configurazione del host, utilizzando il campo *Monitored by*.



L'aggiornamento massivo dei host è un altro modo per specificare che i host debbano essere monitorati da un proxy o da un gruppo di proxy.

1 Sincronizzazione della configurazione del monitoraggio

Panoramica

Questa pagina fornisce dettagli sull'aggiornamento della configurazione di monitoraggio per il proxy, cioè su come le modifiche apportate alla configurazione di monitoraggio sul server vengano 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 (quindi, se non è stata modificata alcuna entità, non verrà inviato nulla). Questo approccio consente di risparmiare risorse e di impostare un intervallo più piccolo (quasi istantaneo) per l'aggiornamento della configurazione del proxy.

Le modifiche alla configurazione del proxy vengono tracciate utilizzando numeri di revisione. Solo le entità con revisioni maggiori della revisione della configurazione del proxy vengono incluse nei dati di configurazione inviati al proxy.

Le entità per 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, le interfacce, l'inventario, gli item, il preprocessing degli item, i parametri degli item, gli scenari web di un host.
macro host	Tutte le macro definite su un host e tutti gli ID dei template collegati ad esso.
regola di discovery del proxy	Regole di discovery e controlli assegnati a un proxy.

Ciò significa:

- Se un item viene modificato su un **host**, verrà sincronizzata tutta la configurazione di quell'host.
- Se un'**espressione regolare** viene modificata, verranno sincronizzate tutte le espressioni regolari.

Fanno eccezione le 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 avverrà all'avvio/riavvio di un proxy, in caso di 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 si è interrotta durante la ricezione dei dati di configurazione.

Parametri di configurazione

Il parametro **ProxyConfigFrequency** determina con quale frequenza la configurazione del proxy viene sincronizzata con il server (10 secondi per impostazione predefinita).

Si noti che ProxyConfigFrequency è:

- un parametro del server per i proxy passivi
- un parametro del proxy per i proxy attivi

Nei proxy attivi, ProxyConfigFrequency è un nuovo parametro introdotto a partire da Zabbix 6.4 e deve essere utilizzato al posto di ConfigFrequency, ora deprecato.

> **Attention:**
> Se vengono utilizzati sia ProxyConfigFrequency sia ConfigFrequency, il proxy registrerà un errore e terminerà.

Calcolo delle code durante la manutenzione

Attention:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; per i dettagli, vedere Calcolo delle code durante la manutenzione.

2 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 dei proxy e l'alta disponibilità.

Il bilanciamento del carico dei proxy e l'alta disponibilità 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 più alto/più basso rispetto ad 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 in un gruppo che soddisfano le seguenti condizioni:

- I proxy eseguono Zabbix 7.0 o versioni successive.
- La versione del proxy corrisponde alla versione di Zabbix server. Se si utilizza Zabbix agent (passivo), 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 di salute del gruppo di proxy può essere monitorato con controlli interni da qualsiasi host assegnato a un gruppo di proxy. Tuttavia, per monitorare lo stato di salute di un singolo proxy in un gruppo, assegnare l'host a quel proxy; in caso contrario, i risultati potrebbero essere incoerenti.

Ridistribuzione 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 distribuzione dei loro host.

L'alta disponibilità dei proxy all'interno di un gruppo è garantita tramite il failover dei proxy: quando un proxy va offline, i suoi 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 2 (eccesso 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 utilizzando la seguente logica:

1. Calcolare il numero medio di host per proxy.
2. Per i proxy con un eccesso di host, spostare gli host in eccesso nel pool non assegnato dei proxy.
3. Per i proxy con un 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:

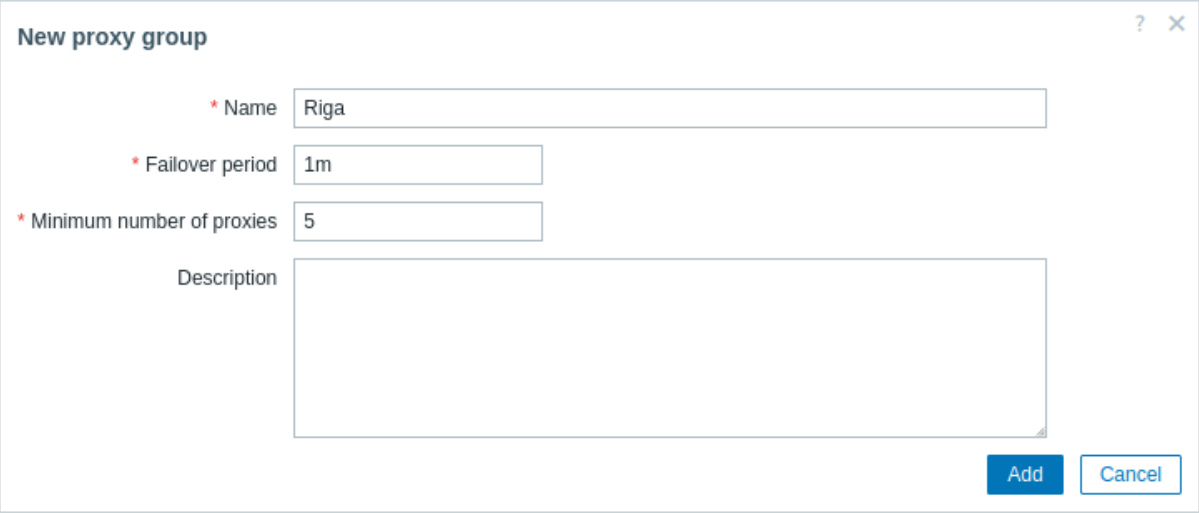
Host sul proxy	Media del gruppo	Riassegnazione degli host
100	50	Sì
60	50	No
40	50	No
25	50	Sì
15	5	Sì
10	5	No

Attention:
Avere meno di 10 host monitorati da un gruppo di proxy 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. Vai a *Amministrazione* > *Gruppi di proxy*
2. Fai clic su *Crea gruppo di proxy*



Parametro	Descrizione
Nome	Nome del gruppo di proxy.
Periodo di failover	Periodo in secondi durante il quale un proxy nel gruppo di proxy deve comunicare con Zabbix server per essere considerato online (predefinito: 1m; intervallo: 10s–15m). Se il proxy non comunica entro questo periodo, il suo stato viene cambiato in *Offline* e i suoi host vengono immediatamente ridistribuiti ad altri proxy. Il bilanciamento del carico del proxy inizia dopo 10 volte questo periodo. Supporta suffissi temporali (ad esempio, 30s, 1m) e macro utente.
Numero minimo di proxy	Numero minimo di proxy online richiesti per mantenere il gruppo di proxy online (predefinito: 1; intervallo: 1–1000). Supporta 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 a 10 farà sì che il gruppo vada offline se un qualsiasi proxy si guasta. Nota che i proxy online in un gruppo offline continuano a funzionare normalmente, ma il bilanciamento del carico/l'alta disponibilità non avranno luogo.
Descrizione	Descrizione del gruppo di proxy.
Proxy	Visualizza un elenco di massimo cinque proxy (come link o come testo semplice, a seconda dei permessi dell'utente sui proxy) durante la modifica di un gruppo con proxy.

Configurazione del bilanciamento del carico del proxy

Per utilizzare il bilanciamento del carico del proxy, è necessario configurare un gruppo di proxy nel frontend di Zabbix (vedi sopra) e assicurarsi che gli host siano monitorati da un gruppo di proxy, non da singoli proxy (è possibile usare l'aggiornamento di massa degli host per spostare gli host dai proxy al gruppo di proxy).

Se si utilizza Zabbix agent, configurarlo inoltre come segue:

- Per i controlli passivi, elencare tutti i proxy del gruppo di proxy nel parametro Server.
- Per i controlli attivi, si consiglia di elencare tutti i proxy del gruppo di proxy oppure Zabbix server nel parametro ServerActive. Si noti che solo Zabbix agent 7.0 (o versioni successive) funzionerà con i gruppi di proxy in modalità attiva.

Se il parametro `ServerActive` include un solo proxy di un gruppo di proxy (oppure Zabbix server), l'agent sarà comunque in grado di connettersi al proxy corretto. Quando il servizio agent si avvia e si connette al proxy specificato, l'agent riceverà e memorizzerà nella cache l'elenco completo degli IP dei proxy e il loro carico corrente all'interno del gruppo. Successivamente, i controlli attivi verranno reindirizzati al proxy online corretto per l'host, in base all'assegnazione corrente host-proxy all'interno del gruppo di proxy.

Warning:
Avere un solo proxy specificato nel parametro `ServerActive` di Zabbix agent può causare la perdita di dati di monitoraggio se l'agent viene avviato/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 host-proxy all'interno del gruppo di proxy. Tuttavia, se si stanno inviando valori di più host da un file di input, utilizzare l'opzione −g per evitare di inviare dati al proxy sbagliato.

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 quel proxy è bloccato da un firewall, la comunicazione si bloccherà in attesa di una risposta.
- In configurazioni stabili ad alta disponibilità 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 singoli proxy (è possibile usare 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 *Administration* > *Proxies*.

Note importanti

- Le trap SNMP non sono supportate dai proxy in un gruppo di proxy.
- I controlli che dipendono da una configurazione esterna (ad esempio, script per i controlli esterni o 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 permessi estesi sull'oggetto database/server.
- Gli host monitorati da un gruppo di proxy verranno distribuiti casualmente tra i proxy del gruppo. Ciò fa sì che ogni proxy memorizzi nella cache tutti i dati VMware, con conseguente carico aggiuntivo su vCenter.
- Gli host creati in base ai dati di autoregistrazione da un proxy in un gruppo di proxy vengono impostati per essere monitorati da quel gruppo di proxy. Tuttavia, gli host creati in base ai dati di individuazione della rete da un proxy in un gruppo di proxy vengono impostati per essere monitorati da quel proxy.

15 Crittografia

Panoramica Zabbix supporta comunicazioni crittografate tra i componenti Zabbix utilizzando il protocollo Transport Layer Security (TLS) v.1.2 e 1.3 (a seconda della libreria crittografica). Sono supportate la crittografia basata su certificati e quella basata su chiavi precondivise.

La crittografia può essere configurata per le connessioni:

- Tra Zabbix server, Zabbix proxy, Zabbix agent, Zabbix web service, le utility zabbix_sender e zabbix_get
- Al database Zabbix da Zabbix frontend e server/proxy
- Tra Zabbix frontend e Zabbix server

La crittografia è facoltativa e configurabile per i singoli componenti:

- Alcuni proxy e agent possono essere configurati per utilizzare la crittografia basata su certificati con il server, mentre altri possono usare la crittografia basata su chiavi precondivise, e altri ancora possono continuare con comunicazioni non crittografate (come in precedenza).
- Il server (proxy) può utilizzare configurazioni di crittografia diverse per host differenti.

I programmi daemon di Zabbix utilizzano un'unica porta di ascolto per le connessioni in ingresso crittografate e non crittografate. L'aggiunta della crittografia non richiede l'apertura di nuove porte nei 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 viene aperta 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, altrimenti alcuni item e azioni che eseguono script remoti sugli agent potrebbero funzionare con connessioni non crittografate, ma fallire per timeout con quelle crittografate.
- La crittografia non è supportata da network discovery. I controlli di Zabbix agent eseguiti dal network discovery non saranno crittografati e, se Zabbix agent è configurato per rifiutare le connessioni non crittografate, tali controlli non avranno successo.

Compilazione di Zabbix con supporto alla crittografia Per supportare la crittografia, Zabbix deve essere compilato e collegato con 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
- 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 specifiche di LibreSSL `tls_*()` non vengono utilizzate. I componenti di Zabbix compilati con LibreSSL non saranno in grado di usare PSK, potranno essere usati solo i certificati.

> **Note:**
> Puoi trovare maggiori informazioni sulla configurazione di SSL per il frontend di Zabbix facendo riferimento a queste best practices.

La libreria viene selezionata specificando la rispettiva opzione nello script "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (usato anche per LibreSSL)

Ad esempio, per configurare i sorgenti per server e agent con *OpenSSL* puoi usare qualcosa come:

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

Diversi componenti di Zabbix possono essere compilati con librerie crittografiche diverse (ad esempio un server con *OpenSSL*, un agent con *GnuTLS*).

> **Attention:**
> Se prevedi di utilizzare chiavi precondivise (PSK), prendi in considerazione l'uso delle librerie *GnuTLS* o *OpenSSL 1.1.0* (o successive) nei componenti di 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 PSK, ma le suite di cifratura PSK disponibili non forniscono Perfect Forward Secrecy.

Gestione della cifratura delle connessioni Le connessioni in Zabbix possono utilizzare:

- nessuna cifratura (predefinita)
- cifratura basata su certificati RSA
- cifratura basata su PSK

Esistono due parametri importanti utilizzati per specificare la cifratura tra i componenti di Zabbix:

- TLSConnect - specifica quale cifratura utilizzare per le connessioni in uscita (non cifrate, PSK o certificato)
- TLSAccept - specifica quali tipi di connessioni sono consentiti per le connessioni in ingresso (non cifrate, PSK o certificato). È possibile specificare uno o più valori.

`TLSConnect` viene utilizzato nei file di configurazione di Zabbix proxy (in modalità attiva, specifica solo le connessioni verso server) e di Zabbix agent (per i controlli attivi). In Zabbix frontend l'equivalente di TLSConnect è il campo *Connessioni verso host* nella scheda *Raccolta dati* → *Host* → *<some host>* → *Cifratura* e il campo *Connessioni verso proxy* nella scheda *Amministrazione* → *Proxy* → *<some proxy>* → *Cifratura*. Se il tipo di cifratura configurato per la connessione fallisce, non verranno provati altri tipi di cifratura.

`TLSAccept` viene utilizzato nei file di configurazione di Zabbix proxy (in modalità passiva, specifica solo le connessioni da server) e di Zabbix agent (per i controlli passivi). In Zabbix frontend l'equivalente di TLSAccept è il campo *Connessioni da host* nella scheda *Raccolta dati* → *Host* → *<some host>* → *Cifratura* e il campo *Connessioni da proxy* nella scheda *Amministrazione* → *Proxy* → *<some proxy>* → *Cifratura*.

Normalmente si configura un solo tipo di cifratura per le connessioni in ingresso. Tuttavia, potrebbe essere necessario cambiare il tipo di cifratura, ad esempio da non cifrata a basata su certificato, con un tempo di inattività minimo e la possibilità di rollback. Per ottenere questo risultato:

- Impostare `TLSAccept=unencrypted,cert` nel file di configurazione di agent e riavviare Zabbix agent
- Testare la connessione con zabbix_get verso agent utilizzando un certificato. Se funziona, è possibile riconfigurare la cifratura per quell'agent in Zabbix frontend nella scheda *Raccolta dati* → *Host* → *<some host>* → *Cifratura* impostando *Connessioni verso host* su "Certificato".
- Quando la cache di configurazione di server viene aggiornata (e la configurazione di proxy viene aggiornata se l'host è monitorato da proxy), allora le connessioni verso quell'agent saranno cifrate
- Se tutto funziona come previsto, è possibile impostare `TLSAccept=cert` nel file di configurazione di agent e riavviare Zabbix agent. Ora agent accetterà solo connessioni cifrate basate su certificato. Le connessioni non cifrate e quelle basate su PSK verranno rifiutate.

In modo simile funziona su server e proxy. Se in Zabbix frontend nella configurazione dell'host *Connessioni da host* è impostato su "Certificato", allora saranno accettate dall'agent solo connessioni cifrate basate su certificato (controlli attivi) e zabbix_sender (item trapper).

Molto probabilmente si configureranno le connessioni in ingresso e in uscita per utilizzare lo stesso tipo di cifratura oppure nessuna cifratura. Tuttavia, tecnicamente è possibile configurarle in modo asimmetrico, ad esempio cifratura basata su certificato per le connessioni in ingresso e basata su PSK per quelle in uscita.

La configurazione della cifratura per ciascun host viene visualizzata in Zabbix frontend, in *Raccolta dati* → *Host* nella colonna *Cifratura agent*. Ad esempio:

Esempio	Connessioni verso host	Connessioni consentite da host	Connessioni rifiutate da host
NONE	Non cifrate	Non cifrate	Cifrate, basate su certificato e basate su PSK
CERT NONE PSK CERT	Cifrate, basate su certificato	Cifrate, basate su certificato	Non cifrate e cifrate basate su PSK
PSK NONE PSK CERT	Cifrate, basate su PSK	Cifrate, basate su PSK	Non cifrate e cifrate basate su certificato
PSK NONE PSK CERT	Cifrate, basate su PSK	Non cifrate e cifrate basate su PSK	Cifrate basate su certificato
CERT NONE PSK CERT	Cifrate, basate su certificato	Non cifrate, oppure cifrate basate su PSK o su certificato	-

Attention:
Le connessioni non sono cifrate per impostazione predefinita. La cifratura deve essere configurata individualmente per ciascun host e proxy.

zabbix_get e zabbix_sender con crittografia Vedere le pagine man di zabbix_get e zabbix_sender per informazioni sul loro utilizzo con la crittografia.

Ciphersuite Per impostazione predefinita, le ciphersuite vengono configurate internamente durante l'avvio di Zabbix.

Sono supportate anche 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 comunque 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):

Libreria	Ciphersuite con certificato	Ciphersuite PSK
GnuTLS 3.1.18	TLS_ECDHE_RSA_AES_128_GCM_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA256 TLS_ECDHE_RSA_AES_128_CBC_SHA1 TLS_RSA_AES_128_GCM_SHA256 TLS_RSA_AES_128_CBC_SHA256 TLS_RSA_AES_128_CBC_SHA1	TLS_ECDHE_PSK_AES_128_CBC_SHA256 TLS_ECDHE_PSK_AES_128_CBC_SHA1 TLS_PSK_AES_128_GCM_SHA256 TLS_PSK_AES_128_CBC_SHA256 TLS_PSK_AES_128_CBC_SHA1
OpenSSL 1.0.2c	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-SHA256 AES128-SHA	PSK-AES128-CBC-SHA
OpenSSL 1.1.0	ECDHE-RSA-AES128-GCM-SHA256 ECDHE-RSA-AES128-SHA256 ECDHE-RSA-AES128-SHA AES128-GCM-SHA256 AES128-CCM8 AES128-CCM AES128-SHA256 AES128-SHA	ECDHE-PSK-AES128-CBC-SHA256 ECDHE-PSK-AES128-CBC-SHA PSK-AES128-GCM-SHA256 PSK-AES128-CCM8 PSK-AES128-CCM PSK-AES128-CBC-SHA256 PSK-AES128-CBC-SHA
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 configurate dall'utente I criteri integrati di selezione delle ciphersuite possono essere sovrascritti con ciphersuite configurate dall'utente.

> **Attention:**
> Le ciphersuite configurate dall'utente sono una funzionalità destinata a utenti avanzati che comprendono le ciphersuite TLS, la loro sicurezza e le conseguenze di eventuali errori, e che hanno familiarità con la risoluzione dei problemi TLS.

I criteri integrati di selezione delle ciphersuite possono essere sovrascritti utilizzando i seguenti parametri:

Ambito di sovrascrittura	Parametro	Valore	Descrizione
Selezione delle ciphersuite per i certificati	TLSCipherCert13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i relativi 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.
	TLSCipherCert	Stringhe di cifratura OpenSSL valide cipher strings per TLS 1.2 oppure stringhe di priorità GnuTLS valide priority strings. I relativi 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)

Ambito di sovrascrittura	Parametro	Valore	Descrizione
Selezione delle ciphersuite per PSK	TLSCipherPSK13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i relativi 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 oppure stringhe di priorità GnuTLS valide priority strings. I relativi 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 di ciphersuite per certificato e PSK	TLSCipherAll13	Stringhe di cifratura OpenSSL 1.1.1 valide cipher strings per il protocollo TLS 1.3 (i relativi 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 oppure stringhe di priorità GnuTLS valide priority strings. I relativi 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 sovrascrivere la selezione delle ciphersuite nelle utility zabbix_get e zabbix_sender, utilizzare i parametri della riga di comando:

- `--tls-cipher13`
- `--tls-cipher`

I nuovi parametri sono facoltativi. Se un parametro non è specificato, viene utilizzato 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, allora server, proxy o agent non si avvieranno e verrà registrato un errore.

È importante comprendere 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, utilizzare TLSCipherCert13 o TLSCipherCert
- Per le connessioni in uscita con PSK, utilizzare TLSCipherPSK13 o TLSCipherPSK
- Nel caso delle utility zabbix_get e zabbix_sender, è possibile usare i parametri della 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 complicata con le connessioni in ingresso, perché le regole sono specifiche per i componenti e per la configurazione.

Per Zabbix **agent**:

Configurazione della connessione dell'agent	Configurazione dei cifrari
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Per Zabbix **server** e **proxy**:

Configurazione della connessione	Configurazione dei cifrari
Connessioni in uscita che usano PSK	TLSCipherPSK, TLSCipherPSK13
Connessioni in ingresso che usano certificati	TLSCipherAll, TLSCipherAll13
Connessioni in ingresso che usano PSK se il server non ha certificato	TLSCipherPSK, TLSCipherPSK13
Connessioni in ingresso che usano PSK se il server ha un certificato	TLSCipherAll, TLSCipherAll13

Nelle due tabelle sopra è possibile osservare un certo schema:

- TLSCipherAll e TLSCipherAll13 possono essere specificati solo se viene usato un elenco combinato di suite di cifratura basate su certificato **e** su PSK. Ci sono due casi in cui questo avviene: server (proxy) con un certificato configurato (le suite di cifratura PSK sono sempre configurate su server e proxy se la libreria crittografica supporta PSK), agent configurato per accettare connessioni in ingresso sia basate su certificato sia su PSK
- negli altri casi sono sufficienti TLSCipherCert* e/o TLSCipherPSK*

Le tabelle seguenti mostrano i valori predefiniti integrati di `TLSCipher*`. Possono essere un buon punto di partenza per i propri 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 OpenSSL meno recenti (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

Vedere di seguito i seguenti esempi di suite di cifratura configurate dall'utente:

- Test delle stringhe dei cifrari e autorizzazione delle sole suite di cifratura PFS
- Passaggio da AES128 ad AES256

Test delle stringhe di cifratura e autorizzazione delle sole suite di cifratura PFS

Per vedere quali suite di cifratura sono state selezionate, è necessario impostare 'DebugLevel=4' nel file di configurazione oppure usare l'opzione `-vv` per zabbix_sender.

Potrebbe essere necessario fare alcune prove con i parametri `TLSCipher*` prima di ottenere le suite di cifratura desiderate. Riavviare Zabbix server, proxy o agent più volte solo per modificare i parametri `TLSCipher*` è scomodo. Opzioni più pratiche sono usare zabbix_sender o il comando `openssl`. Mostriamo entrambe.

1. Uso 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 agent validi e una PSK. Adattare i percorsi e i nomi dei file dei certificati e della PSK al proprio ambiente.

Se non si usano certificati, ma solo 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 si vedono 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 OpenSSL meno recenti (a partire dalla 1.0.1).

Con sistemi più recenti è possibile scegliere di rafforzare la sicurezza consentendo solo alcune suite di cifratura, ad esempio solo quelle con PFS (Perfect Forward Secrecy). Proviamo a consentire solo suite di cifratura con PFS usando i parametri `TLSCipher*`.

Attention:
Il risultato non sarà interoperabile con sistemi che usano OpenSSL 1.0.1 e 1.0.2, se viene usata la 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 di nuovo 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 - sono più brevi di prima e contengono solo suite di cifratura TLS 1.3 e 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 sopra. Per regolare TLSCipherAll e TLSCipherAll13 è necessario fare prove con l'agent, il proxy o il server.

Quindi, per consentire solo 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 ognuno di essi ha un certificato configurato e l'agent ha anche una PSK.

Se l'ambiente Zabbix usa solo cifratura basata su PSK e non certificati, allora ne basta solo uno:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Ora che è chiaro come funziona, è possibile testare la selezione delle suite di cifratura anche al di fuori di Zabbix, con il comando `openssl`. Proviamo tutti e tre i valori del parametro `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
```

Si potrebbe preferire `openssl ciphers` con l'opzione `-V` per 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
```

In modo analogo, 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 utilizza AES128 come valore predefinito integrato per i dati. Supponiamo che si stiano utilizzando certificati e che si voglia passare ad AES256, su OpenSSL 1.1.1.

Questo può essere ottenuto aggiungendo i rispettivi parametri 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:
Sebbene verranno utilizzate solo suite di cifratura relative ai certificati, vengono definiti anche i parametri `TLSCipherPSK*` per evitare i loro valori predefiniti, che includono cifrari meno sicuri per una più ampia 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
```

1 Utilizzo dei 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 sulla gestione di una CA interna, sulla generazione e firma delle richieste di certificato e sulla revoca dei certificati, fare riferimento a tutorial come OpenSSL PKI Tutorial v2.0.

Valutare attentamente e testare le estensioni dei certificati. Per maggiori dettagli, vedere Limitazioni sull'utilizzo delle estensioni dei certificati X.509 v3.

Parametri di configurazione dei certificati

I seguenti parametri di configurazione sono supportati per impostare i certificati sui componenti Zabbix.

Parametro	Obbligatorio	Descrizione
TLSCAFile	sì	Percorso completo di un file contenente i certificati della/e CA di livello superiore per la verifica del certificato del peer. Se si utilizza una catena di certificati con più membri, ordinare prima i certificati della/e CA di livello inferiore, seguiti dai certificati della/e 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 Liste di revoca dei certificati (CRL).
TLSCertFile	sì	Percorso completo di un file contenente il certificato. Se si utilizza una catena di certificati con più membri, ordinare prima il certificato del server, proxy o agent, seguito dai certificati della/e CA di livello inferiore, e infine dai certificati della/e CA di livello superiore.
TLSKeyFile	sì	Percorso completo di un file contenente la chiave privata. Assicurarsi che questo file sia leggibile solo dallo utente Zabbix impostando diritti di accesso appropriati.
TLSServerCertIssuer	no	Autorità emittente consentita del certificato del server.
TLSServerCertSubject	no	Soggetto consentito del certificato del server.

Esempi di configurazione

Dopo aver configurato i certificati necessari, configurare i componenti Zabbix per utilizzare la crittografia basata su certificati.

Di seguito sono riportati i passaggi dettagliati per la configurazione di:

- Zabbix server
- Zabbix proxy
- Zabbix agent

Zabbix server

1. Preparare il file del certificato CA.

Per verificare i certificati del peer, il server Zabbix deve avere accesso al file contenente i certificati root CA autofirmati di livello superiore. Ad esempio, se sono necessari certificati di due root CA 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
....
9wEzdN8uTrqoyU78gi12npLj08LegRKjb5hFTVmO
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
            ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root2 CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ....
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE
            ....
-----BEGIN CERTIFICATE-----
MIID3DCCAsSgAwIBAgIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQB
...
vdGNYoSfvu41GQAR5Vj5FnRJRzv5XQOZ3B6894GY1zY=
-----END CERTIFICATE-----
```

2. Inserire il certificato/la catena di certificati del server Zabbix in un file, ad esempio in `/home/zabbix/zabbix_server.crt`. Il primo certificato è il certificato del server Zabbix, seguito dal certificato CA intermedio:

```
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 sopra menzionati sia per i certificati client che per quelli server, per evitare di influire sul processo di verifica del certificato. Ad esempio, OpenSSL potrebbe non riuscire a stabilire una connessione cifrata se vengono utilizzate le estensioni *X509v3 Subject Alternative Name* o *Netscape Cert Type*. Per ulteriori informazioni, vedere Limitations on using X.509 v3 certificate extensions.

3. Inserire la chiave privata del server Zabbix 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 del server Zabbix:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix proxy

1. Preparare i file con i certificati CA di livello superiore, il certificato/la catena di certificati del proxy Zabbix 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 proxy Zabbix.

2. Modificare i parametri TLS aggiuntivi nel file di configurazione del proxy Zabbix:

- Per proxy attivo: `TLSConnect=cert`
- Per proxy passivo: `TLSAccept=cert`

Note:
Per migliorare la sicurezza del proxy, è possibile impostare anche i parametri `TLSServerCertIssuer` e `TLSServerCertSubject`. Per ulteriori informazioni, vedere Limitazione dell'autorità emittente e del soggetto del certificato consentiti.

I parametri TLS nel file di configurazione finale del proxy possono apparire come segue:

```
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
```

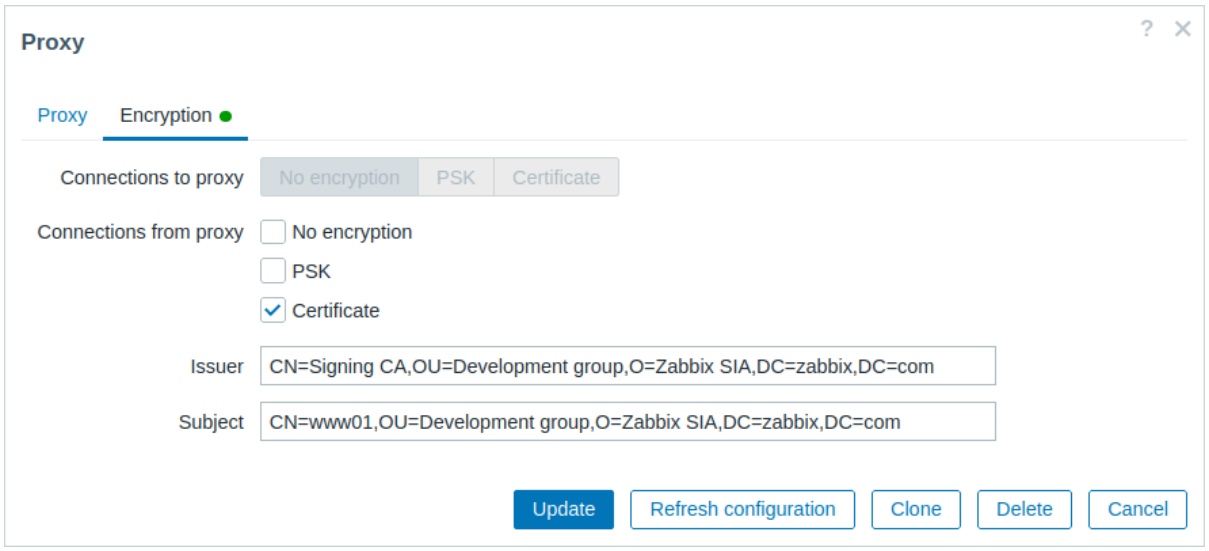
3. Configurare la crittografia per questo proxy nel frontend Zabbix:

- Andare in: *Administration* → *Proxies*.

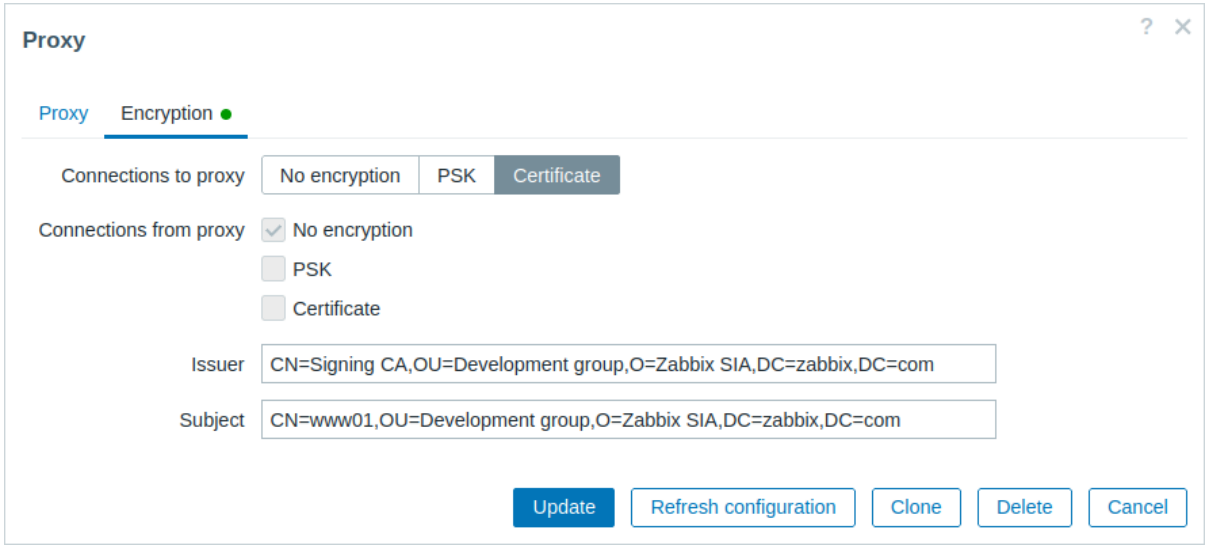
- Selezionare il proxy e fare clic sulla scheda *Encryption*.

Negli esempi seguenti, i campi *Issuer* e *Subject* sono compilati. Per ulteriori informazioni sul perché e su come utilizzare questi campi, vedere Limitazione dell'autorità emittente e del soggetto del certificato consentiti.

Per proxy attivo:



Per proxy passivo:



Zabbix agent

1. Preparare i file con i certificati CA di livello superiore, il certificato/la catena di certificati di 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 di Zabbix agent.

2. Modificare i parametri TLS aggiuntivi nel file di configurazione di Zabbix agent:

- Per agent attivo: `TLSConnect=cert`
- Per 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 soggetto del certificato consentiti.

I parametri TLS nel file di configurazione finale dell'agent possono apparire come segue. Si noti che l'esempio presuppone che l'host sia monitorato da un proxy, pertanto questo è specificato come Subject del certificato:

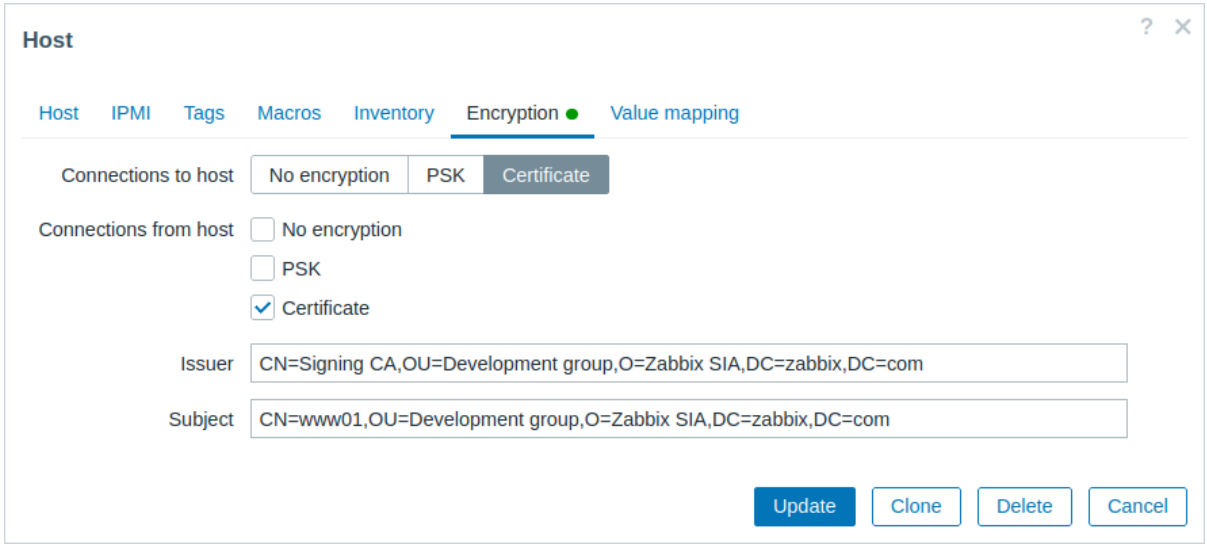
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
```

```
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 in Zabbix frontend per l'host monitorato da questo agent.

- Andare in: *Raccolta dati* → *Host*.
- Selezionare l'host e fare clic sulla scheda *Crittografia*.

Nell'esempio seguente, i campi *Issuer* e *Subject* sono compilati. Per ulteriori informazioni sul perché e su come utilizzare questi campi, vedere Limitazione dell'emittente e del soggetto del certificato consentiti.



Servizio web Zabbix

1. Preparare i file con i certificati CA di livello superiore, il certificato/la catena di certificati del servizio web Zabbix e la chiave privata come descritto nella sezione Zabbix server. Quindi, modificare i parametri `TLSCAFile`, `TLSCertFile` e `TLSKeyFile` nel file di configurazione del servizio web Zabbix di conseguenza.

2. Modificare un parametro TLS aggiuntivo nel file di configurazione del servizio web Zabbix: `TLSAccept=cert`

I parametri TLS nel file di configurazione finale del servizio web possono apparire come segue:

```
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 per connettersi al servizio web Zabbix configurato con TLS modificando il parametro `WebServiceURL` nel file di configurazione di Zabbix server:

```
WebServiceURL=https://example.com:443/report
```

Limitazione dell'autorità emittente e del soggetto del certificato consentiti

Quando due componenti Zabbix (ad esempio, server e agent) stabiliscono una connessione TLS, verificano reciprocamente i rispettivi 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 emittente e il soggetto del certificato non vengono verificati.

Tuttavia, questo comporta un rischio: chiunque disponga di un certificato valido può impersonare chiunque altro (ad esempio, un certificato host potrebbe essere usato per impersonare un server). Sebbene ciò possa essere accettabile in ambienti piccoli, 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 più sensibili dal punto di vista della sicurezza.

Se la CA di livello superiore emette certificati che non dovrebbero essere accettati da Zabbix oppure se si desidera ridurre il rischio di impersonificazione, è possibile limitare i certificati consentiti specificandone l'autorità emittente e il soggetto.

Ad esempio, nel file di configurazione di Zabbix proxy, 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 emittente o un soggetto diversi. 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 controllate 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.
- I caratteri UTF-8 sono supportati. Tuttavia, i caratteri jolly (*) o le espressioni regolari non sono supportati.
- Sono implementati i seguenti requisiti di RFC 4514 - 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 non riuscirà.
- 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 preceduti da escape:

```
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), passando ai 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` nell'ordine dal livello più alto a quello più basso. Nell'esempio seguente, i campi `Issuer` e `Subject` iniziano dal 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, eseguire 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'utilizzo delle estensioni dei certificati X.509 v3

Quando si implementano certificati X.509 v3 in Zabbix, alcune estensioni potrebbero non essere pienamente supportate oppure potrebbero causare comportamenti incoerenti.

Estensione Subject Alternative Name

Zabbix non supporta l'estensione *Subject Alternative Name*, utilizzata per specificare nomi DNS alternativi come indirizzi IP o indirizzi email. Zabbix può convalidare solo il valore nel campo *Subject* del certificato (vedere Restricting Allowed Certificate Issuer and Subject). Se i certificati includono il campo `subjectAltName`, il risultato della convalida del certificato può variare a seconda

degli specifici toolkit crittografici utilizzati 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 utilizzata, in genere è richiesto che siano specificati entrambi gli attributi *clientAuth* (per l'autenticazione client TLS WWW) e *serverAuth* (per l'autenticazione server TLS WWW). Ad esempio:

- Nei controlli passivi, in cui Zabbix agent opera come server TLS, l'attributo *serverAuth* deve essere incluso nel certificato dell'agent.
- Per i controlli attivi, in cui l'agent opera come client TLS, l'attributo *clientAuth* deve essere incluso nel certificato dell'agent.

Sebbene GnuTLS possa emettere un avviso per violazioni dell'utilizzo della chiave, in genere consente comunque alla comunicazione di procedere nonostante tali 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 dello specifico toolkit in uso.

Liste di revoca dei certificati (CRL)

Se un certificato è compromesso, l'Autorità di certificazione (CA) può revocarlo includendo il certificato in una Lista di revoca dei certificati (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 apparire così:

```
-----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, riavviare Zabbix.

Attention:
Se i componenti Zabbix sono compilati con OpenSSL e vengono utilizzate le CRL, assicurarsi che ogni CA di primo livello e intermedia nelle catene di certificati abbia una CRL corrispondente (anche se vuota) inclusa nel `TLSCRLFile`.

2 Utilizzo delle chiavi precondivise

Panoramica

Ogni chiave pre-condivisa (PSK) in Zabbix è in realtà una coppia composta da:

- stringa di identità PSK non segreta,
- valore della stringa PSK segreta.

La stringa di identità 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 riferito dai componenti di Zabbix. Non inserire informazioni sensibili nella stringa di identità PSK: viene trasmessa sulla rete senza crittografia.

Il valore PSK è una stringa di cifre esadecimali difficile da indovinare, ad esempio, "e560cb0d918d26d31b4f642181f5f570ad89a390931102e53

Limiti di dimensione

In Zabbix esistono limiti di dimensione per l'identità e il valore PSK; in alcuni casi una libreria crittografica può avere limiti inferiori:

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 successive	127 byte (può includere caratteri UTF-8)	-	2048 bit (PSK di 256 byte, inserita come 512 cifre esadecimali)

Attention:
Il frontend di 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 per tali componenti.
Il superamento dei limiti di lunghezza provoca errori di comunicazione tra i componenti Zabbix.

Prima che Zabbix server si connetta all'agent usando PSK, il server cerca nel database l'identità PSK e il valore PSK configurati per quell'agent (in realtà nella cache di configurazione). Quando riceve una connessione, l'agent utilizza l'identità PSK e il valore PSK dal suo file di configurazione. Se entrambe le parti hanno la stessa stringa di identità PSK e lo stesso valore PSK, la connessione può andare a buon fine.

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 valori diversi. In caso contrario, si possono verificare errori imprevedibili o interruzioni della comunicazione tra i componenti Zabbix che utilizzano PSK con questa stringa di identità PSK.

Generazione della PSK

Ad esempio, una PSK a 256 bit (32 byte) può essere generata utilizzando i seguenti comandi:

- con *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- con *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Generazione di una chiave casuale per l'utente 'psk_identity'
Chiave salvata in database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Si noti che "psktool" sopra genera un file di database con un'identità PSK e la PSK associata. Zabbix si aspetta solo una PSK nel file PSK, quindi la stringa di identità e i due punti (':') devono essere rimossi dal file.

Configurazione della PSK per la comunicazione server-agent (esempio)

Sull'host dell'agent, scrivere il valore PSK in un file, ad esempio `/home/zabbix/zabbix_agentd.psk`. Il file deve contenere la PSK nella prima stringa di testo, ad esempio:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

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 `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 la PSK. L'identità PSK sarà "PSK 001".

Riavviare l'agent. Ora è possibile testare la connessione utilizzando `zabbix_get`, ad esempio:

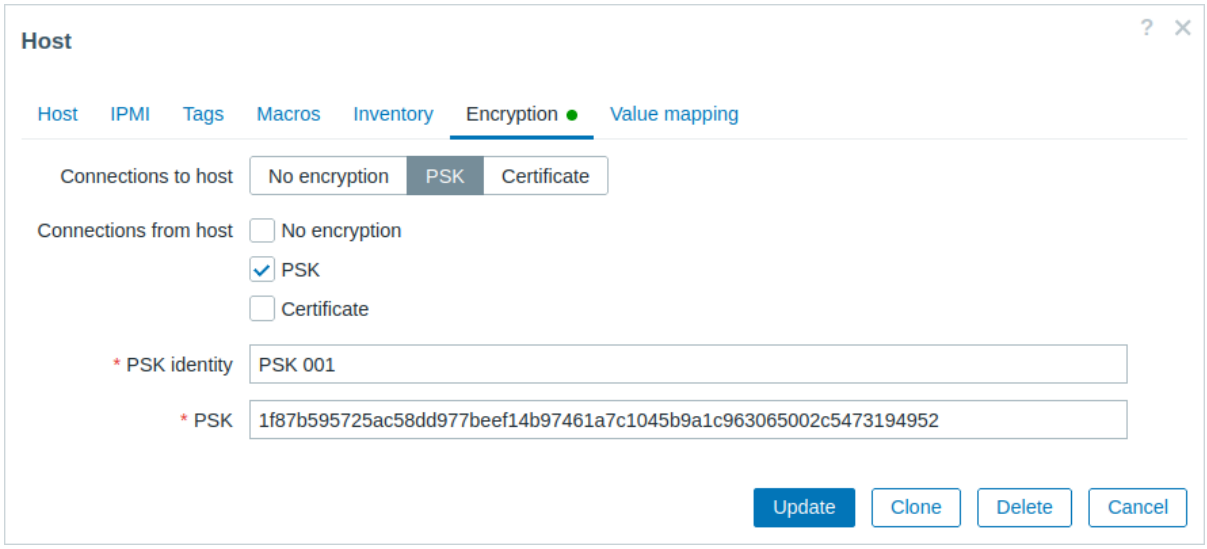
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Per ridurre al minimo il downtime, vedere come modificare il tipo di connessione in Gestione della crittografia della connessione).

Configurare la crittografia PSK per questo agent nel frontend di Zabbix:

- Andare in: *Data collection → Hosts*
- Selezionare l'host e fare clic sulla scheda **Encryption**

Esempio:



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

Quando la cache di configurazione viene sincronizzata con il database, le nuove connessioni utilizzeranno la PSK. Controllare i file di log del server e dell'agent per eventuali messaggi di errore.

Configurazione di PSK per la comunicazione server - proxy attivo (esempio)

Sul proxy, scrivere il valore PSK in un file, ad esempio `/home/zabbix/zabbix_proxy.psk`. Il file deve contenere la PSK nella prima stringa 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 `zabbix_proxy.conf`, ad esempio impostando:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Il proxy si connetterà al server utilizzando PSK. L'identità PSK sarà "PSK 002".

(Per ridurre al minimo il downtime, vedere come modificare il tipo di connessione in Connection encryption management).

Configurare la PSK per questo proxy nel frontend di Zabbix. Andare in *Administration→Proxies*, selezionare il proxy, quindi aprire la scheda "Encryption". In "Connections from proxy" selezionare PSK. Incollare nel campo "PSK identity" "PSK 002" 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 eventuali messaggi di errore.

Per un proxy passivo la procedura è molto simile. L'unica differenza è: impostare `TLSAccept=psk` nel file di configurazione del proxy e impostare "Connections to proxy" nel frontend di Zabbix su PSK.

3 Risoluzione dei problemi

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 di loro, possono tutti funzionare sia come server TLS sia come client TLS.
  Per esempio, Zabbix server che si connette all'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 dal 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 il proprio peer e può rifiutare la connessione.
  Per esempio, Zabbix server che si connette all'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 entrambi i lati, nel client TLS e nel server TLS.
  Il lato che rifiuta la connessione può registrare un motivo preciso del rifiuto. L'altro lato spesso riporta un errore piuttosto generico (ad esempio "Connection closed by peer", "connection was non-properly terminated").
- Talvolta una configurazione errata della crittografia produce messaggi di errore fuorvianti che non indicano in alcun modo la causa reale.
  Nelle sottosezioni seguenti cerchiamo di fornire una raccolta di messaggi e possibili cause (ben lontana dall'essere esaustiva) che potrebbe aiutare nella risoluzione dei problemi.
  Si noti che diversi toolkit crittografici (OpenSSL, GnuTLS) spesso producono messaggi di errore diversi nelle stesse situazioni problematiche.
  Talvolta i messaggi di errore dipendono perfino dalla particolare combinazione di toolkit crittografici su entrambi i lati.

1 Tipo di connessione o problemi di autorizzazione

Il server è configurato per connettersi con PSK all'agent, ma l'agent accetta solo connessioni non cifrate

Nel log del server o del 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 del server o del 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*):

```
impossibile accettare una connessione in ingresso: da 127.0.0.1: zbx_tls_accept(): gnutls_handshake() non
    -21 Impossibile negoziare una suite di cifratura supportata.
```

In qualsiasi log (con *OpenSSL* 1.0.2c):

```
impossibile accettare una connessione in ingresso: da 127.0.0.1: TLS handshake ha restituito il codice di
    file .\ssl\s3_srvr.c riga 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS ha scritto l'avviso fatale "handshake failure"
```

Tentativo di utilizzare il mittente Zabbix compilato con il supporto TLS per inviare dati al server/proxy Zabbix 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"
...End of zbx_tls_connect():FAIL error:'connessione chiusa dal peer'
...errore di invio del valore: TCP riuscito, impossibile stabilire TLS con [[localhost]:10051]: connession
```

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() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'errore di I/O di SSL_connect(): [0x00000000] Operazione completata
...errore di invio del valore: TCP riuscito, impossibile stabilire TLS con [[192.168.1.2]:10051]: errore d
```

Nel registro lato accettazione:

```
...impossibile accettare una connessione in entrata: da 127.0.0.1: il supporto per TLS non è stato compila
```

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 che avvia la connessione:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

Nel log del lato che accetta la connessione:

```
...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 avvia la connessione:

```
...TCP riuscito, impossibile stabilire TLS con [[192.168.1.2]:10050]: SSL_connect() ha impostato il codice
```

Nel log del lato che accetta la connessione:

```
...impossibile accettare una connessione in ingresso: da 192.168.1.2: l'handshake TLS ha impostato il codi
```

2 Problemi con i certificati

OpenSSL utilizzato con CRL e per alcune CA nella catena del certificato la relativa CRL non è inclusa in TLSCRLFile

Nel log del server TLS nel caso di un 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 un 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 scaduta o in scadenza 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 una CRL valida un certificato revocato viene segnalato come ”certificate revoked”. Quando la 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 è 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 stato firmato dalla 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.

3 Problemi PSK

PSK contiene un numero dispari di cifre esadecimali

Il proxy o l’agente non si avvia, messaggio nel registro del proxy o dell’agente:

```
PSK non valido nel file "/home/zabbix/zabbix_proxy.psk"
```

Stringa di identità PSK più lunga di 128 byte viene passata a GnuTLS

Nel log lato client TLS:

```
gnutls_handshake() failed: -110 The TLS connection was non-properly terminated.
```

Nel log lato server TLS:

```
gnutls_handshake() failed: -90 The SRP username supplied is illegal.
```

Valore PSK troppo lungo utilizzato con OpenSSL 1.1.1

Nel log del lato che avvia la 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 che accetta la connessione:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Questo problema si verifica in genere durante l’aggiornamento di OpenSSL dalla versione 1.0.x o 1.1.0 alla 1.1.1 e se il valore PSK è più lungo di 512 bit (PSK da 64 byte, inserito come 128 cifre esadecimali).

Vedere anche: Limiti della dimensione del valore

16 Interfaccia web

Panoramica Per un accesso semplice a Zabbix da ovunque e da qualsiasi piattaforma, viene fornita un'interfaccia web-based.

> **Note:**
> Se utilizzi più di un'istanza frontend assicurati che le impostazioni di localizzazione e le librerie (LDAP, SAML ecc.) sono installate e configurate identicamente per tutti i frontend.

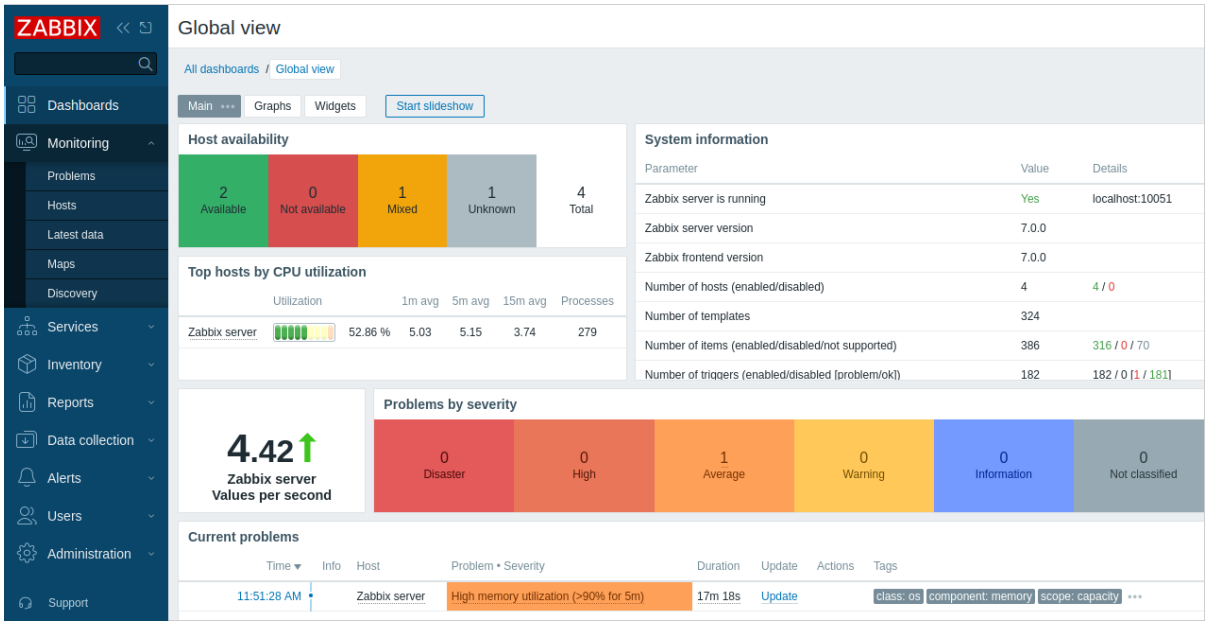
Guida del frontend Un link alla guida è presente nei form del frontend di Zabbix con collegamenti diretti alle parti corrispondenti della documentazione.

1 Menù

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.

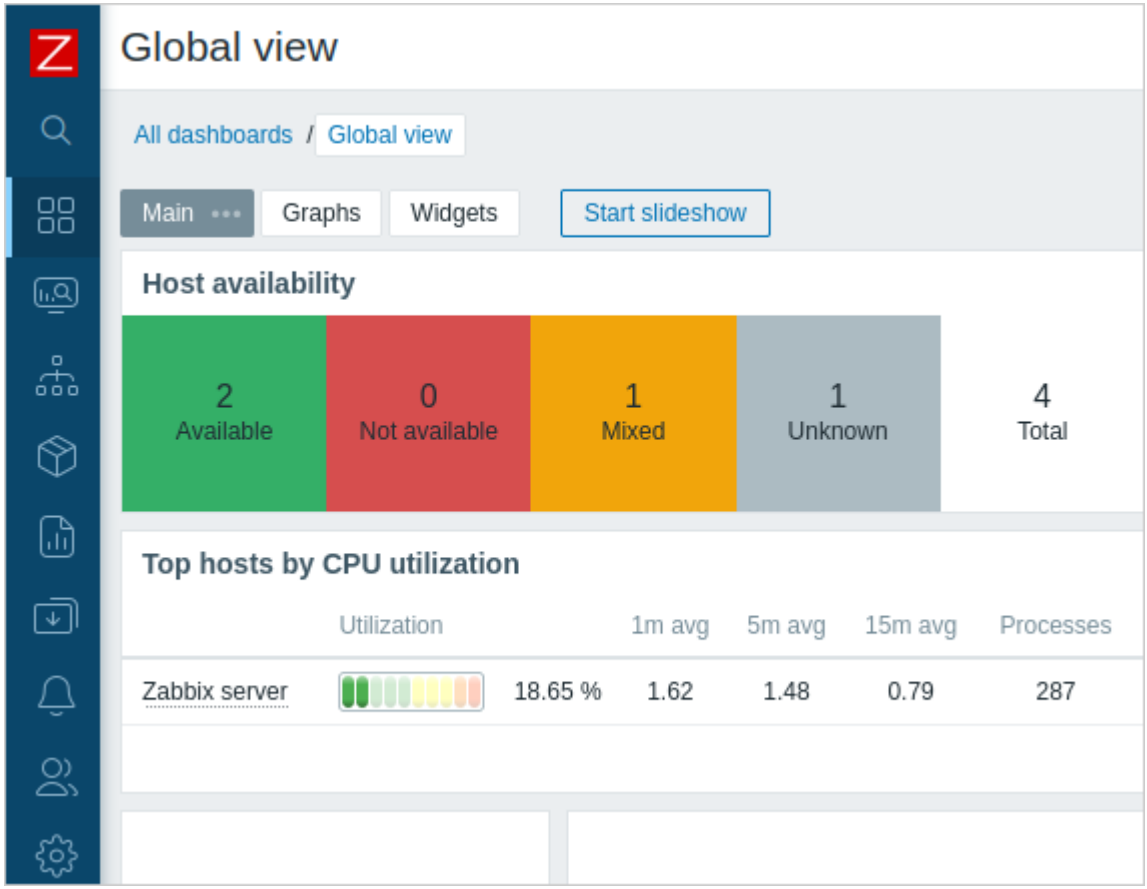


Utilizzo del menu

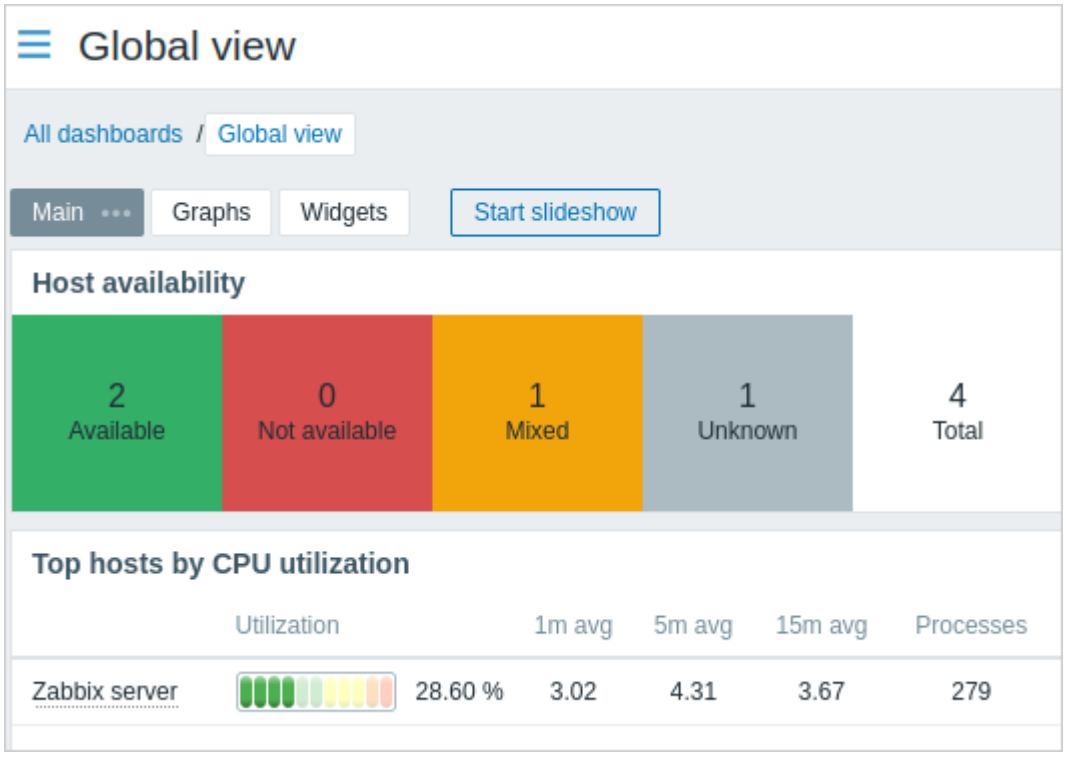
Una casella di ricerca globale si trova sotto il logo di Zabbix.

Il menu può essere compresso o nascosto completamente:

- Per comprimere, fare clic su accanto al logo di Zabbix. Nel menu compresso sono visibili solo le icone.



- Per nascondere, fare clic su  accanto al logo di 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 vi si posiziona sopra il cursore del mouse. Si noti che 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 posizionato fuori dal menu completo, il menu si comprimerà di nuovo dopo due secondi.

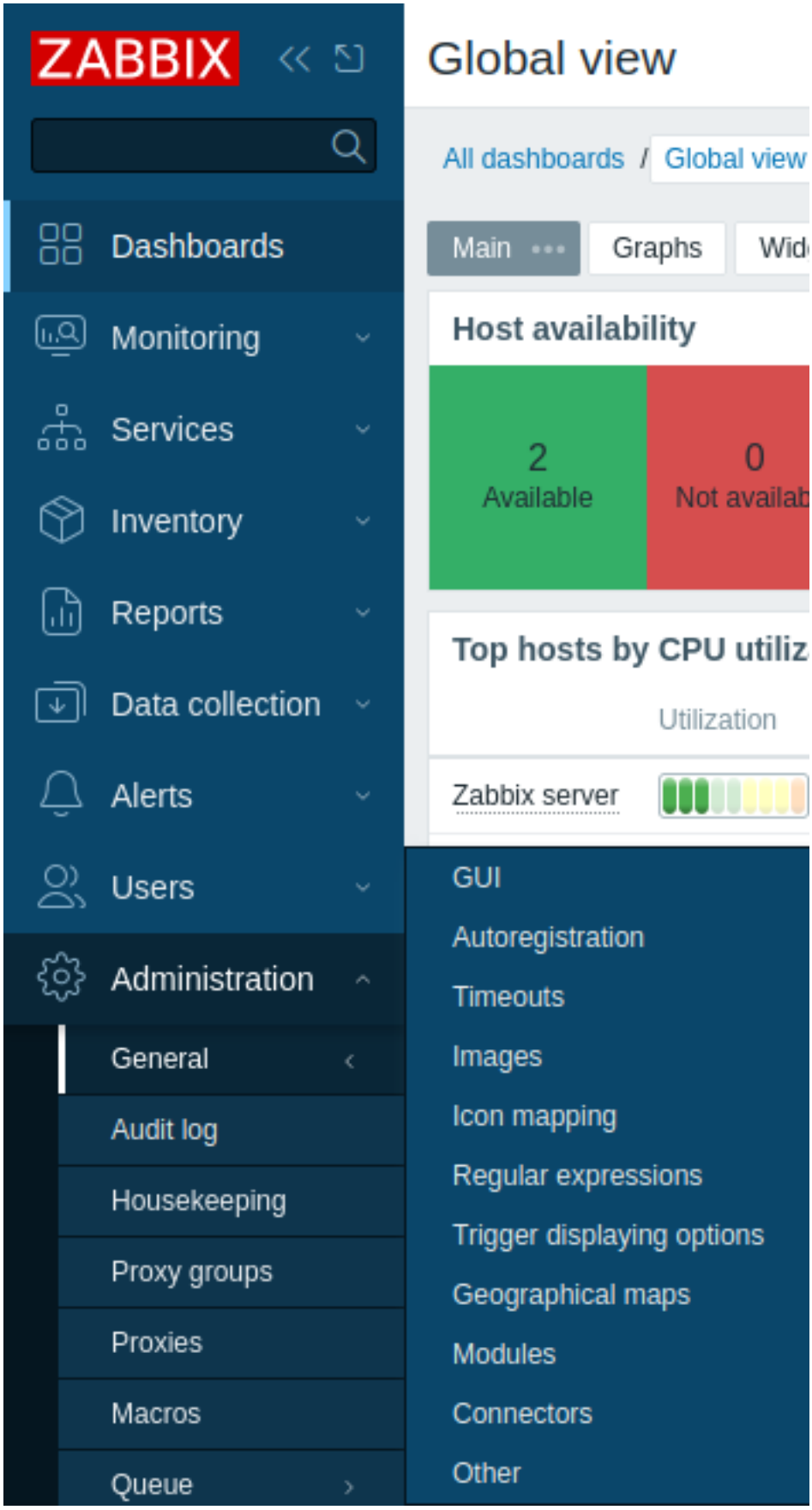
È anche possibile far riapparire completamente un menu compresso premendo il tasto Tab. Premendo ripetutamente il tasto Tab sarà possibile spostare il focus sull'elemento successivo del menu.

Menu nascosto

Anche quando il menu è completamente nascosto, il menu completo è a portata di un clic del mouse, facendo clic sull'icona hamburger  . Si noti che ricompare sopra il contenuto della pagina; per spostare il contenuto della pagina verso destra è necessario mostrare nuovamente il menu facendo clic sul pulsante per visualizzare la barra laterale.

Livelli del menu

Nel menu sono presenti fino a tre livelli.



Menu contestuali

Oltre al menu principale, Zabbix fornisce menu contestuali per host, item e event per un accesso rapido alle entità usate più di frequente, come gli ultimi valori, il grafico semplice, il modulo di configurazione, gli script correlati o i link esterni.

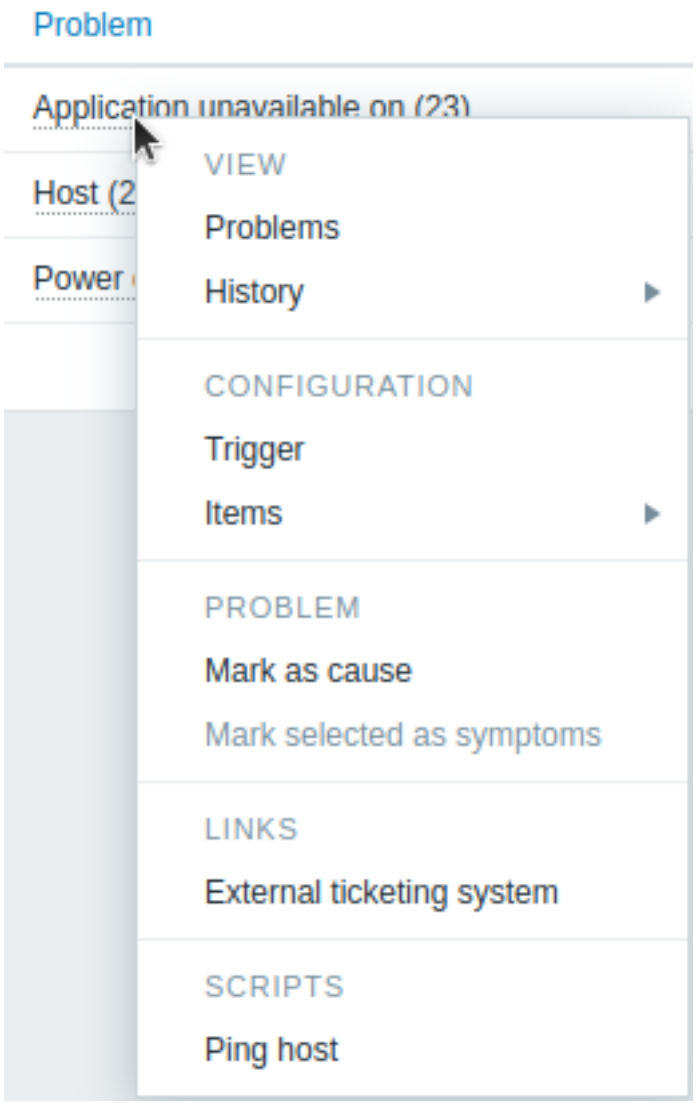
I menu contestuali sono accessibili facendo clic sul nome dell'host, dell'item o del problema/trigger nelle posizioni supportate.

1 Menu Eventi

Panoramica

Il menu evento contiene scorciatoie ad azioni o sezioni del frontend che sono frequentemente richieste per un evento.

Il menu evento può essere aperto facendo clic sul nome dell'evento.



Contenuto

Il menu contestuale dell'evento ha sei sezioni: *Vista*, *Azioni*, *Configurazione*, *Problema*, *Collegamenti* e *Script*. Per le entità che non sono configurate, i collegamenti sono disabilitati e visualizzati in grigio. Le sezioni *Script* e *Collegamenti* sono visualizzate se le relative entità sono configurate.

La sezione *Vista* contiene collegamenti a:

- **Problemi** - apre l'elenco dei problemi irrisolti del trigger sottostante;
- **Cronologia** - porta alla cronologia del grafico/item in *Ultimi dati* per gli item sottostanti. Se un trigger utilizza più item, saranno disponibili collegamenti per ciascuno di essi.

La sezione *Azioni* è disponibile solo nei widget *Panoramica 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:**
> Si noti che 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 *Collegamenti* contiene collegamenti per:

- accedere a un URL del trigger configurato;
- accedere a collegamenti personalizzati configurati in Script globali (con ambito 'Azione manuale sull'evento' e tipo 'URL');
- accedere a un ticket esterno configurato per il problema (vedere l'opzione *Include event menu entry* 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 usati 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 widget, come il widget *Problems*, il widget *Trigger overview*, ecc.
- Monitoraggio → Problems
- Monitoraggio → Problems → Dettagli evento
- Report → Top 100 triggers (gli script globali e l'accesso a ticket esterni non sono supportati in questa posizione)

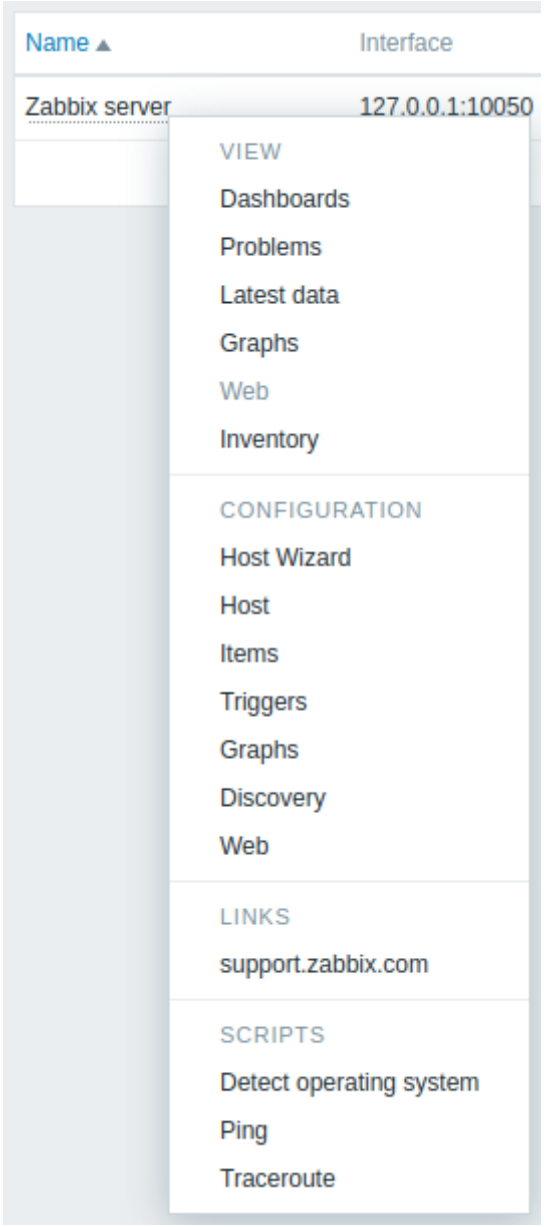
2 Menu Host

Panoramica

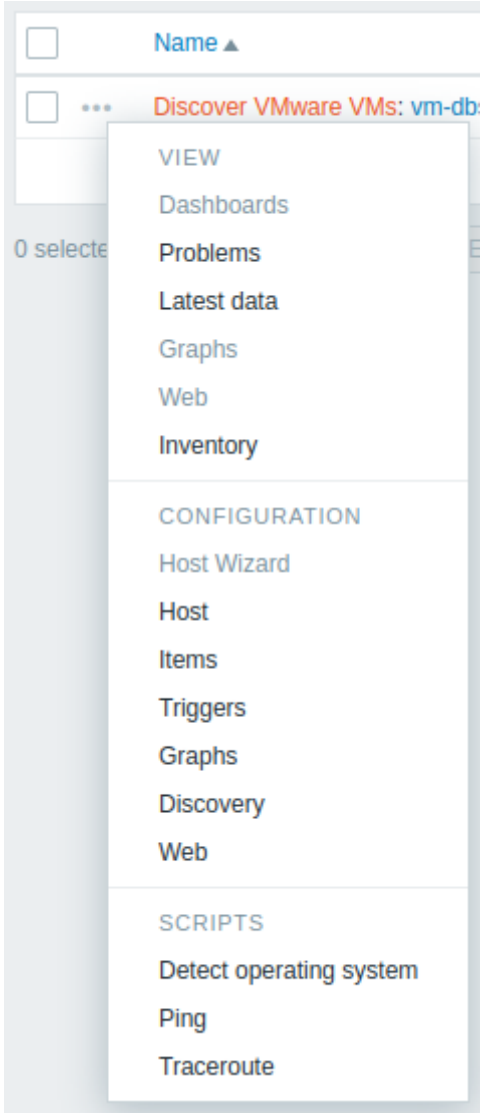
Il menu dell'host contiene scorciatoie ad azioni o sezioni del frontend frequentemente richieste per un host.

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 è composto da quattro sezioni: *Visualizza*, *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 *Visualizza* 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 automaticamente).
- **Host** - il modulo di configurazione dell'host corrente.
- **Item** - l'elenco degli item dell'host corrente.
- **Trigger** - l'elenco dei trigger dell'host corrente.
- **Grafici** - i grafici semplici dell'host corrente.
- **Discovery** - l'elenco delle regole di low-level discovery dell'host corrente.

- **Web** - l'elenco degli scenari web dell'host corrente.

> **Note:**
> Si noti che la sezione di configurazione è disponibile solo per gli utenti Admin e Super admin.

La sezione *Collegamenti* contiene collegamenti per:

- accedere a un URL del trigger configurato.
- accedere a collegamenti personalizzati configurati in Global scripts (con ambito *Manual host action* e tipo 'URL').

La sezione *Script* consente di eseguire global scripts configurati per l'host corrente. Per essere disponibili nel menu host, questi script devono avere l'ambito impostato su *Manual host action*.

Posizioni supportate

Il menu host è accessibile facendo clic sul nome di un host in varie sezioni del frontend, ad esempio:

- Dashboard widget, come Problemi, Top item, Panoramica trigger, ecc.
- Monitoraggio → Problemi
- Monitoraggio → Problemi → Dettagli evento
- Monitoraggio → Host
- Monitoraggio → Host → Monitoraggio web
- Monitoraggio → Ultimi dati
- Monitoraggio → Mappe
- Inventario → Host
- Report → Top 100 trigger

Il menu host è accessibile facendo clic sull'icona con i tre puntini in Raccolta dati → Host.

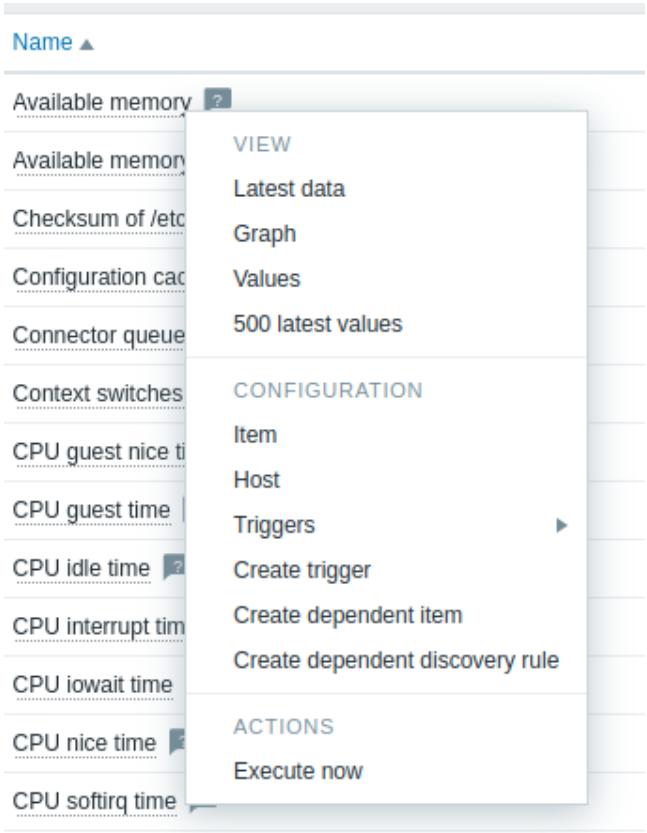
3 Menu item

Panoramica

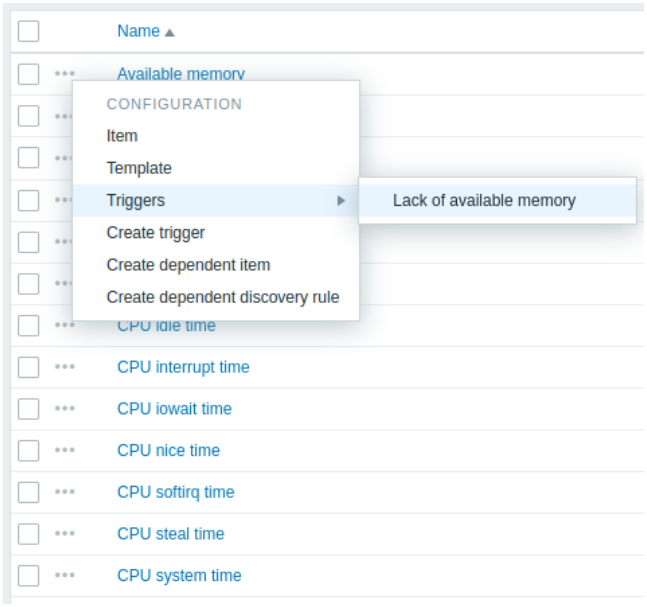
Il menu dell'item contiene scorciatoie ad azioni o sezioni del frontend frequentemente necessarie per un item.

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 ha tre sezioni: *Visualizza*, *Configurazione* e *Azioni*.

La sezione *Visualizza* 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 per l'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 dell'item corrente;
- **Template** - apre il modulo di configurazione del template del template a cui appartiene l'item (disponibile solo quando si accede al menu dell'item da *Raccolta dati* → *Templates* → *Items*);
- **Host** - apre il modulo di configurazione dell'host 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:

- Monitoraggio → Ultimi dati
- Raccolta dati → Host → Item
- Raccolta dati → Host → Regole di individuazione → Prototipi di item

Il menu dell'item è accessibile facendo clic sul valore di un item nel widget della dashboard Top items.

2 Sezioni di frontend

Struttura del menu Il menu del frontend di Zabbix ha la seguente struttura:

- Dashboard
- Monitoraggio
  - Problemi
  - Host
  - Ultimi dati
  - Mappe
  - Discovery
- Servizi
  - Servizi
  - SLA
  - Report SLA
- Inventario
  - Panoramica
  - Host
- Report
  - Informazioni di sistema
  - Report pianificati
  - Report di disponibilità
  - Top 100 trigger
  - Log di audit
  - Log delle azioni
  - Notifiche
- Raccolta dati
  - Gruppi di template
  - Gruppi di host
  - Template
  - Host
  - Manutenzione
  - Correlazione eventi
  - Discovery
- Avvisi
  - Azioni
    - Azioni trigger
    - Azioni servizio
    - Azioni discovery
    - Azioni di autoregistrazione
    - Azioni interne
  - Tipi di supporto
  - Script
- Utenti
  - Gruppi utenti
  - Ruoli utente
  - Utenti
  - Token API
  - Autenticazione
- Amministrazione
  - Generale
    - GUI
    - Autoregistrazione
    - Timeout
    - Immagini
    - Mappatura icone
    - Espressioni regolari
    - Opzioni di visualizzazione dei trigger
    - Mappe geografiche
    - Moduli
    - Connettori

    - Altro
- Log di audit
- Housekeeping
- Gruppi proxy
- Proxy
- Macro
- Coda
    - Panoramica della coda
    - Panoramica della coda per proxy
    - Dettagli della coda

1 Dashboard

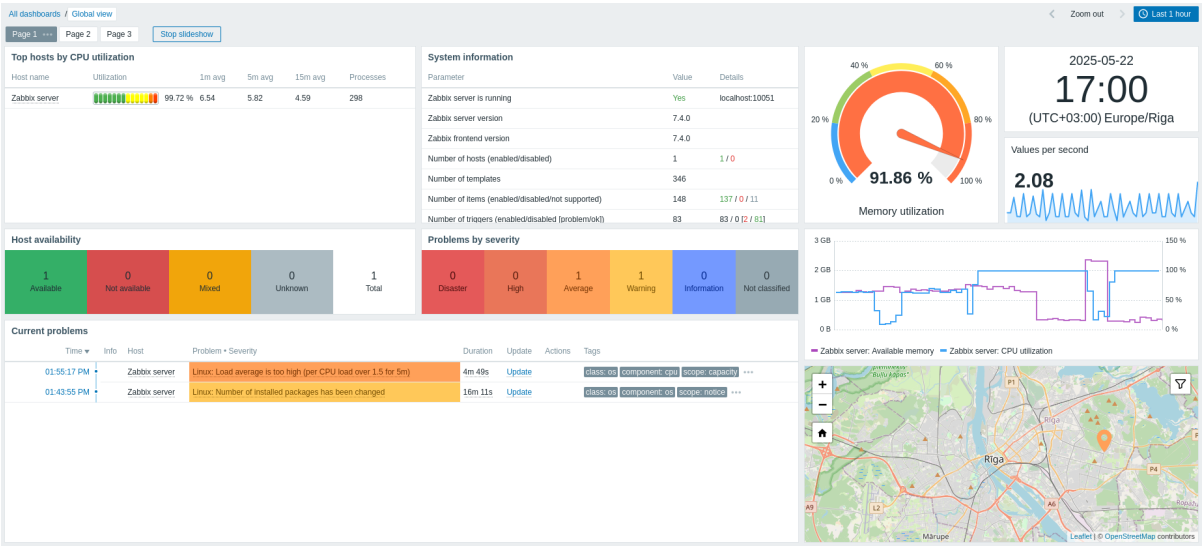
Panoramica

La sezione *Dashboard* è progettata per visualizzare riepiloghi di tutte le informazioni importanti in una **dashboard**.

Sebbene possa essere visualizzata una sola dashboard alla volta, è possibile configurarne diverse. Ogni dashboard può contenere una o più pagine che possono essere ruotate in una presentazione.

Una pagina della dashboard è composta da widget e ciascun widget è progettato per visualizzare informazioni di un determinato tipo e provenienza, che possono essere un riepilogo, una mappa, un grafico, l'orologio, ecc.

L'accesso agli host nei widget dipende dai permessi.



Le pagine e i widget vengono aggiunti alla dashboard e modificati nella modalità di modifica della dashboard. Le pagine possono essere visualizzate e ruotate nella modalità di visualizzazione.

L'intervallo di tempo visualizzato nei widget grafico è controllato dal selettore dell'intervallo di tempo situato sopra i widget. La sua etichetta mostra l'intervallo di tempo attualmente selezionato. Facendo clic sull'etichetta è possibile espandere e comprimere il selettore dell'intervallo di tempo.

Dimensione della dashboard

La larghezza minima di una dashboard è di 1200 pixel. La dashboard non si ridurrà al di sotto di questa larghezza; se la finestra del browser è più piccola, verrà invece visualizzata una barra di scorrimento orizzontale.

La larghezza massima di una dashboard corrisponde alla larghezza della finestra del browser. I widget della dashboard si estendono orizzontalmente per adattarsi alla finestra. Allo stesso tempo, un widget della dashboard non può essere esteso orizzontalmente oltre i limiti della finestra.

Orizzontalmente, la dashboard è composta da 72 colonne di larghezza sempre uguale, che si estendono/si riducono dinamicamente (ma non fino a meno di 1200 pixel complessivi).

Verticalmente, la dashboard può contenere un massimo di 64 righe; ogni riga ha un'altezza fissa di 70 pixel.

Un widget può quindi avere una larghezza massima di 72 colonne e un'altezza massima di 64 righe.

Visualizzazione delle dashboard

Per visualizzare tutte le dashboard configurate, fare clic su *Tutte le dashboard* appena sotto il titolo della sezione.



Le dashboard sono visualizzate con un tag di condivisione:

- *Mie* - indica una dashboard di proprietà dell'utente corrente
- *Condivise* - indica una dashboard pubblica o una dashboard privata condivisa con qualsiasi utente o gruppo di utenti

Si noti che i tag di condivisione sono mostrati solo per le dashboard di proprietà dell'utente corrente. Le dashboard di proprietà di altri utenti sono elencate senza tag di condivisione.

Il filtro situato in alto a destra sopra l'elenco consente di filtrare le dashboard per nome e per quelle create dall'utente corrente.

Per eliminare una o più dashboard, selezionare le caselle di controllo delle dashboard corrispondenti e fare clic su *Elimina* sotto l'elenco.

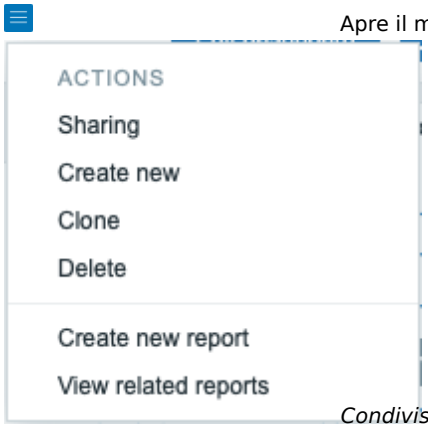
Visualizzazione di una dashboard

Per visualizzare una singola dashboard, fare clic sul suo nome nell'elenco delle dashboard.

Quando si **visualizza** una dashboard, sono disponibili le seguenti opzioni:

Edit dashboard

Passa alla modalità di **modifica** della dashboard.
La modalità di modifica si apre anche quando viene creata una nuova dashboard e quando si fa clic sul pulsante di modifica di un widget.



Apre il menu delle azioni (vedere le descrizioni delle azioni di seguito).

Condivisione - modifica le preferenze di condivisione della dashboard.
Crea nuovo - crea una nuova dashboard.
Clona - crea una nuova dashboard copiando le proprietà di quella esistente. Per prima cosa viene richiesto di inserire i parametri della dashboard. Quindi, la nuova dashboard si apre in modalità di modifica con tutti i widget della dashboard originale.
Elimina - elimina la dashboard.
Crea nuovo report - apre una finestra pop-up con il modulo di configurazione del report. Disabilitato se l'utente non dispone dell'autorizzazione per gestire i report pianificati.
Visualizza report correlati - apre una finestra pop-up con un elenco dei report esistenti basati sulla dashboard corrente. Disabilitato se non ci sono report correlati o se l'utente non dispone dell'autorizzazione per visualizzare i report pianificati.

	Visualizza solo il contenuto della pagina (modalità kiosk). È possibile accedere alla modalità kiosk anche con i seguenti parametri URL: `/zabbix.php?action=dashboard.view&kiosk=1`. Per tornare alla modalità normale: `/zabbix.php?action=dashboard.view&kiosk=0`.

Condivisione

Le dashboard possono essere rese pubbliche o private.

Le dashboard pubbliche sono visibili a tutti gli utenti. Le dashboard private sono visibili solo al loro proprietario e agli utenti/gruppi di utenti aggiunti nelle preferenze di condivisione.

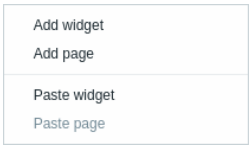
Per modificare lo stato di condivisione di una dashboard, fare clic sull'opzione *Condivisione* nel menu delle azioni durante la visualizzazione di una singola dashboard:

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.
Elenco delle condivisioni dei gruppi di utenti	Selezionare i gruppi di utenti a cui la dashboard è accessibile. È possibile consentire l'accesso in sola lettura o in lettura-scrittura.
Elenco delle condivisioni degli utenti	Selezionare gli utenti a cui la dashboard è accessibile. È possibile consentire l'accesso in sola lettura o in lettura-scrittura.

Vedere Autorizzazioni per le dashboard per scoprire in che modo le opzioni di condivisione limitano le azioni disponibili agli utenti.

Modifica di una dashboard

Quando si **modifica** una dashboard, sono disponibili le seguenti opzioni:

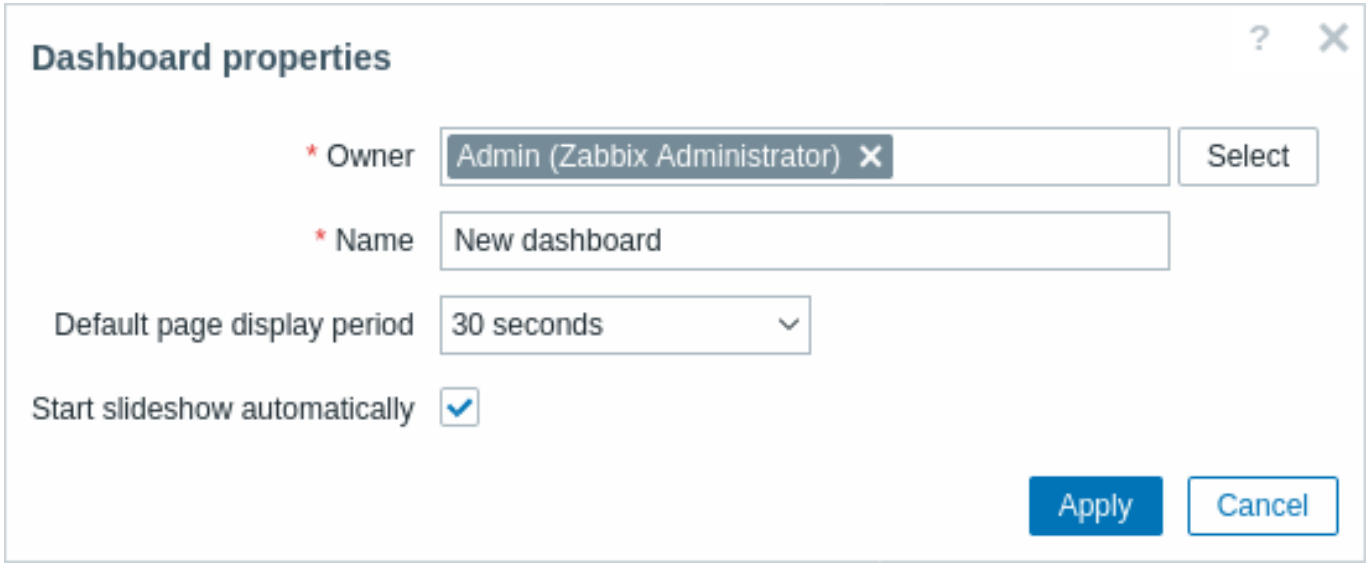
	Modificare i parametri generali della dashboard. Aggiungere un nuovo widget. Facendo clic sul pulsante con la freccia si aprirà il menu delle azioni (vedere le descrizioni delle azioni di seguito).
	*Aggiungi widget* - aggiungere un nuovo widget. *Aggiungi pagina* - aggiungere una nuova pagina. *Incolla widget* - incollare un widget copiato. Questa opzione è disattivata se non è stato copiato alcun widget. È possibile copiare una sola entità alla volta (widget o pagina). *Incolla pagina* - incollare una pagina copiata. Questa opzione è disattivata se non è stata copiata alcuna pagina.
	Salvare le modifiche alla dashboard.
	Annullare le modifiche alla dashboard.

Creazione di una dashboard

È possibile creare una nuova dashboard in due modi:

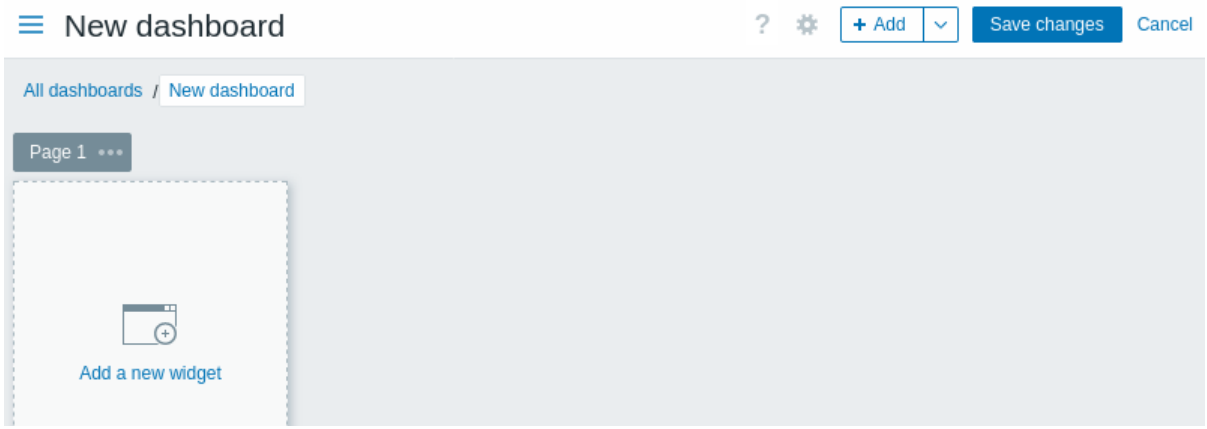
- Fare clic su *Create dashboard* durante la visualizzazione di tutte le dashboard
- Selezionare *Create new* dal menu delle azioni durante la visualizzazione di una singola dashboard

Verrà innanzitutto richiesto di inserire i parametri generali della dashboard:



Parameter	Description
Owner	Selezionare l'utente di sistema che sarà il proprietario della dashboard.
Name	Inserire il nome della dashboard.
Default page display period	Selezionare il periodo di tempo per cui una pagina della dashboard viene visualizzata prima di passare alla pagina successiva in una presentazione.
Start slideshow automatically	Selezionare questa casella di controllo per avviare automaticamente una presentazione quando esiste più di una pagina della dashboard.

Quando si fa clic su *Apply*, viene aperta una dashboard vuota:



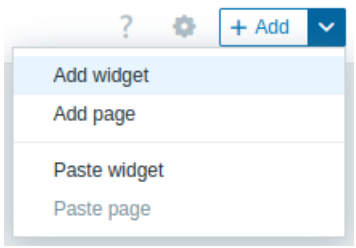
Per popolare la dashboard, è possibile aggiungere widget e pagine.

Fare clic sul pulsante *Save changes* per salvare la dashboard. Se si fa clic su *Cancel*, la dashboard non verrà creata.

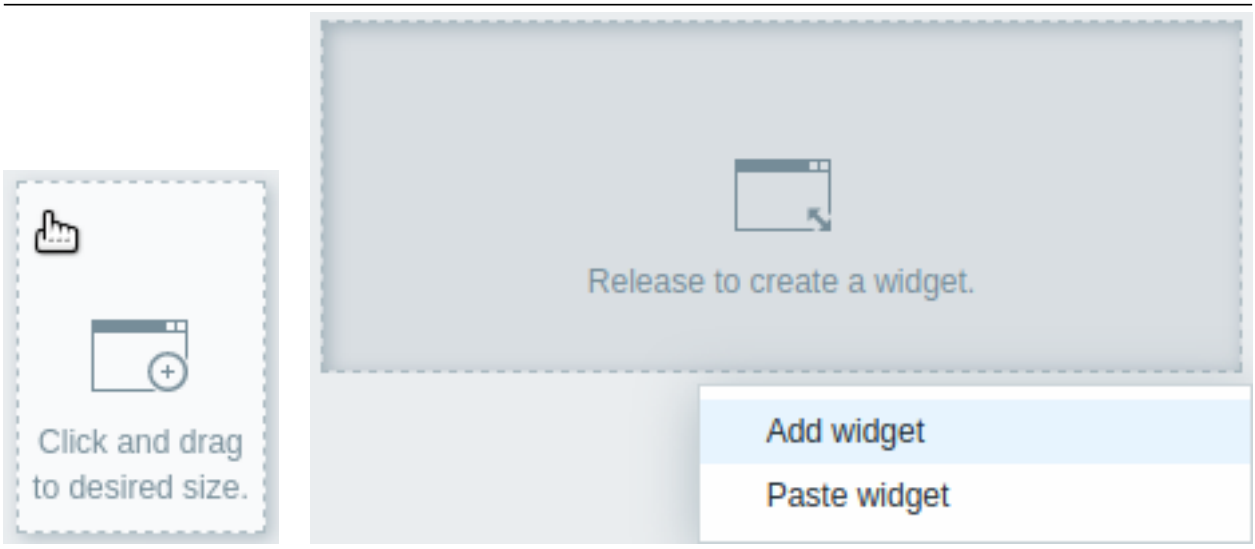
Aggiunta di widget

Puoi aggiungere vari widget (Registro azioni, Orologio, Stato di discovery, ecc.) a una dashboard. I widget possono essere aggiunti in due modi:

- Fai clic sul pulsante *Aggiungi* o sulla freccia accanto ad esso, quindi seleziona *Aggiungi widget* dal menu delle azioni. Dopo aver configurato il widget, questo verrà aggiunto con le dimensioni predefinite e posizionato dopo eventuali widget già esistenti.

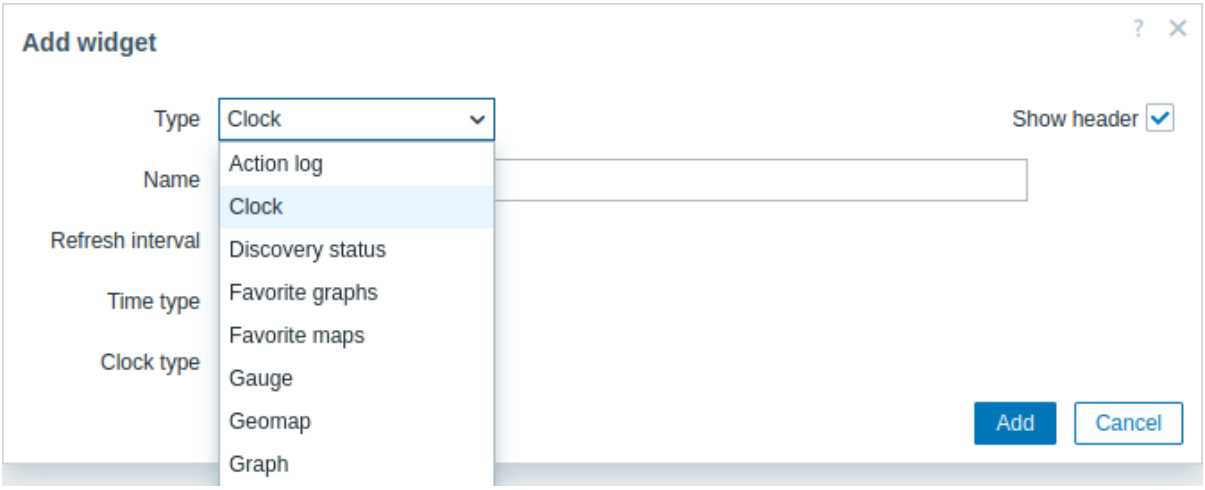


- Sposta il mouse in un punto vuoto della dashboard. Noterai comparire un segnaposto; fai quindi clic su di esso per aprire il modulo di configurazione del widget. Dopo aver configurato il widget, questo verrà aggiunto con le dimensioni predefinite oppure ridimensionato per adattarsi allo spazio disponibile. In alternativa, trascina il segnaposto per determinare la dimensione del widget e rilascialo per aprire il modulo di configurazione del widget (se un widget è stato copiato negli appunti, ti verrà prima chiesto di scegliere tra *Aggiungi widget* o *Incolla widget*).



Nel modulo di configurazione del widget:

1. Seleziona il tipo di widget.
2. Configura i parametri del widget.
3. Fai clic su *Aggiungi* per aggiungere il widget alla dashboard.



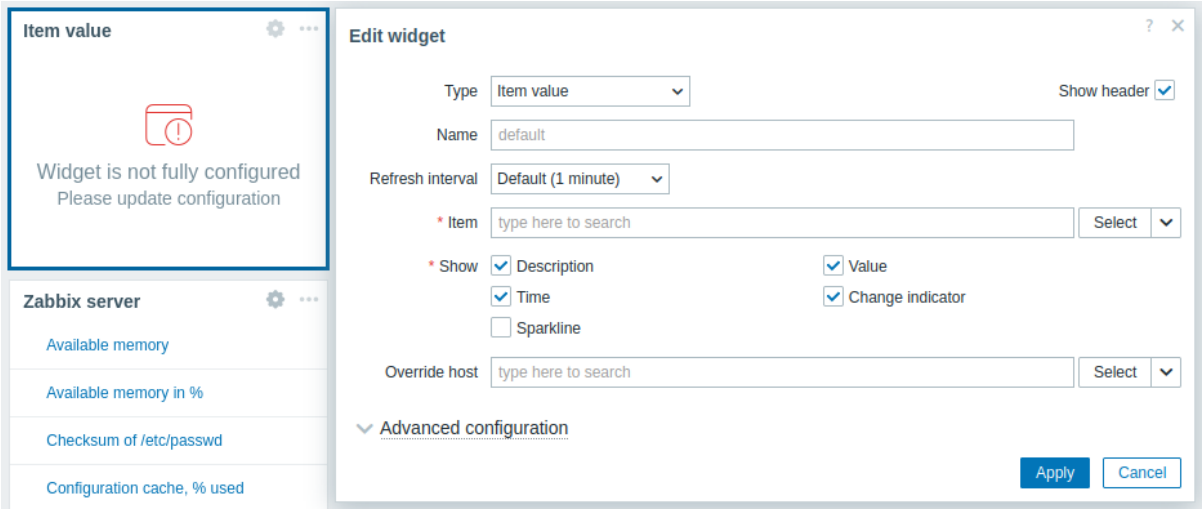
Modifica dei widget

I widget possono essere ridimensionati e riposizionati nella dashboard in modalità di modifica trascinando la barra del titolo. Puoi anche riposizionare il modulo di configurazione del widget trascinando la relativa barra del titolo.

Ogni widget include dei controlli nell'angolo superiore destro:

- - modifica il widget
- - apre il menu del widget

Durante la modifica, il widget viene evidenziato e le modifiche vengono visualizzate in anteprima in tempo reale. Se i parametri obbligatori del widget non sono configurati, il widget entra nello stato *Il widget non è completamente configurato*.



Dopo la modifica, fai clic su *Applica* nel widget, quindi su *Salva modifiche* nell'angolo superiore destro della dashboard per applicare le modifiche.

Copiare/incollare i widget

È possibile copiare e incollare i widget del dashboard per creare rapidamente nuovi widget con la stessa configurazione. I widget possono essere copiati e incollati all'interno dello stesso dashboard oppure tra dashboard aperti in schede diverse.

Per copiare un widget, utilizzare il menu del widget. Per incollare un widget copiato (disponibile nella modalità di modifica del dashboard):

- Fare clic sulla freccia accanto al pulsante *Aggiungi* e selezionare *Incolla widget*.
- In alternativa, fare clic su un'area vuota del dashboard e selezionare *Incolla widget*.

È inoltre possibile incollare un widget copiato sopra un widget esistente utilizzando l'opzione *Incolla* nel relativo menu del widget.

Creazione di una presentazione

Una presentazione verrà eseguita automaticamente se la dashboard contiene due o più pagine (vedere Aggiunta di pagine) e se una delle seguenti condizioni è vera:

- L'opzione *Avvia automaticamente la presentazione* è selezionata nelle proprietà della dashboard
- L'URL della dashboard contiene un parametro `slideshow=1`

Le pagine ruotano in base agli intervalli specificati nelle proprietà della dashboard e delle singole pagine. Fare clic su:

- *Interrompi presentazione* - per interrompere la presentazione
- *Avvia presentazione* - per avviare la presentazione



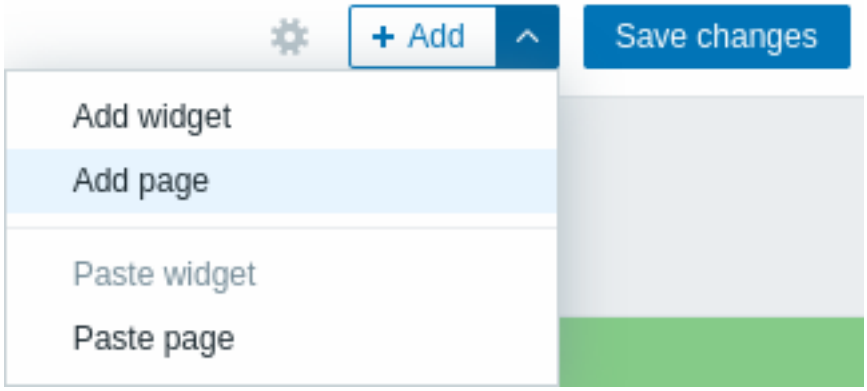
I controlli relativi alla presentazione sono disponibili anche in modalità chiosco (dove viene mostrato solo il contenuto della pagina):

-  - interrompi presentazione
-  - avvia presentazione
-  - torna indietro di una pagina
-  - vai alla pagina successiva

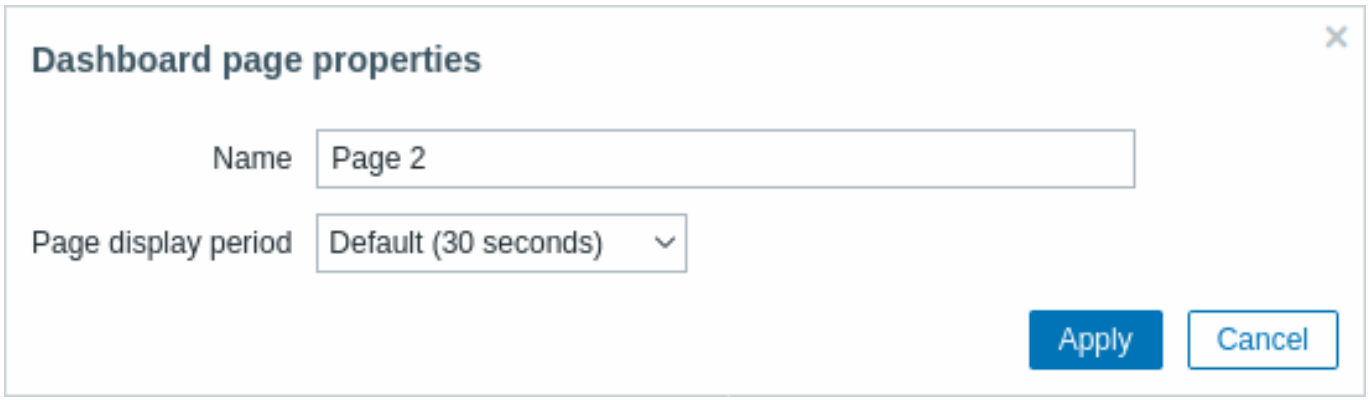
Aggiunta di pagine

Per aggiungere una nuova pagina a una dashboard:

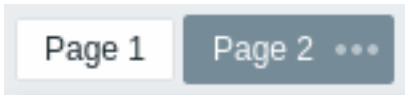
- Assicurati che la dashboard sia in modalità di modifica
- Fai clic sulla freccia accanto al pulsante *Aggiungi* e seleziona l'opzione *Aggiungi pagina*



- Compila i parametri generali della pagina e fai clic su *Applica*. Se lasci il nome vuoto, la pagina verrà aggiunta con il nome `Page N`, dove 'N' è il numero progressivo della pagina. Il periodo di visualizzazione della pagina consente di personalizzare per quanto tempo una pagina viene mostrata in una presentazione.



Verrà aggiunta una nuova pagina, indicata da una nuova scheda (*Page 2*).



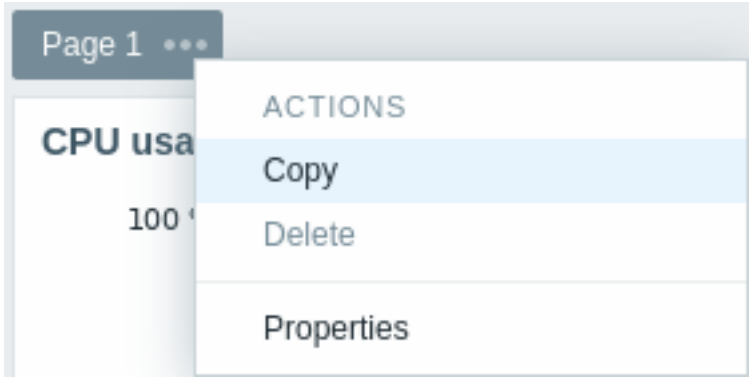
Le pagine possono essere riordinate trascinando le relative schede. Il riordinamento mantiene la denominazione originale delle pagine. È sempre possibile passare a ciascuna pagina facendo clic sulla relativa scheda.

Quando viene aggiunta una nuova pagina, questa è vuota. Puoi aggiungervi widget come descritto sopra.

Copiare/incollare le pagine

Le pagine della dashboard possono essere copiate e incollate, consentendo di creare una nuova pagina con le proprietà di una pagina esistente. Possono essere incollate dalla stessa dashboard o da una dashboard diversa.

Per incollare una pagina esistente nella dashboard, copiarla prima utilizzando il menu della pagina:

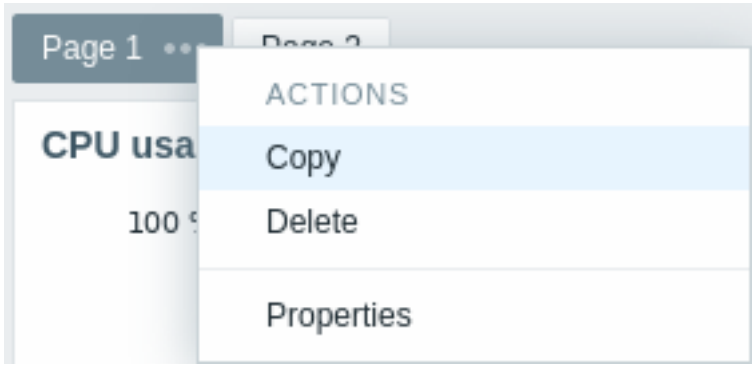


Per incollare la pagina copiata:

- Assicurarsi che la dashboard sia in modalità di modifica
- Fare clic sulla freccia accanto al pulsante *Aggiungi* e selezionare l'opzione *Incolla pagina*

Menu della pagina

Il menu della pagina può essere aperto facendo clic sui tre puntini  accanto al nome della pagina:

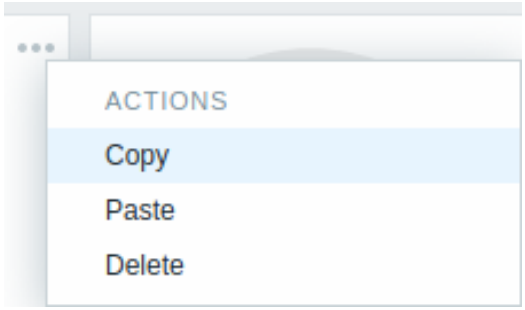
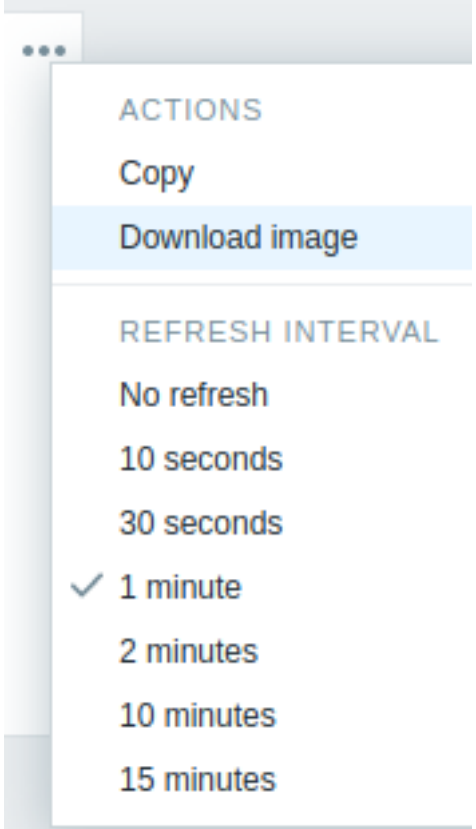


Contiene le seguenti opzioni:

- *Copia* - copia la pagina
- *Elimina* - elimina la pagina (le pagine possono essere eliminate solo nella modalità di modifica della dashboard)
- *Proprietà* - personalizza i parametri della pagina (il nome e il periodo di visualizzazione della pagina in una presentazione)

Menu del widget

Il menu del widget contiene diverse opzioni a seconda che la dashboard sia in modalità di modifica o di visualizzazione:

Menu del widget	Opzioni
In modalità di modifica della dashboard: 	*Copia* - copia il widget. *Incolla* - incolla un widget copiato sopra questo widget. Questa opzione è disattivata se non è stato copiato alcun widget. *Elimina* - elimina il widget.
In modalità di visualizzazione della dashboard: 	*Copia* - copia il widget. *Scarica immagine* - scarica il widget come immagine PNG. Questa opzione è disponibile per i widget che supportano questa funzionalità (vedere le singole pagine dei widget). A seconda delle impostazioni del browser, l'immagine viene salvata automaticamente nella cartella Download oppure si apre una finestra di dialogo Salva, che consente di scegliere la posizione. *Intervallo di aggiornamento* - selezionare la frequenza di aggiornamento del contenuto del widget

Autorizzazioni per le dashboard

Le autorizzazioni per le dashboard per gli utenti normali e per gli utenti di tipo 'Admin' (a condizione che l'accesso alle dashboard e l'opzione *Create and edit dashboards* siano abilitati per il ruolo utente loro assegnato) sono limitate nel modo descritto di seguito.

- Possono vedere e clonare una dashboard pubblica anche quando nessun utente o gruppo di utenti è stato aggiunto nelle preferenze di condivisione della dashboard.
- Possono vedere e clonare una dashboard privata se dispongono di almeno diritti di *Read* su di essa impostati tramite le preferenze di condivisione.
- Possono modificare ed eliminare una dashboard solo se dispongono di diritti di *Read-write* su di essa impostati tramite le preferenze di condivisione.
- Non possono cambiare il proprietario della dashboard.

1 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.

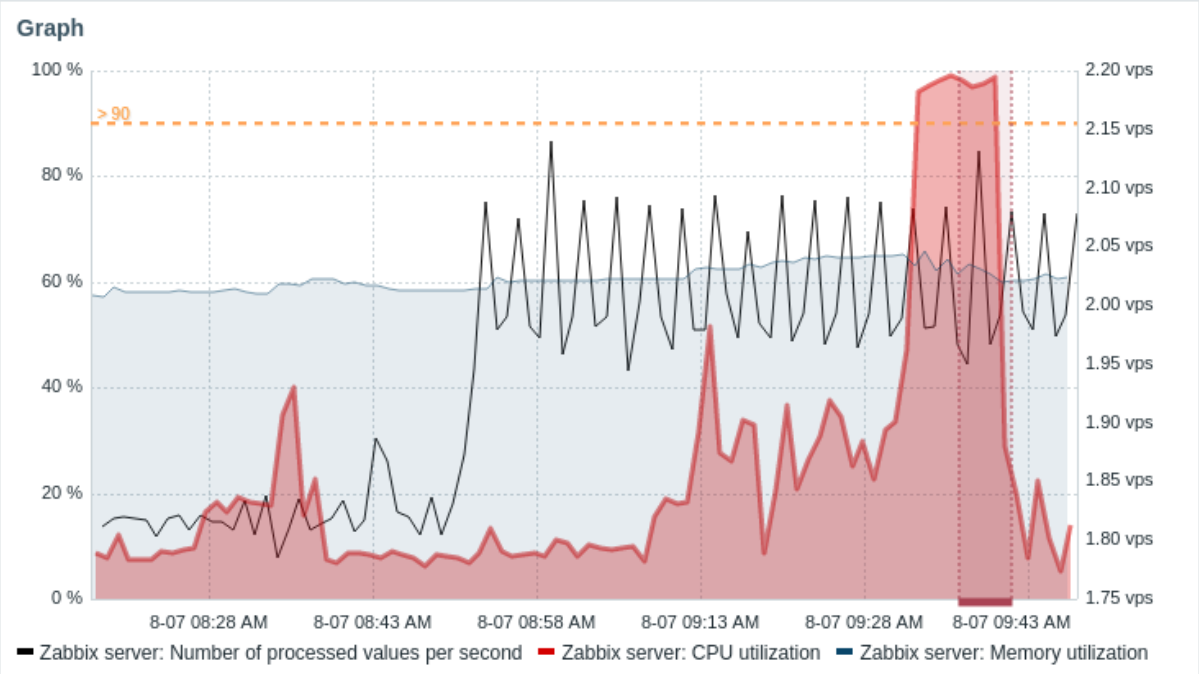
Per i dettagli sulla configurazione dei widget, vedere Dashboard widget parameters.

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.

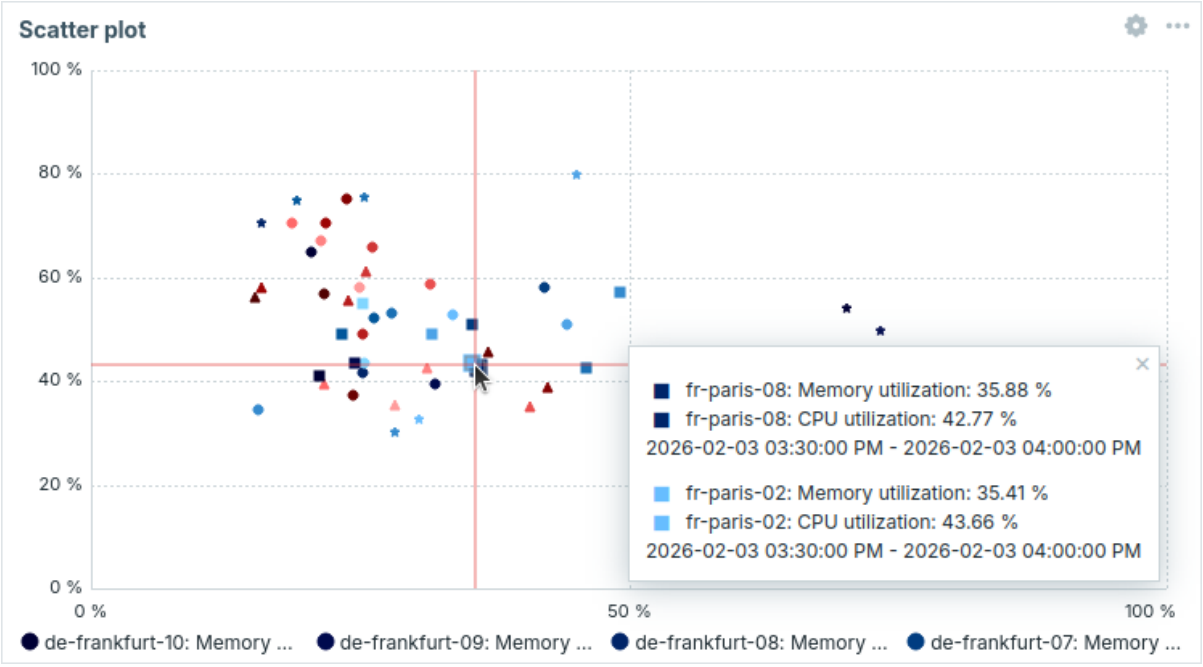
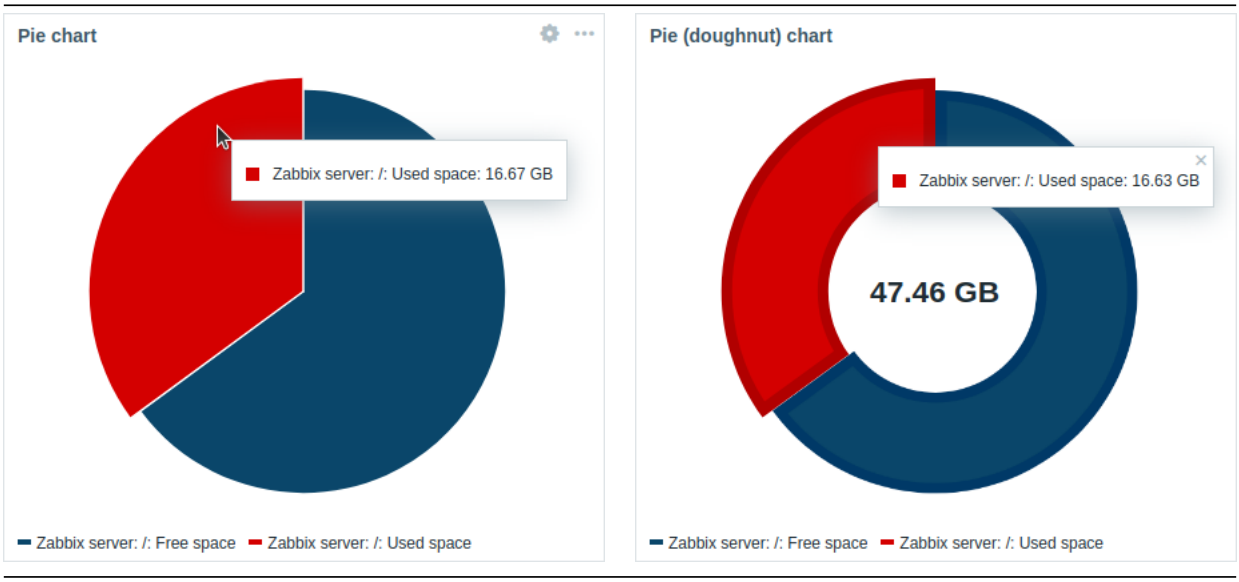


Grafico a torta

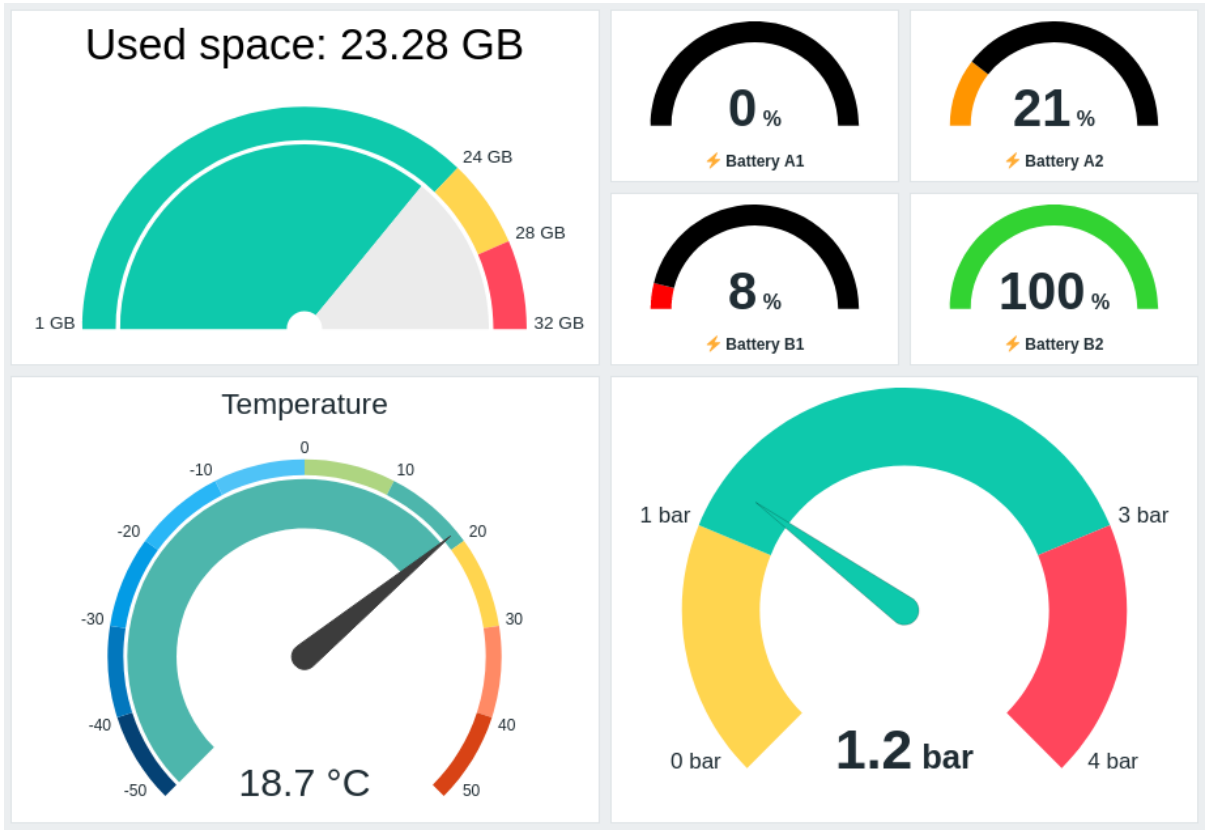
Il widget Pie chart visualizza i dati numerici degli item come un grafico a torta o ad anello vettoriale. Questo widget può aiutarti a visualizzare come gli item o gli host contribuiscono al dataset complessivo.



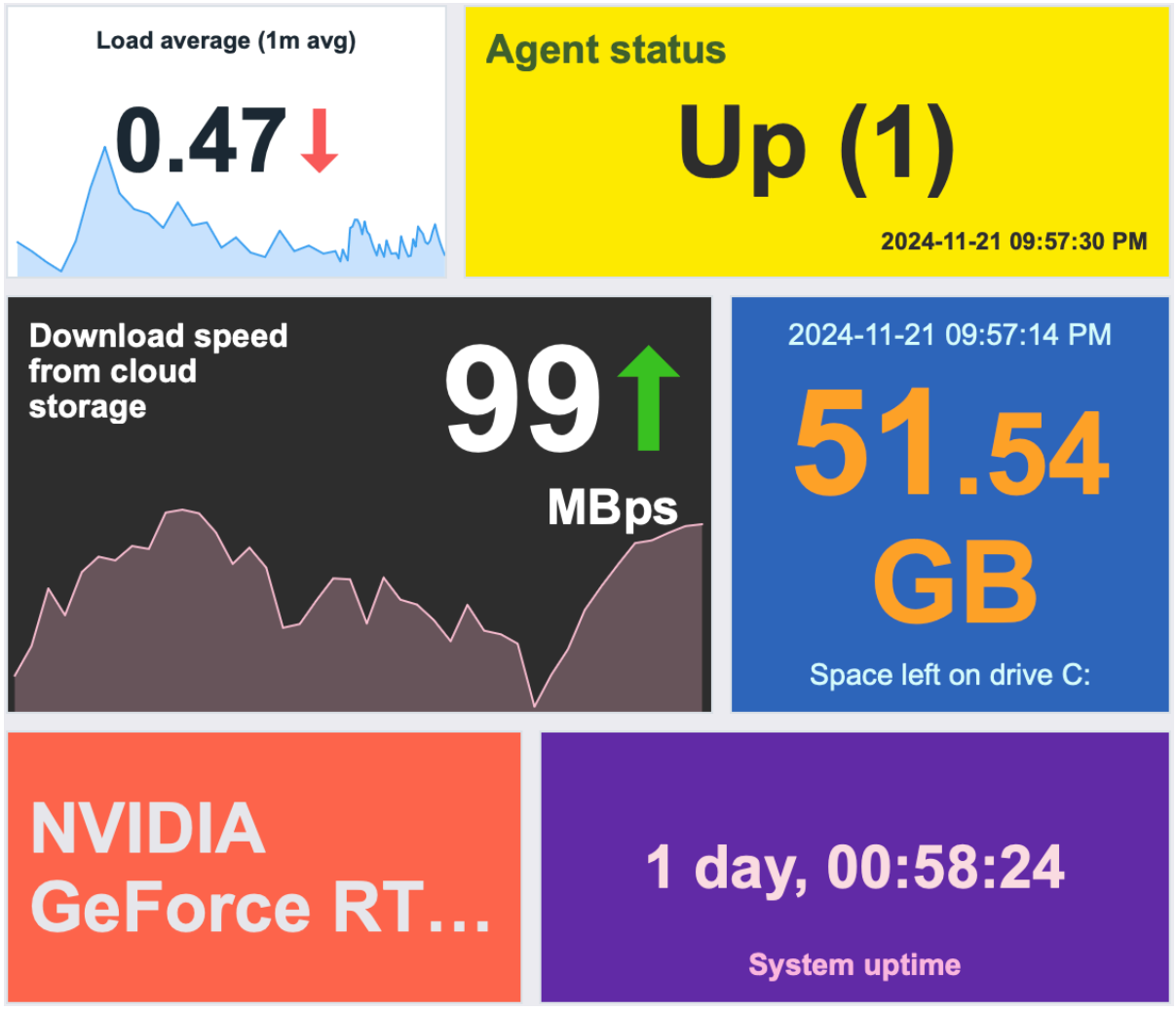
Valore indicatore/Valore item

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:

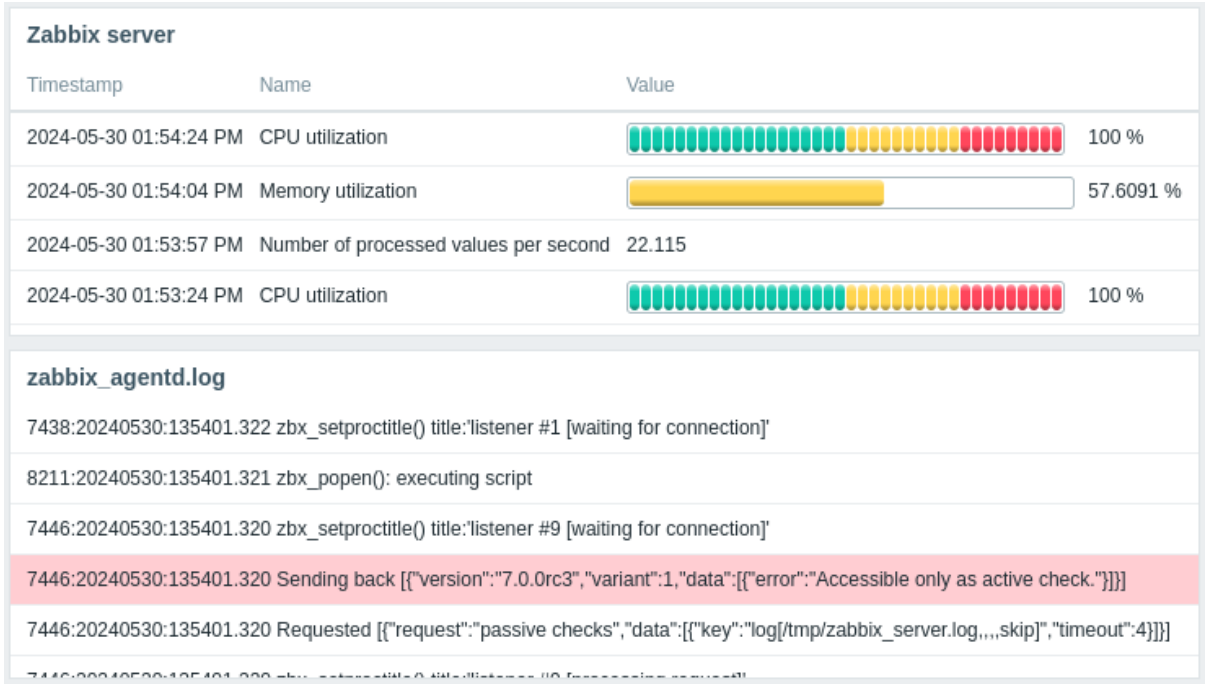


Widget Valore item:



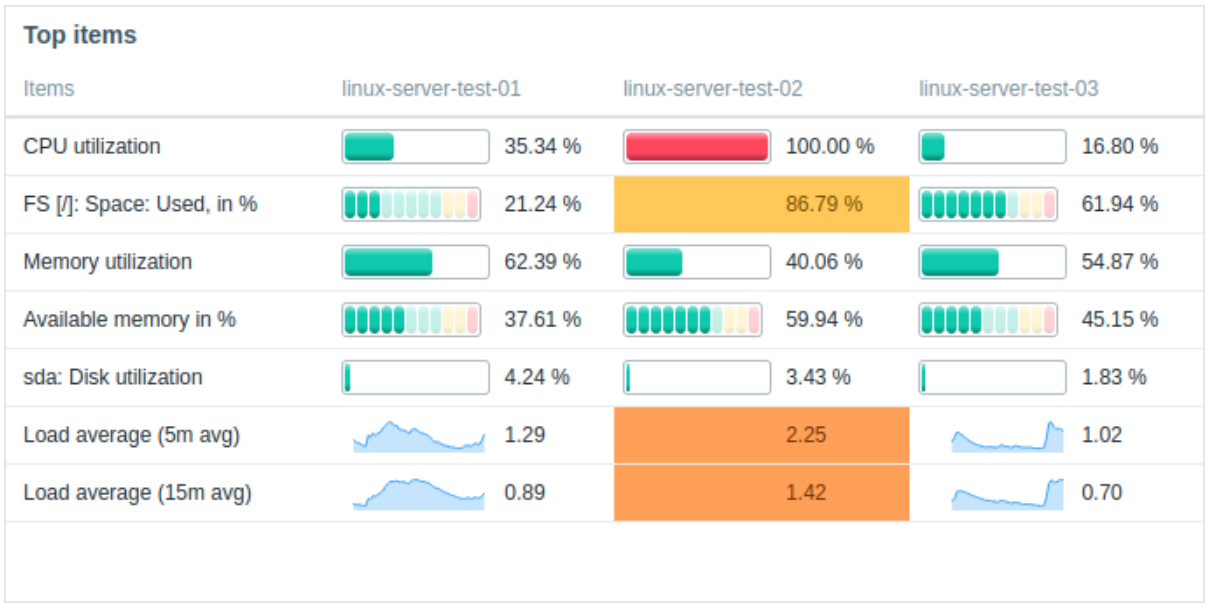
Cronologia degli item

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.

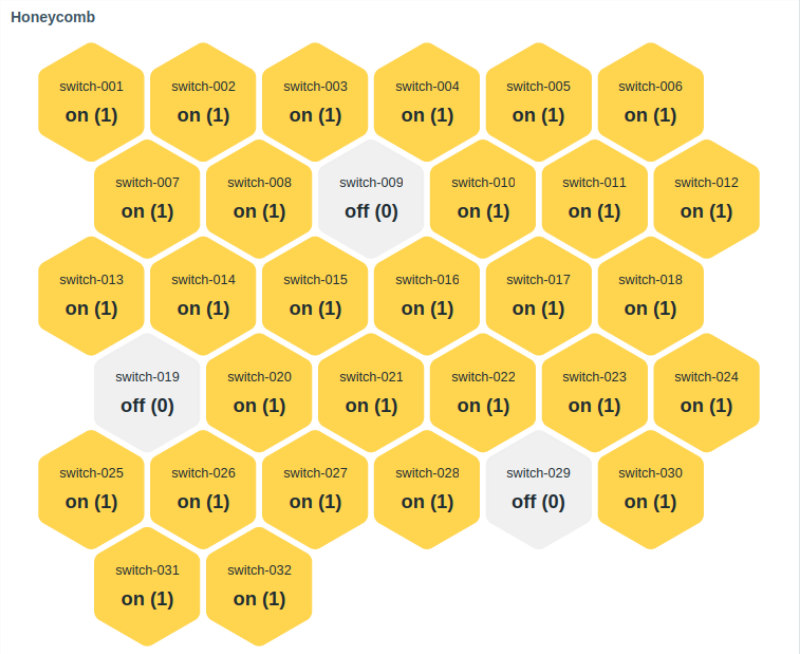


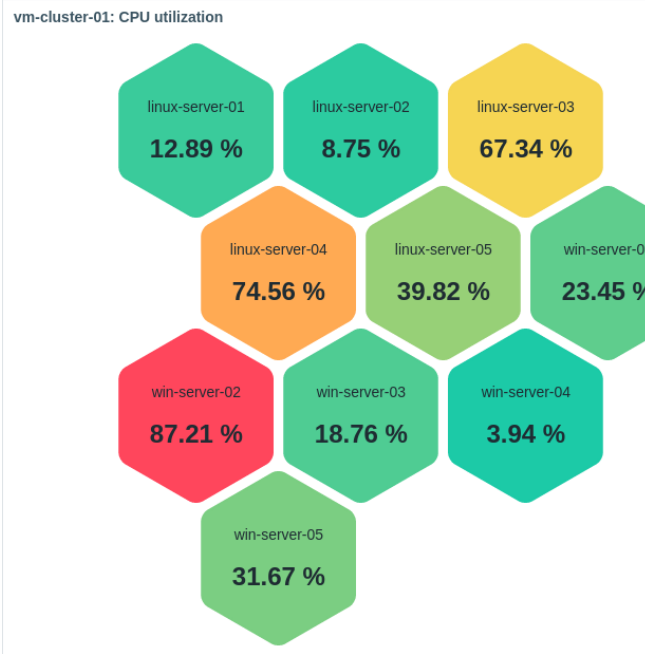
Vedi anche:

- 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.



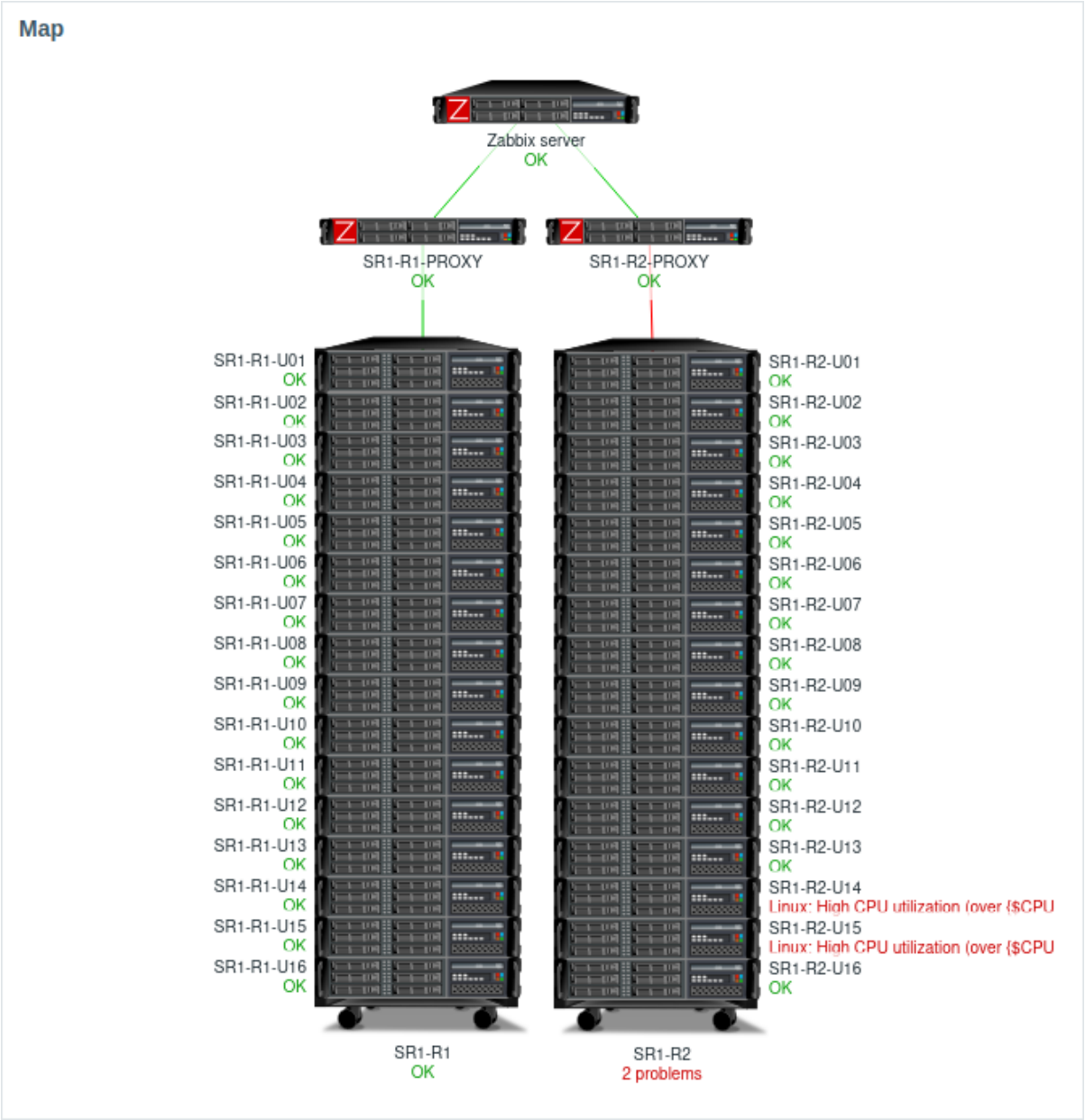


Mappe

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.

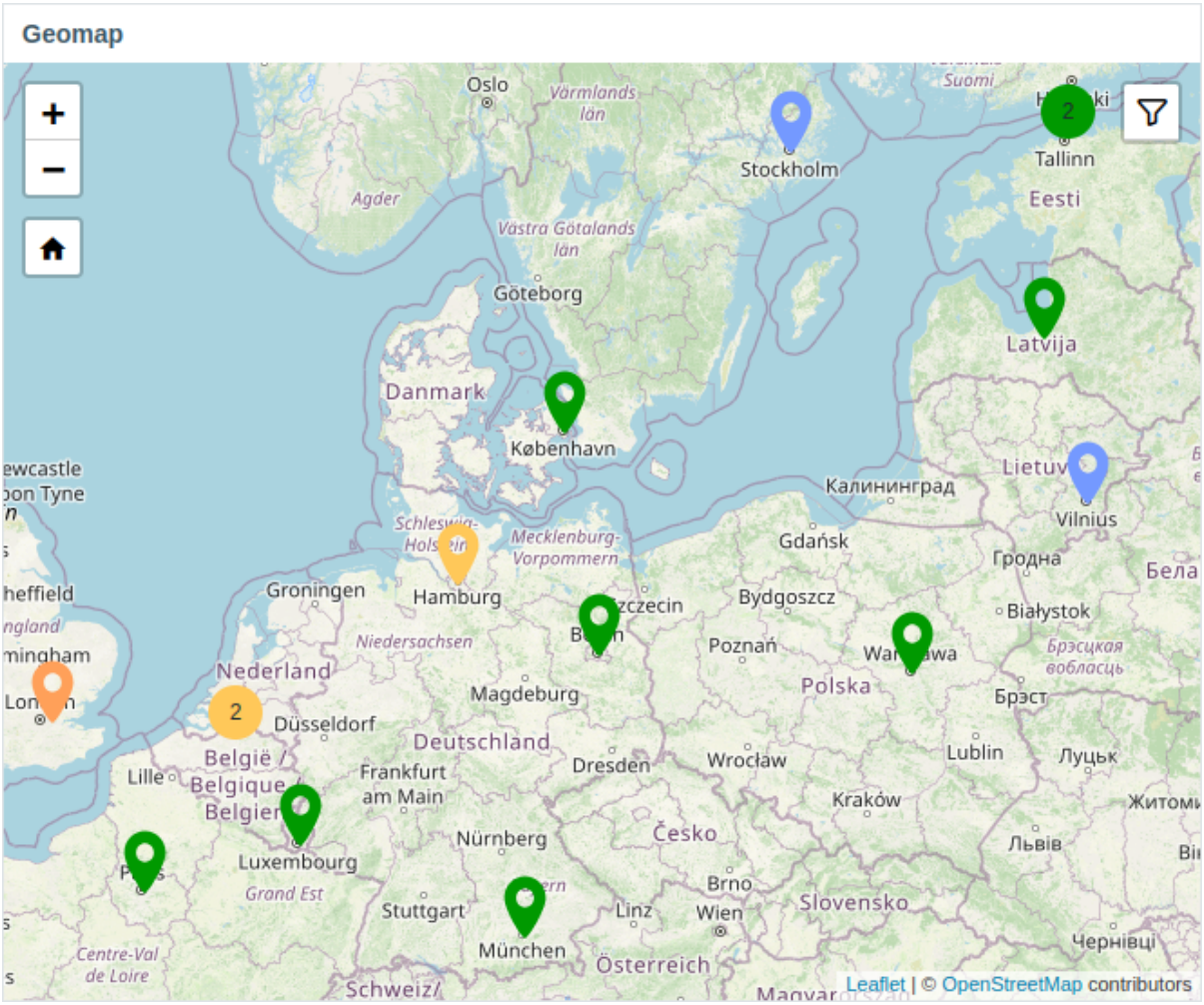


Vedi anche:

- 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.

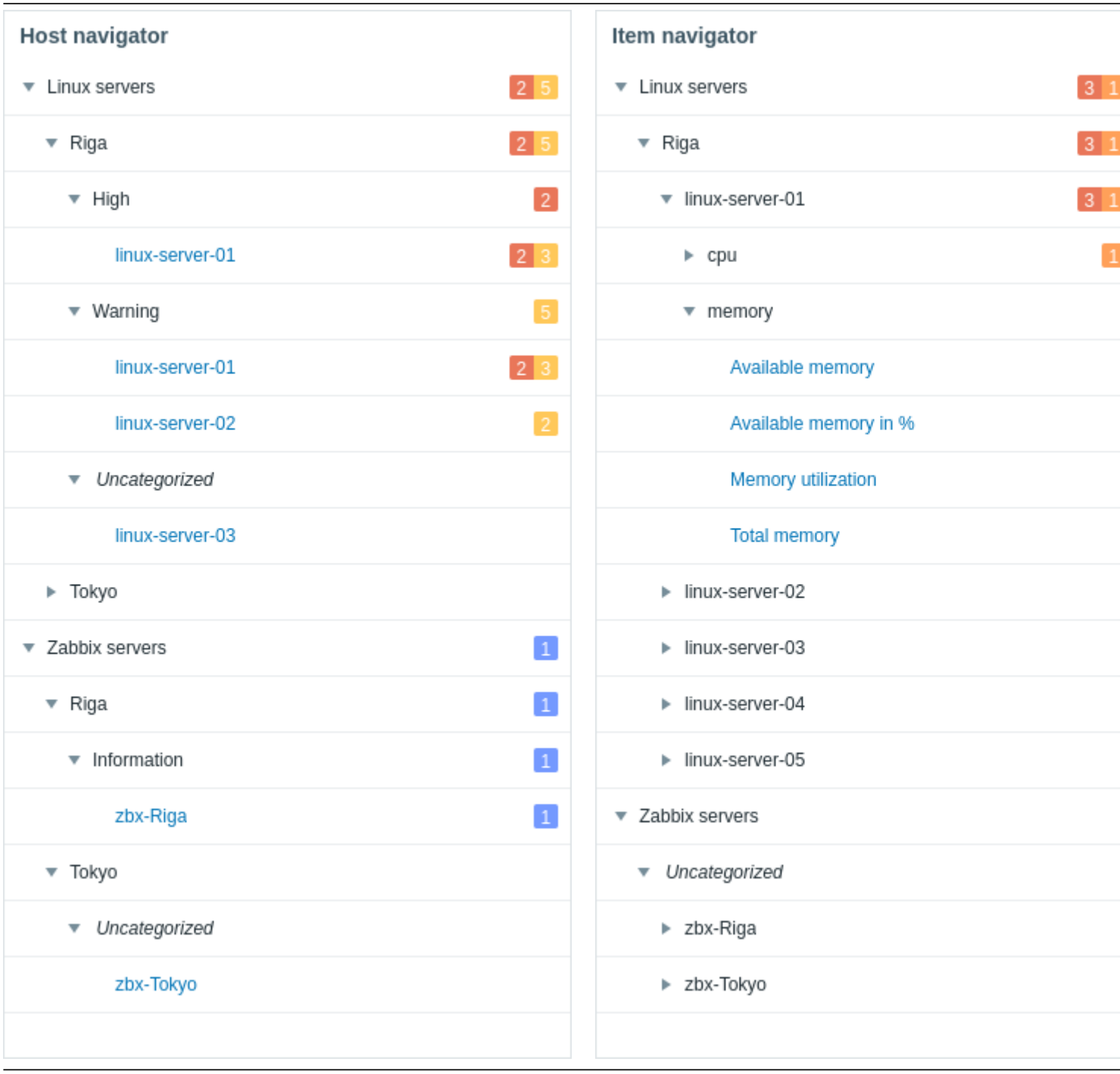


Dettagli di host/item

Questi widget forniscono elenchi e viste dettagliate di host o item.

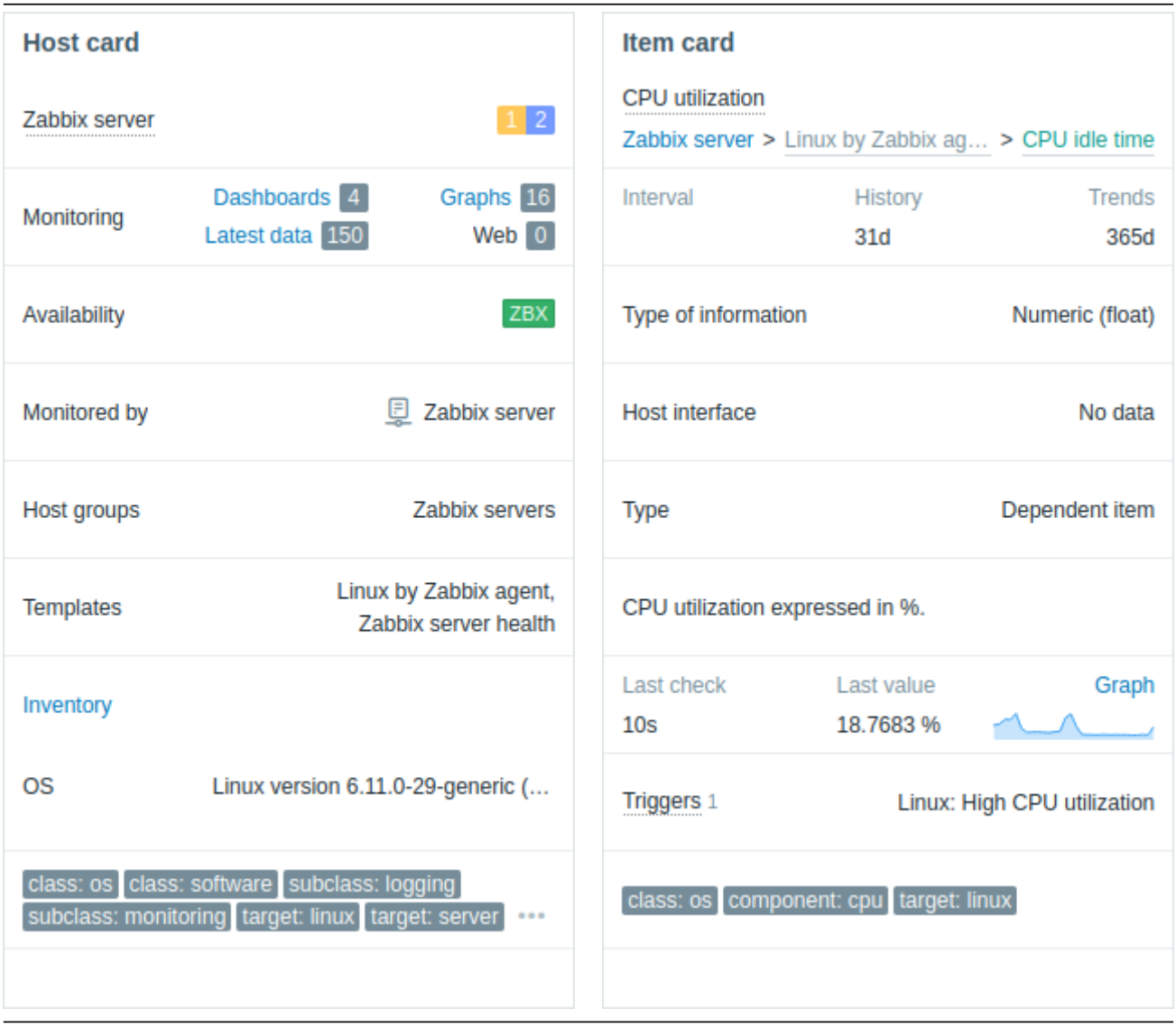
Navigatore host/item

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.).

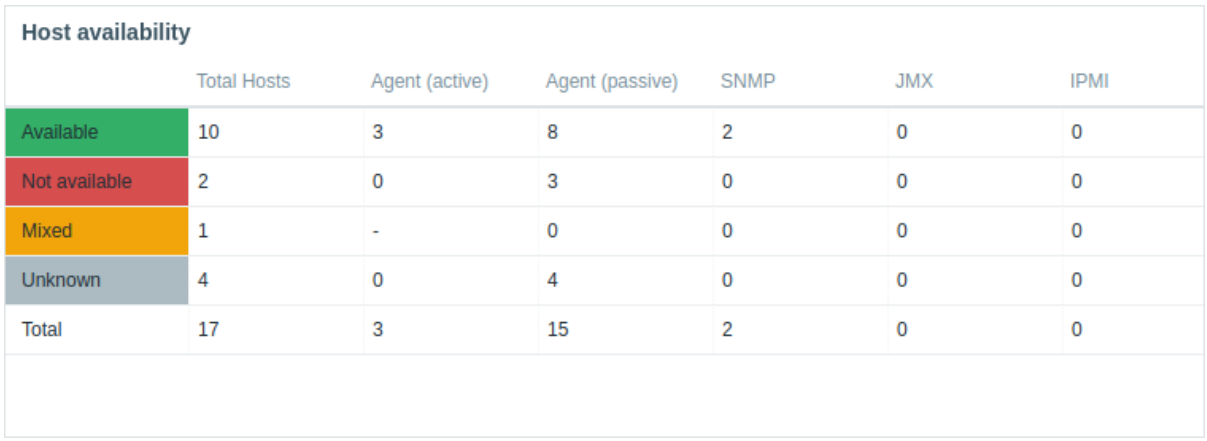


Stato e report

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.

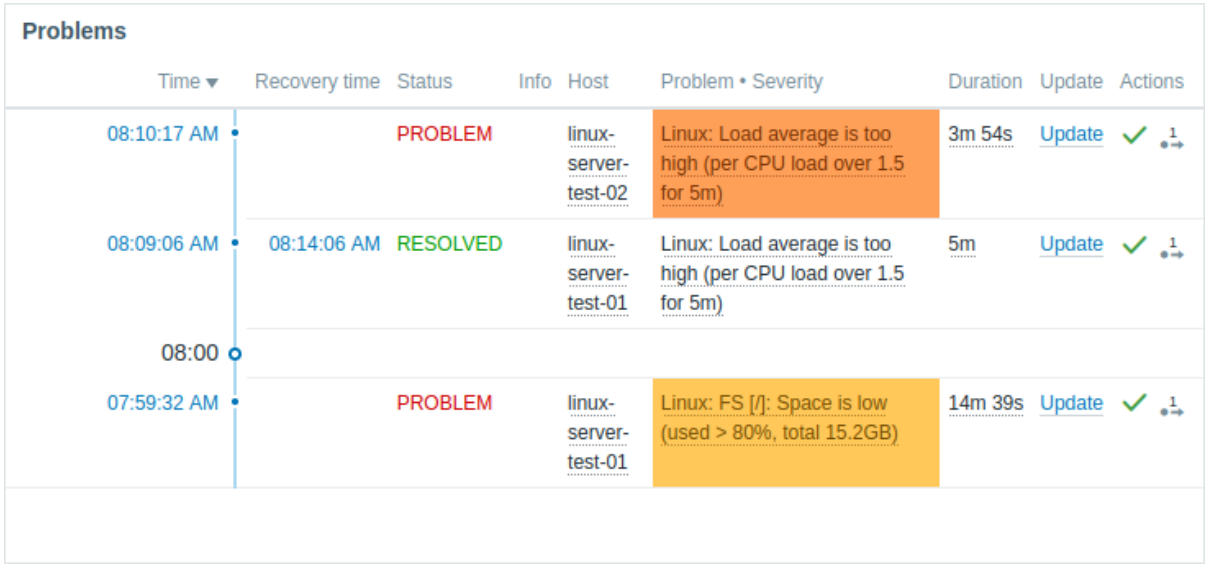


Host availability

	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

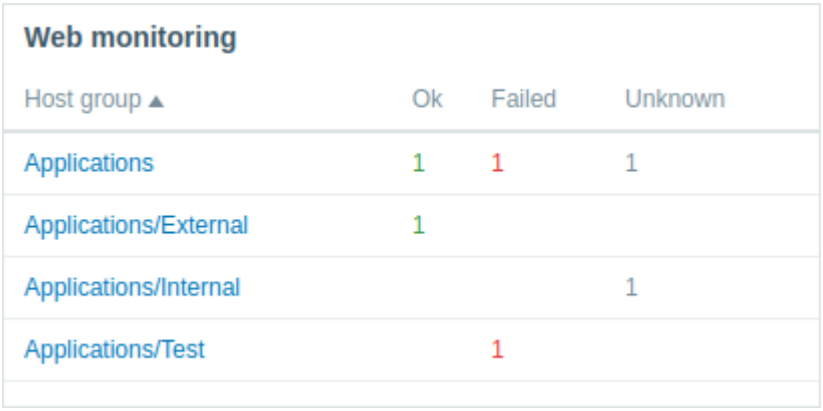
Il widget SLA report visualizza report SLA, aiutandoti a monitorare le prestazioni del servizio rispetto agli obiettivi definiti.

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

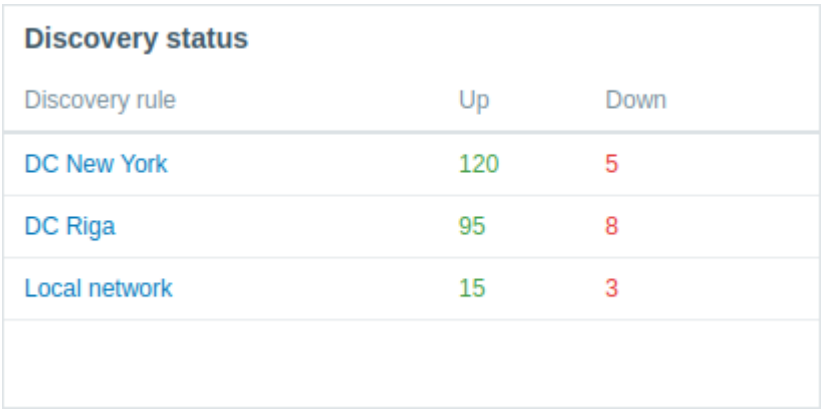
Monitoraggio web

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

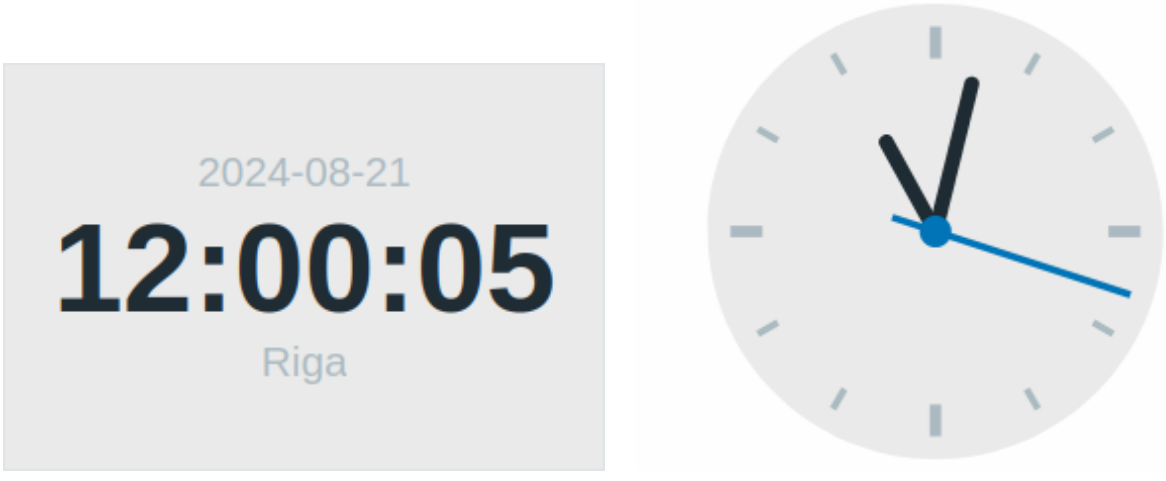
Il widget Informazioni di sistema visualizza un riepilogo dei principali dati del server Zabbix e del sistema oppure i dettagli dei nodi ad alta disponibilità.

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	

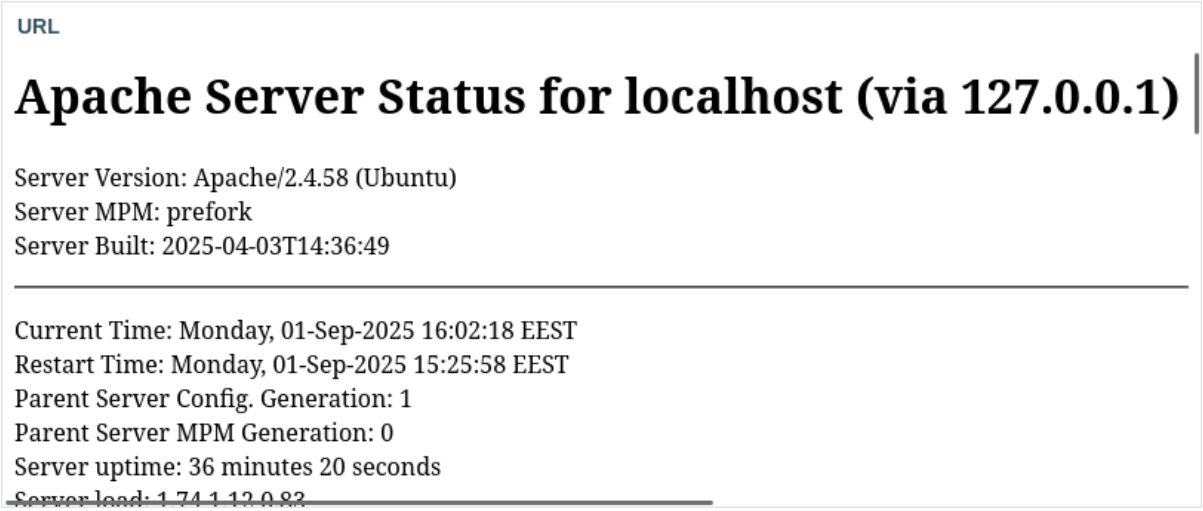
Orologio

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.



2 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 singolo widget:

Name	Inserire un nome per il widget.
Refresh interval	Configurare l'intervallo di aggiornamento predefinito. Gli intervalli di aggiornamento predefiniti per i widget vanno da *No refresh* a *15 minutes* a seconda del tipo di widget. Ad esempio: - *No refresh* per il widget *URL*; - *1 minute* per il widget *Action log*; - *15 minutes* per il widget *Clock*. Gli intervalli di aggiornamento possono essere impostati su un valore predefinito per tutti gli utenti. Passare la dashboard in 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 dashboard in modalità di visualizzazione, fare clic sul pulsante con i tre puntini su un widget e selezionare l'intervallo di aggiornamento desiderato dall'elenco a discesa. Si noti che l'intervallo di aggiornamento personalizzato di un utente ha la priorità sull'impostazione del widget e viene mantenuto anche quando l'impostazione del widget viene modificata.
Show header	Selezionare la casella di controllo per visualizzare permanentemente l'intestazione del widget. Se non selezionata, l'intestazione viene nascosta per risparmiare spazio e diventa visibile solo al passaggio del 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*.

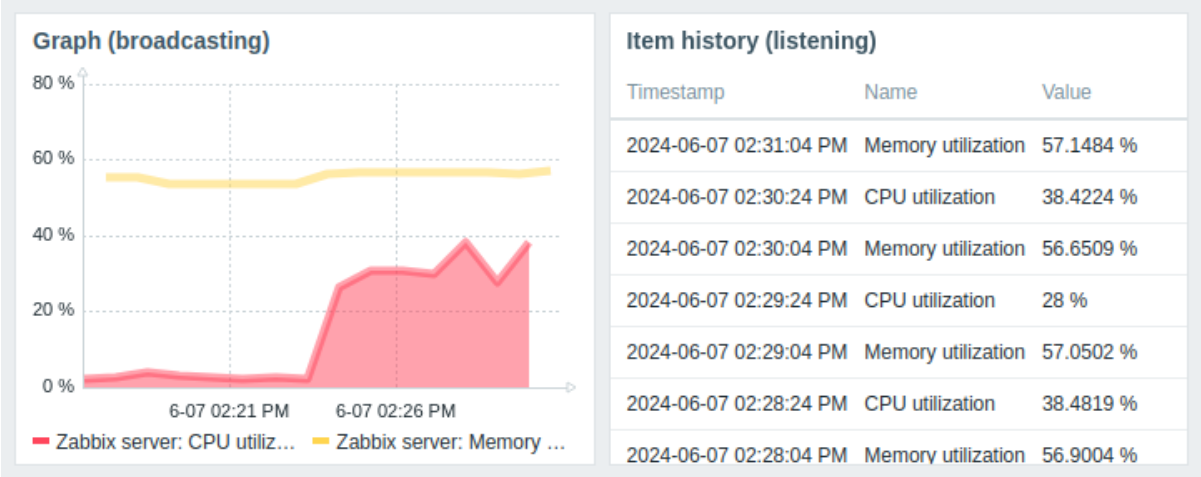
- Il widget *Geomap* può trasmettere dati dei host ai widget che li ricevono (*Honeycomb*, *Top items*, ecc.) e può anche ricevere dati di gruppi di host e host dai widget che li trasmettono (*Honeycomb*, *Problem hosts*, ecc.).
- Il widget *Clock* non può né trasmettere né ricevere alcun dato.

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

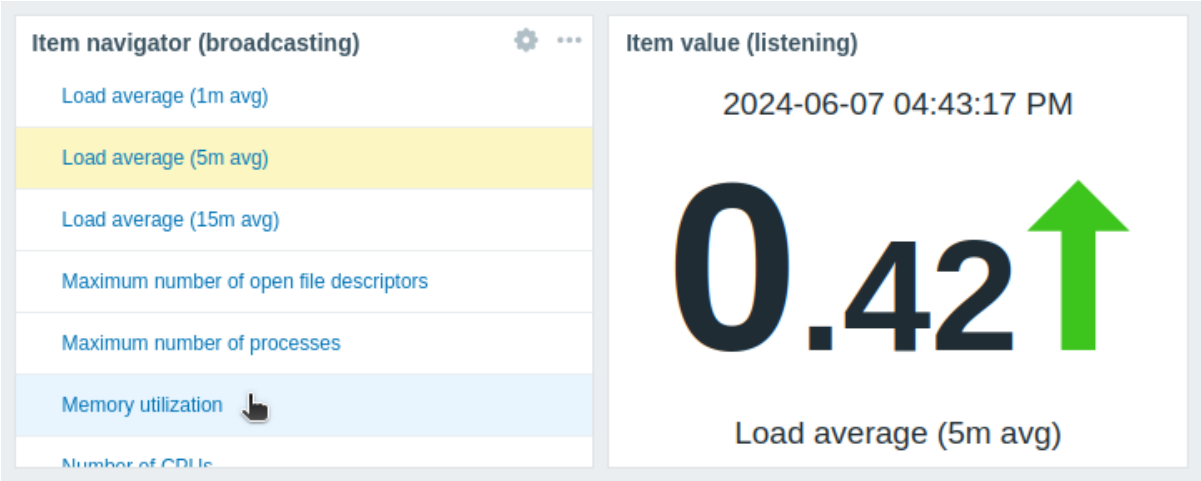
I widget differiscono nel modo in cui **trasmettono** dati ad altri widget.

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.

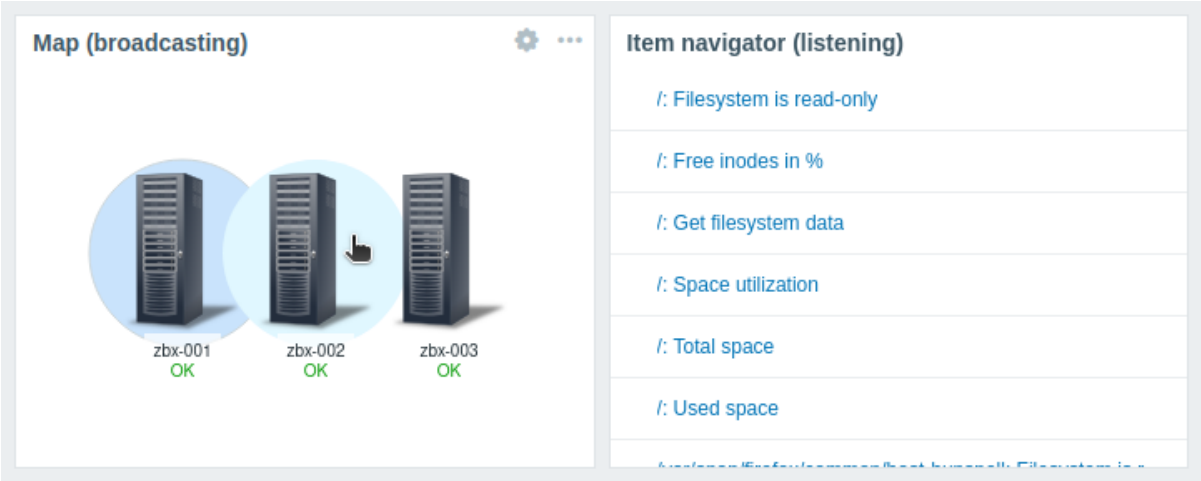


I widget che supportano la selezione di entità trasmettono automaticamente i dati per la prima entità disponibile. Ad esempio, il widget *Item navigator* trasmette i dati per il primo item nel proprio elenco di item. Quando viene selezionato un item diverso,

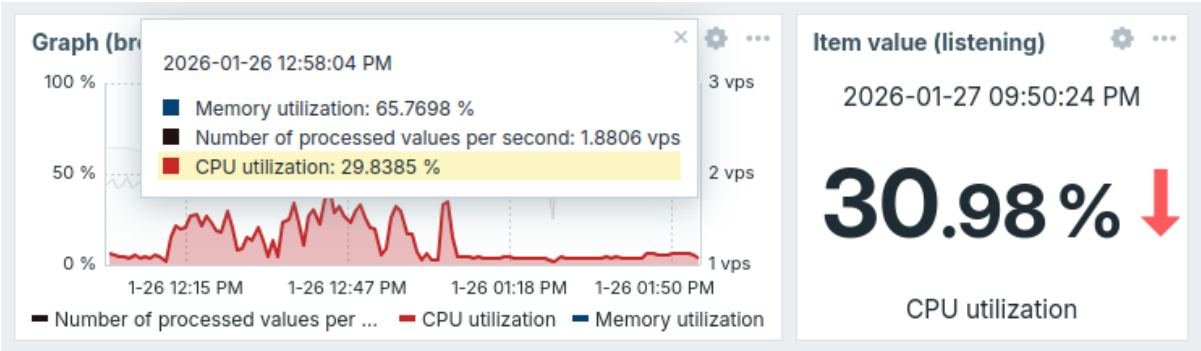
trasmette i dati di quell'item. Al passaggio del mouse, l'item viene evidenziato in azzurro chiaro; quando viene selezionato, viene evidenziato in giallo.



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.

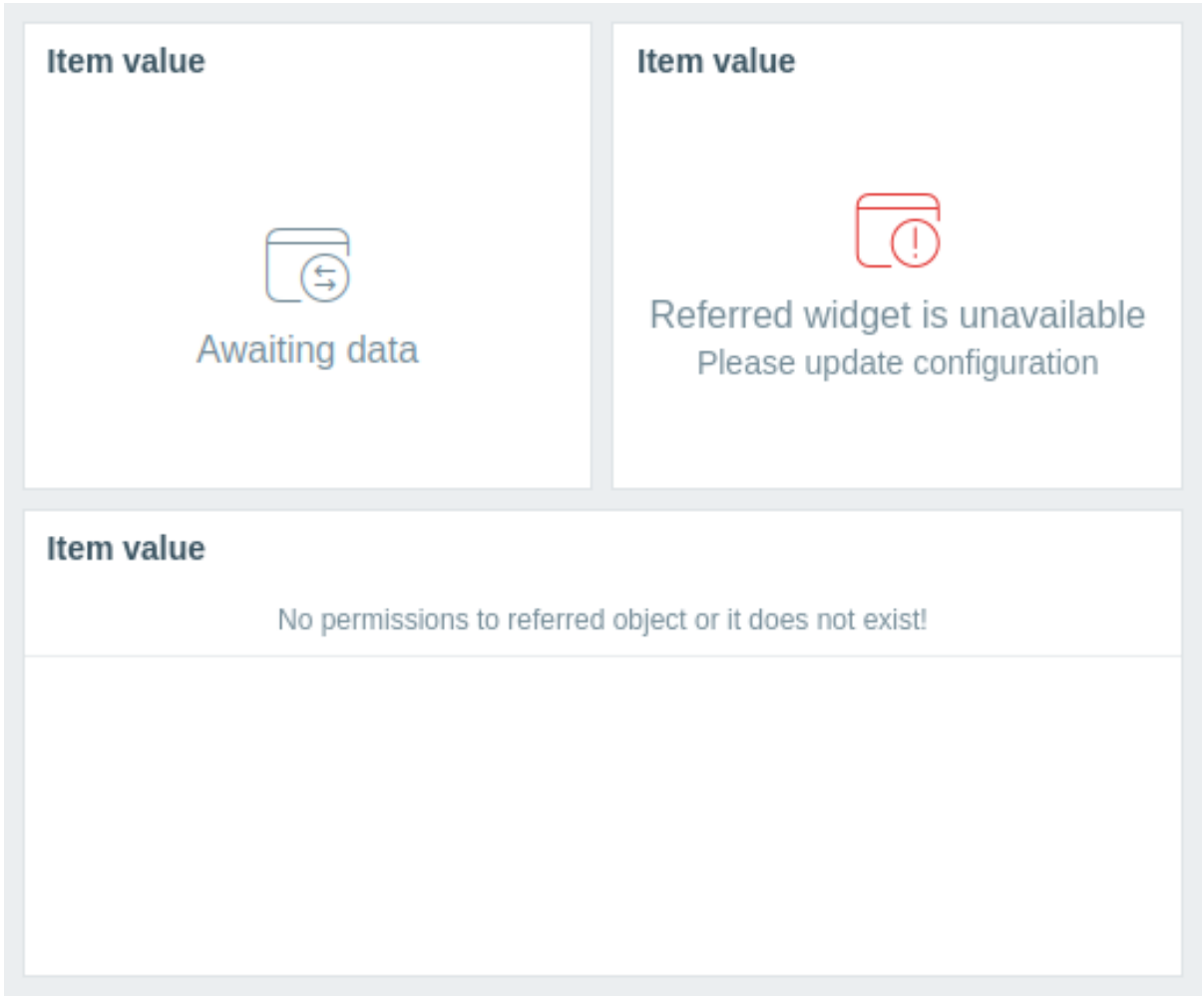


Widget come *Graph* e *Scatter plot* possono trasmettere dati degli item quando un item viene selezionato nei relativi tooltip:



I widget hanno anche comportamenti specifici quando **ricevono** dati da altri widget:

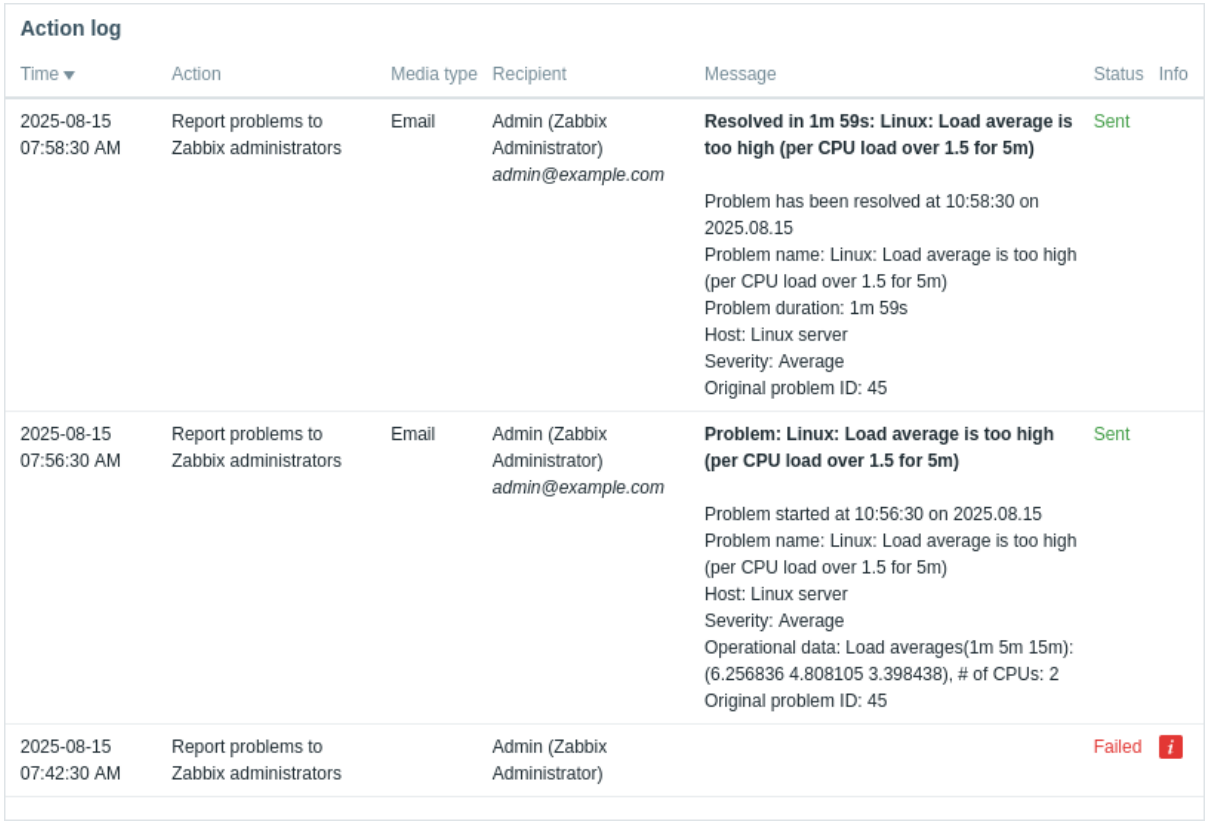
- Se il widget origine dati non sta trasmettendo dati, il widget in ascolto entra nello stato *In attesa di dati*.
- Se il widget origine dati è stato eliminato, sostituito con un widget incompatibile oppure spostato in un'altra pagina della dashboard, il widget in ascolto entra nello stato *Il widget 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!"*



1 Registro delle azioni

Panoramica

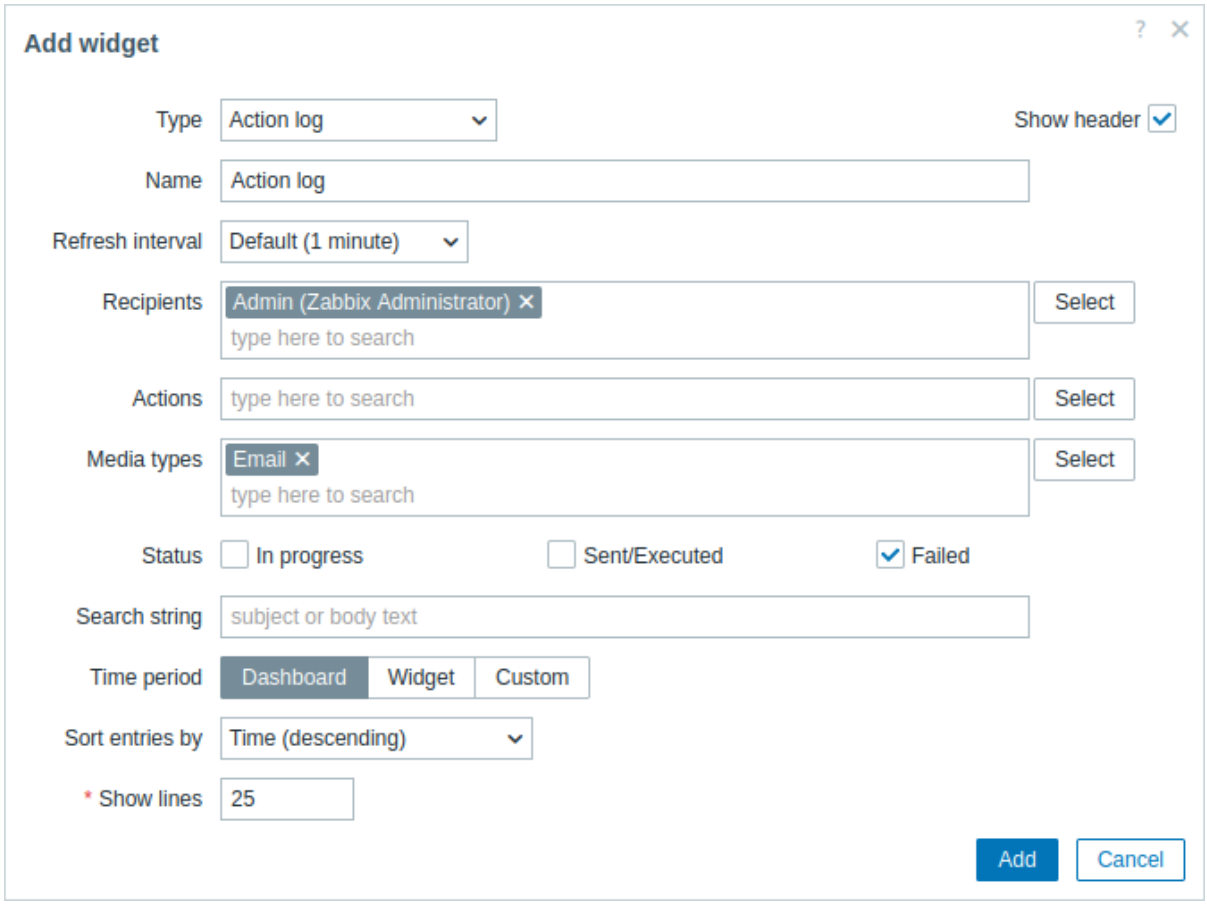
Il widget *Log azioni* visualizza i dettagli delle operazioni (notifiche, comandi remoti) eseguite all'interno di un'azione. Il log dettagliato aiuta nelle attività di audit, risoluzione dei problemi e monitoraggio dell'esecuzione di ciascuna azione.



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:



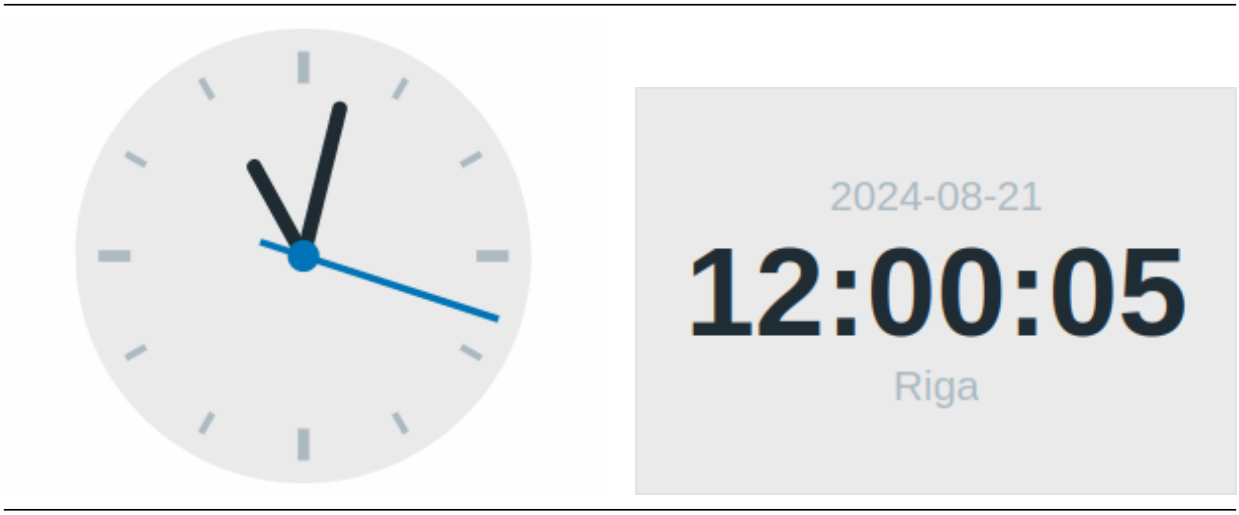
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.

2 Orologio

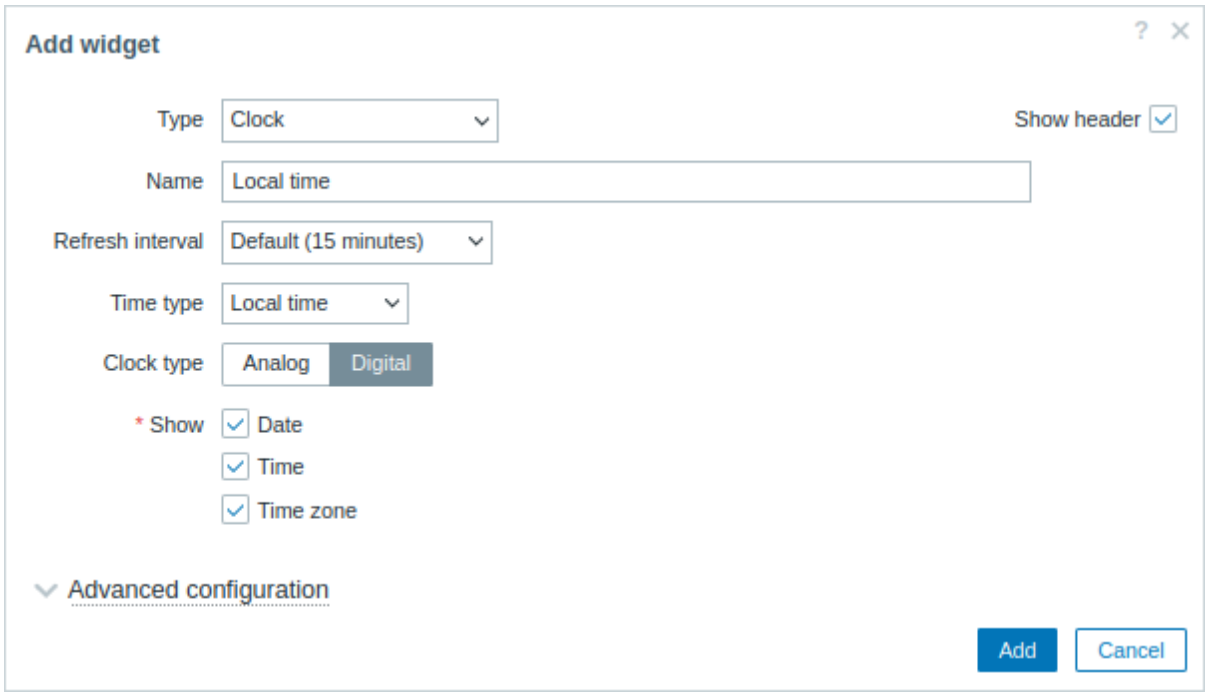
Panoramica

Il widget *Clock* visualizza l'ora locale, del server o dell'host in formato analogico o digitale.



Configurazione

Per configurare, selezionare *Clock* come tipo:



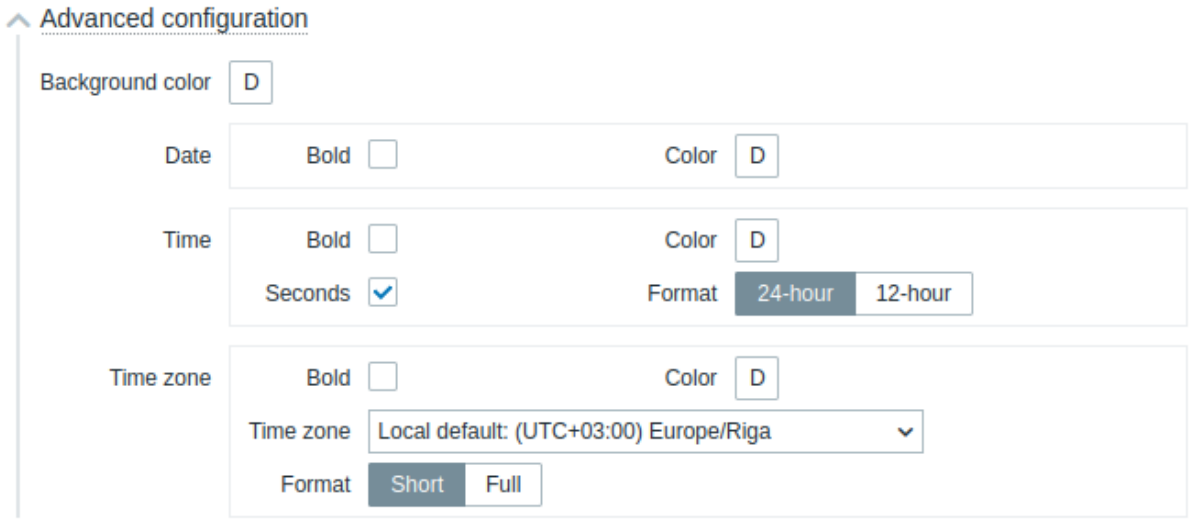
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Time type	Selezionare la sorgente dell'ora: **Local** - ora di sistema del dispositivo che esegue il browser; **Server** - fuso orario impostato globalmente o per l'utente Zabbix; **Host** - ora recuperata da un item del host.
Item	Selezionare l'item per l'ora del host (utilizzare l'item system.localtime[local]). Questo parametro è disponibile se *Time type* è impostato su "Host time".
Clock type	Selezionare il tipo di orologio (analogico o digitale).
Show	Selezionare una o più unità di informazione (data, ora, fuso orario) da visualizzare nell'orologio digitale. La dimensione dell'unità si adatta automaticamente alle unità visualizzate, alla dimensione del widget e alla scala di visualizzazione. Questo parametro è disponibile se *Clock type* è impostato su "Digital".
Advanced configuration	Fare clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata per l'orologio digitale. Questa sezione è disponibile se *Clock type* è impostato su "Digital".

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*, e solo per quegli elementi selezionati nel campo *Mostra* (vedi sopra).

Inoltre, la configurazione avanzata consente di modificare il colore di sfondo dell'intero widget.

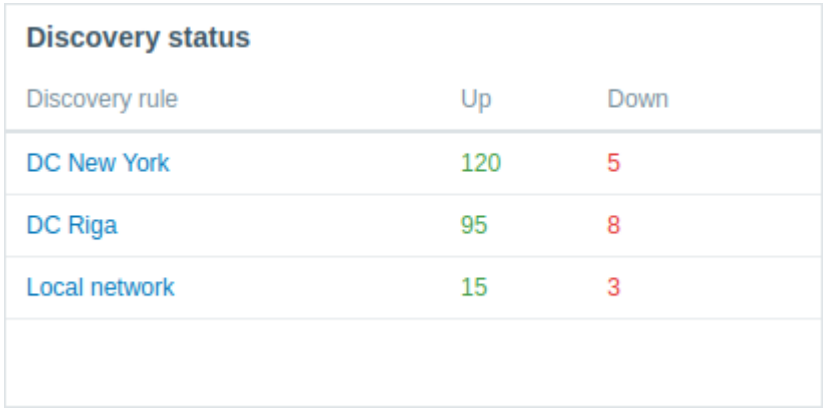


Colore di sfondo	Selezionare il colore di sfondo dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per tornare al valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Data	
Grassetto	Selezionare la casella di controllo per visualizzare la data in grassetto.
Colore	Selezionare il colore della data 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	
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.
Secondi	Selezionare la casella di controllo per visualizzare i secondi. In caso contrario verranno visualizzate solo ore e minuti.
Formato	Selezionare se visualizzare l'ora nel formato a 24 ore o a 12 ore.
Fuso orario	
Grassetto	Selezionare la casella di controllo per visualizzare il fuso orario in grassetto.
Colore	Selezionare il colore del fuso orario 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.
Fuso orario	Selezionare il fuso orario.
Formato	Selezionare se visualizzare il fuso orario nel formato breve (ad es. `New York`) o nel formato completo (ad es. `(UTC-04:00) America/New York`).

3 Stato del discovery

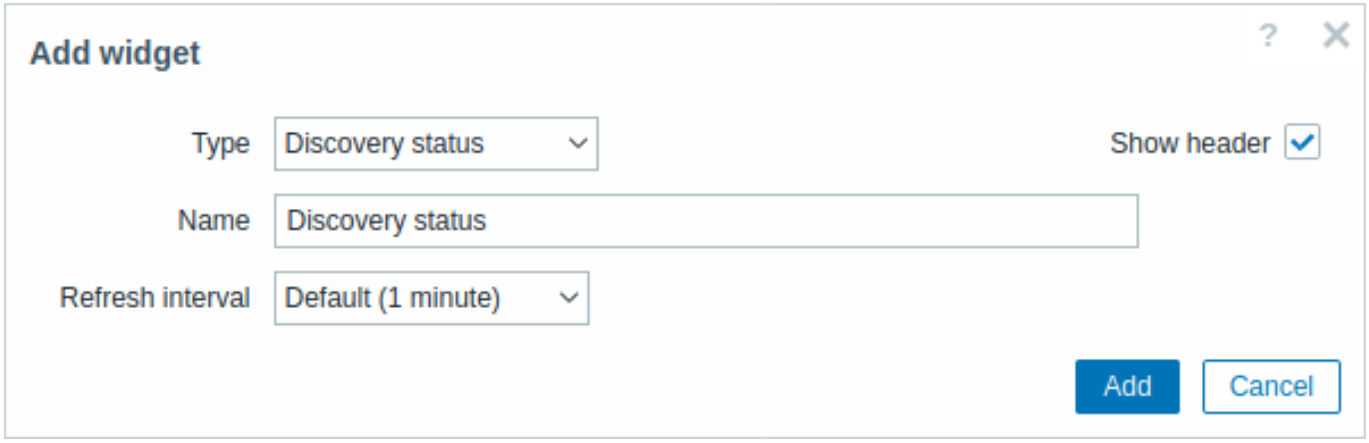
Panoramica

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

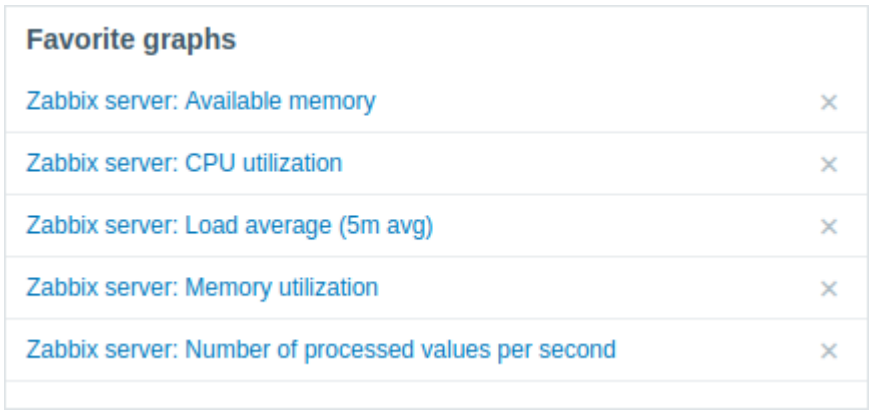
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



4 Grafici preferiti

Panoramica

Il widget *Grafici preferiti* visualizza un elenco alfabetico di collegamenti ai grafici contrassegnati come preferiti dall'utente corrente.



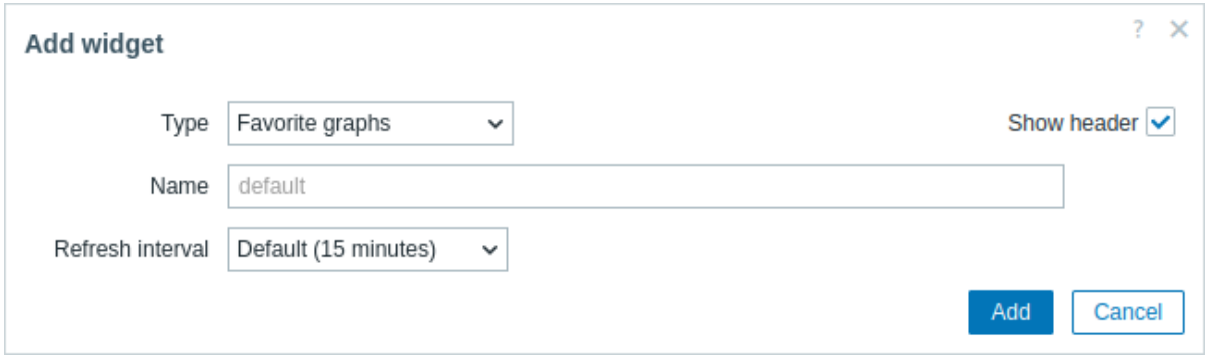
Facendo clic su x, il grafico viene rimosso dall'elenco e dai preferiti.

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

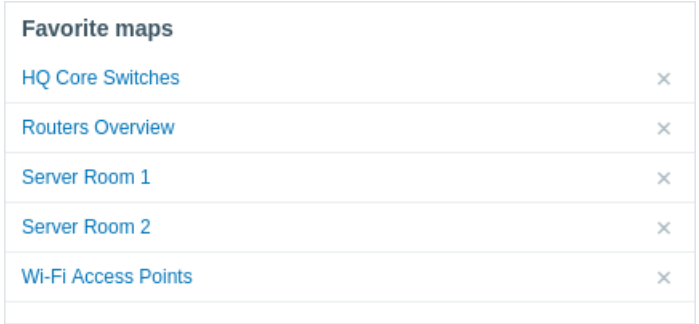
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



5 Mappe preferite

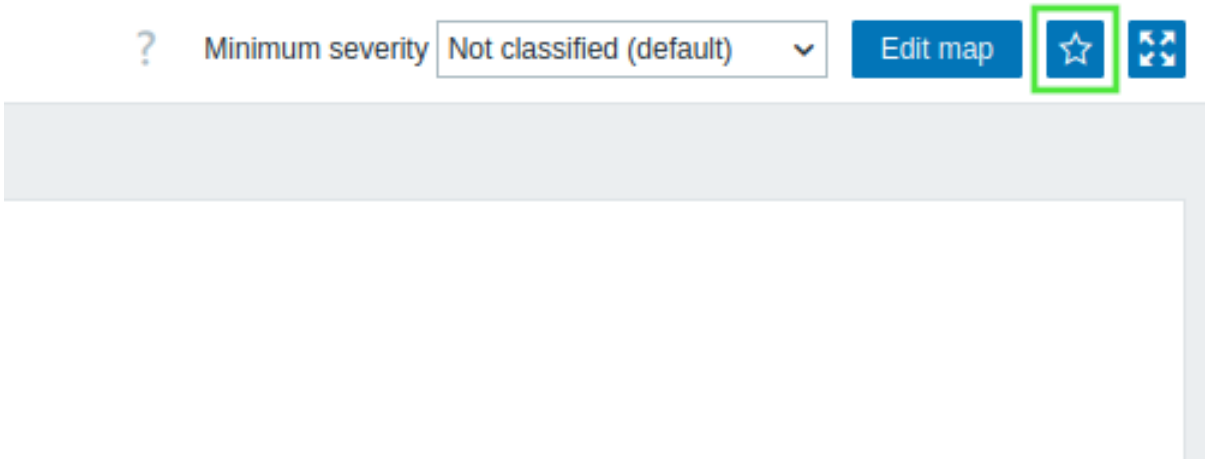
Panoramica

Il widget *Mappe preferite* visualizza un elenco di scorciatoie alle mappe contrassegnate come preferite dall'utente corrente, ordinate alfabeticamente.



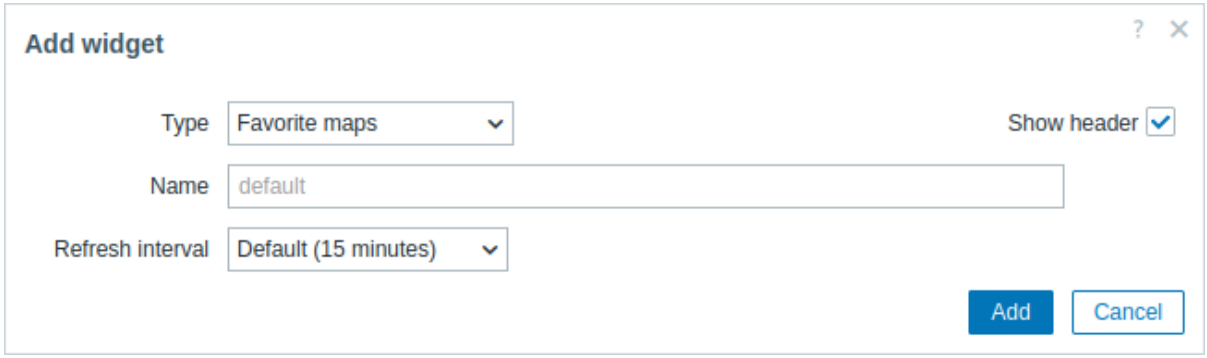
Facendo clic su x la mappa viene rimossa dai preferiti.

Le mappe vengono contrassegnate come preferite facendo clic sul pulsante  *Aggiungi ai preferiti* durante la visualizzazione delle mappe in *Monitoring* > *Maps*.



Configurazione

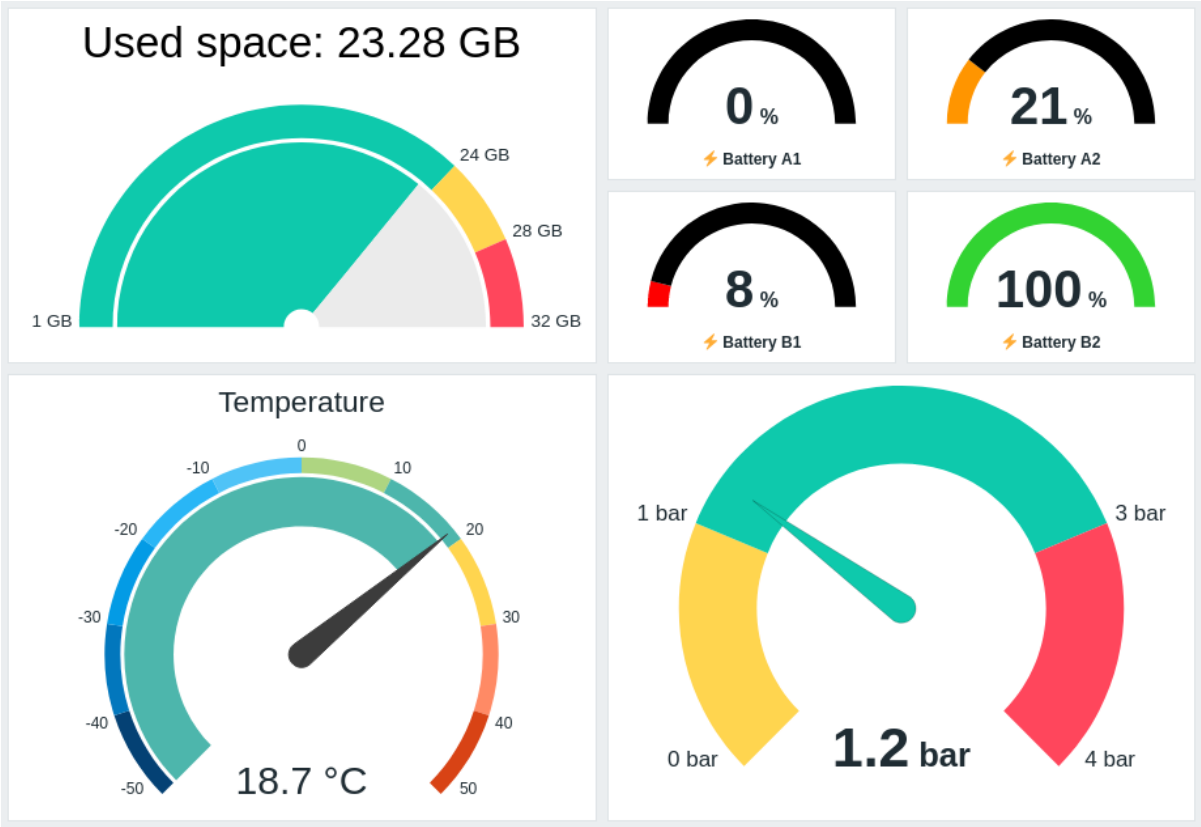
Tutti i parametri di configurazione per questo widget sono comuni a tutti i widget.



6 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.



Puoi configurare il widget per visualizzare quanto segue:

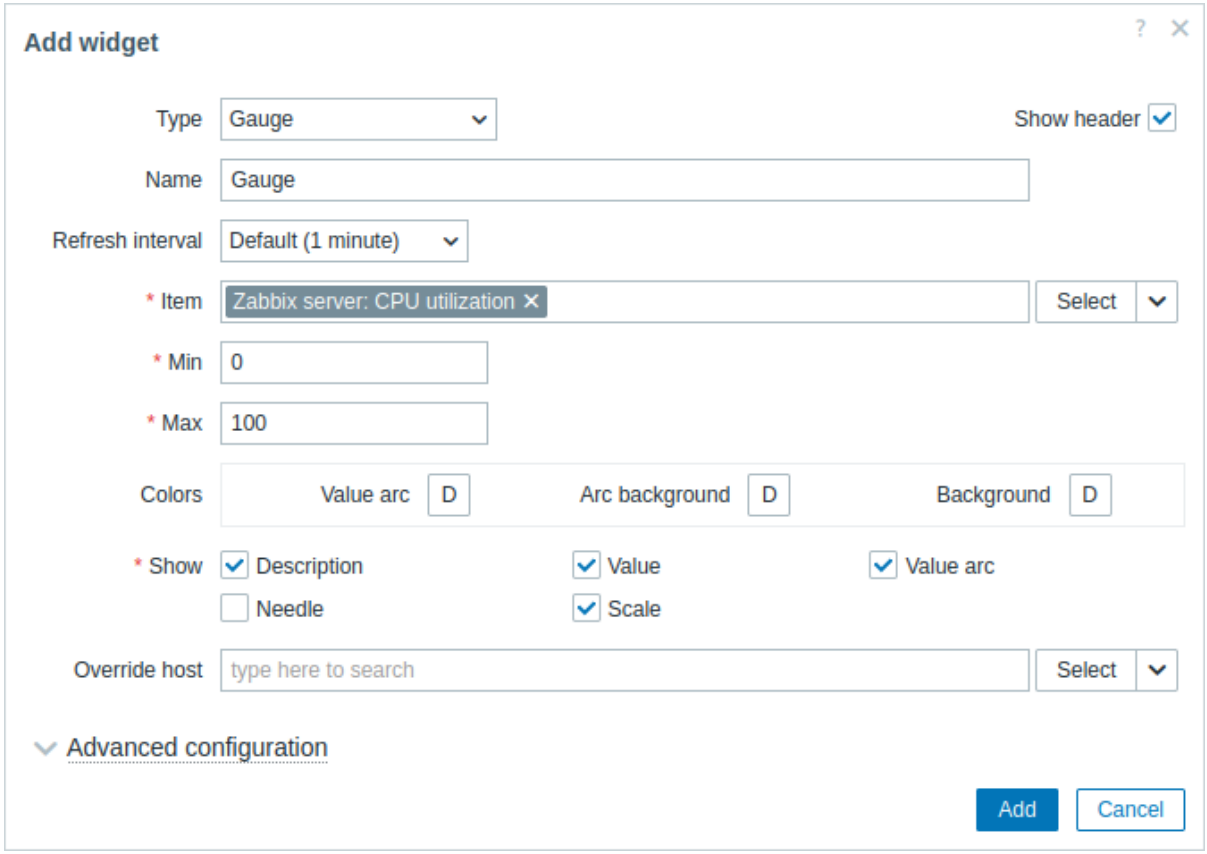
- Descrizione dell'item (Spazio utilizzato: {ITEM.LASTVALUE}, Temperatura)
- Valore dell'item e unità (21 %, 18.7 °C, 1.2 bar)
- Scala dell'indicatore (1GB/24GB/28GB/32GB, -50/-40/-30/ecc.)
- Arco dell'indicatore (arco del valore dell'indicatore e arco delle soglie dell'indicatore)
- Lancetta dell'indicatore

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 *Download image* nel menu del widget.

Configurazione

Per configurare il widget, seleziona *Gauge* come tipo:

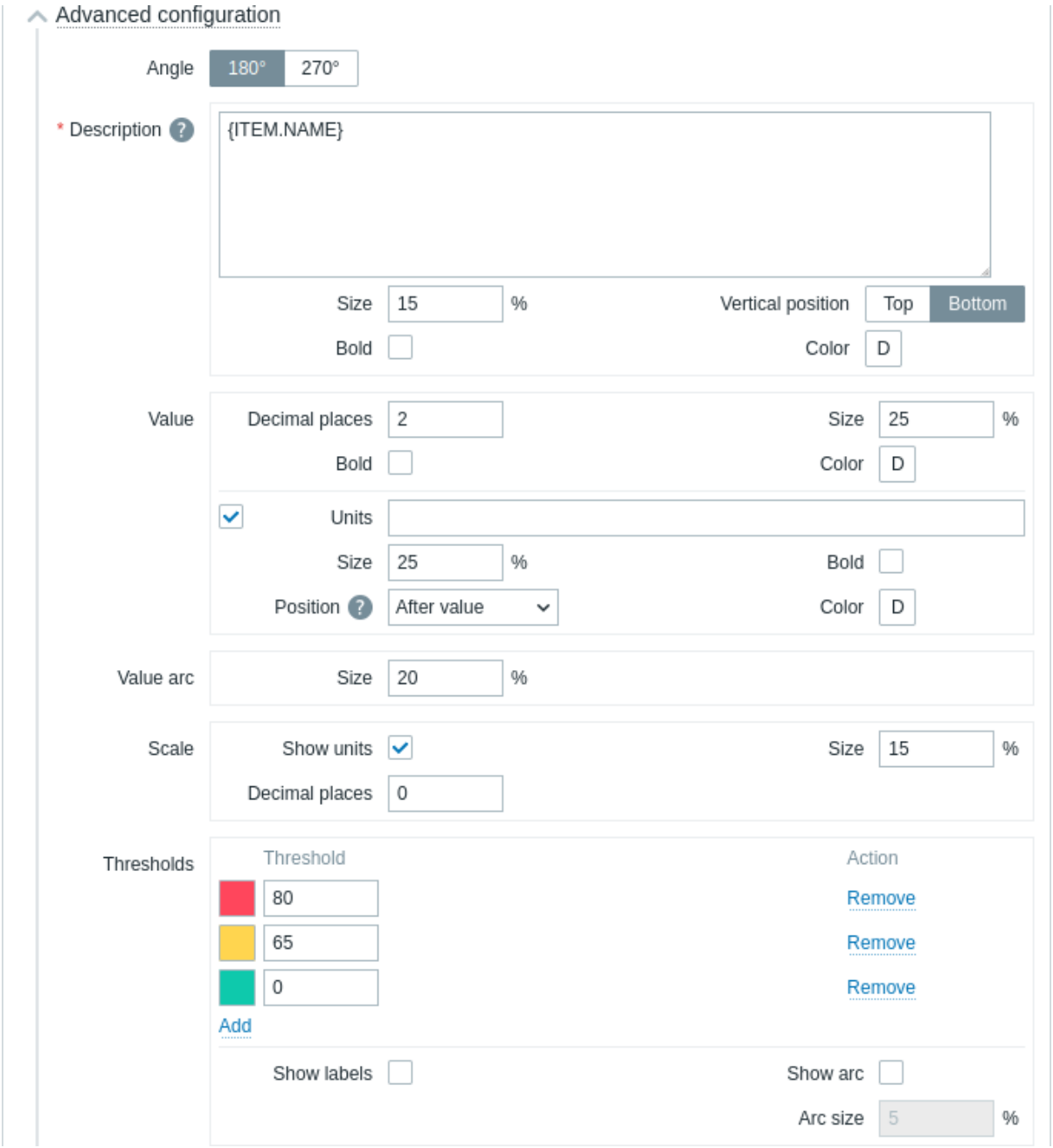


Oltre ai parametri comuni a tutti i widget, puoi impostare le seguenti opzioni specifiche:

Item	Seleziona l'item. In alternativa, seleziona un widget compatibile come origine dati per gli item. Questo campo supporta l'autocompletamento, quindi iniziando a digitare il nome di un item verrà mostrato un menu a discesa con gli item corrispondenti. Nota che puoi selezionare solo item che restituiscono dati numerici (eccetto i dati binari).
Min	Inserisci il valore minimo del gauge. Sono supportati i suffissi (ad esempio, "1d", "2w", "4K", "8G"). Sono supportate le mappature dei valori.
Max	Inserisci il valore massimo del gauge. Sono supportati i suffissi (ad esempio, "1d", "2w", "4K", "8G"). Sono supportate le mappature dei valori.
Colors	Seleziona il colore dal selettore colori: **Arco del valore** - seleziona il colore dell'arco del valore del gauge; **Sfondo dell'arco** - seleziona il colore di sfondo dell'arco del valore del gauge e dell'arco delle soglie del gauge; **Sfondo** - seleziona 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, fai clic sul pulsante *Use default* nel selettore colori.
Show	Seleziona 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). Deselezionala per nasconderlo. Deve essere selezionato almeno un elemento. Nota 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 (vedi le opzioni di configurazione avanzata). Nota 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	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 in una dashboard di template.
Advanced configuration	Fai clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata. Qui potrai anche regolare gli elementi del gauge selezionati nel campo *Show*.

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili nella sezione comprimibile *Configurazione avanzata*:



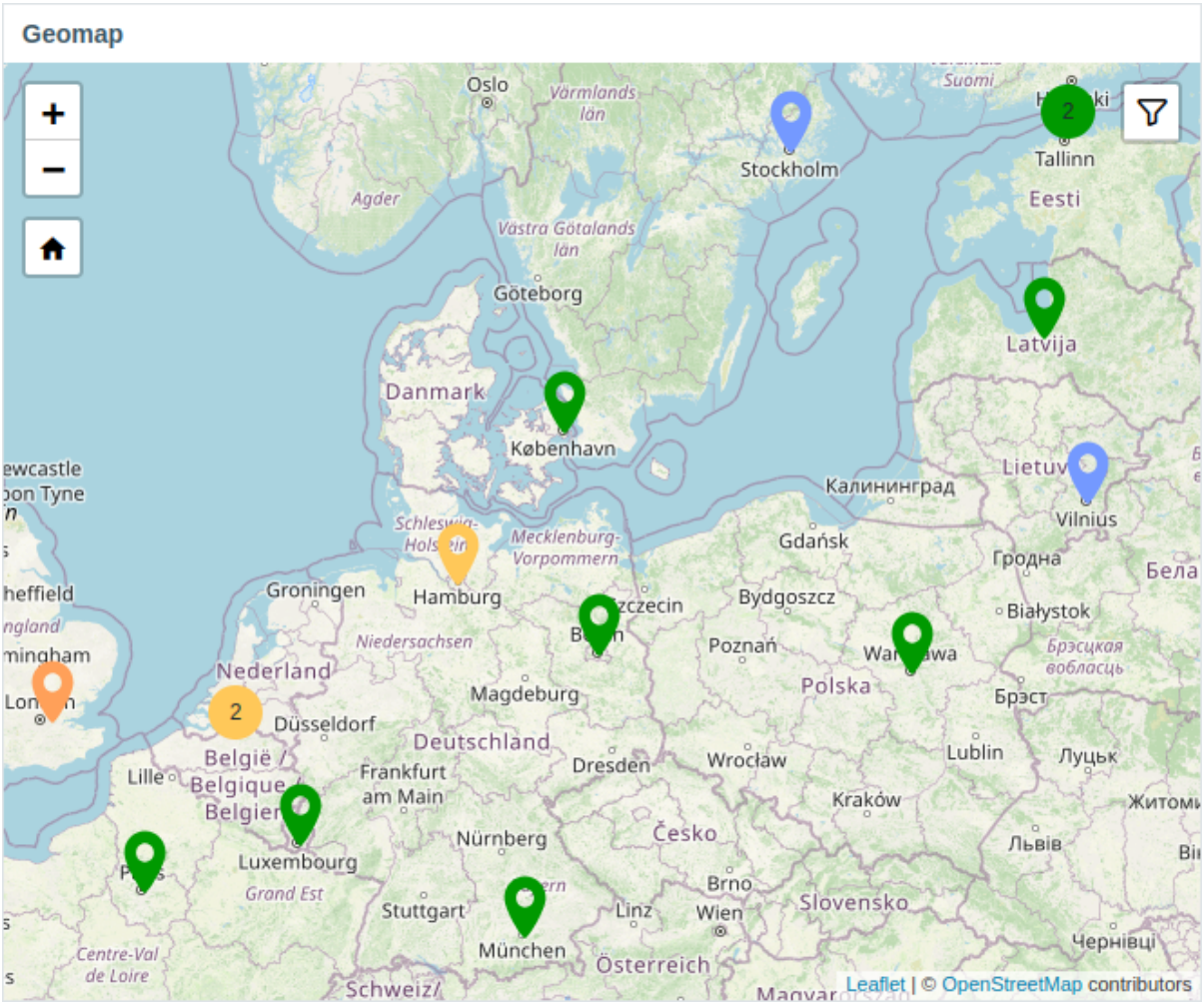
Angolo	Selezionare l'angolo dell'indicatore (180° o 270°).
Descrizione	
Descrizione	Inserire la descrizione dell'item. Questa descrizione può sovrascrivere il nome predefinito dell'item. Sono supportate descrizioni su più righe. È possibile una combinazione di testo e macro supportate. Sono supportate {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).

7 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.

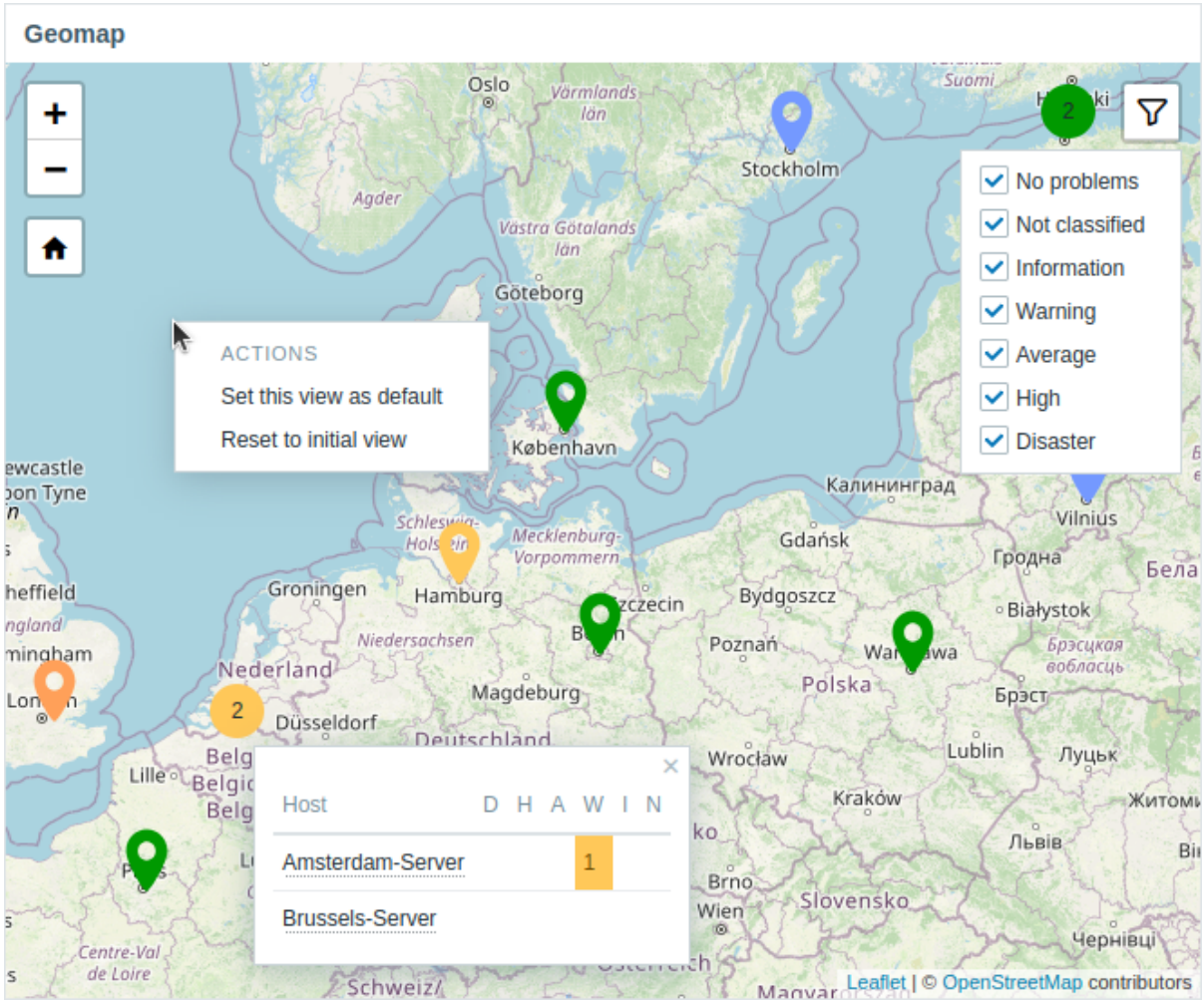


Note:
Zabbix offre diversi provider di tile della mappa predefiniti e consente di aggiungerne di personalizzati in *Administration* > *General* > *Geographical maps*.

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.

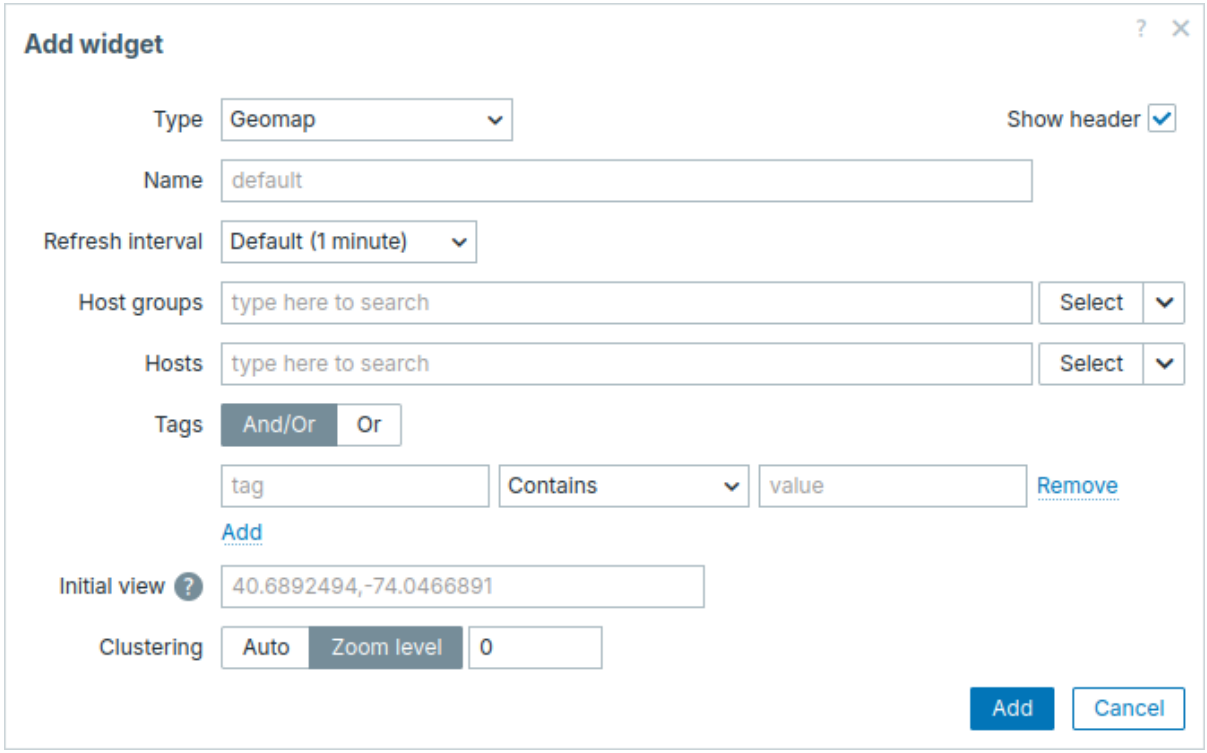
Utilizzare le seguenti opzioni per interagire con il widget:

- 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.



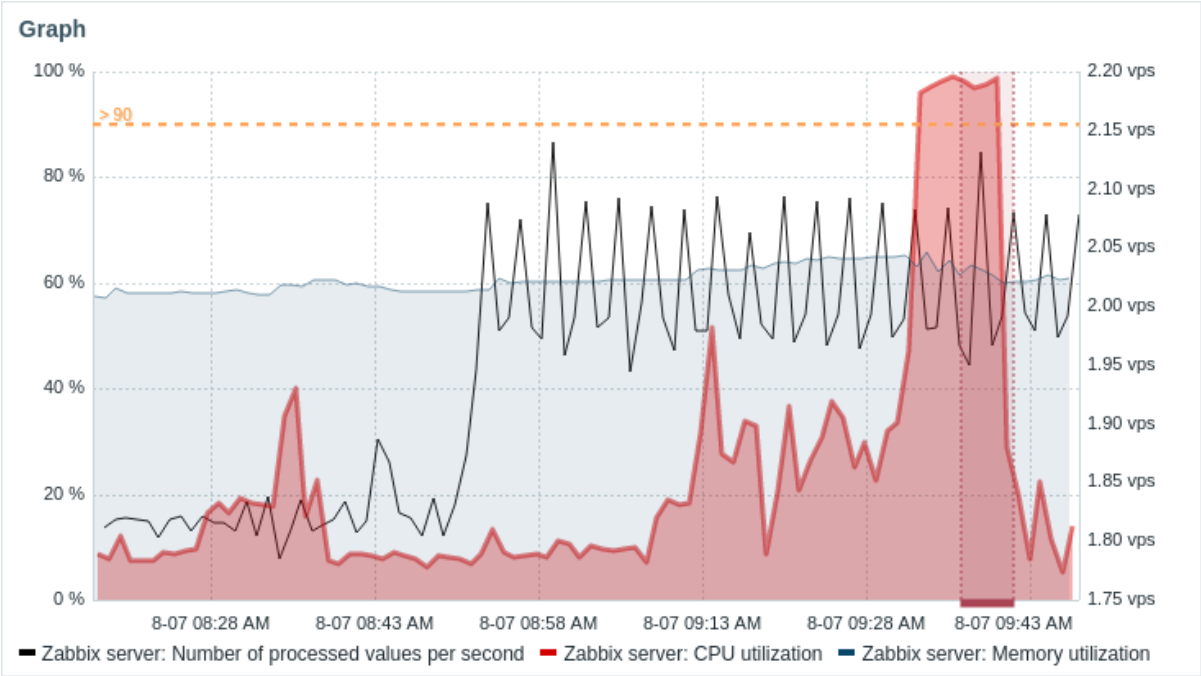
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Gruppi host	Selezionare i gruppi host da visualizzare sulla mappa. 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 elenco a discesa con i gruppi corrispondenti. Se non viene selezionato nulla nei campi *Gruppi host* e *Host*, verranno visualizzati tutti gli host con coordinate valide. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Host	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 *Gruppi host* e *Host*, verranno visualizzati tutti gli host con coordinate valide. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Tag	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 più 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). 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. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Vista iniziale	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 livello di zoom massimo 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 singolo marcatore con un conteggio: **Auto** - i marcatori vengono combinati automaticamente; **Livello di zoom** - 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.

8 Grafico

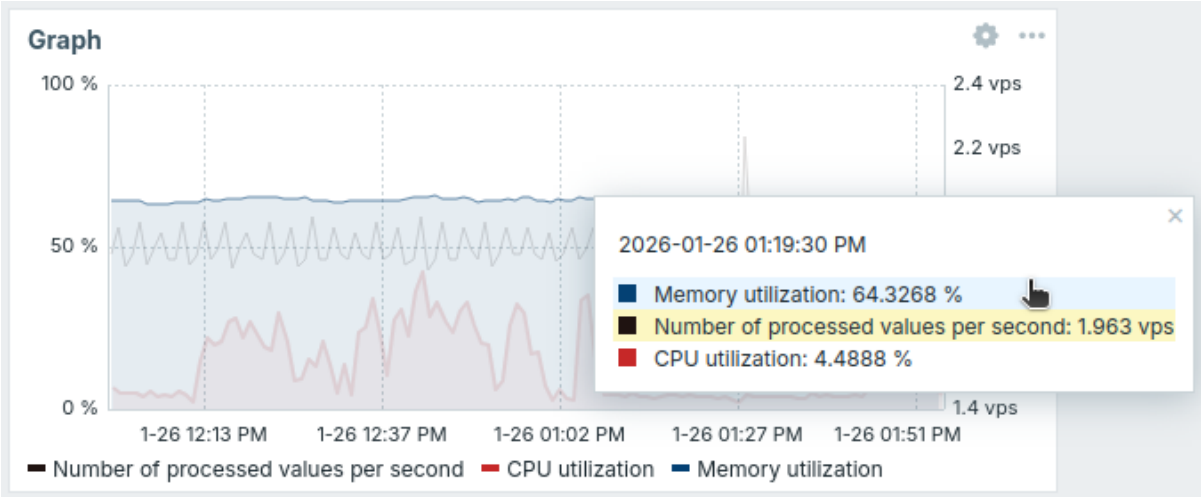
Panoramica

Il widget *Graph* visualizza i dati numerici degli item come un grafico vettoriale. Può aiutarti a monitorare le tendenze, individuare problemi e confrontare i valori nel tempo e tra host diversi.

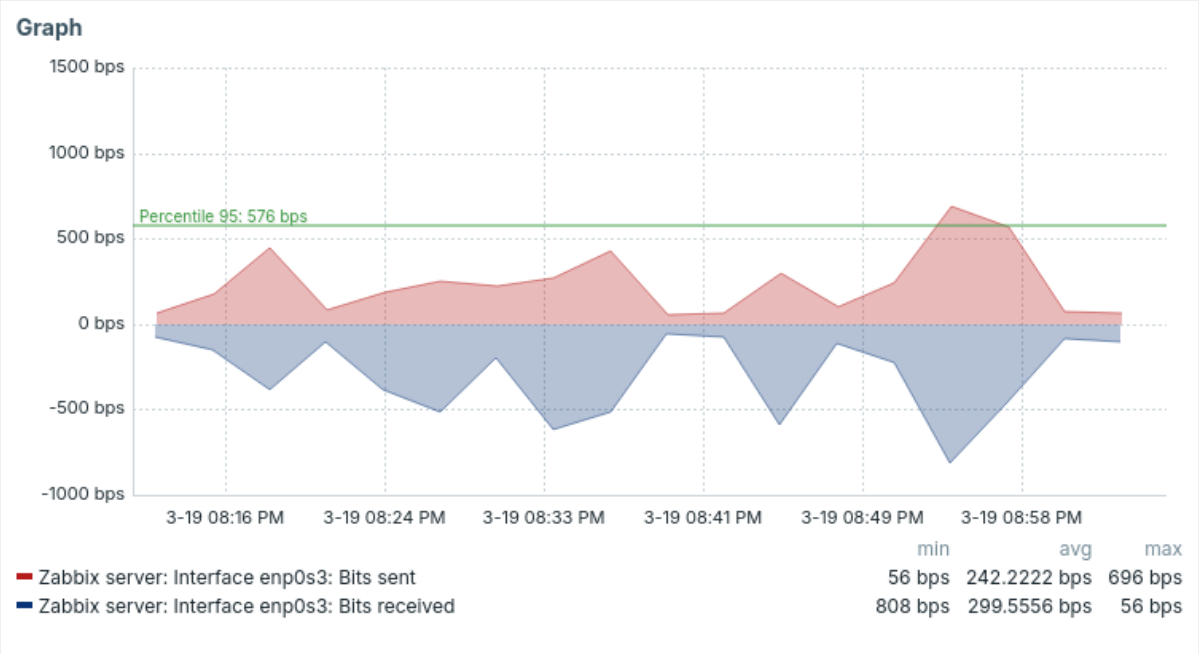


Passando il cursore sul grafico viene visualizzato un tooltip con un elenco di item ordinati per valore in ordine decrescente.

Passando il cursore su un item nel tooltip, il relativo grafico viene evidenziato (gli altri vengono attenuati); selezionando l'item, i suoi dati vengono trasmessi ad 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.

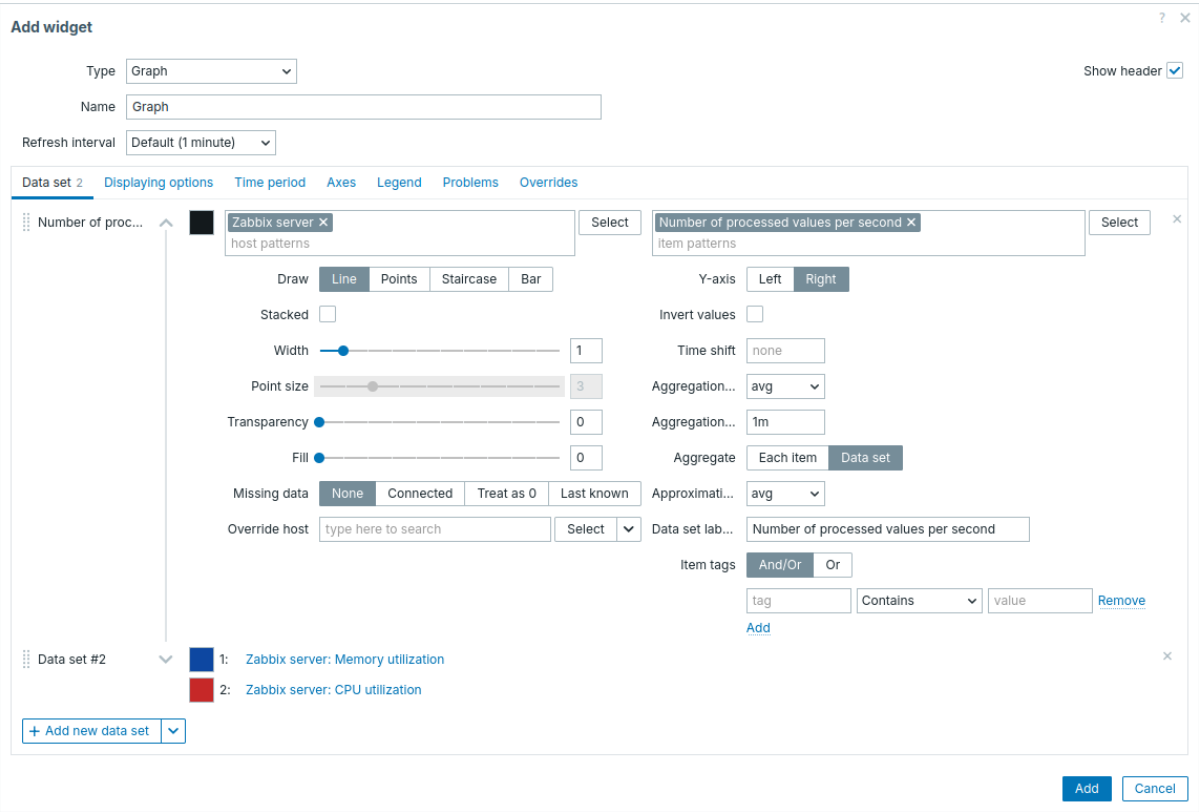


È disponibile anche il widget legacy Graph (classic).

Le informazioni visualizzate nel widget *Graph* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu del widget.

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 palette per assegnare colori distinti a ciascun item corrispondente.
- *Elenco di item* - vengono visualizzati i dati degli item selezionati. È possibile scegliere singolarmente il colore di ciascun 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 inserire i modelli di host e item; i dati degli item che corrispondono a questi modelli verranno visualizzati nel grafico; possono essere visualizzati fino a 50 item. Per la selezione possono essere utilizzati modelli con caratteri jolly (ad esempio, ∗ restituirà risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, inserire manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, quindi non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se esistono 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 di 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 di configurazione del set di dati. Si noti 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 di selezionare solo gli item configurati nel template.
Disegno	Scegliere il tipo di disegno della metrica. Tipi di disegno possibili: *Linea* (impostato per impostazione predefinita), *Punti*, *Scala* e *Barra*. Si noti che se nel grafico linea/scala 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 per visualizzare i dati come impilati (vengono visualizzate 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 *Scala*.
Dimensione 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 *Scala*.
Dati mancanti	Selezionare l'opzione per visualizzare i dati mancanti: **Nessuno** - l'intervallo viene lasciato vuoto; **Connesso** - i due valori ai bordi vengono collegati; **Tratta come 0** - i dati mancanti vengono visualizzati come valori 0; **Ultimo noto** - i dati mancanti vengono visualizzati con lo stesso valore dell'ultimo valore noto; non applicabile al tipo di disegno *Punti* e *Barra*.
Sostituisci 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.
Asse Y	Selezionare il lato del grafico in cui verrà visualizzato l'asse Y.
Inverti valori	Selezionare la casella per moltiplicare tutti i valori dell'asse Y per -1 prima di visualizzarli nel grafico. Ciò influisce solo sulla rappresentazione visiva; i dati originali rimangono invariati. Questa opzione non influisce sul tooltip, sulla legenda o sulla linea del percentile. Se per l'asse è selezionata l'opzione *Trigger semplici*, le soglie del trigger verranno visualizzate invertite.
Spostamento temporale	Specificare uno spostamento temporale, se necessario. In questo campo è possibile utilizzare i suffissi temporali. Sono consentiti valori negativi.

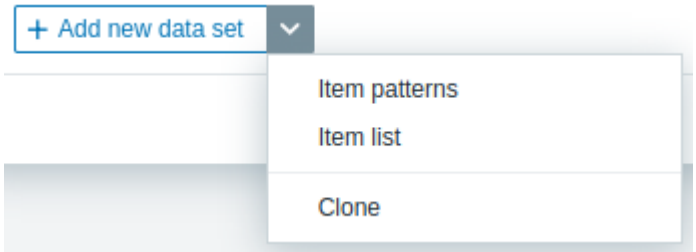
Funzione di aggregazione	Specificare quale funzione di aggregazione utilizzare: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **sum** - visualizza la somma dei valori; **count** - visualizza il conteggio dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore; **none** - visualizza tutti i valori (nessuna aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo scelto (5 minuti, un'ora, un giorno), invece di tutti i valori. 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 verrà considerato in secondi. Si noti che se il widget è configurato per visualizzare dati storici basati su trend (*Selezione dati storici* è impostata su *Trend* o *Auto*), si consiglia di utilizzare un intervallo di aggregazione multiplo di 1 ora (ad esempio, 3600, 60m, 1h, 3h, ecc.). I trend memorizzano valori aggregati orari, quindi l'utilizzo di un intervallo di aggregazione che non sia multiplo di 1 ora (ad esempio, 100s, 7min, 15min, 90min, ecc.) può portare a risultati difficili da interpretare.
Aggrega	Specificare se aggregare: **Ogni item** - ogni item nel set di dati verrà aggregato e visualizzato separatamente; **Set di dati** - tutti gli item del set di dati verranno aggregati e visualizzati come un unico valore.
Approssimazione	Specificare quale valore visualizzare quando esiste più di un valore per pixel verticale del grafico: **all** - visualizza il valore più piccolo, il più grande e il valore medio; **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio. Questa impostazione è utile quando si visualizza un grafico per un lungo periodo di tempo con un intervallo di aggiornamento frequente (ad esempio un anno di valori raccolti ogni 10 minuti).
Etichetta set di dati	Specificare l'etichetta del set di dati visualizzata nella configurazione *Set di dati* del grafico e nella *Legenda* del grafico (per i set di dati aggregati). Tutti i set di dati sono numerati, compresi quelli con un'*Etichetta set di dati* specificata. Se non viene specificata alcuna etichetta, il set di dati verrà etichettato automaticamente in base al suo numero (ad esempio "Set di dati #2", "Set di dati #3", ecc.). La numerazione dei set di dati viene ricalcolata dopo il riordinamento/trascinamento dei set di dati. Le etichette dei set di dati troppo lunghe verranno abbreviate per adattarsi allo spazio disponibile (ad esempio "Numero di proc...").

Tag item	Specificare i tag per filtrare gli 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 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). 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.

Dettagli della configurazione del set di dati

I set di dati esistenti sono 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 espanso, 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 usato come base per le sfumature generate in *Item patterns*. Usare 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 pattern di host e item. Se si fa clic sull'icona con la freccia 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.

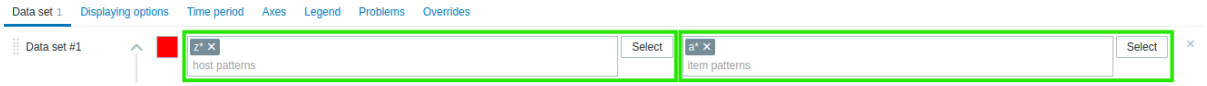


Il set di dati **Pattern degli item** contiene i campi *Pattern degli host* e *Pattern degli item*, entrambi in grado di riconoscere nomi completi o pattern contenenti un simbolo jolly (*). Questa funzionalità consente di selezionare tutti i nomi host e i nomi degli item che contengono il pattern selezionato. 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.

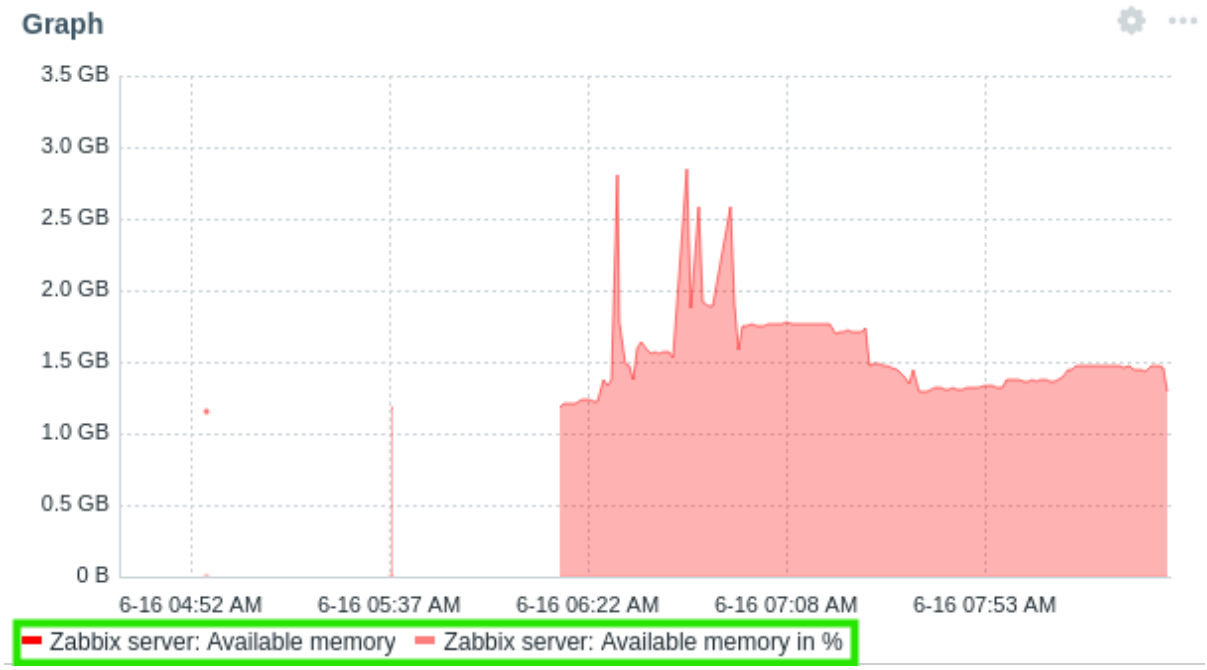
Ad esempio, dopo aver digitato il pattern **z*** nel campo *Pattern degli host*, l'elenco a discesa mostra tutti i nomi host che contengono questo pattern: **z*, Zabbix server, Zabbix proxy**. Dopo aver premuto *Invio*, questo pattern viene accettato e visualizzato come **z***. Allo stesso modo, dopo 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 %**.



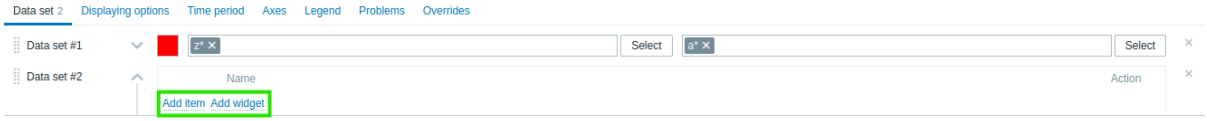
Dopo aver premuto *Invio*, il pattern viene accettato e mostrato come **a***.



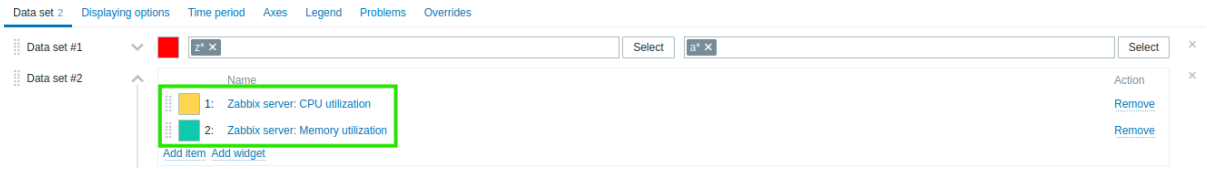
Il grafico mostra quindi tutti gli item che appartengono ai nomi host selezionati.



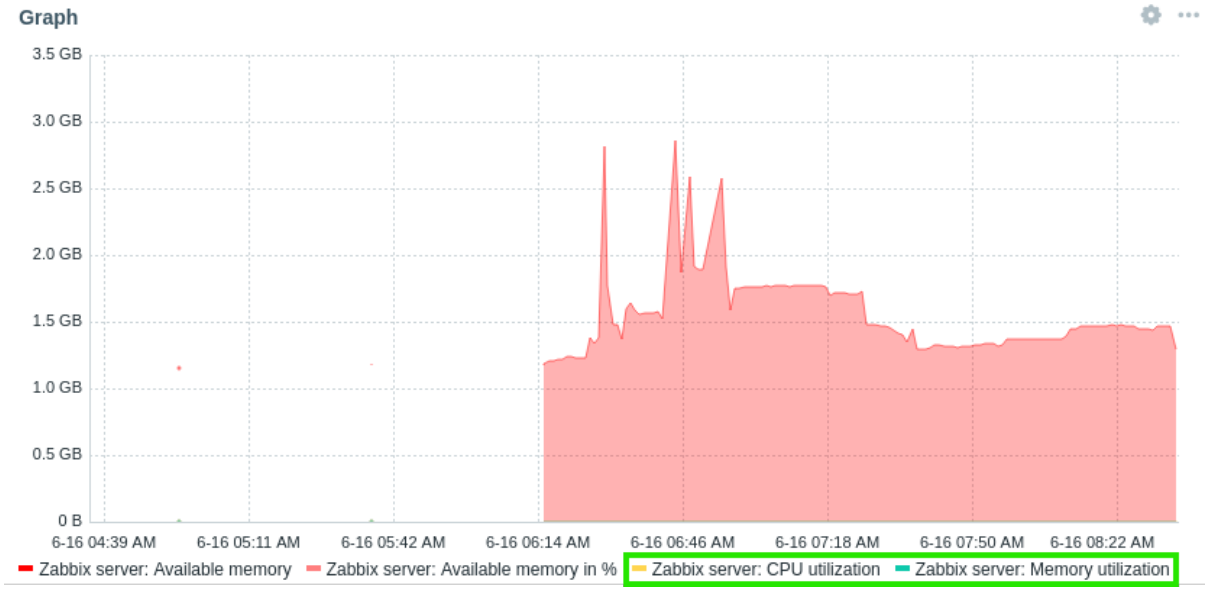
Il set di dati **Elenco item** contiene il pulsante *Aggiungi item*, che consente di aggiungere item da visualizzare nel grafico. È inoltre possibile aggiungere widget compatibili come origine dati per gli item facendo clic sul pulsante *Aggiungi widget*.



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.

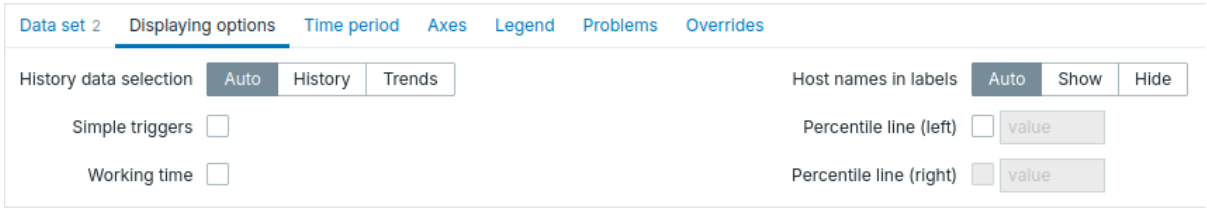


Dopo aver selezionato uno o più item, questi verranno visualizzati nell'elenco degli item del set di dati e nel grafico.



Opzioni di visualizzazione

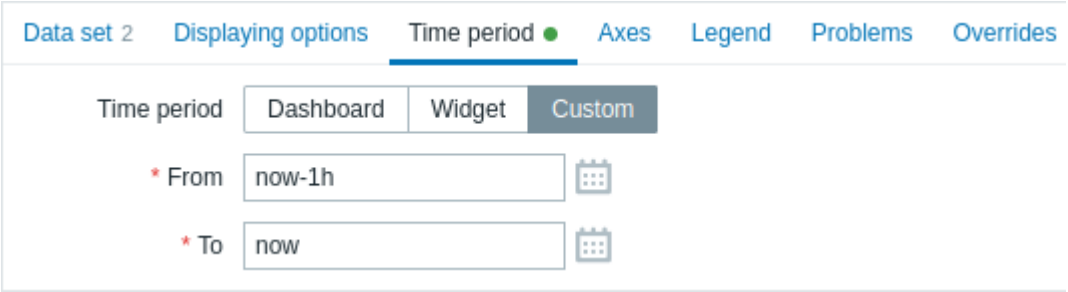
La scheda **Opzioni di visualizzazione** consente di definire la selezione dei dati dello storico:



Selezione dei dati dello storico	Imposta la sorgente dei dati del grafico: **Auto** - i dati vengono recuperati in base al classico algoritmo del grafico (predefinito); **Storico** - dati dallo storico; **Trend** - dati dai trend.
Trigger semplici	Selezionare la casella per mostrare le soglie dei trigger semplici. Le soglie verranno disegnate come linee tratteggiate utilizzando il colore di gravità del trigger. Un trigger semplice è un trigger con una funzione (solo `last`, `max`, `min`, `avg`) per un item nell'espressione. È possibile disegnare un massimo di tre trigger. Si noti che il trigger deve trovarsi all'interno dell'intervallo disegnato per essere visibile.
Orario di lavoro	Selezionare la casella per mostrare l'orario di lavoro nel grafico. 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*).
Nomi degli host nelle etichette	Scegliere se visualizzare i nomi degli host nella legenda del grafico e nel tooltip che appare passando il cursore sui dati del grafico: **Auto** - i nomi degli host vengono visualizzati solo se nei set di dati è presente più di un host (predefinito). **Mostra** - i nomi degli host vengono visualizzati. **Nascondi** - i nomi degli host vengono nascosti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Linea del percentile (sinistra)	Selezionare la casella e inserire il valore del percentile per mostrare il percentile specificato come linea sull'asse Y sinistro del grafico. Se, ad esempio, è impostato un percentile del 95%, la linea del percentile si troverà al livello sotto il quale ricade il 95 percento dei valori.
Linea del percentile (destra)	Selezionare la casella e inserire il valore del percentile per mostrare il percentile specificato come linea sull'asse Y destro del grafico. Se, ad esempio, è impostato un percentile del 95%, la linea del percentile si troverà al livello sotto il quale ricade il 95 percento dei valori.

Periodo di tempo

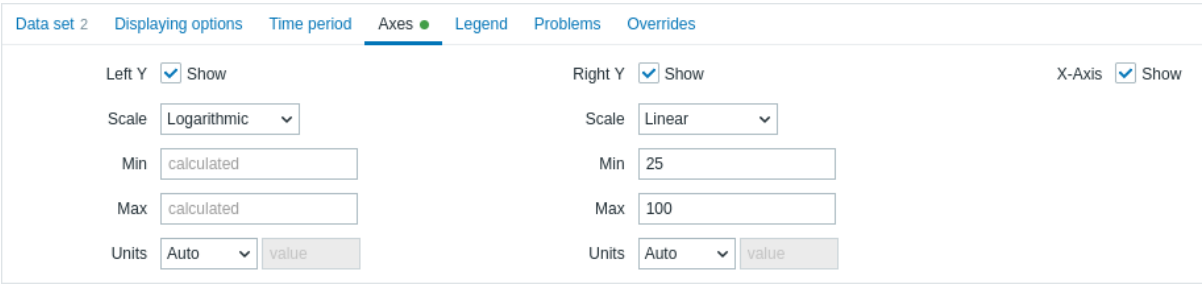
La scheda **Periodo di tempo** consente di impostare un periodo di tempo per il quale visualizzare i dati nel grafico:



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	Inserire 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	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

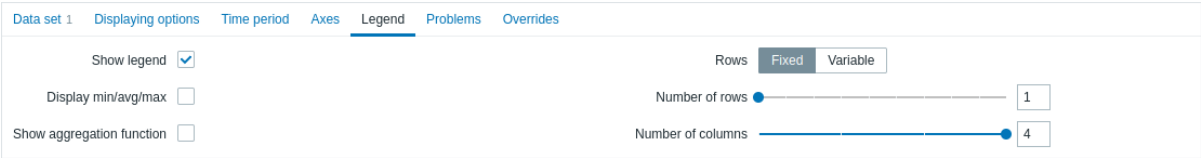
La scheda **Assi** consente di personalizzare la visualizzazione degli assi:



Y sinistro	Selezionare questa casella per rendere visibile l'asse Y sinistro. La casella può essere disabilitata se non è selezionata né nella scheda *Set di dati* né nella scheda *Sostituzioni*.
Y destro	Selezionare questa casella per rendere visibile l'asse Y destro. La casella può essere disabilitata se non è selezionata né nella scheda *Set di dati* né nella scheda *Sostituzioni*.
Asse X	Deselezionare questa casella per nascondere l'asse X (selezionata per impostazione predefinita).
Scala	Scegliere la scala per i valori dell'asse del grafico dal menu a discesa: **Lineare** - i valori dell'asse aumentano di una quantità fissa (ad esempio 10, 20, 30), adatta a 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 a 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 dell'asse del grafico dal menu a discesa: **Auto** - i valori dell'asse vengono visualizzati utilizzando l'unità del primo item nel set di dati; **Statica** - i valori dell'asse vengono visualizzati utilizzando l'unità specificata nel campo di input *valore*; se il campo viene lasciato vuoto, vengono visualizzati solo valori numerici.

Legenda

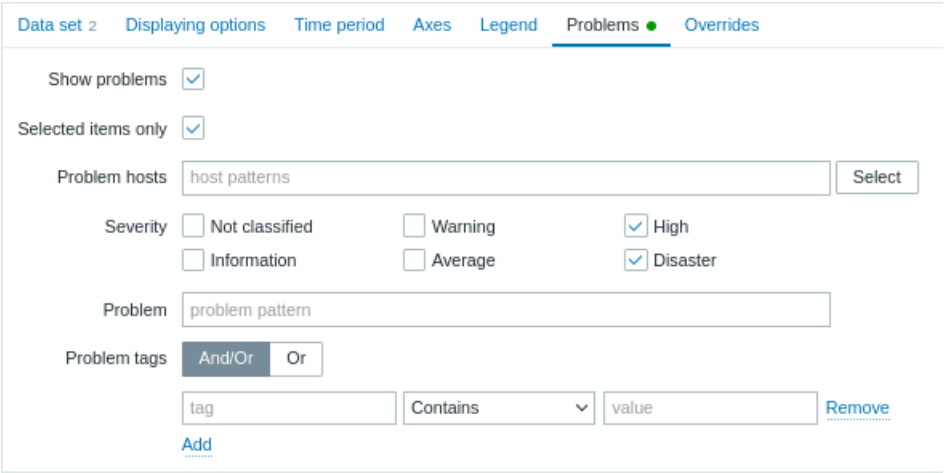
La scheda **Legenda** consente di personalizzare la legenda del grafico:



Mostra legenda	Deseleziona questa casella di controllo per nascondere la legenda nel grafico (selezionata per impostazione predefinita).
Visualizza min/media/max	Seleziona questa casella di controllo per visualizzare nella legenda i valori minimo, medio e massimo dell'item.
Mostra funzione di aggregazione	Seleziona questa casella di controllo per mostrare nella legenda la funzione di aggregazione.
Righe	Seleziona la modalità di visualizzazione per le 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", imposta il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su "Variabile", imposta il numero massimo di righe della legenda da visualizzare (1-10).
Numero di colonne	Imposta 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:

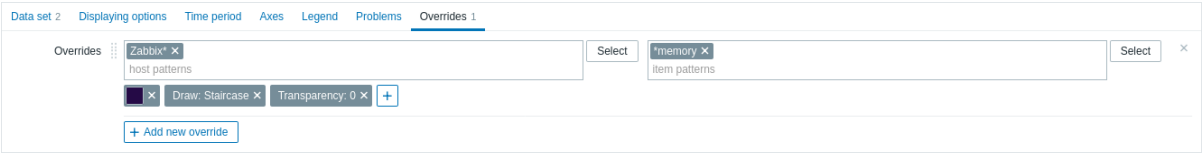


Mostra problemi	Selezionare questa casella per abilitare la visualizzazione dei problemi nel grafico (non selezionata, cioè disabilitata per impostazione predefinita).
Solo item selezionati	Selezionare questa casella per includere nel grafico solo i problemi relativi agli item selezionati.
Host dei problemi	Selezionare gli host dei problemi da visualizzare nel grafico. È possibile utilizzare modelli con caratteri jolly (ad esempio, * restituirà risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, inserire semplicemente la stringa manualmente e premere *Invio*. Durante la digitazione, notare come tutti gli host corrispondenti vengano visualizzati nel menu a discesa. Questo parametro non è disponibile durante la configurazione del 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 del problema	Specificare i tag del problema per limitare il numero di problemi 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. Sono disponibili diversi operatori per ogni condizione: **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 del tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una sola condizione.

Sostituzioni

La scheda **Sostituzioni** consente di aggiungere sostituzioni personalizzate per i set di dati:



Le sostituzioni sono utili quando vengono selezionati più 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).

Le sostituzioni esistenti (se presenti) vengono visualizzate in un elenco. Per aggiungere una nuova sostituzione:

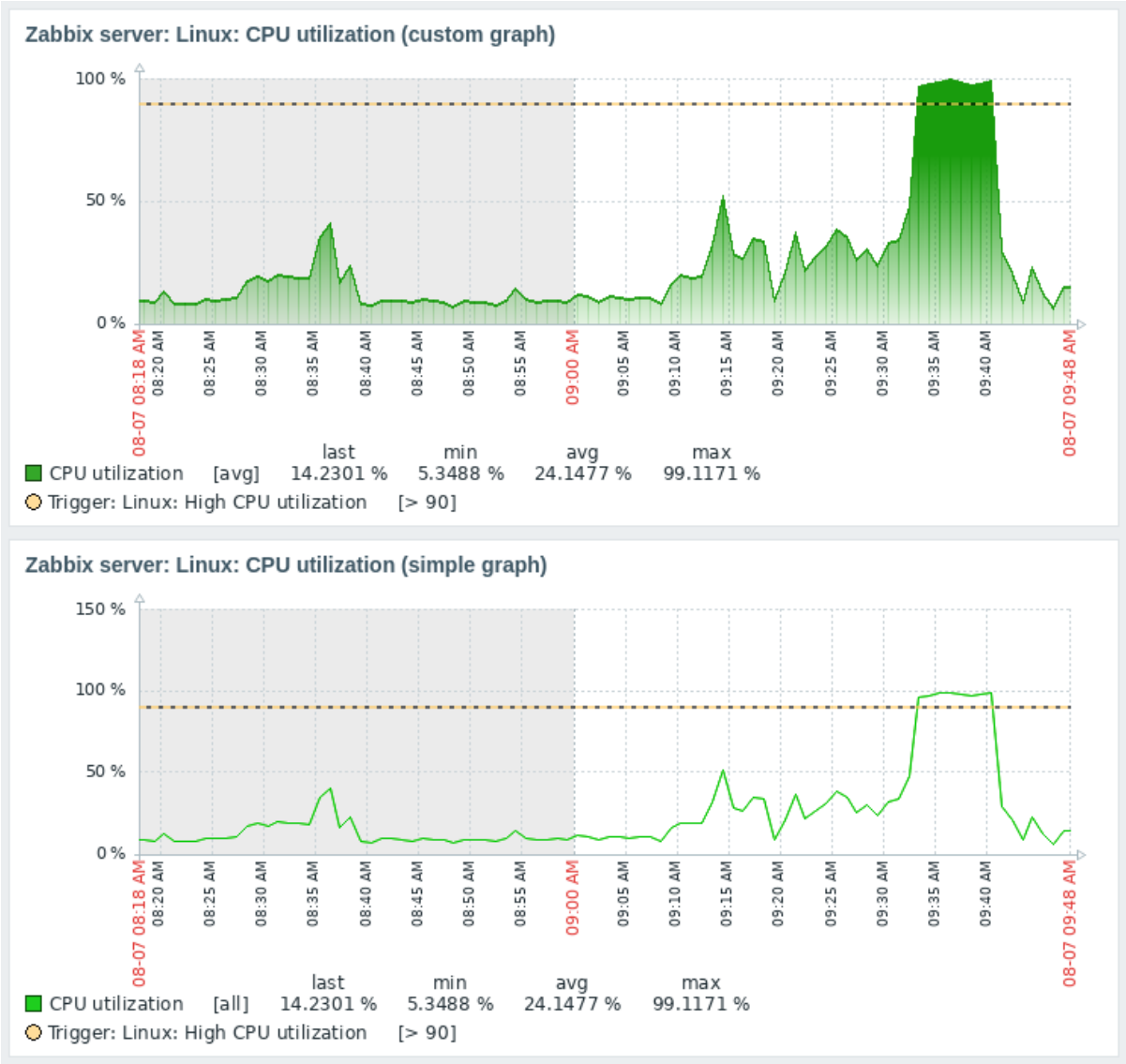
- Fare clic sul pulsante 

- Selezionare host e item per la sostituzione. In alternativa, è possibile inserire modelli di host e item. È possibile utilizzare modelli con caratteri jolly (ad esempio, ∗ restituirà risultati che corrispondono a zero o più caratteri). Per specificare un modello con caratteri jolly, è sufficiente inserire manualmente la stringa e premere *Invio*. Durante la digitazione, notare come tutti gli host corrispondenti vengano visualizzati nel menu a discesa. 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). I parametri del modello host e del modello item sono obbligatori. Il parametro per specificare i modelli host non è disponibile durante la configurazione del widget in una dashboard di template. Il parametro per specificare un elenco di item consente di selezionare solo gli item configurati nel template durante la configurazione del widget in una dashboard di template.
- Fare clic su  per selezionare i parametri della sostituzione. Deve essere selezionato almeno un parametro di sostituzione. Per la descrizione dei parametri, vedere la scheda *Set di dati* sopra.

9 Grafico (classico)

Panoramica

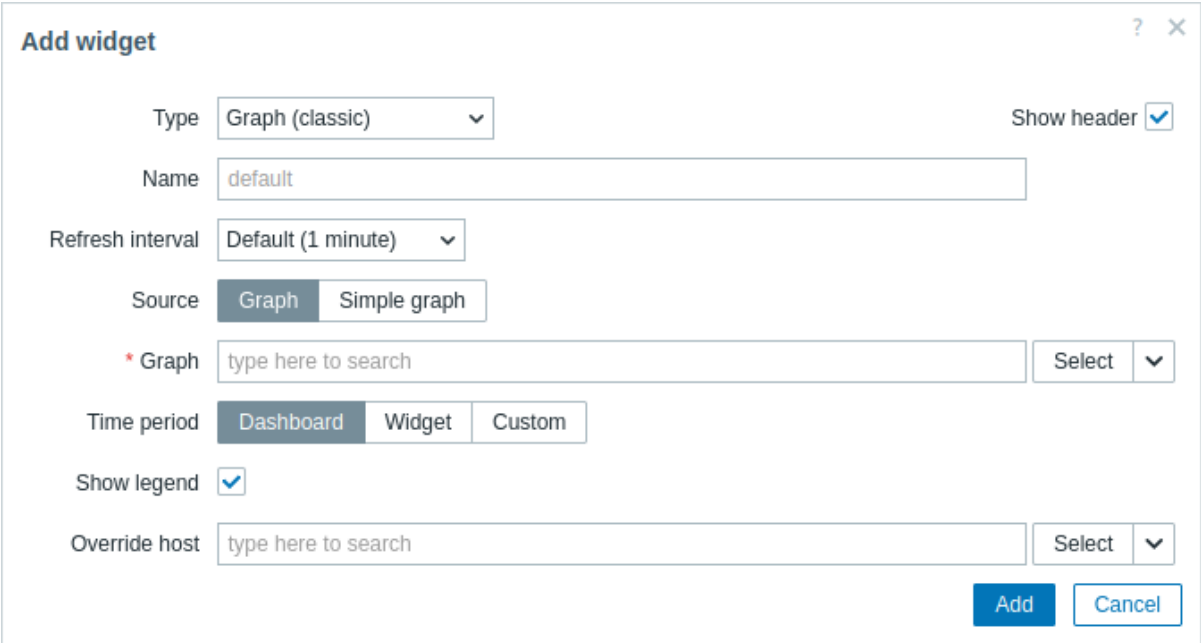
Il widget *Graph (classic)* visualizza i dati numerici degli item come un’immagine basata su un grafico personalizzato o un grafico semplice. Supporta la selezione dell’intervallo di tempo, le linee dei trigger e la visualizzazione facoltativa della legenda.



Le informazioni visualizzate nel widget *Graph (classic)* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu del widget.

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)

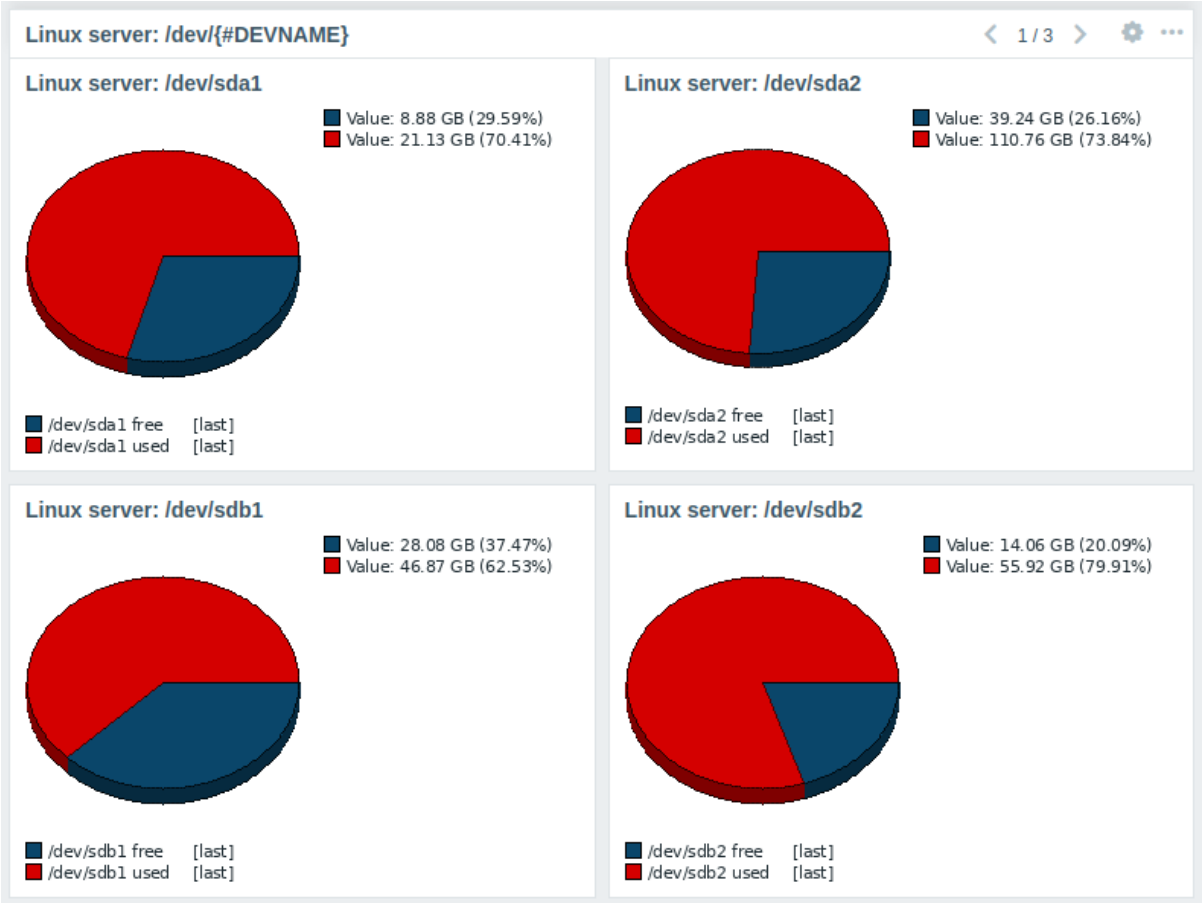
Se l’altezza del grafico all’interno del widget è insufficiente, la legenda potrebbe non essere visualizzata oppure essere visualizzata solo parzialmente. I trigger e i percentili vengono nascosti per primi, seguiti dalla legenda degli item. Per visualizzare la legenda completa, aumentare la dimensione 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.

10 Prototipo di grafico

Panoramica

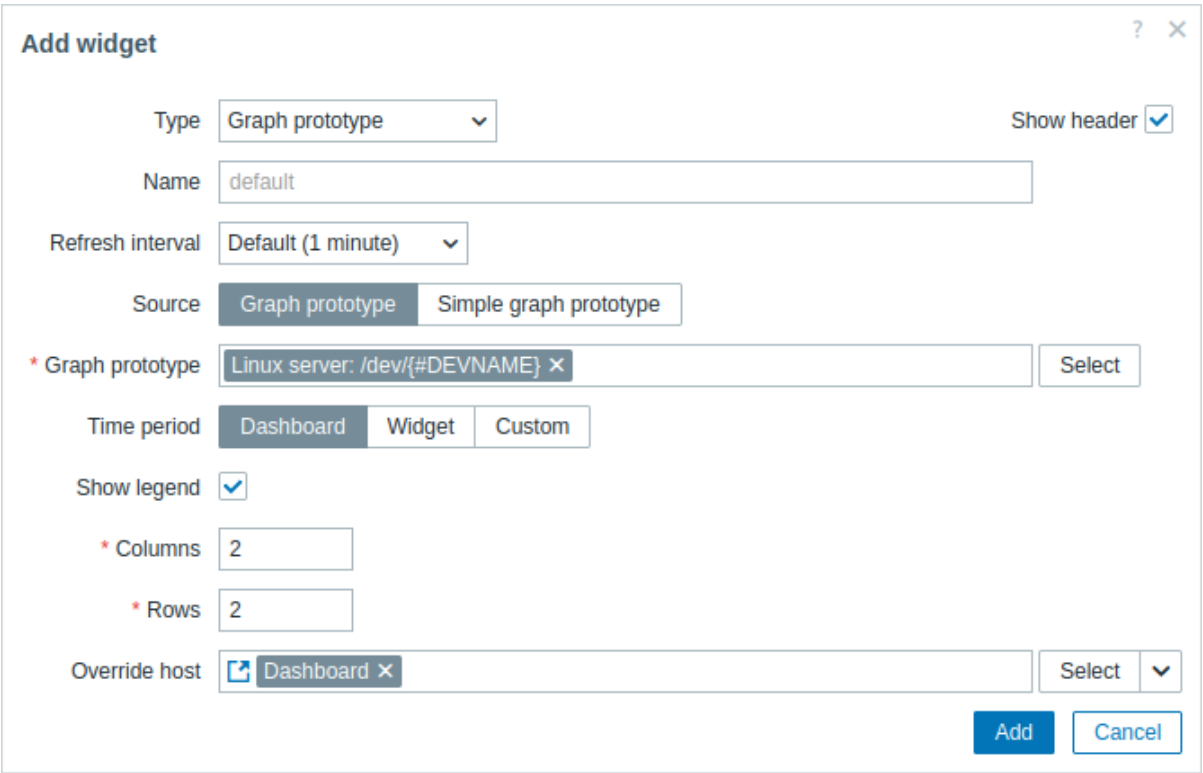
Il widget *Prototipo di grafico* visualizza una griglia di grafici personalizzati creati automaticamente in base ai prototipi di grafico o ai prototipi di item della regola di low-level discovery.



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 la sorgente dei grafici: **Graph prototype** oppure **Simple graph prototype**.
Graph prototype	Selezionare un prototipo di grafico per visualizzare i grafici individuati 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 individuati dal prototipo di item. Questo parametro è disponibile se *Source* è impostato su "Simple graph prototype".
Time period	Impostare un intervallo di tempo per il quale visualizzare i dati nei grafici. Selezionare la sorgente dati per l'intervallo di tempo: **Dashboard** - utilizza il selettore dell'intervallo di tempo della dashboard; **Widget** - utilizza un widget compatibile (impostato nel parametro *Widget*); **Custom** - utilizza un intervallo 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 sorgente dati per l'intervallo di tempo.
Widget	Immettere o selezionare un widget compatibile come sorgente dati per l'intervallo di tempo. Questo parametro è disponibile se *Time period* è impostato su "Widget".
From	Immettere 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 *Time period* è impostato su "Custom".
To	Immettere 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 *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 della dashboard come sorgente dati per gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template.
Columns	Immettere il numero di colonne di grafici da visualizzare all'interno di un widget di tipo graph prototype.
Rows	Immettere il numero di righe di grafici da visualizzare all'interno di un widget di tipo graph prototype.

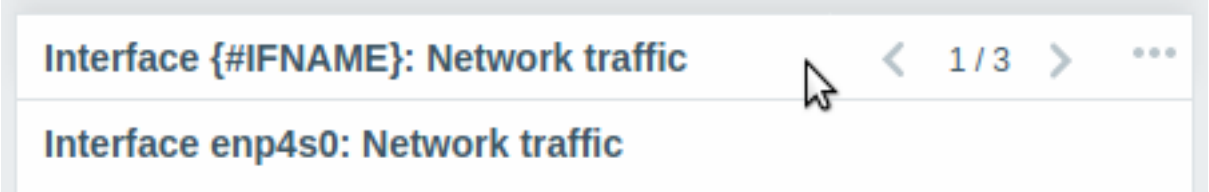
La legenda del grafico è composta da tre sezioni:

- Item e i 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 righe di trigger. Se sono presenti più trigger, per la visualizzazione vengono prioritizzati i trigger con gravità inferiore.

Sebbene i parametri *Columns* e *Rows* consentano di inserire più di un grafico nel widget, potrebbero comunque esserci più grafici individuati rispetto al numero di colonne/righe presenti nel widget. In questo caso, nel widget diventa disponibile la paginazione e un'intestazione a comparsa consente di passare da una pagina all'altra utilizzando le frecce sinistra e destra:

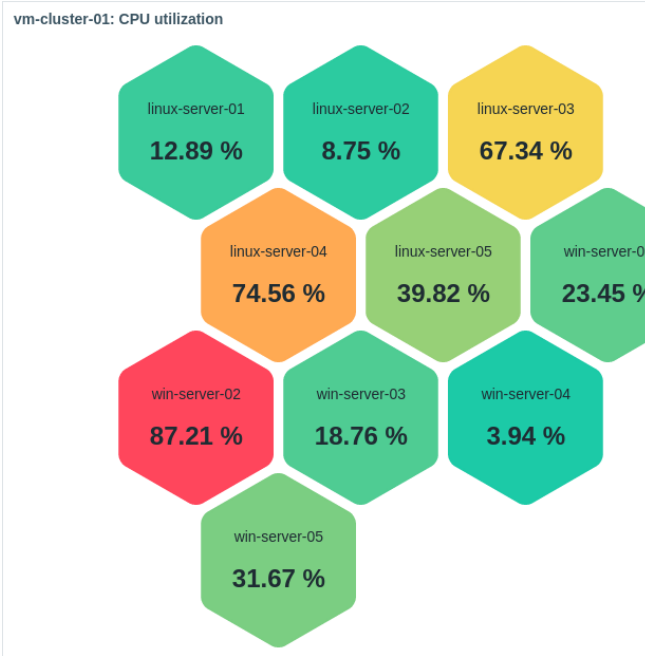


11 Honeycomb

Panoramica

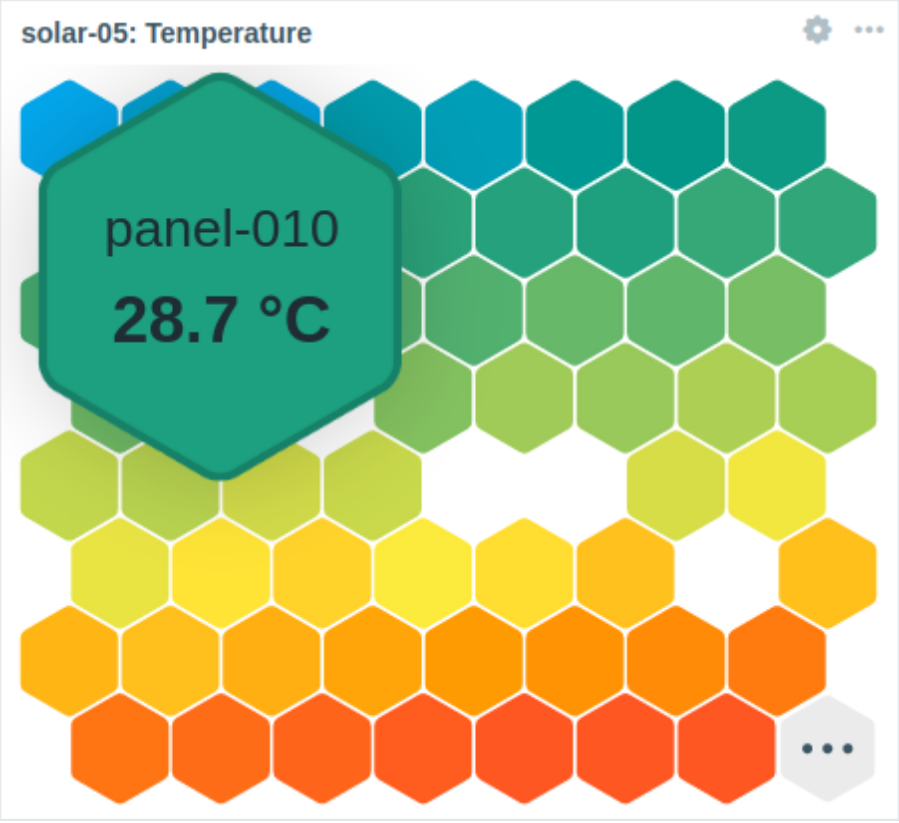
Il widget *Honeycomb* visualizza host o item come celle esagonali in una griglia. Questa disposizione consente di individuare facilmente host/item con problemi, identificare gruppi di problemi, confrontare gruppi a colpo d'occhio e altro ancora.





Il numero di celle visualizzate è limitato dalle dimensioni del widget e dalla dimensione minima della cella (32px). Se non tutte le celle entrano nello spazio disponibile, l'ultima cella mostra dei puntini di sospensione. Le dimensioni del widget possono essere modificate e il numero di celle, la loro dimensione e il loro posizionamento si adattano dinamicamente.

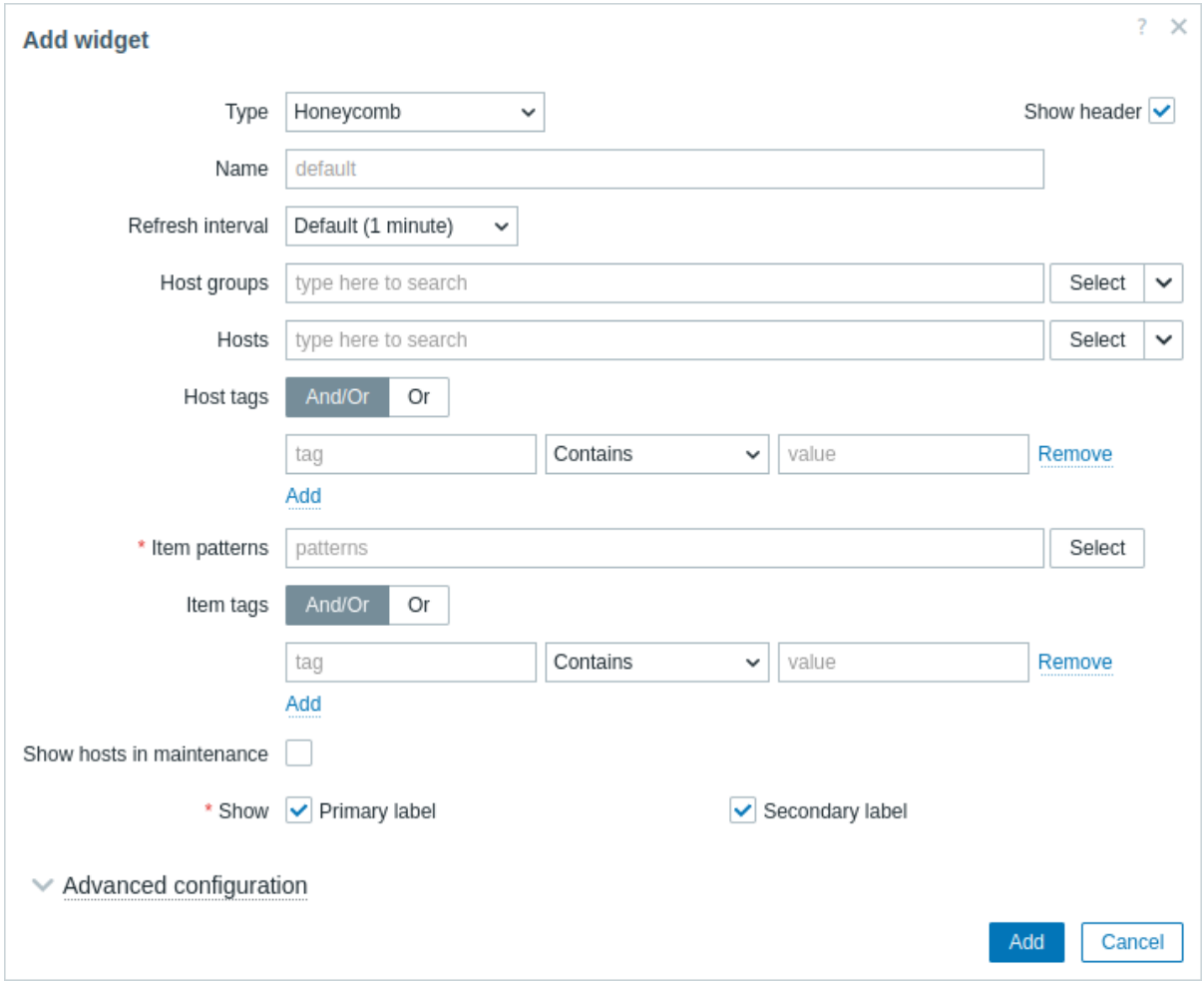
Passando il mouse sopra una cella, la cella evidenziata si ingrandisce per migliorare la visibilità. Facendo clic su una cella, il suo bordo viene evidenziato finché non viene selezionata un'altra cella.



Le informazioni visualizzate nel widget *Honeycomb* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu del widget.

Configurazione

Per configurare, selezionare *Honeycomb* come tipo:



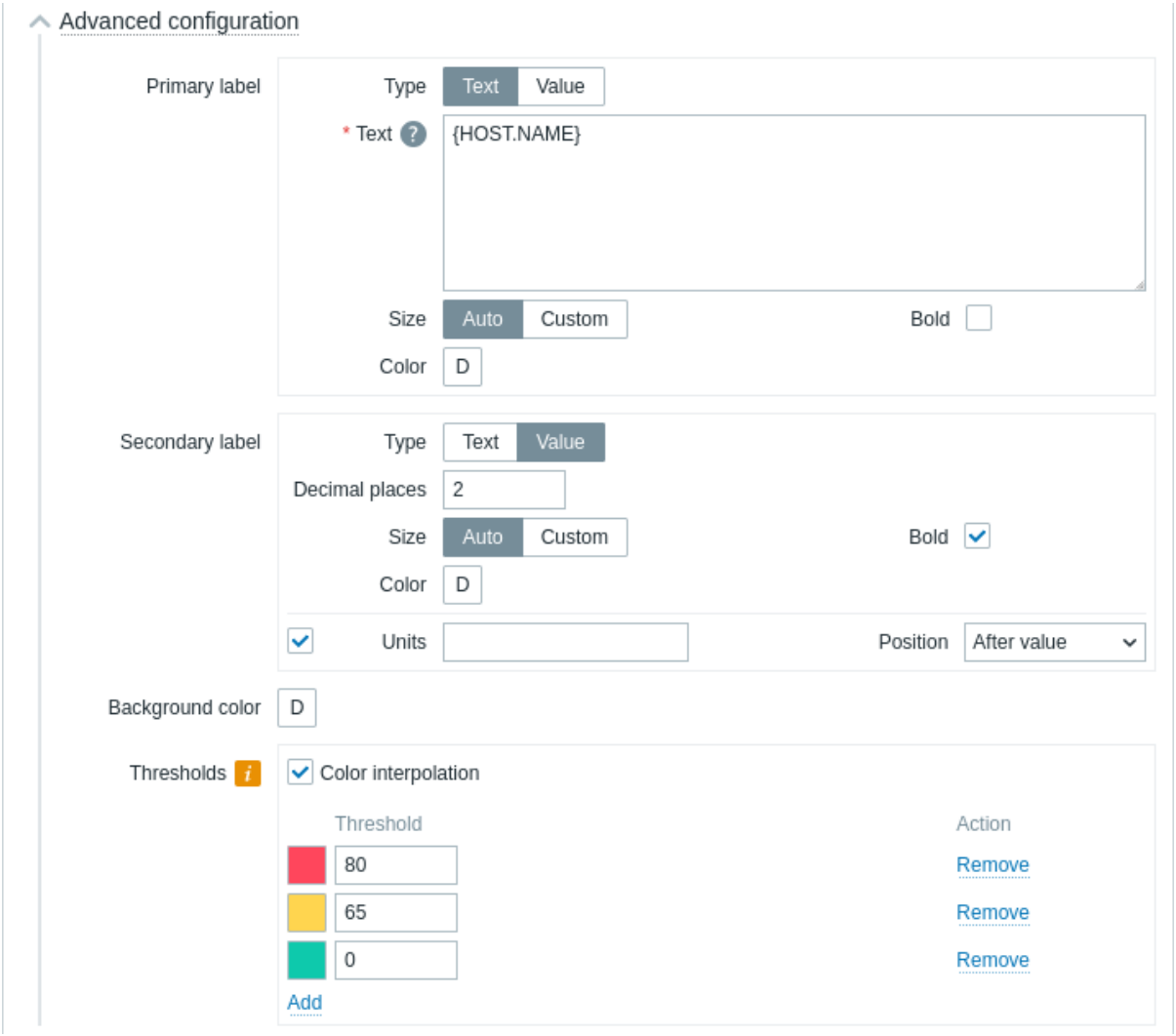
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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 mostrato 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 contenenti host con item corrispondenti al *Pattern item* selezionato (vedere sotto). Questo parametro non è disponibile durante la configurazione del 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 mostrato 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 durante la configurazione del 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 dei 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 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 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 template.
Item patterns	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 item che corrispondono a zero o più caratteri; `Zabbix*` restituirà 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, ad esempio, un item denominato *item** singolarmente, se esistono altri item corrispondenti (ad esempio, item2, item3). Durante la configurazione del 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* durante la configurazione del 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'etichetta *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 nel parametro *Testo*;
Valore - l'etichetta visualizzerà il valore dell'item con il numero di decimali specificato nel 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 supportate {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 unità dell'item in grassetto.

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.

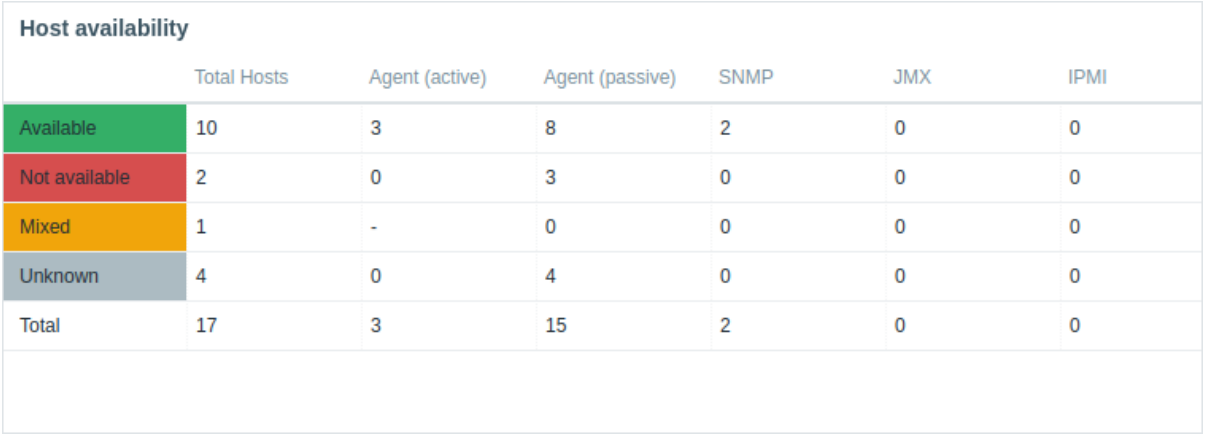
Unità

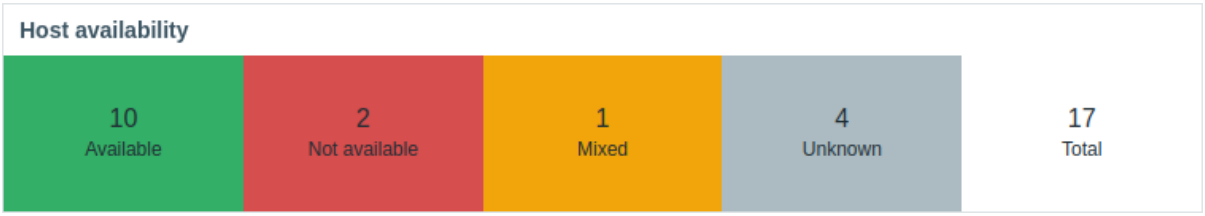
Unità	Selezionare la casella di controllo per visualizzare le unità con il valore dell'item. Se si inserisce un nome di unità, questo sostituirà le unità impostate nella configurazione dell'item. Questo parametro è disponibile se *Tipo* è impostato su "Testo".
Posizione	Selezionare la posizione delle 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 "Testo".
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 di soglia 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 di 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.

12 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.



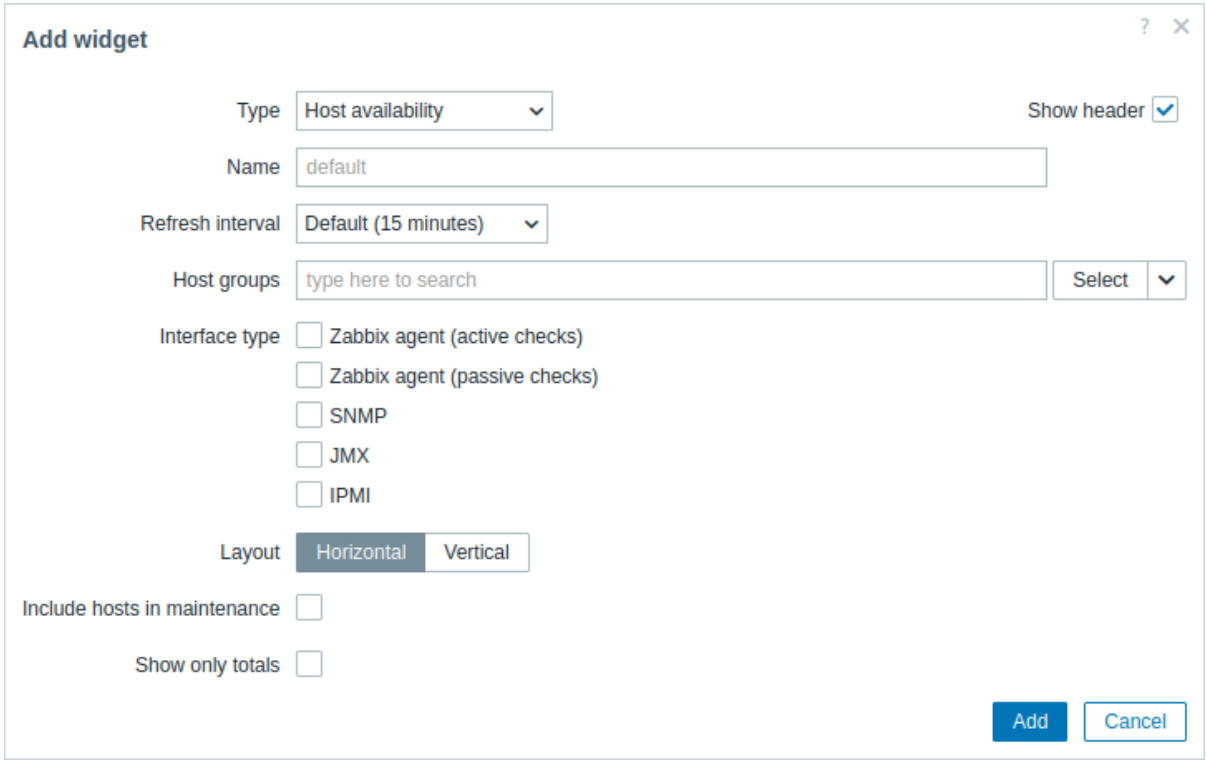


La disponibilità degli host viene conteggiata come segue:

- *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:



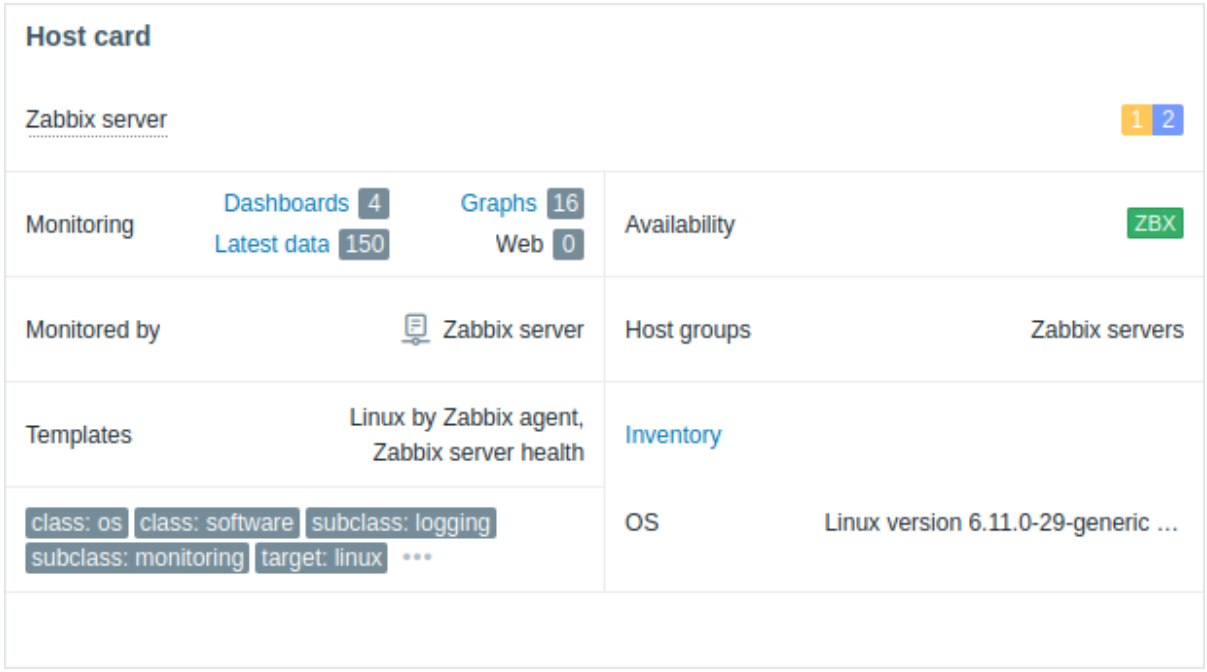
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.

13 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.).

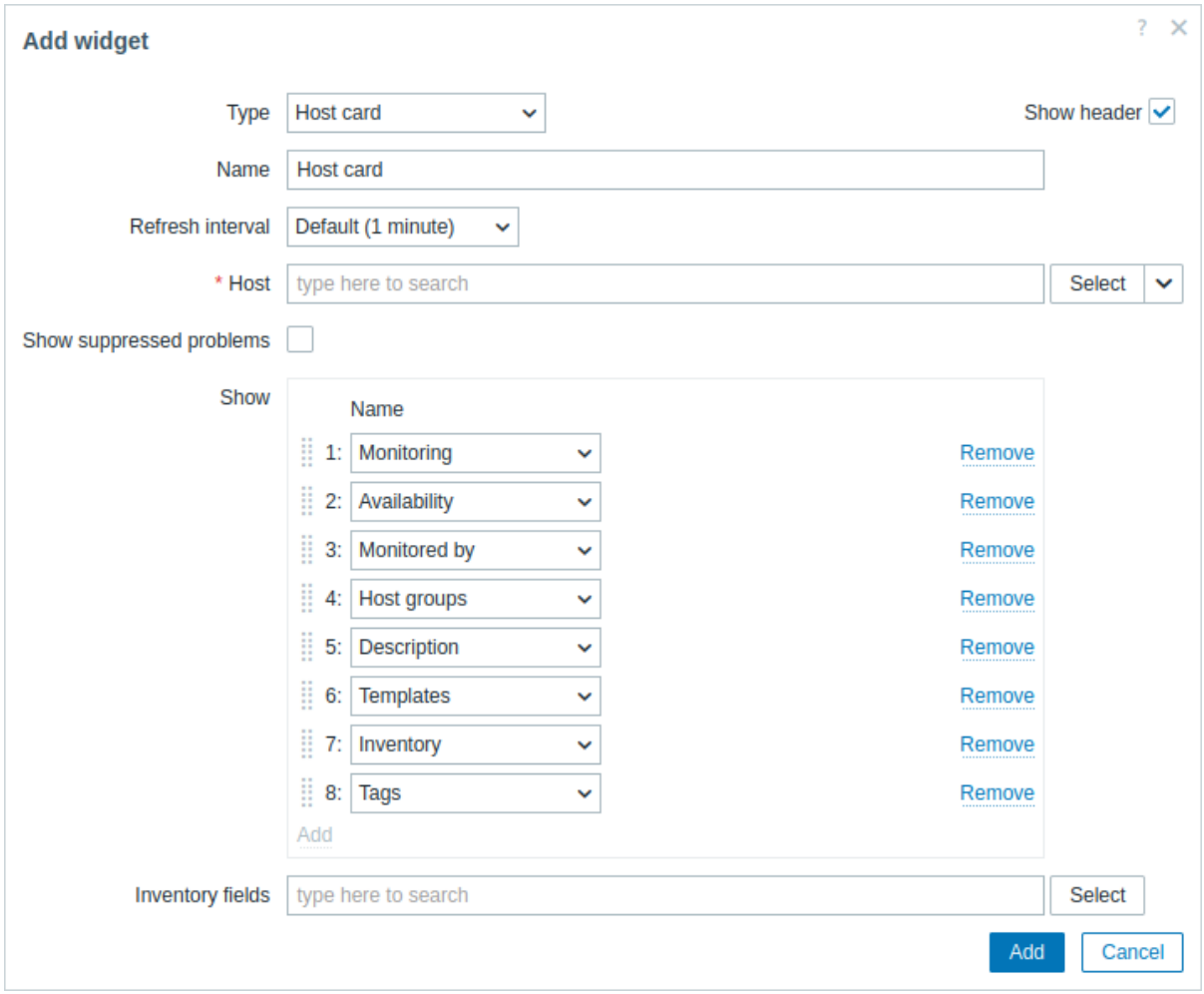


L’host può essere impostato direttamente nella configurazione del widget oppure selezionato da un widget compatibile o dal selettore host nella dashboard.

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:



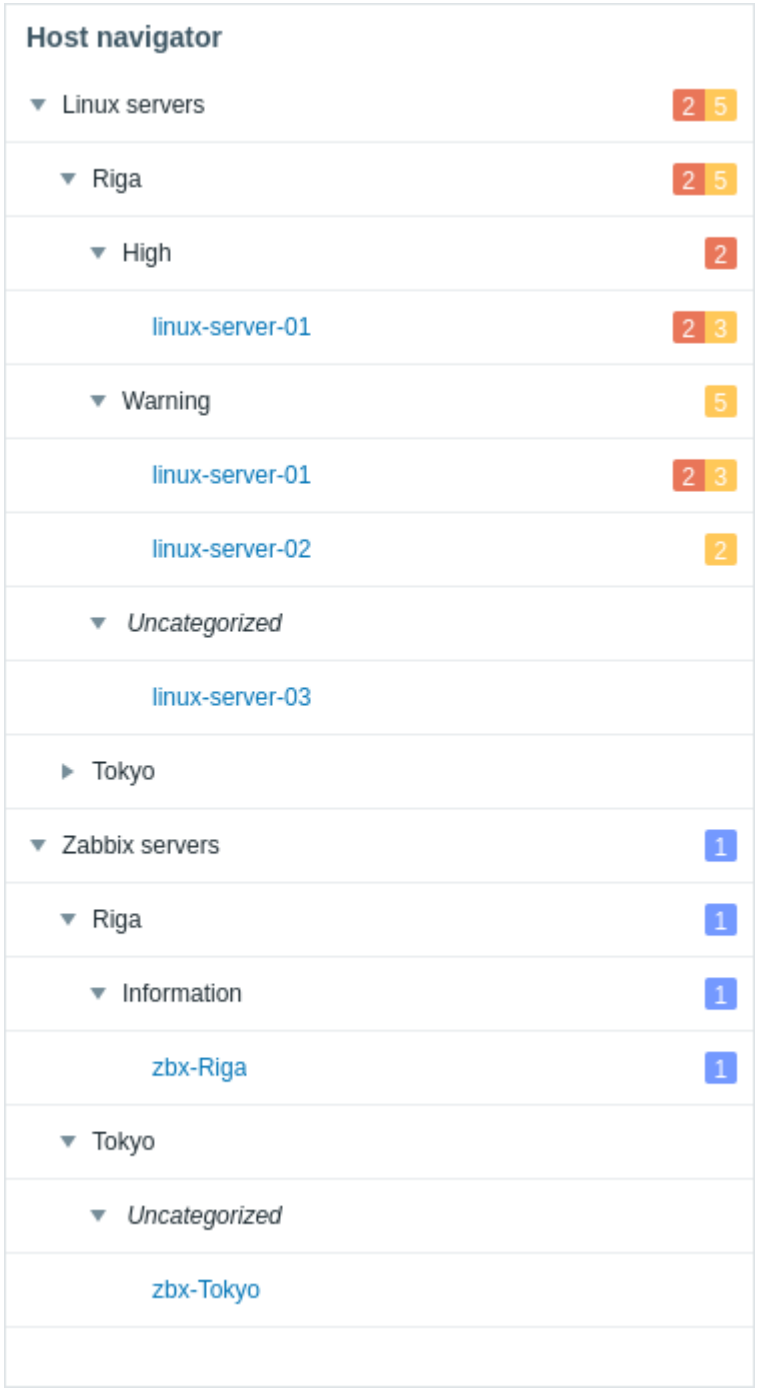
Oltre ai parametri comuni a tutti i widget, è possibile impostare i seguenti parametri specifici:

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*.

14 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.

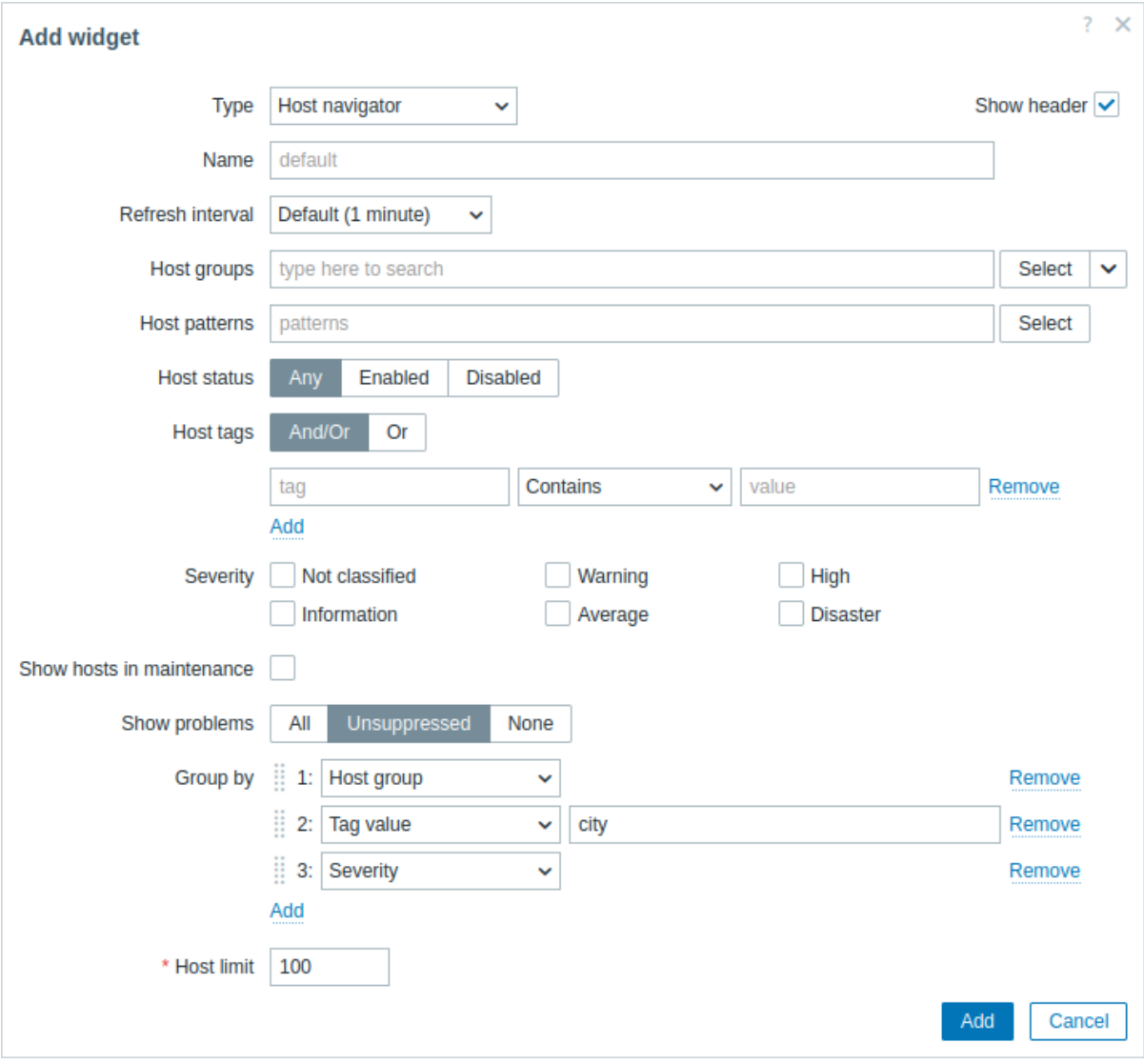
Per i gruppi, i problemi e gli host in manutenzione, ulteriori dettagli sono accessibili tramite suggerimenti al passaggio del mouse.

Il widget è particolarmente utile per controllare ciò che altri widget visualizzano in base all'host selezionato.



Configurazione

Per configurare il widget, selezionare *Host navigator* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

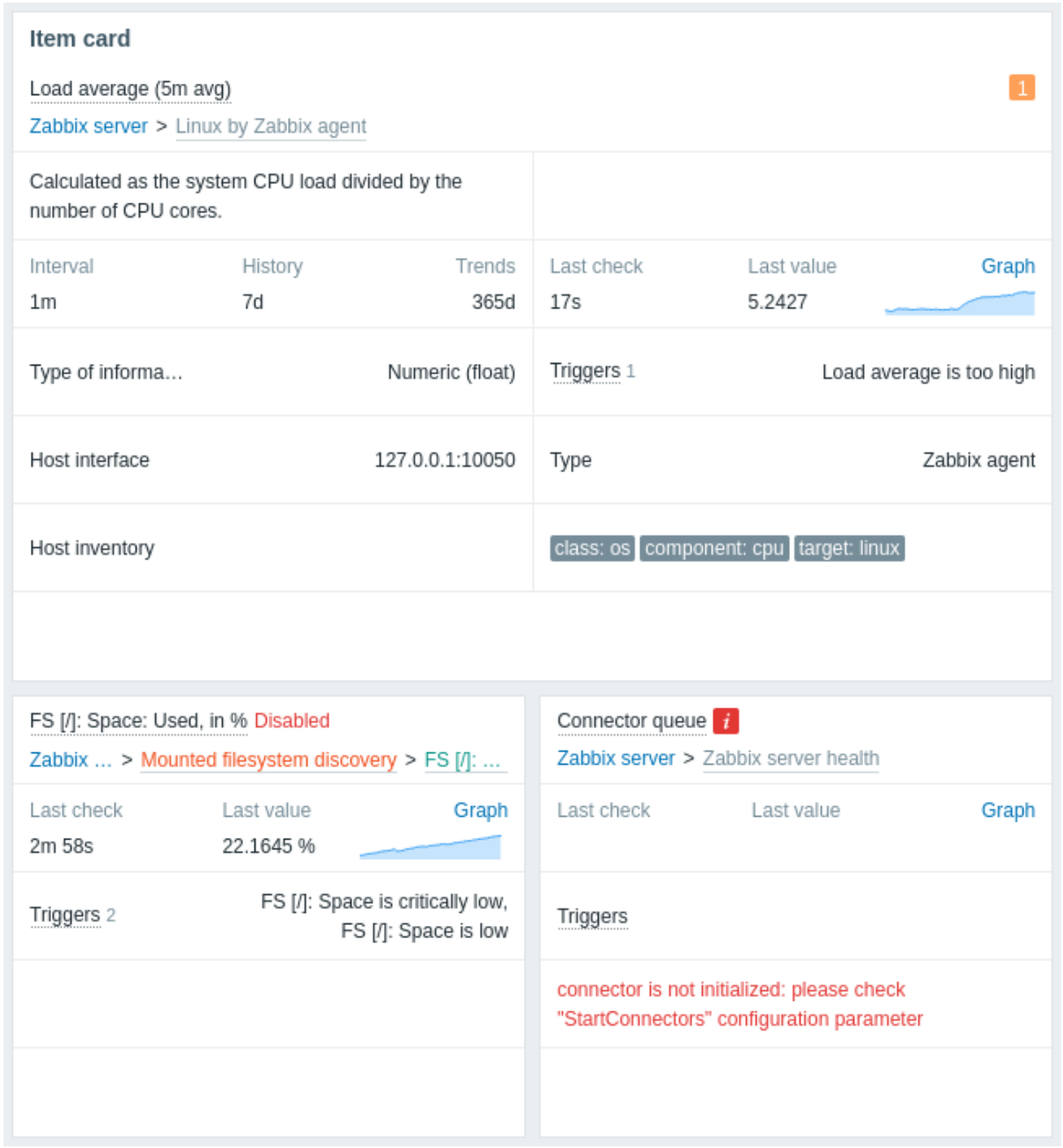
Host groups	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi corrispondenti. La selezione 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 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 pattern di host oppure selezionare host esistenti come pattern di host. Gli host che corrispondono ai pattern specificati verranno visualizzati nel navigatore 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 selezionato alcun host, il widget visualizzerà tutti gli host. Per la selezione possono essere utilizzati pattern wildcard (ad esempio, `*` restituirà host che corrispondono a zero o più caratteri; `Zabbix*` restituirà host che iniziano con "Zabbix"). Per specificare un pattern wildcard, inserire manualmente la stringa e premere *Invio*. Quando si inizia a digitare, un elenco a discesa mostrerà gli host corrispondenti limitati a quelli appartenenti agli host nei *Host groups* selezionati, se presenti. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Host status	Filtrare quali host visualizzare in base al loro stato (qualsiasi, abilitato, disabilitato).
Host tags	Questo parametro non è disponibile quando si configura il widget in una dashboard di template. Specificare i tag per filtrare gli 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 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). 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.
Severity	Questo parametro non è disponibile quando si configura il widget in una dashboard di template. Contrassegnare le severity dei problemi per filtrare gli host con problemi da visualizzare nel widget. Se non viene contrassegnata alcuna severity, verranno visualizzati tutti gli host con tutti i problemi.
Show hosts in maintenance	Selezionare questa casella 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 di template.
Show problems	

Group by	Aggiungere un attributo di raggruppamento in base al quale raggruppare gli host selezionati: **Host group** - raggruppa gli host in base al loro 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 alle severity dei loro problemi. Se *Show problems* è configurato per visualizzare i problemi, questi vengono mostrati come segue: - per ciascun gruppo di severity, viene visualizzato solo il conteggio dei problemi corrispondente; - per ciascun host, vengono visualizzati tutti i conteggi dei suoi problemi. Si noti che gli host verranno raggruppati solo in base alle severity contrassegnate nel parametro *Severity*; se non viene contrassegnata alcuna severity, tutti gli host verranno raggruppati in base a tutte le severity. Gli attributi di raggruppamento possono essere riordinati trascinandoli verso l’alto o verso il basso tramite la maniglia prima del nome del gruppo. Si noti che l’ordine degli attributi di raggruppamento determina l’ordine di nidificazione dei gruppi. Ad esempio, specificando più nomi di tag (1: color, 2: city), gli host verranno raggruppati per colore (red, blue, ecc.) e poi per città (Riga, Tokyo, ecc.). Un host può essere visualizzato in più gruppi a seconda degli 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 verranno 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 e tutti devono essere 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 più host da visualizzare rispetto al limite impostato, sotto gli host visualizzati viene mostrato un messaggio corrispondente (ad esempio, ”100 of 100+ hosts are shown”). Si noti che 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 di tag (più di 200), nel widget verranno visualizzati solo i primi 100 valori di 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.

15 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.).

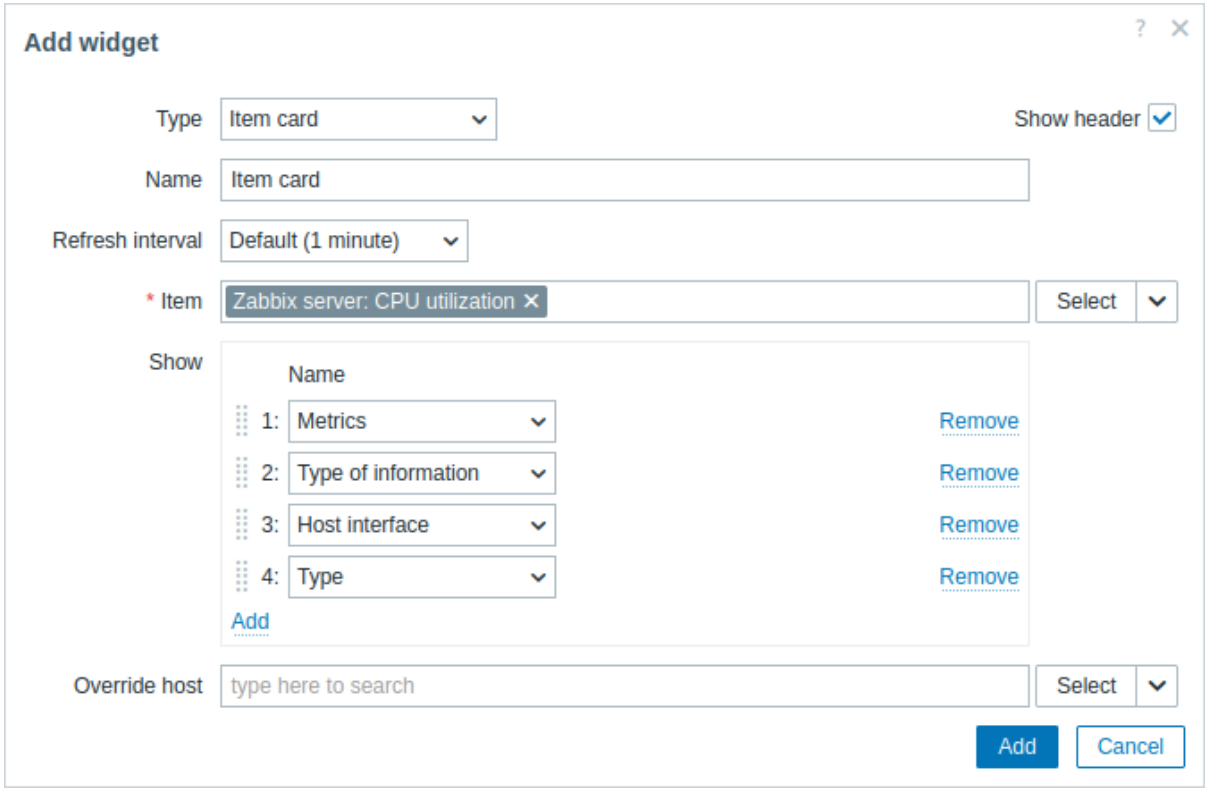


L'item può essere impostato direttamente nella configurazione del widget oppure selezionato da un widget compatibile nella dashboard.

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:



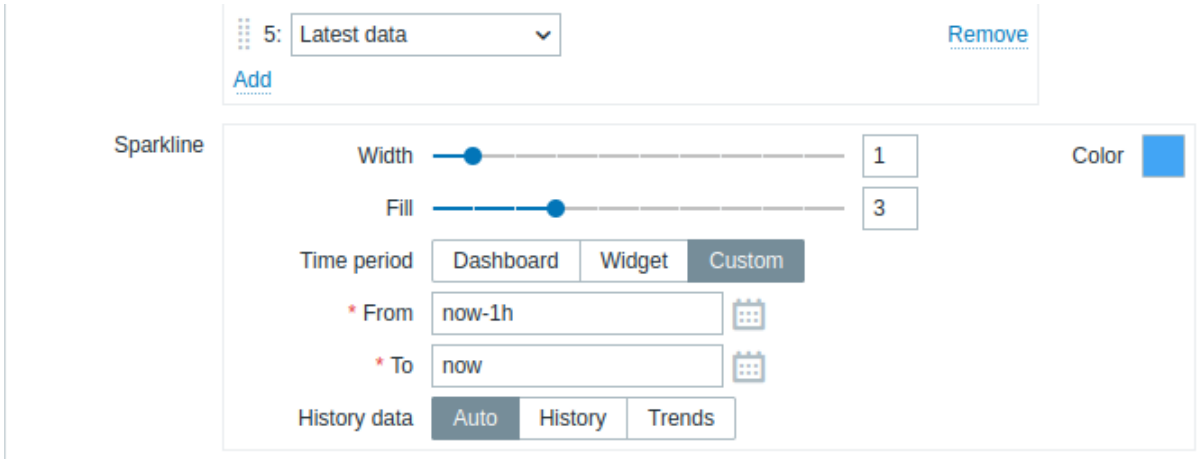
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.



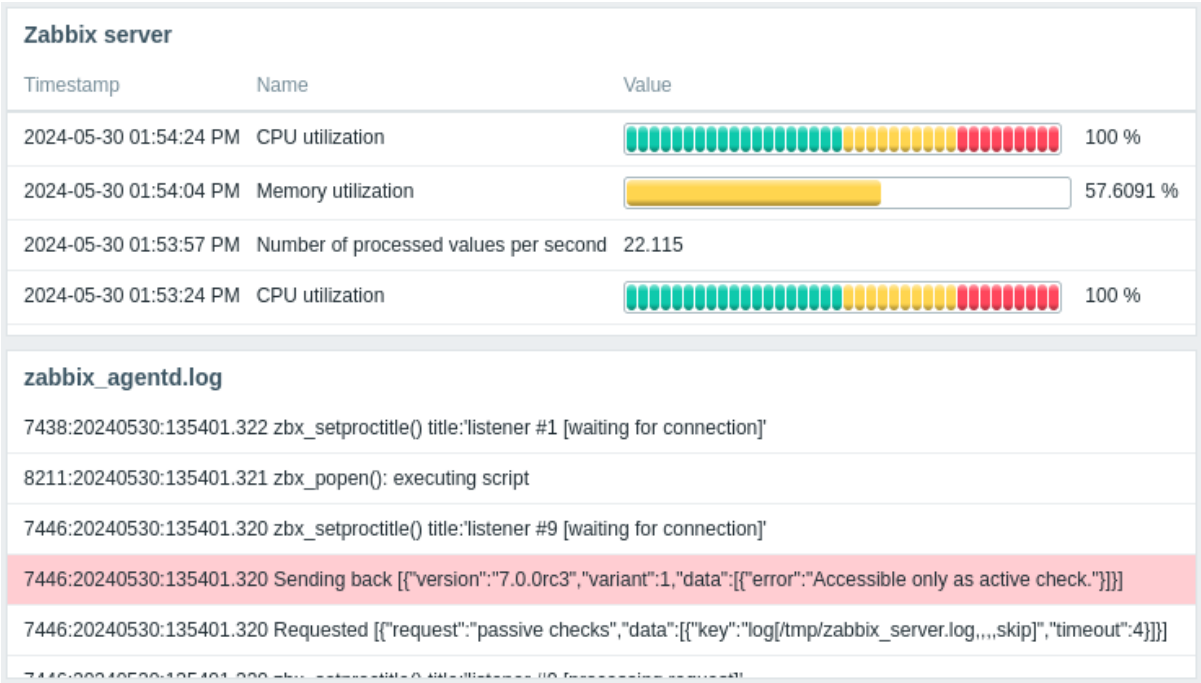
Larghezza	Imposta lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Colore	Seleziona il colore della linea e del riempimento.
Riempimento	Imposta il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo 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.

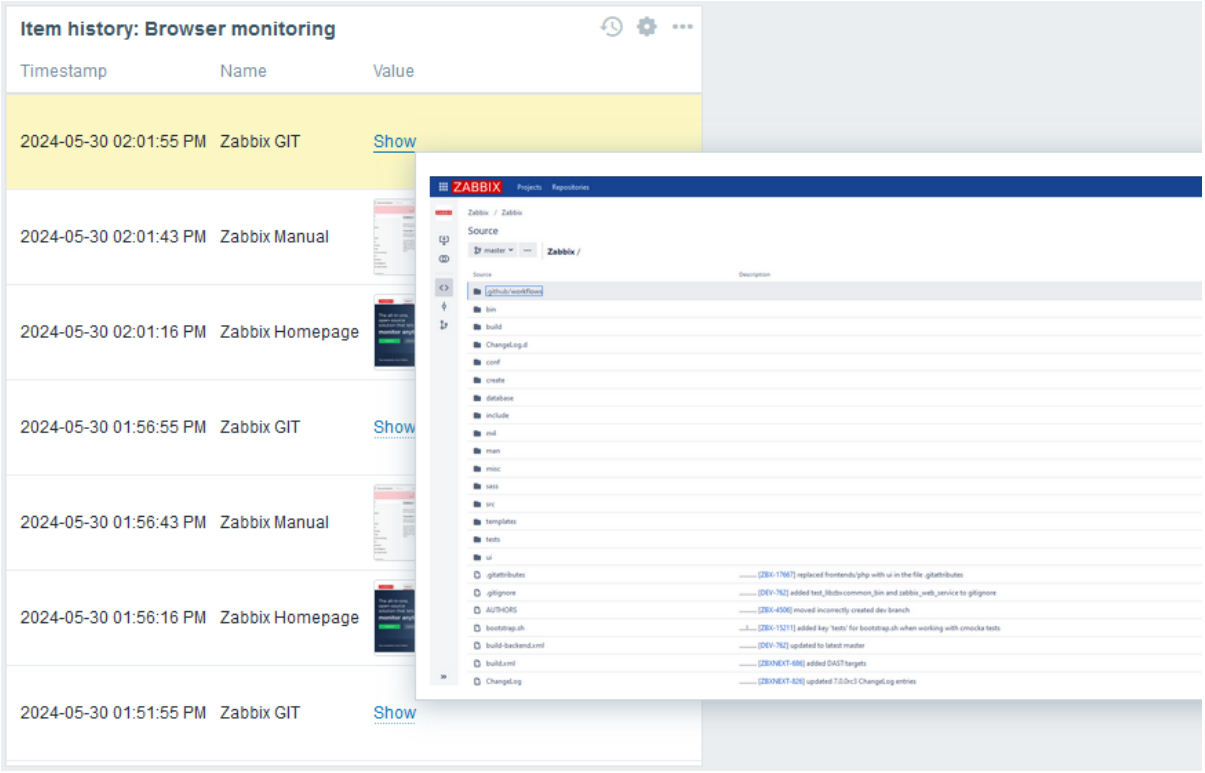
Widget	Inserisci o seleziona un widget compatibile come origine dati per il periodo di tempo. Questo parametro è disponibile se *Periodo di tempo* è impostato su "Widget".
Da	Inserisci o seleziona l'inizio del periodo di tempo. La sintassi del tempo relativo (`now`, `now/d`, `now/w-1w`, ecc.) è supportata. 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.

16 Cronologia degli item

Panoramica

Il widget *Cronologia item* visualizza gli ultimi dati per tutti i tipi di item (numerici, testuali, ecc.) in formato tabellare. Può mostrare barre di avanzamento e immagini (utile per i browser items) e può evidenziare i valori (utile per il monitoraggio dei file di log).

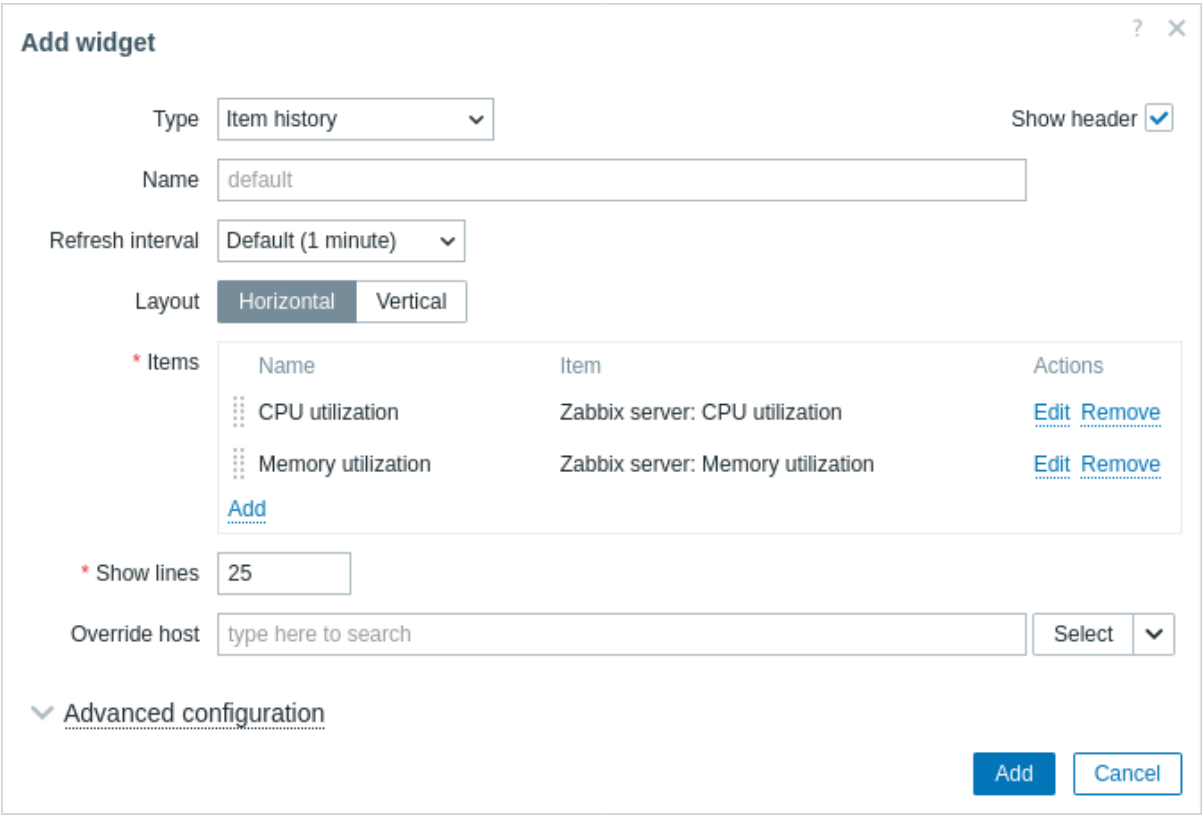




È possibile visualizzare fino a 1000 record.

Configurazione

Per configurare il widget, selezionare *Cronologia item* come tipo:



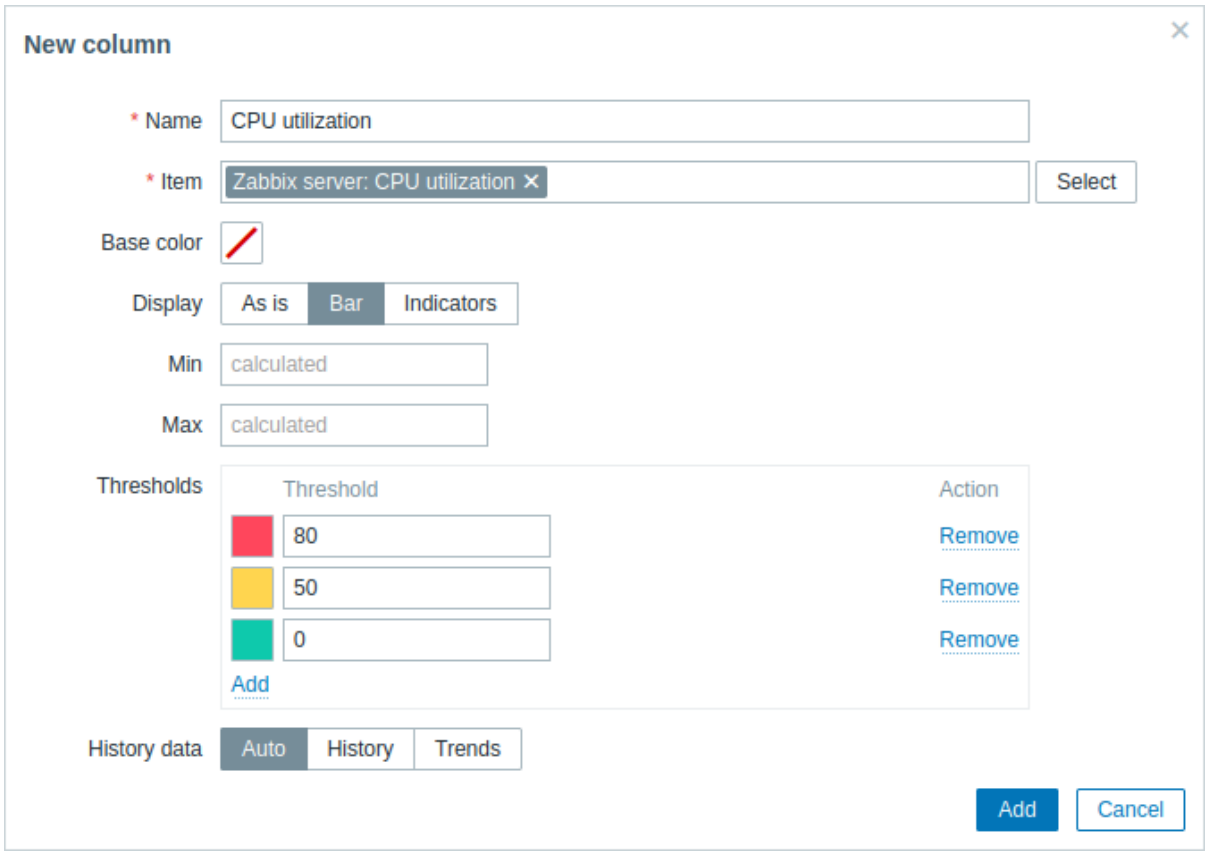
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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 gli item di tipo carattere, testo, log e JSON:

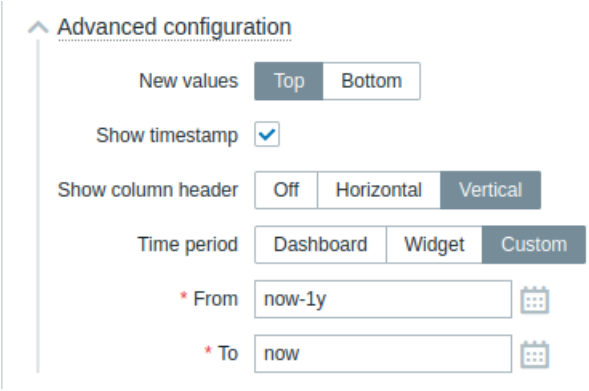
Evidenziazioni	Fare clic su *Aggiungi* per aggiungere un'evidenziazione, selezionare un colore di evidenziazione dal selettore colori e specificare un'espressione regolare. Il colore selezionato verrà utilizzato come colore di sfondo per i valori dell'item in cui l'espressione regolare specificata corrisponde al testo.
Visualizzazione	Selezionare come deve essere visualizzato il valore dell'item: **Così com'è** - esattamente come ricevuto. Le interruzioni di riga vengono mantenute se il valore contiene ritorni a capo. L'andata a capo automatica viene applicata se *Layout* è impostato su "Verticale" e sono configurate più colonne; **HTML** - come testo formattato in HTML; **Riga singola** - come una singola riga, troncata alla lunghezza specificata (1-500 caratteri). Passando il puntatore o facendo clic sul valore troncato si apre una finestra pop-up con il valore completo.
Usa font a spaziatura fissa	Selezionare questa casella di controllo per visualizzare il valore dell'item con un font a spaziatura fissa (non selezionata per impostazione predefinita).
Visualizza ora locale	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 *Mostra timestamp* in configurazione avanzata deve essere selezionata. Questo parametro è disponibile solo per gli 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*:

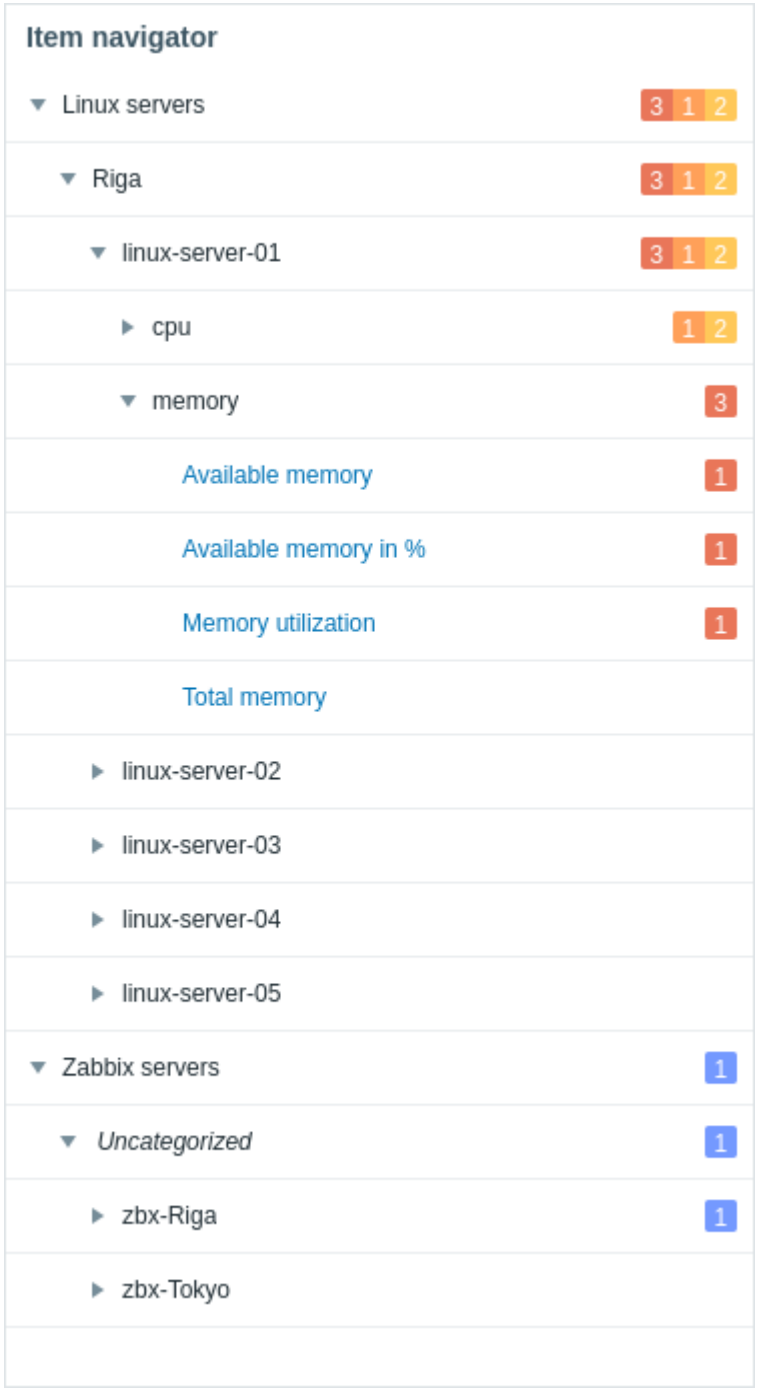


Nuovi valori	Selezionare dove devono essere aggiunti i nuovi valori degli item: **In alto** - nella parte superiore delle colonne; **In basso** - nella parte inferiore delle colonne.
Mostra timestamp	Selezionare questa casella per visualizzare la colonna del timestamp (non selezionata per impostazione 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".

17 Navigatore item

Panoramica

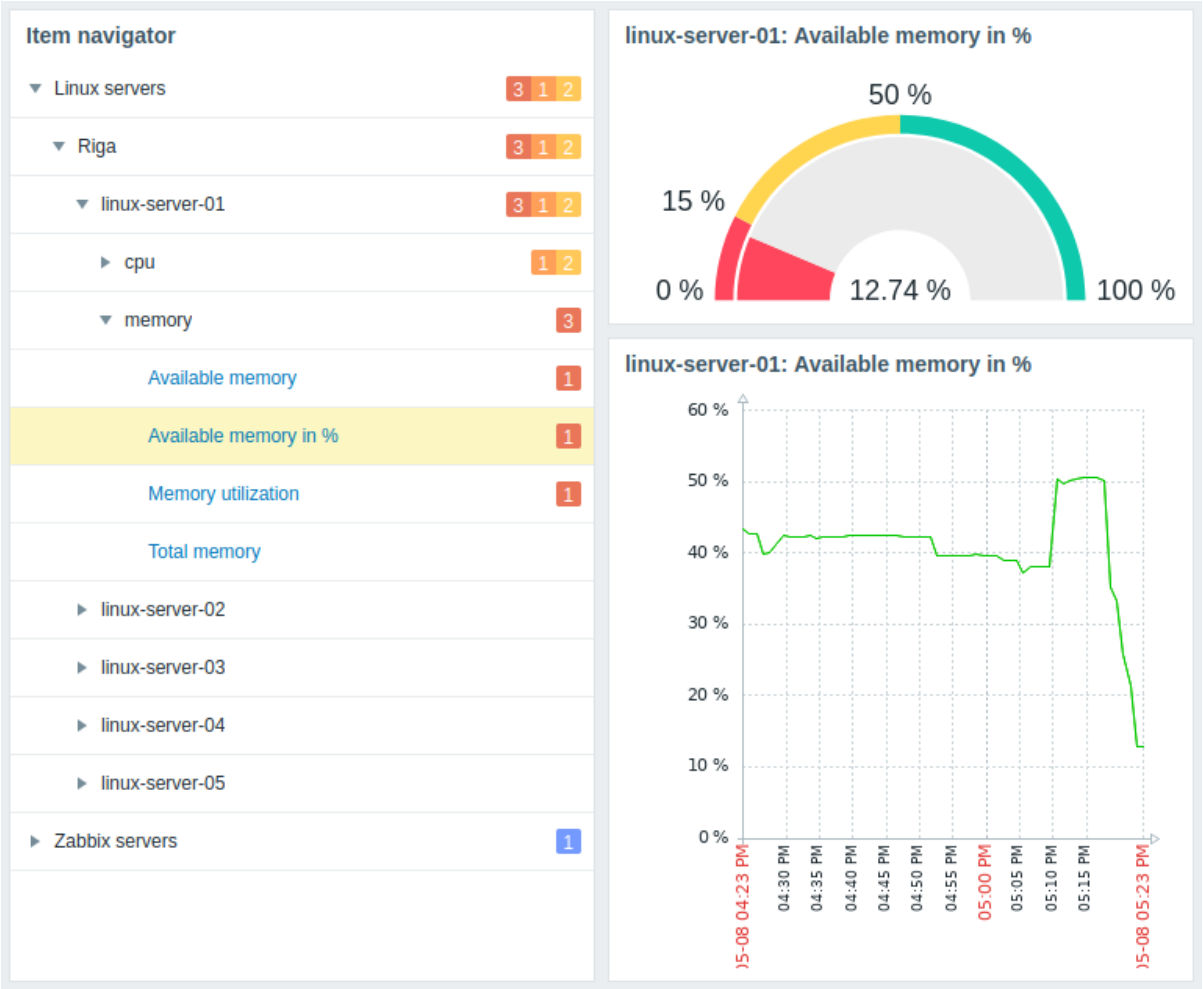
Il widget *Navigatore item* visualizza un elenco di item in base a varie opzioni di filtro e raggruppamento.



I gruppi in base ai quali gli item sono organizzati possono essere espansi o compressi.

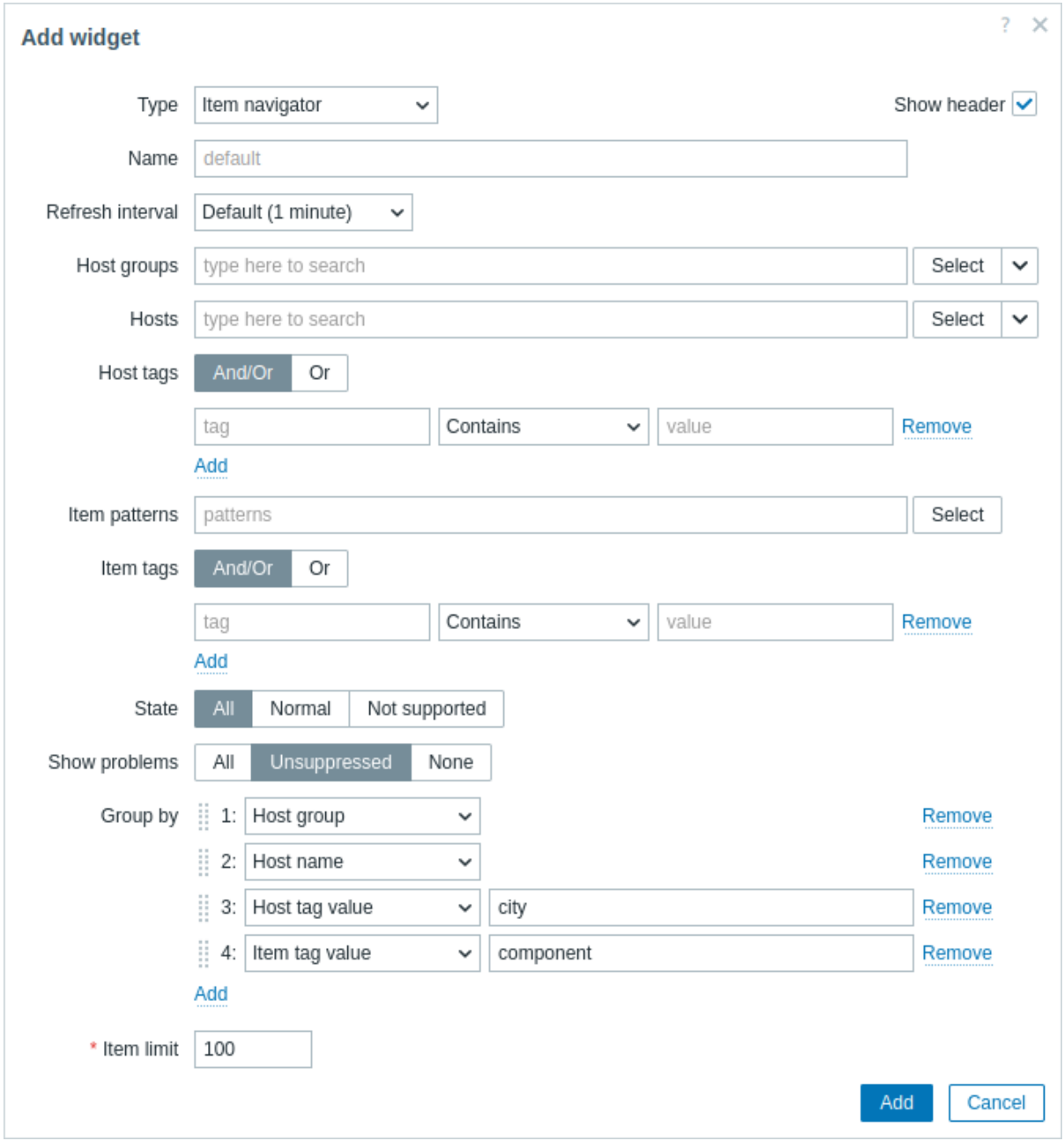
Per i gruppi e i problemi, ulteriori dettagli sono accessibili passando il mouse sui suggerimenti.

Il widget è particolarmente utile per controllare ciò che altri widget visualizzano in base all'item selezionato.



Configurazione

Per configurare il widget, seleziona *Item navigator* 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 il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato 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 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	Seleziona gli host. In alternativa, seleziona 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 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.

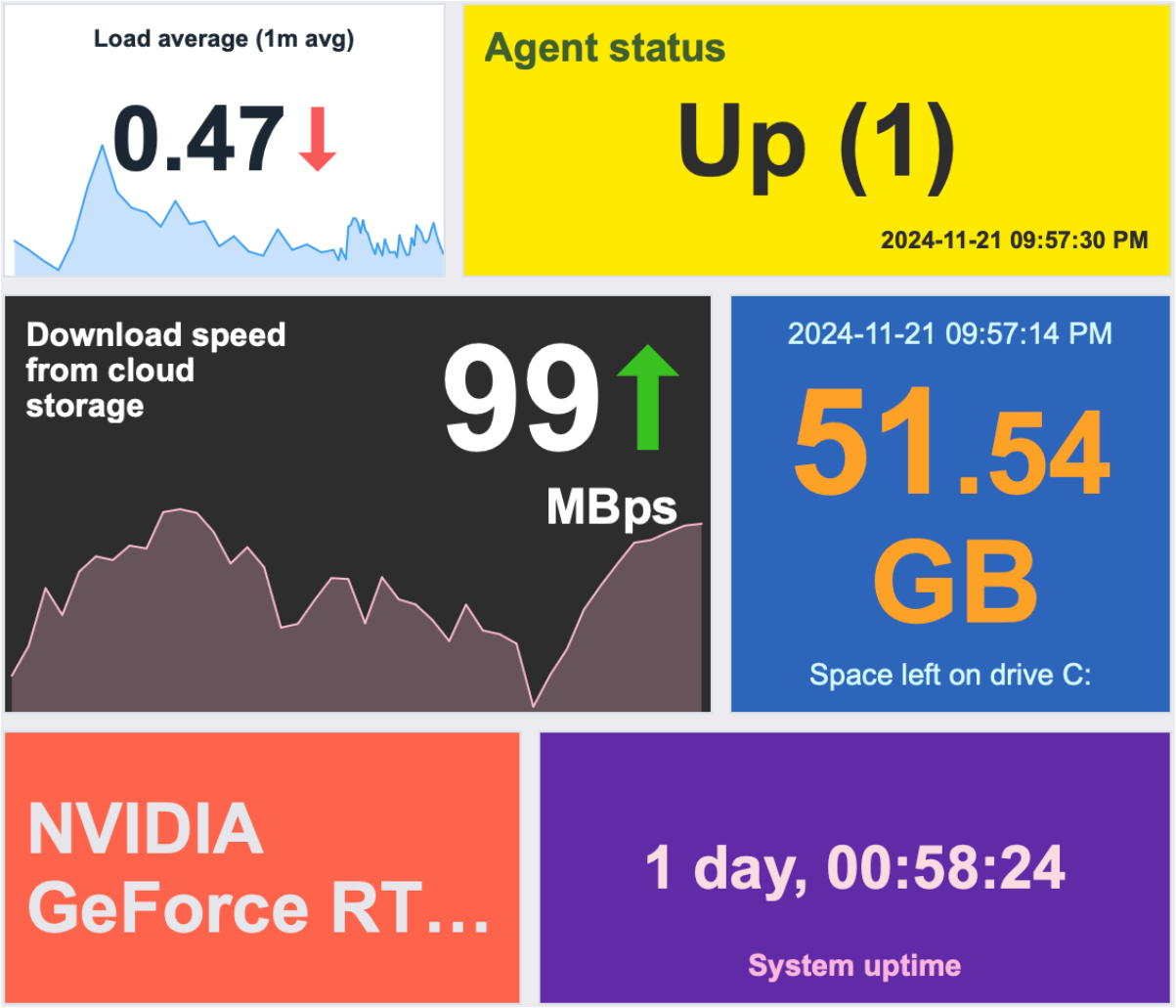
Host tags	Specifica i tag per filtrare gli item visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori 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). 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. Questo parametro non è disponibile quando si configura il widget in una dashboard di template.
Item patterns	Inserisci pattern di item oppure seleziona item esistenti come pattern di item. Gli item che corrispondono ai pattern specificati verranno visualizzati nell'item navigator. 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, inserisci manualmente la stringa e premi *Enter*. Quando inizi a digitare, un elenco a discesa mostrerà gli item corrispondenti limitati a quelli appartenenti agli *Hosts* selezionati o agli host presenti 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 esistono 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	Specifica i tag per filtrare gli 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). 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.
State	Filtra quali item visualizzare in base al loro stato (tutti, normali, non supportati).

Show problems	Filtra quali problemi visualizzare in base al loro stato (tutti, non soppressi, nessuno). I problemi sono indicati con blocchi colorati alla fine della riga dell'item. I colori si basano sulla severità del problema, che può essere personalizzata in *Administration* > *General* > *Trigger displaying options*. Le severità dei problemi possono essere modificate durante l'aggiornamento dei problemi.
Group by	Aggiungi un attributo di raggruppamento in base al quale raggruppare gli item: **Host group** - raggruppa gli item in base ai gruppi di host dei rispettivi host; **Host name** - raggruppa gli item in base ai rispettivi host; **Host tag value** - inserisci un nome di tag per raggruppare gli item in base ai valori di questo tag host (ad esempio, inserisci "city" per raggruppare gli item in base ai valori "Riga", "Tokyo", ecc.); **Item tag value** - inserisci un nome di tag per raggruppare gli item in base ai valori di questo tag item (ad esempio, inserisci "component" per raggruppare gli item in base ai valori "cpu", "memory", ecc.). Gli attributi di raggruppamento possono essere riordinati trascinandoli verso l'alto o verso il basso tramite la maniglia prima del nome del gruppo. Nota che l'ordine degli attributi di raggruppamento determina l'ordine di nidificazione dei gruppi. Ad esempio, specificando più nomi di tag host (1: color, 2: city), gli item verranno raggruppati per colore (red, blue, ecc.) e poi per città (Riga, Tokyo, ecc.). Un item può essere visualizzato in più gruppi a seconda degli 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 verranno 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 mostrati come segue: - per ciascun gruppo, viene visualizzato il conteggio di tutti i problemi degli item subordinati; - per ciascun item, viene visualizzato solo il relativo conteggio dei problemi. È possibile specificare fino a 10 attributi di raggruppamento e tutti devono essere univoci. Se non viene specificato alcun attributo di raggruppamento, gli item non verranno raggruppati.
Item limit	Inserisci il numero massimo di item da visualizzare. I valori possibili vanno da 1 a 9999. Quando sono disponibili più item da visualizzare rispetto al limite impostato, sotto gli item visualizzati viene mostrato un messaggio corrispondente (ad esempio, "100 of 100+ items are shown"). Nota che 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*.

18 Valore dell'item

Panoramica

Il widget *Valore item* visualizza il valore di un singolo item numerico o di tipo stringa. È utile per tenere sotto controllo metriche importanti, visualizzare le soglie e rilevare cambiamenti improvvisi nei dati.



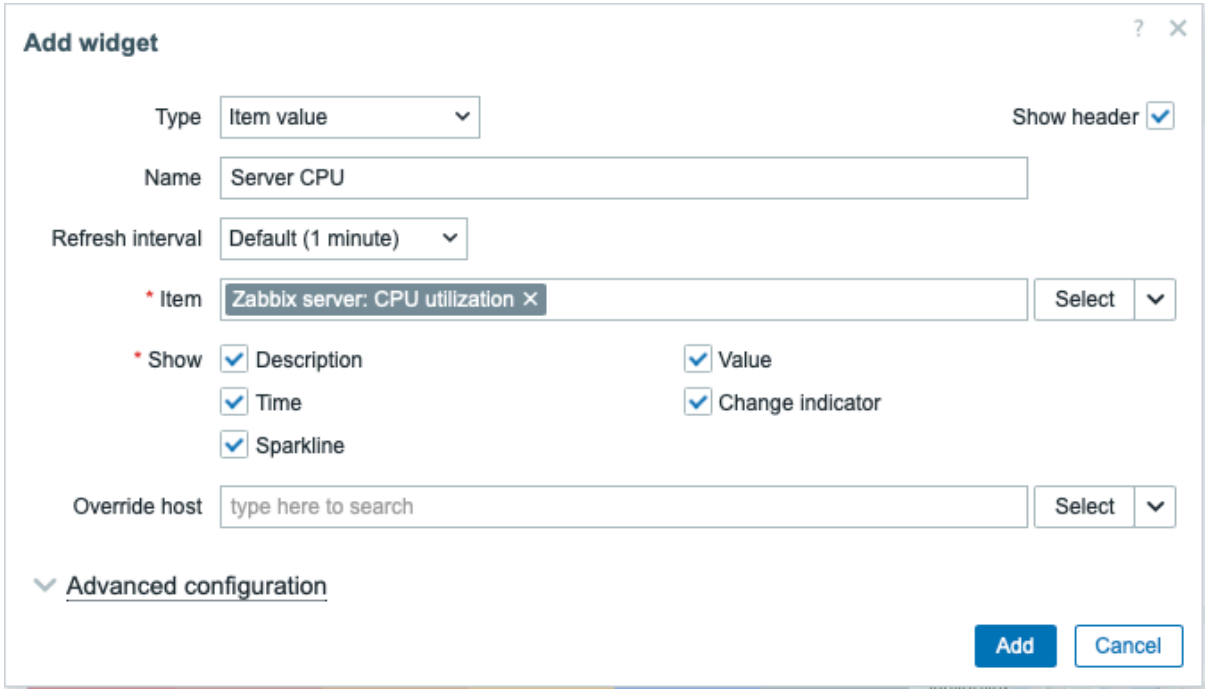
Puoi configurare il widget per visualizzare quanto segue:

- Timestamp della metrica (2024-11-21 09:57:30)
- Descrizione dell'item (Load average (1m), Stato agent)
- 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 gli ultimi dati per gli item di tipo stringa.

Configurazione

Per configurare il widget, 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 gli 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 quando si configura il widget in una dashboard template.
Configurazione avanzata	Fare clic sull'etichetta *Configurazione avanzata* per visualizzare le opzioni di configurazione avanzata.

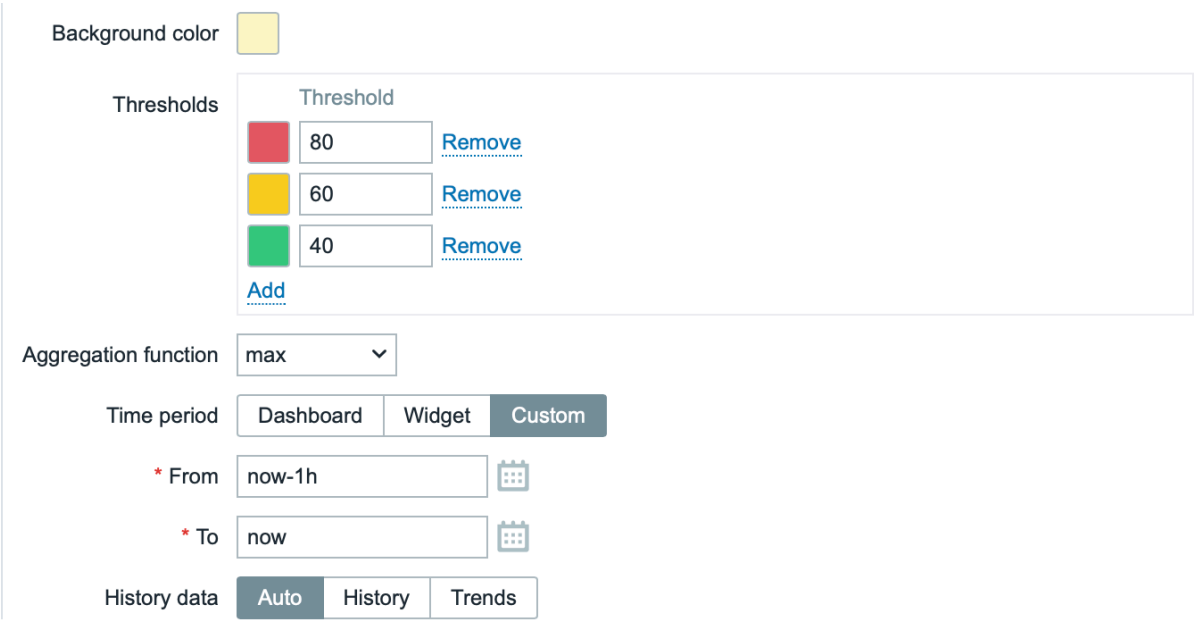
Configurazione avanzata

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) per l'intero widget e una funzione di aggregazione per la visualizzazione dei valori.

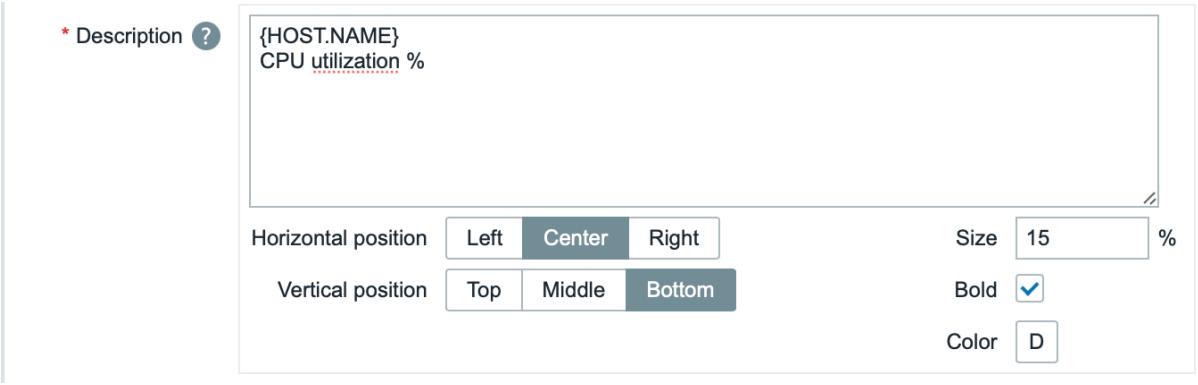


Colore di sfondo	Selezionare il colore di sfondo per l'intero widget dal selettore colori. D indica il colore predefinito (dipende dal tema del frontend). Per tornare al valore predefinito, fare clic sul pulsante *Usa predefinito* nel selettore colori.
Soglie	Configurare il colore di sfondo dinamico per l'intero widget. Fare clic su *Aggiungi* per aggiungere una soglia, selezionare il colore di sfondo dal selettore colori e specificare un valore numerico. Quando il valore dell'item è uguale o superiore al valore di soglia, il colore di sfondo cambierà. L'elenco verrà ordinato in ordine crescente al momento del salvataggio. Si noti che il colore di sfondo dinamico verrà visualizzato correttamente solo per item numerici.
Funzione di aggregazione	Specificare quale funzione di aggregazione utilizzare: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il conteggio dei valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore; **not used** - visualizza il valore più recente (nessuna aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo selezionato (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.
Periodo di tempo	Periodo di tempo da utilizzare per aggregare i valori. 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 *Da* e *A*; se impostato, verrà visualizzata un'icona dell'orologio nell'angolo superiore destro del widget, che indicherà l'ora impostata al passaggio del mouse. 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 impostata su "not used".
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 "Custom".
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 "Custom".

Dati storici	Prelevare i dati dalla cronologia o dai trend: **Auto** - selezione automatica; **History** - preleva i dati dalla cronologia; **Trends** - preleva i dati dai trend. Questa impostazione si applica solo ai dati numerici. I dati non numerici verranno sempre prelevati dalla cronologia.

Descrizione

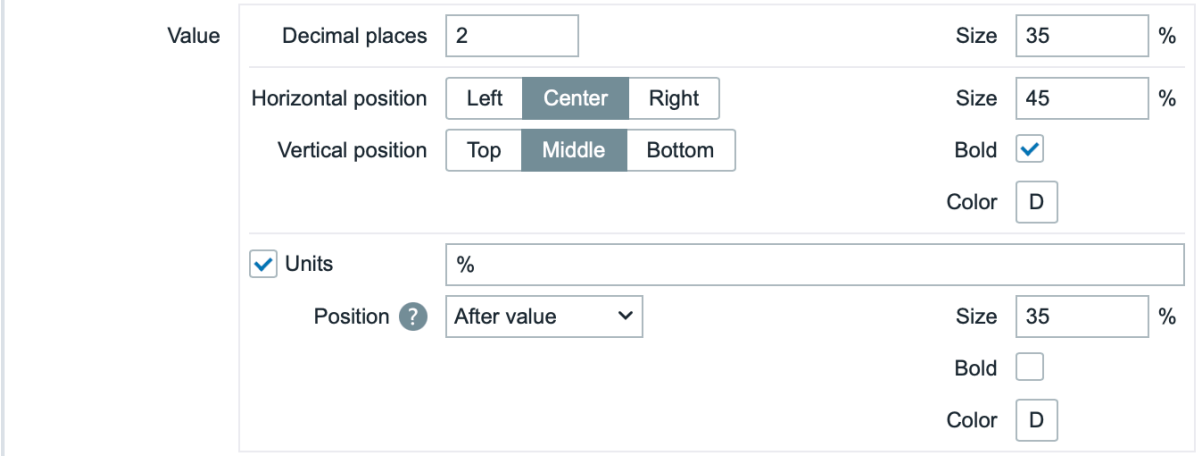
Questi parametri determinano come deve essere visualizzata la descrizione dell'item.



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

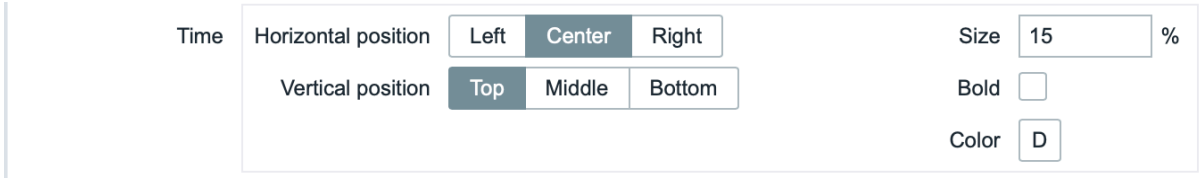
Questi parametri determinano come deve essere visualizzato il valore dell'item.



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

Questa sezione consente di selezionare il colore degli indicatori di variazione dal selettore colori.



Gli indicatori di variazione sono i seguenti:

- ↑ - il valore dell’item è aumentato (per item numerici)
- ↓ - il valore dell’item è diminuito (per item numerici)
- ↕ - il valore dell’item è cambiato (per item stringa e item con mappatura dei valori)

Nota:

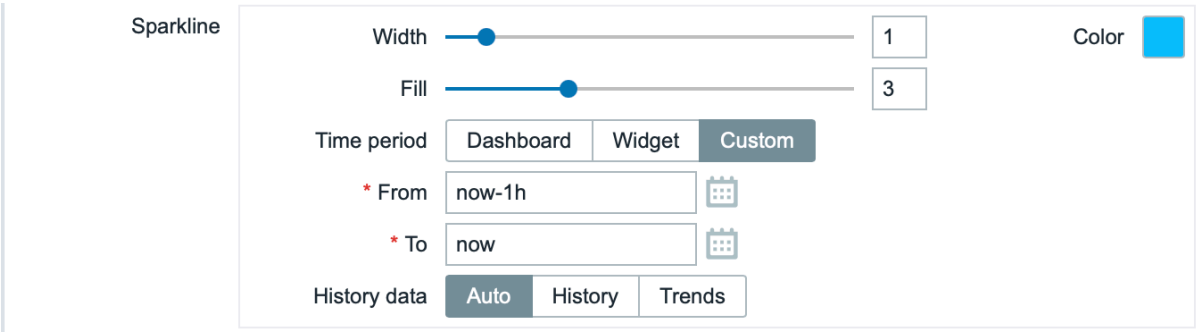
- Gli indicatori di aumento (↑) e diminuzione (↓) non vengono mostrati se è presente un solo valore.
- I valori vengono sempre confrontati all'interno dello stesso ambito; ad esempio:
  - L'ultimo valore viene confrontato con il valore precedente.
  - Un valore mensile viene confrontato con il mese precedente.
  - 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 tornare al valore predefinito, fare clic sul pulsante *Use default* nel selettore 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.

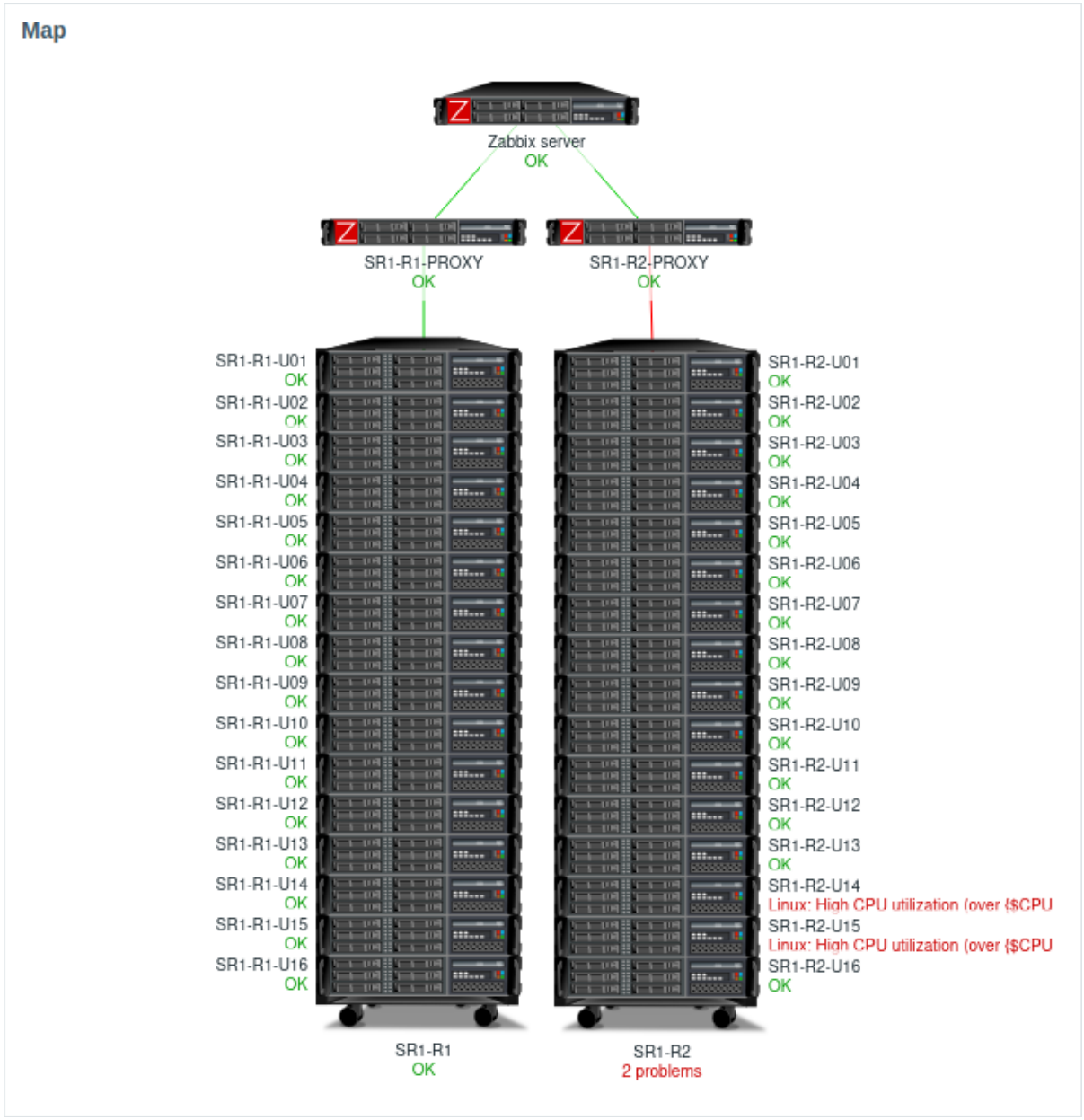


Larghezza	Imposta lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Colore	Seleziona il colore della linea e del riempimento.
Riempimento	Imposta il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo 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.

19 Mappa

Panoramica

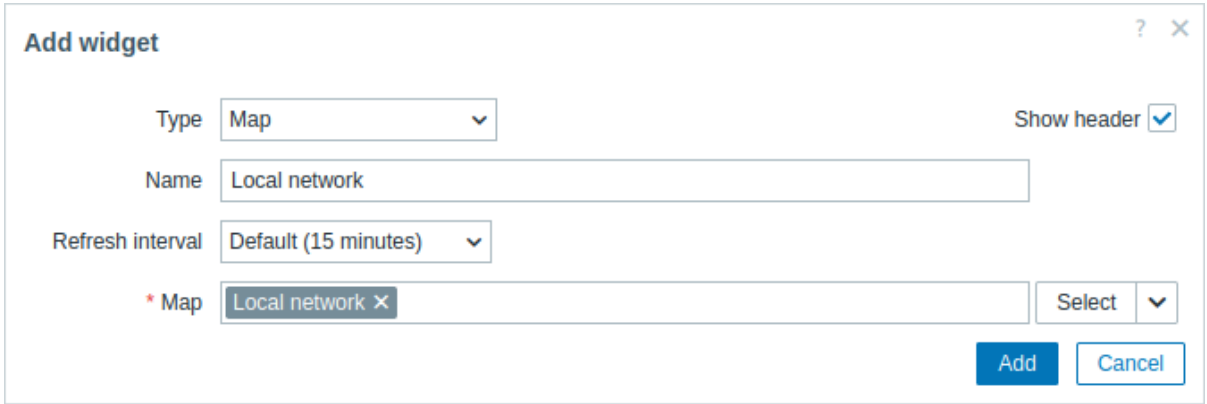
Il widget *Map* può visualizzare una mappa di rete o una visualizzazione simile, fornendoti una panoramica dinamica della tua rete.



Puoi abbinarlo al widget Map navigation tree per visualizzare dinamicamente la mappa selezionata nell'albero.

Configurazione

Per configurare, selezionare *Mappa* come tipo:



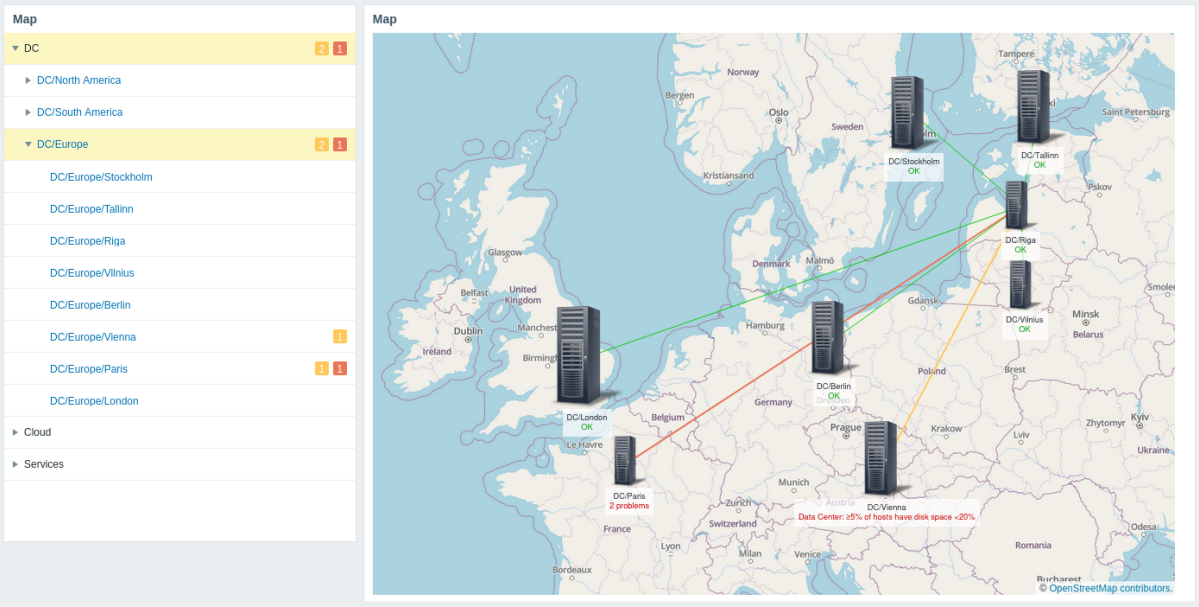
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.

20 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.

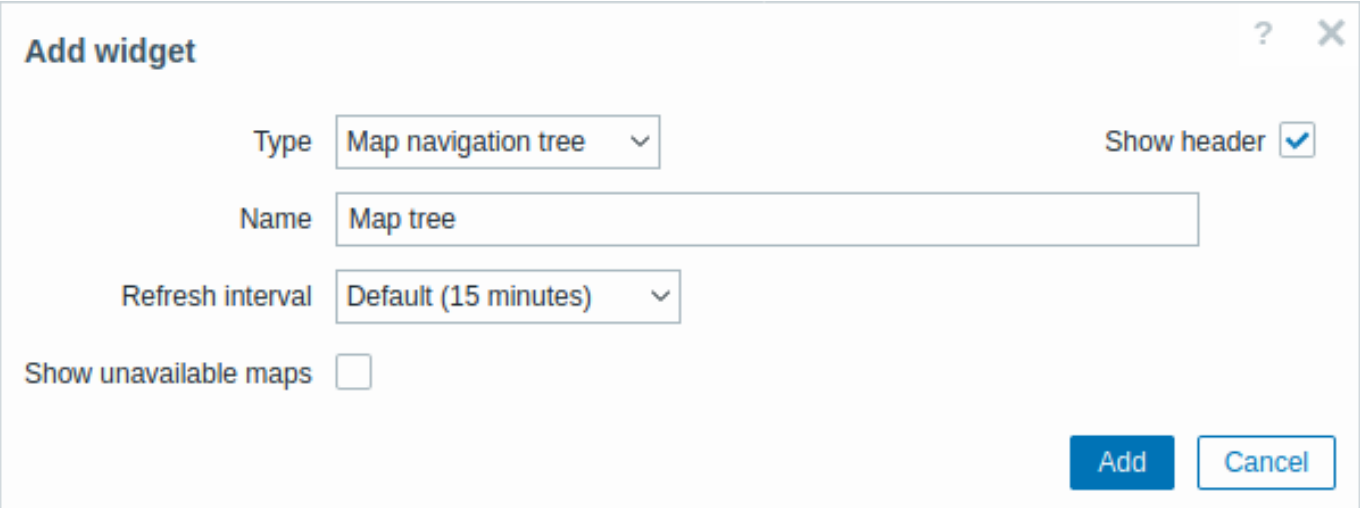


Puoi abbinarlo al widget Map per visualizzare dinamicamente la mappa selezionata nell'albero.

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:

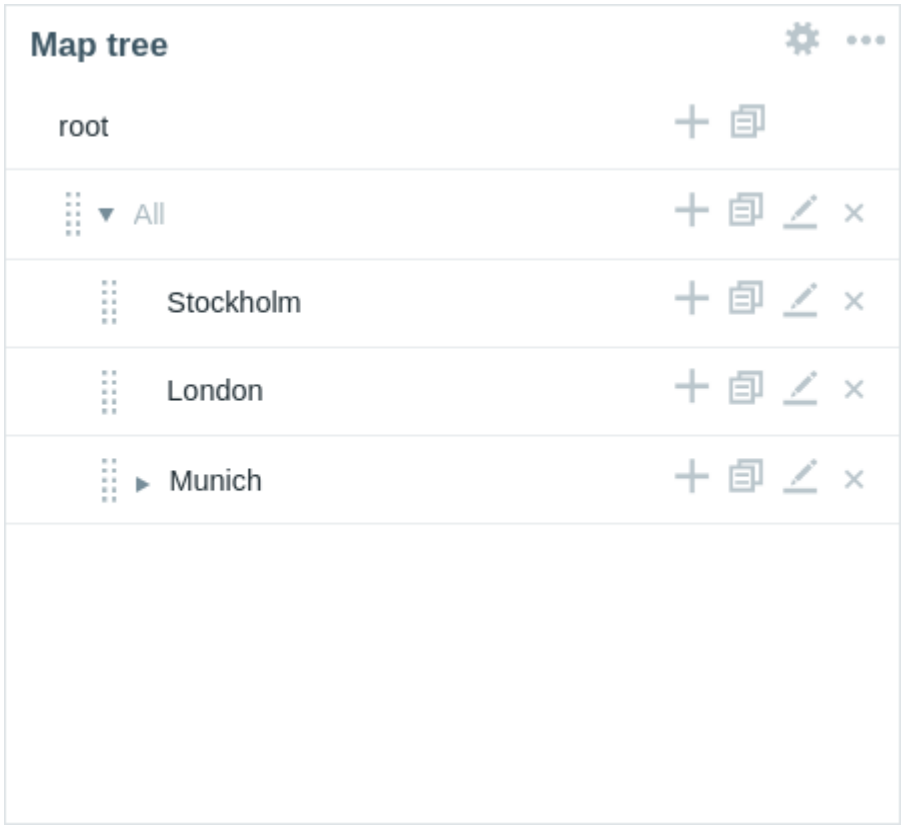


Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.

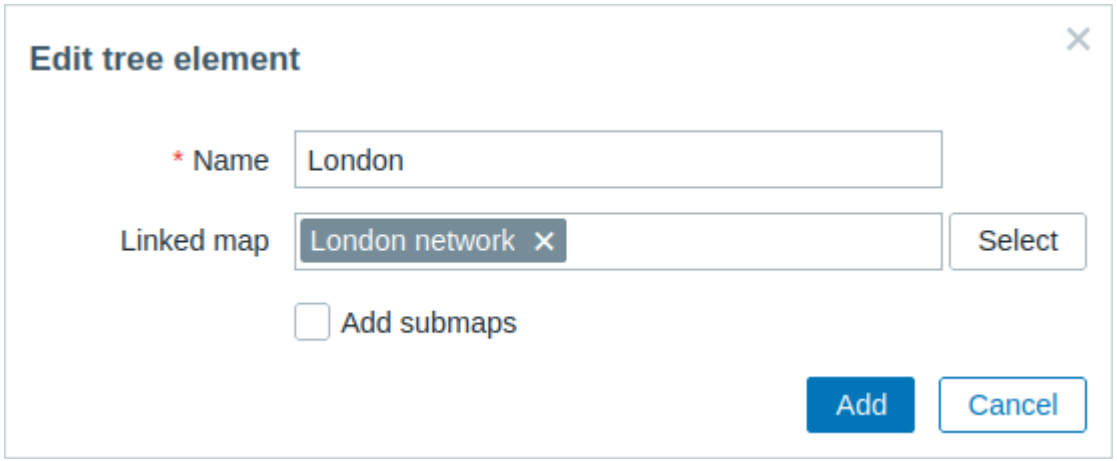
Gli elementi dell'albero di navigazione sono visualizzati in un elenco. È possibile:

- 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.



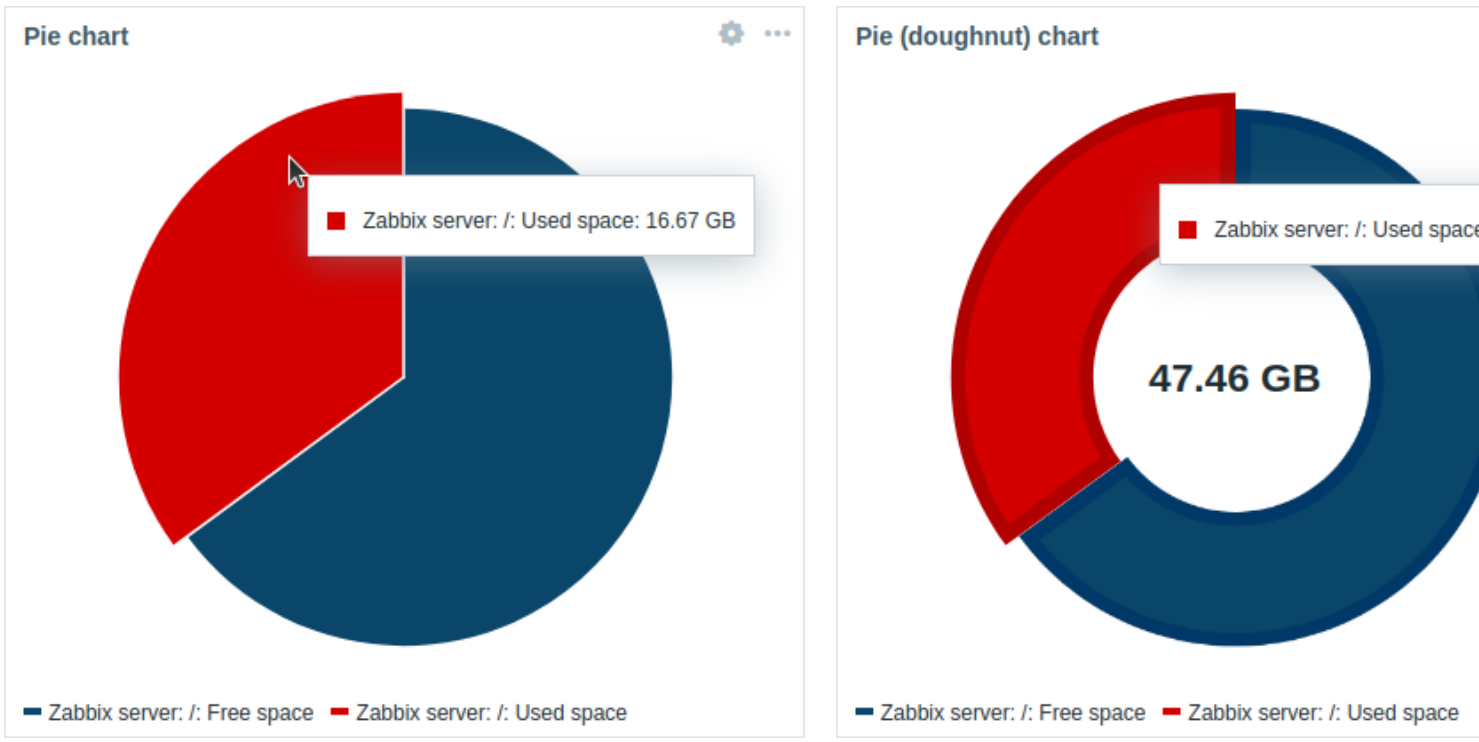
The following navigation tree element configuration parameters are available:

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.

21 Grafico a torta

Panoramica

Il widget *Grafico a torta* visualizza dati numerici degli item come un grafico a torta o ad anello basato su vettori. Questo widget può aiutarti a visualizzare come gli item o gli host contribuiscono al dataset complessivo.

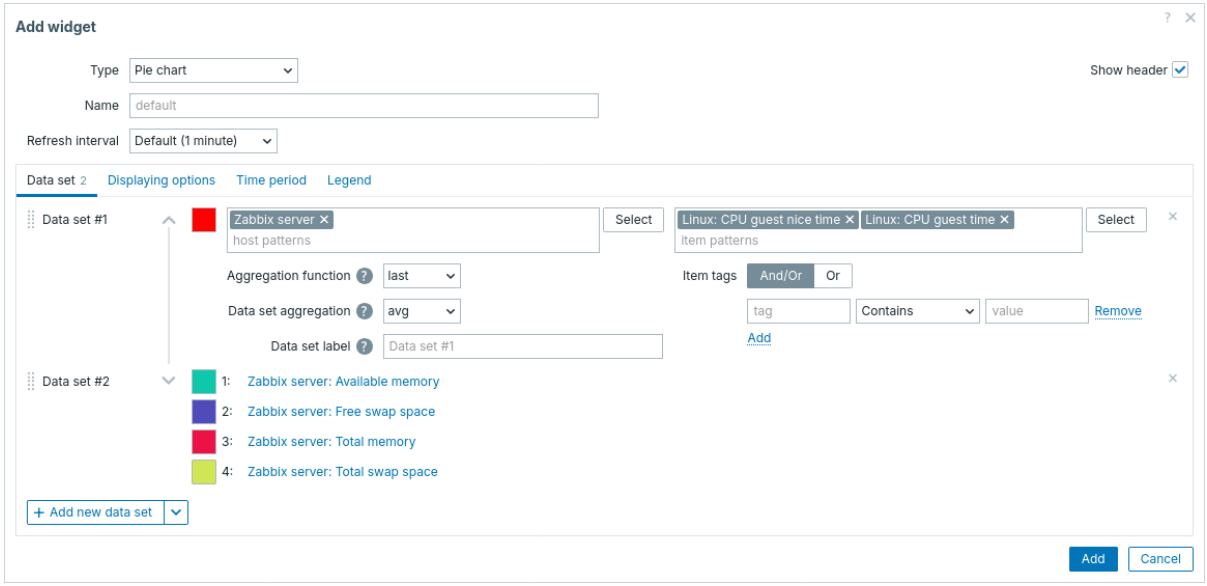


Passando il mouse sopra, il settore evidenziato si espande e mostra la relativa legenda; facendo clic sul settore, questo rimane espanso e la sua legenda resta visibile fino alla chiusura.

Le informazioni visualizzate nel widget *Grafico a torta* possono essere scaricate come immagine PNG selezionando l'opzione *Scarica immagine* nel menu del widget.

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 di item* - vengono visualizzati i dati degli item corrispondenti. È possibile scegliere un singolo colore di base oppure selezionare una riga della palette per assegnare colori distinti a ciascun item corrispondente.
- *Elenco di item* - vengono visualizzati i dati degli item selezionati. È possibile scegliere singolarmente il colore di ciascun item dal selettore.

Per impostazione predefinita, viene aggiunto un set di dati *Pattern di item*.

Set di dati	Per il set di dati **Pattern di item**: Selezionare o inserire i pattern di host e item; i dati degli item che corrispondono ai pattern immessi verranno visualizzati nel grafico a torta; possono essere visualizzati fino a 50 item. Per la selezione possono essere utilizzati pattern con caratteri jolly (ad esempio, * restituirà risultati che corrispondono a zero o più caratteri). Per specificare un pattern con caratteri jolly, inserire manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, quindi non è possibile aggiungere, ad esempio, un item denominato *item** singolarmente se esistono altri item corrispondenti (ad esempio, *item2*, *item3*). La specifica dei pattern di host e item è obbligatoria per i set di dati "Pattern di item". Vedere anche: Dettagli di 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*. L'elenco a discesa *Tipo* dopo il nome dell'item consente di selezionare il tipo di visualizzazione per ciascun item: **Normale** - il valore dell'item è rappresentato proporzionalmente nel grafico a torta (predefinito); **Totale** - il valore dell'item occupa l'intero grafico a torta. Si noti che può esistere un solo item "Totale" per grafico a torta e che 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 sotto) 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 previste per i singoli item. La specifica di item o widget è obbligatoria per i set di dati "Elenco di item". Vedere anche: Dettagli di configurazione del set di dati. Si noti 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 quale funzione di aggregazione utilizzare per ciascun item nel set di dati: **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 conteggio dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore (predefinito). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo (5 minuti, un'ora, un giorno) selezionato nella scheda *Periodo di tempo* o utilizzato per l'intera dashboard.
Aggregazione del set di dati	Specificare quale funzione di aggregazione utilizzare per l'intero set di dati: **not used** - nessuna aggregazione, gli item vengono visualizzati separatamente (predefinito); **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 conteggio dei valori. L'aggregazione consente di visualizzare un valore aggregato per l'intervallo (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 sono numerati, compresi 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 suo numero (ad esempio "Set di dati #2", "Set di dati #3", ecc.). La numerazione dei set di dati viene ricalcolata dopo il riordino/trascinamento dei set di dati. Le etichette dei set di dati troppo lunghe verranno abbreviate per adattarsi allo spazio disponibile nella visualizzazione (ad esempio "Numero di proc...").
Tag di item	Specificare i tag per filtrare gli 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 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). 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.

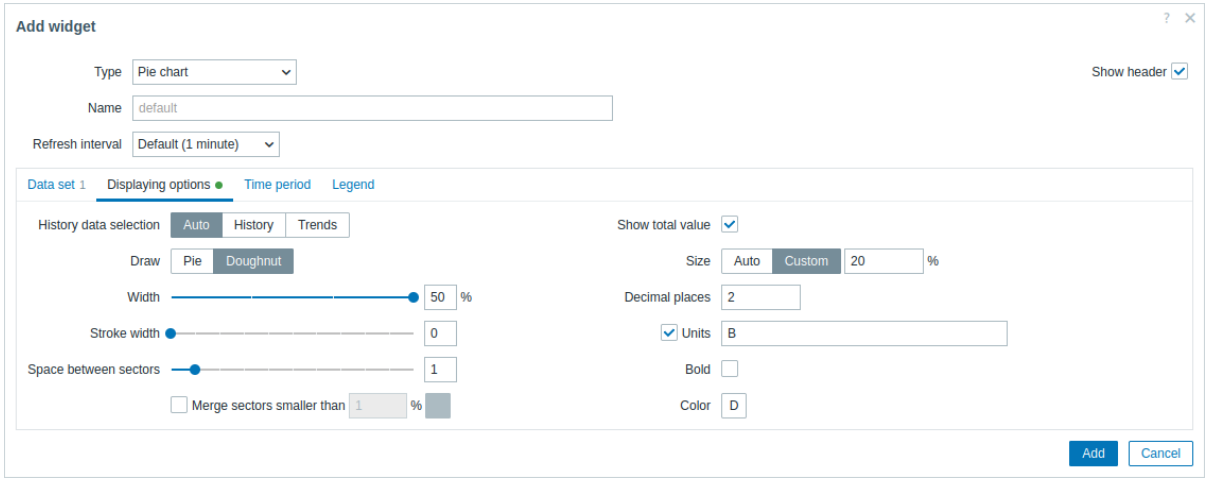
Dettagli di configurazione del set di dati

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

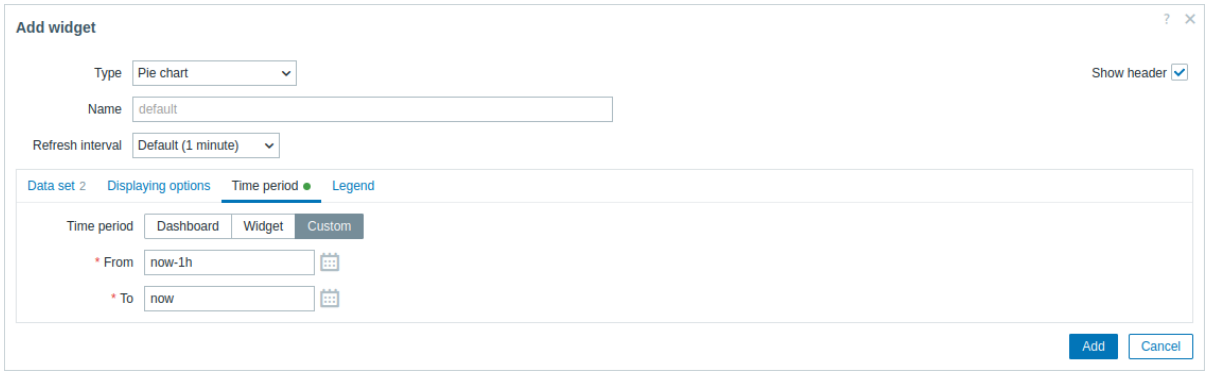
La scheda **Opzioni di visualizzazione** consente di definire la selezione dei dati storici e le opzioni di visualizzazione per il grafico a torta:



Selezione dei dati storici	Seleziona la sorgente dati: **Auto** - i dati vengono recuperati secondo il classico algoritmo (predefinito); **Cronologia** - dati dalla cronologia; **Trend** - dati dai trend.
Disegna	Seleziona lo stile di visualizzazione del grafico a torta: **Torta** - una torta completa (i settori occupano il 100% del raggio); **Ciambella** - una torta con un cerchio vuoto al centro (i settori utilizzano fino al 50% del raggio).
Spazio tra i settori	Seleziona la dimensione dello spazio (in unità da 0 a 10) tra i settori (il valore predefinito è ”1”).
Unisci i settori più piccoli di N%	Seleziona la casella per unire i settori più piccoli di N%. Se abilitato, seleziona il colore per i settori uniti e la soglia percentuale (N) per unire i settori piccoli.
Stile di disegno:	
Ciambella	
Larghezza	Seleziona la larghezza della ciambella: 20, 30, 40 o 50% (predefinito) del raggio.
Larghezza del bordo	Seleziona la larghezza del bordo del settore della ciambella (0-10).
Mostra valore totale	Seleziona la casella per mostrare il valore totale al centro del grafico a ciambella.
Dimensione	Seleziona l’opzione di dimensione per il valore totale: **Auto** - la dimensione del testo viene regolata automaticamente per adattarsi in modo leggibile al centro della ciambella; **Personalizzata** - specifica la dimensione del testo come percentuale dell’altezza totale del widget.
Posizioni decimali	Specifica il numero di posizioni decimali per il valore totale (0-6).
Unità	Specifica le unità per il valore totale.
Grassetto	Seleziona la casella per visualizzare il valore totale in grassetto.
Colore	Seleziona il colore per il valore totale.

Periodo di tempo

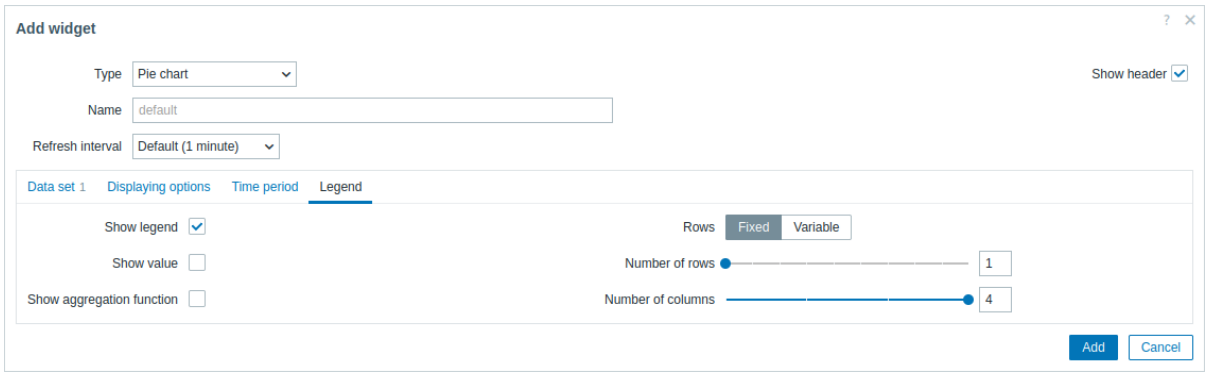
La scheda **Periodo di tempo** consente di impostare un periodo di tempo personalizzato per le impostazioni di aggregazione del grafico a torta:



Periodo di tempo	Selezionare 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 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, 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. 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	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 *Periodo di tempo* è impostato su "Personalizzato".

Legenda

La scheda **Legenda** consente di personalizzare la legenda del grafico a torta:



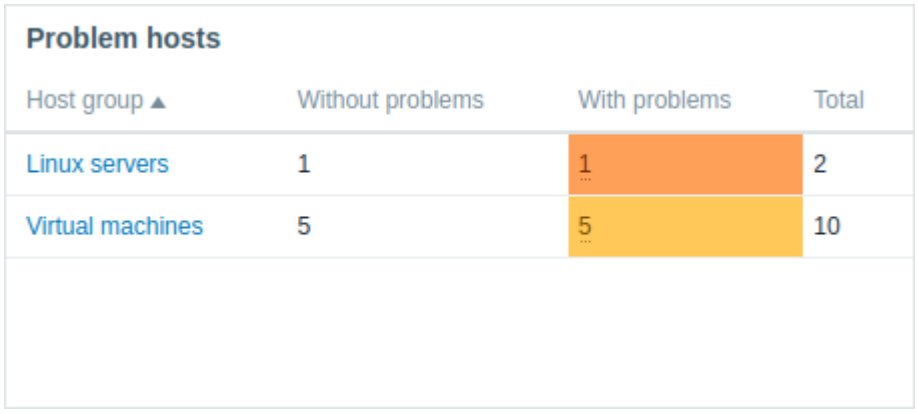
Mostra legenda	Deseleziona questa casella di controllo per nascondere la legenda nel grafico a torta (selezionata per impostazione predefinita).
Mostra valore	Seleziona questa casella di controllo per mostrare il valore dell'item nella legenda.
Mostra funzione di aggregazione	Seleziona questa casella di controllo per mostrare la funzione di aggregazione nella legenda.
Righe	Seleziona 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", imposta il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su "Variabile", imposta il numero massimo di righe della legenda da visualizzare (1-10).

Numero di colonne	Imposta il numero di colonne della legenda da visualizzare (1-4). Questo parametro è disponibile se *Mostra valore* non è selezionato.

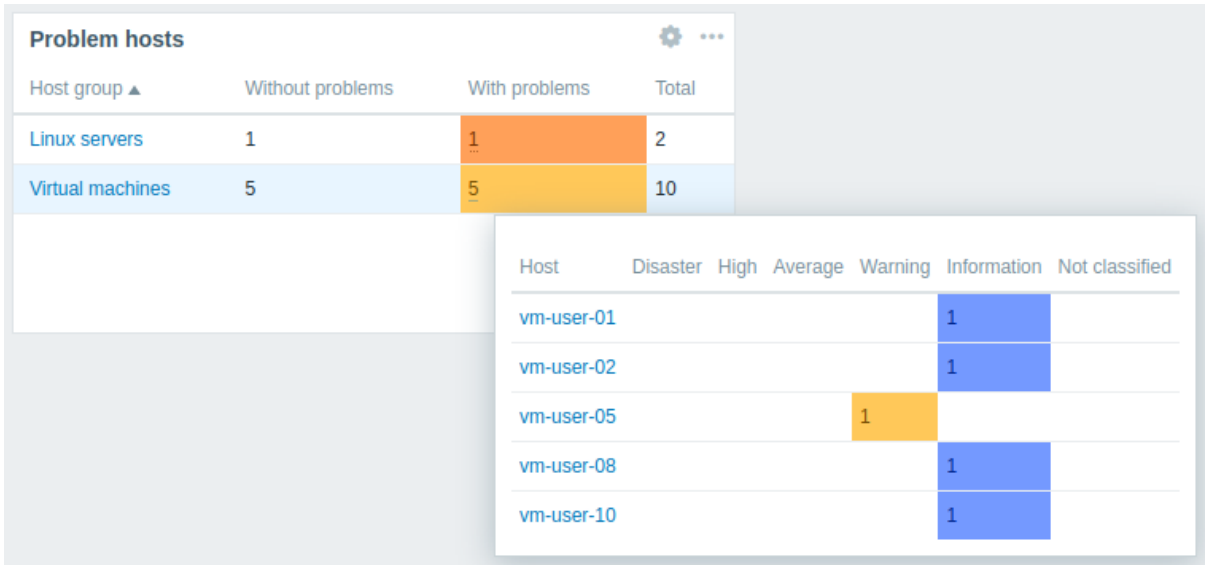
22 Host con problemi

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.).

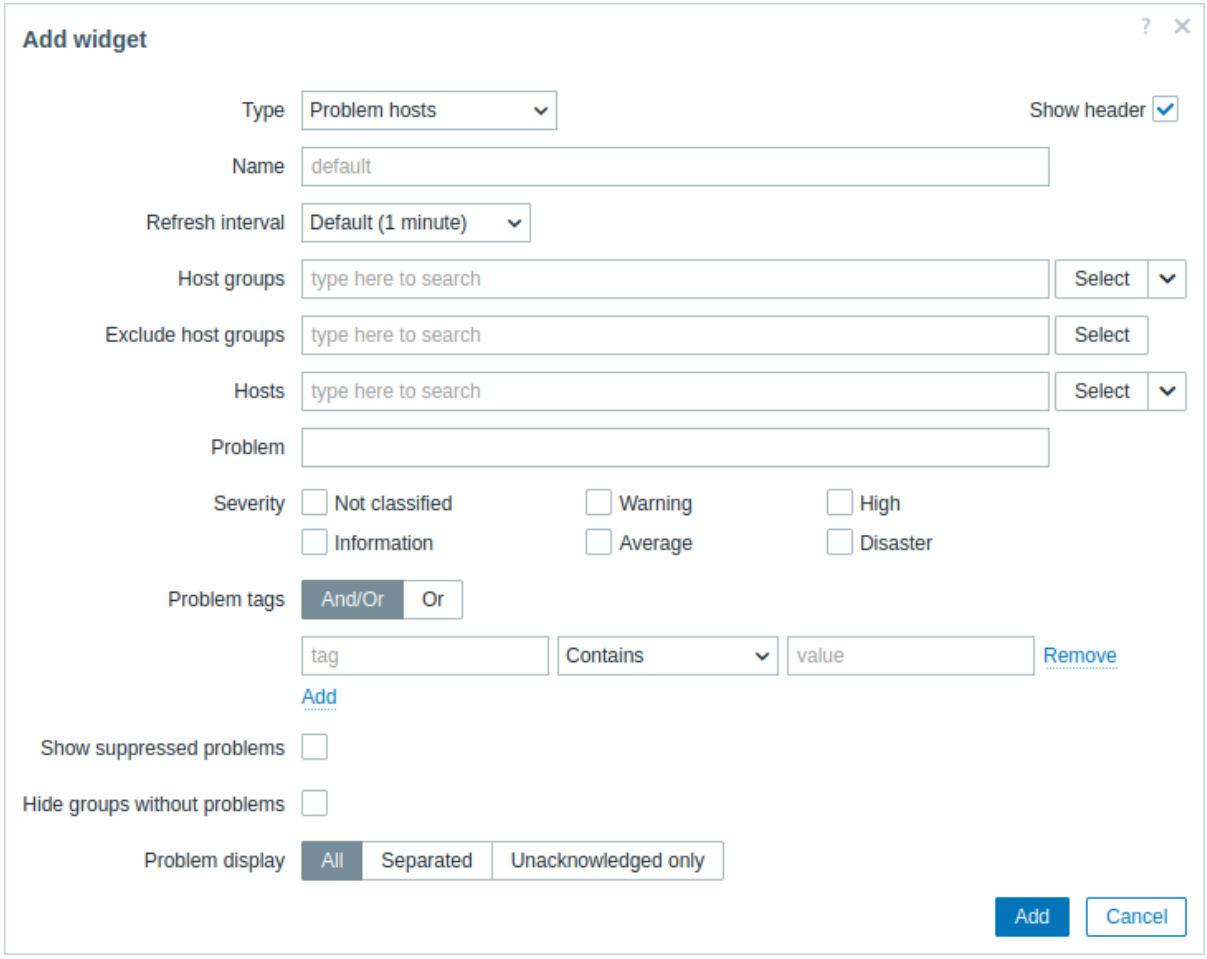


Passando il puntatore sul conteggio dei problemi viene mostrato l'elenco degli host con problemi:



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:

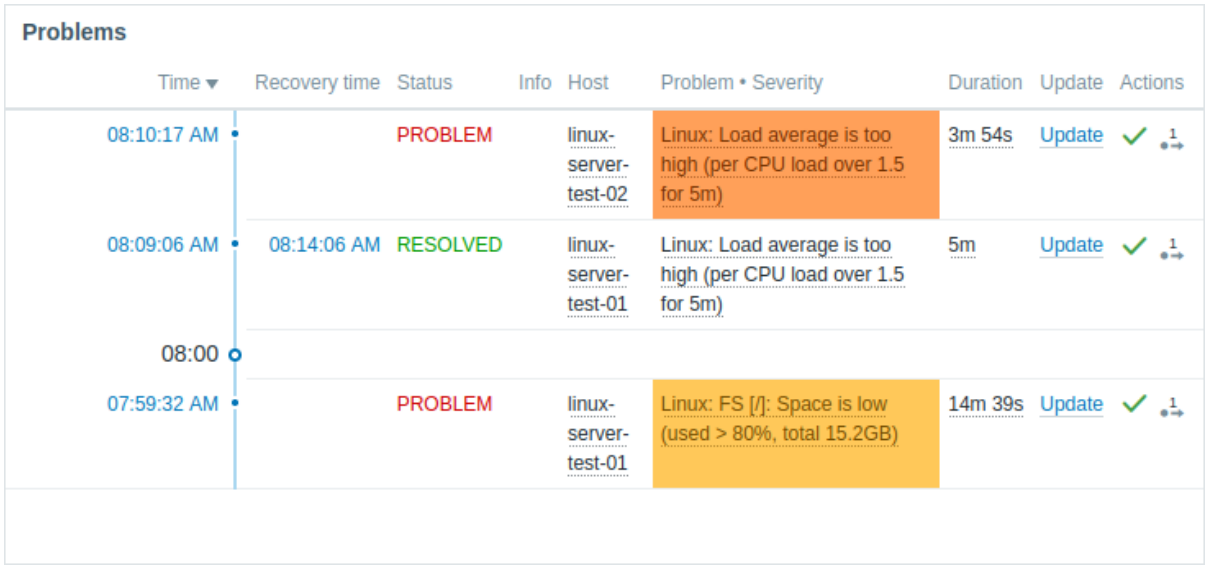
Host groups	Selezionare i gruppi di host da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un menu a discesa con i gruppi 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.

23 Problemi

Panoramica

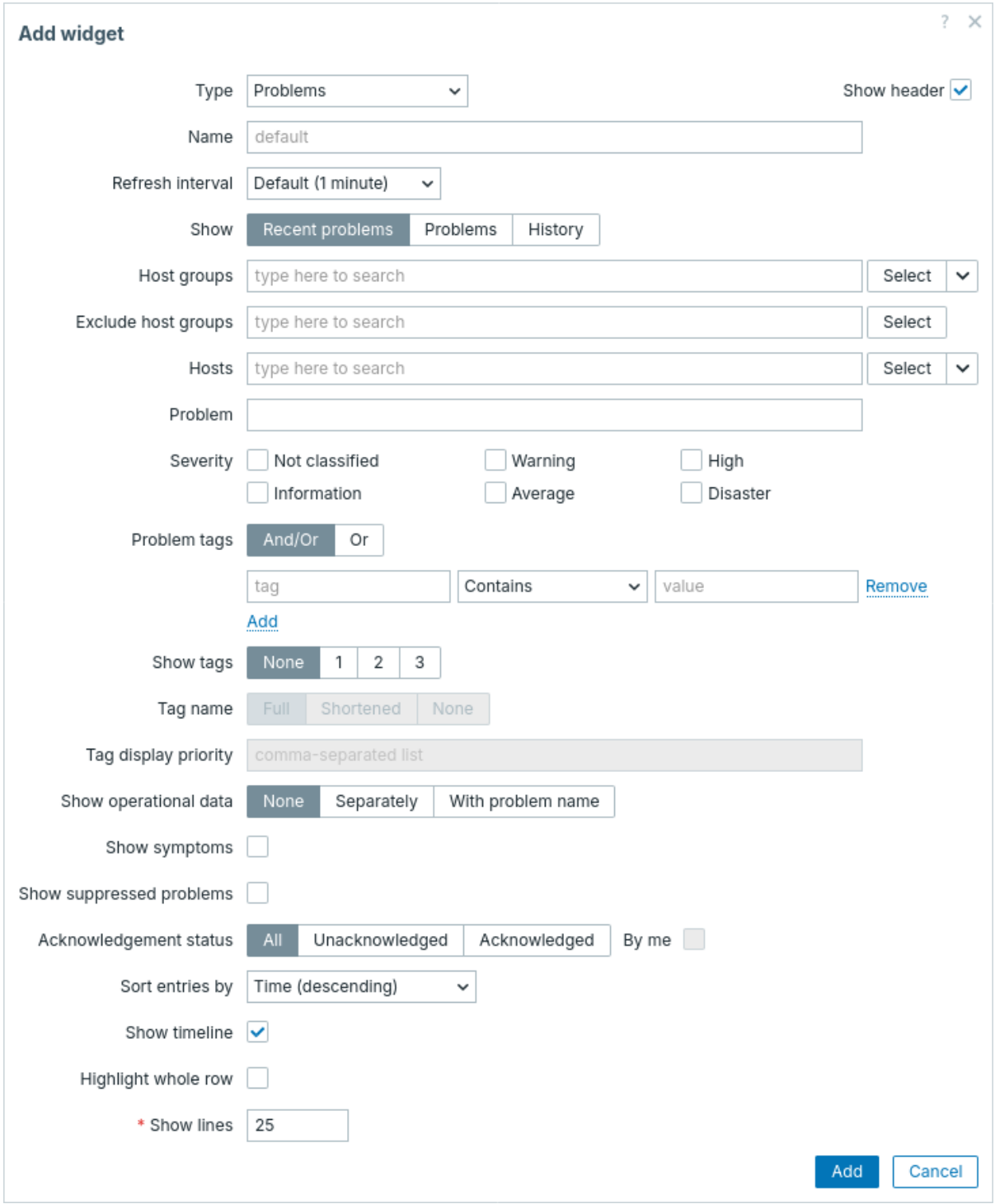
Il widget *Problems* visualizza i problemi correnti filtrati in base a diversi parametri (gruppi di host, host, nomi dei problemi, ecc.), offrendo una vista chiara di ciò che richiede attenzione.



Visualizza gli stessi dati di *Monitoring* > *Problems* e può mostrare fino a 1000 record.

Configurazione

Per configurare il widget, selezionare *Problems* come tipo:



È possibile limitare in vari modi quanti problemi vengono visualizzati nel widget: per stato del problema, nome del problema, gravità, gruppo host, host, tag evento, stato di conferma e così via.

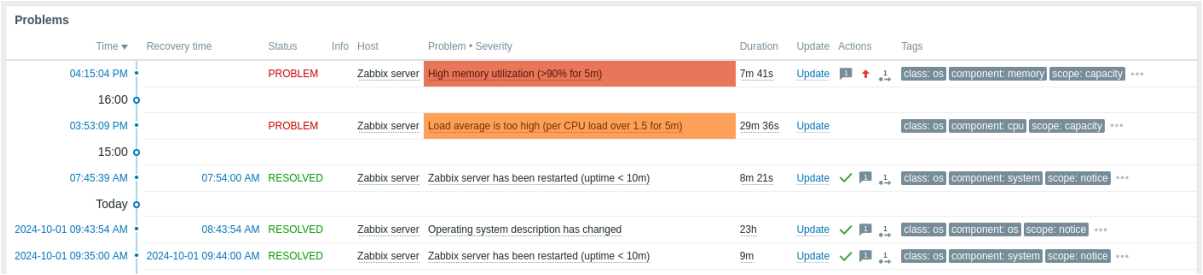
Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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	Selezionare i gruppi host di cui visualizzare i problemi 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 implicitamente selezionati tutti i gruppi host annidati. Nel widget verranno visualizzati i problemi di questi gruppi host; se non viene inserito alcun gruppo host, verranno visualizzati i problemi di tutti i gruppi host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.
Exclude host groups	Selezionare i gruppi host di cui nascondere i problemi 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 implicitamente selezionati tutti i gruppi host annidati. I problemi 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 selezioniamo contemporaneamente di *mostrare* il Gruppo A e di *escludere* il Gruppo B, nel widget verranno visualizzati solo i problemi dell'host 001. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.
Hosts	Selezionare gli host di cui visualizzare i problemi 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 i problemi di tutti gli host. Questo parametro non è disponibile durante la configurazione del widget in una dashboard template.
Problem	È possibile limitare il numero di problemi visualizzati in base al loro nome. Se si inserisce qui una stringa, verranno visualizzati solo i problemi il cui nome contiene la stringa inserita. Le macro non vengono espanse.
Severity	Selezionare le gravità del problema per filtrare i problemi da visualizzare nel widget. Se non viene selezionata alcuna gravità, verranno visualizzati tutti i problemi.
Problem tags	Specificare i tag del problema per limitare il numero di problemi 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. Sono disponibili diversi operatori per ogni condizione: **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 in cui i valori dei tag contengono la stringa inserita (corrispondenza per 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 in cui i valori dei 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 tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una condizione. Quando viene applicato il filtro, i tag qui specificati verranno visualizzati per primi insieme al problema, salvo diversa impostazione nell'elenco *Tag display priority* (vedere sotto).

Show tags	Selezionare 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, passare il mouse sull'icona con i tre puntini.
Tag name	Selezionare 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 vengono abbreviati a 3 simboli, ma i valori dei tag vengono visualizzati per intero; **None** - vengono visualizzati solo i valori dei tag, senza i nomi.
Tag display priority	Inserire la priorità di visualizzazione dei tag per un problema, come elenco di tag separati da virgole. Devono essere utilizzati solo i nomi dei tag, non i valori. Esempio: `Services,Applications,Application` I tag di questo elenco verranno sempre visualizzati per primi, sovrascrivendo l'ordinamento naturale alfabetico.
Show operational data	Selezionare la modalità di visualizzazione dei dati operativi: **None** - non vengono visualizzati dati operativi; **Separately** - i dati operativi vengono visualizzati in una colonna separata; **With problem name** - aggiunge i dati operativi al nome del problema, usando parentesi per i dati operativi.
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 (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 oppure solo i problemi confermati. Selezionare l'ulteriore casella di controllo per escludere i problemi che sono già stati confermati da voi.
Sort entries by	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 durante la configurazione del widget in una dashboard template.
Show timeline	Selezionare la casella di controllo per visualizzare una timeline visiva.
Highlight whole row	Selezionare la casella di controllo per evidenziare l'intera riga per i problemi non risolti. Per l'evidenziazione viene utilizzato il colore della gravità del problema. *Highlight whole row* non è disponibile nei temi ad alto contrasto.
Show lines	Specificare il numero di righe di problemi da visualizzare.

Utilizzo del widget



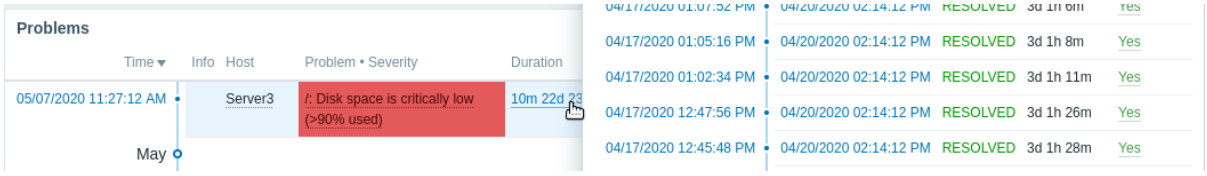
Il widget Problemi offre un accesso rapido a informazioni aggiuntive:

- Fare clic sulla data del problema (nella colonna Ora) o sulla data di ripristino (nella colonna Ora di ripristino) per visualizzare i dettagli dell'evento.
- Se il campo Info non è vuoto, passare il puntatore sull'icona visualizzata per vedere ulteriori dettagli.
- Fare clic sul nome del host per aprire il menu host.
- Fare clic sul nome del problema per aprire il menu evento.
- Passare il puntatore sopra o fare clic sulla durata del problema per visualizzare il popup dell'evento del problema.

- Premendo Aggiorna si apre una finestra Aggiorna problema.
- Passare il puntatore sopra o premere l'icona della freccia grigia nella colonna Azioni 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.



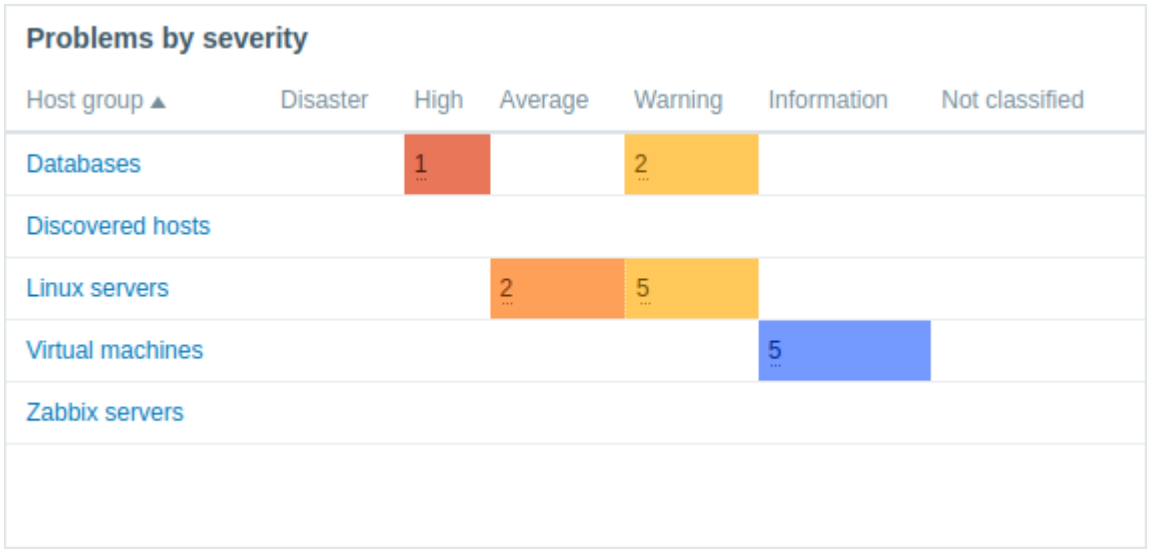
Per visualizzare il popup dell'evento del problema:

- Passare il mouse sulla durata del problema nella colonna *Durata* del widget *Problemi*. Il popup scompare non appena si sposta il mouse dalla durata.
- Fare clic sulla durata nella colonna *Durata* del widget *Problemi*. Il popup scompare solo se si fa nuovamente clic sulla durata.

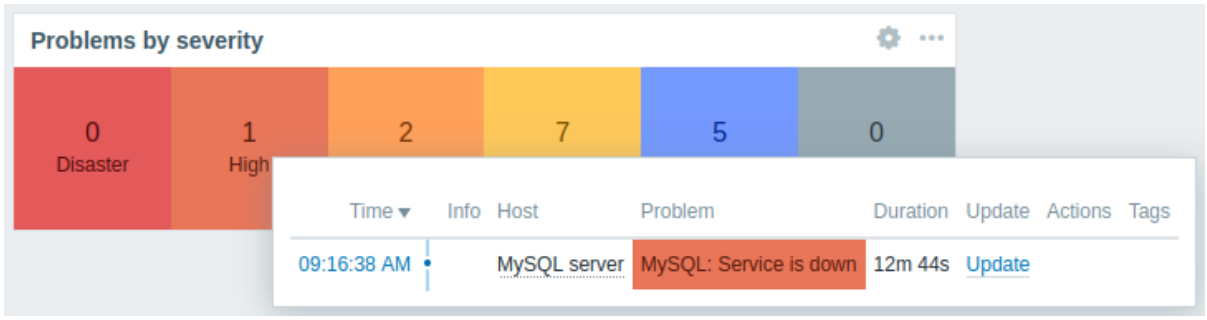
24 Problemi per gravità

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.).

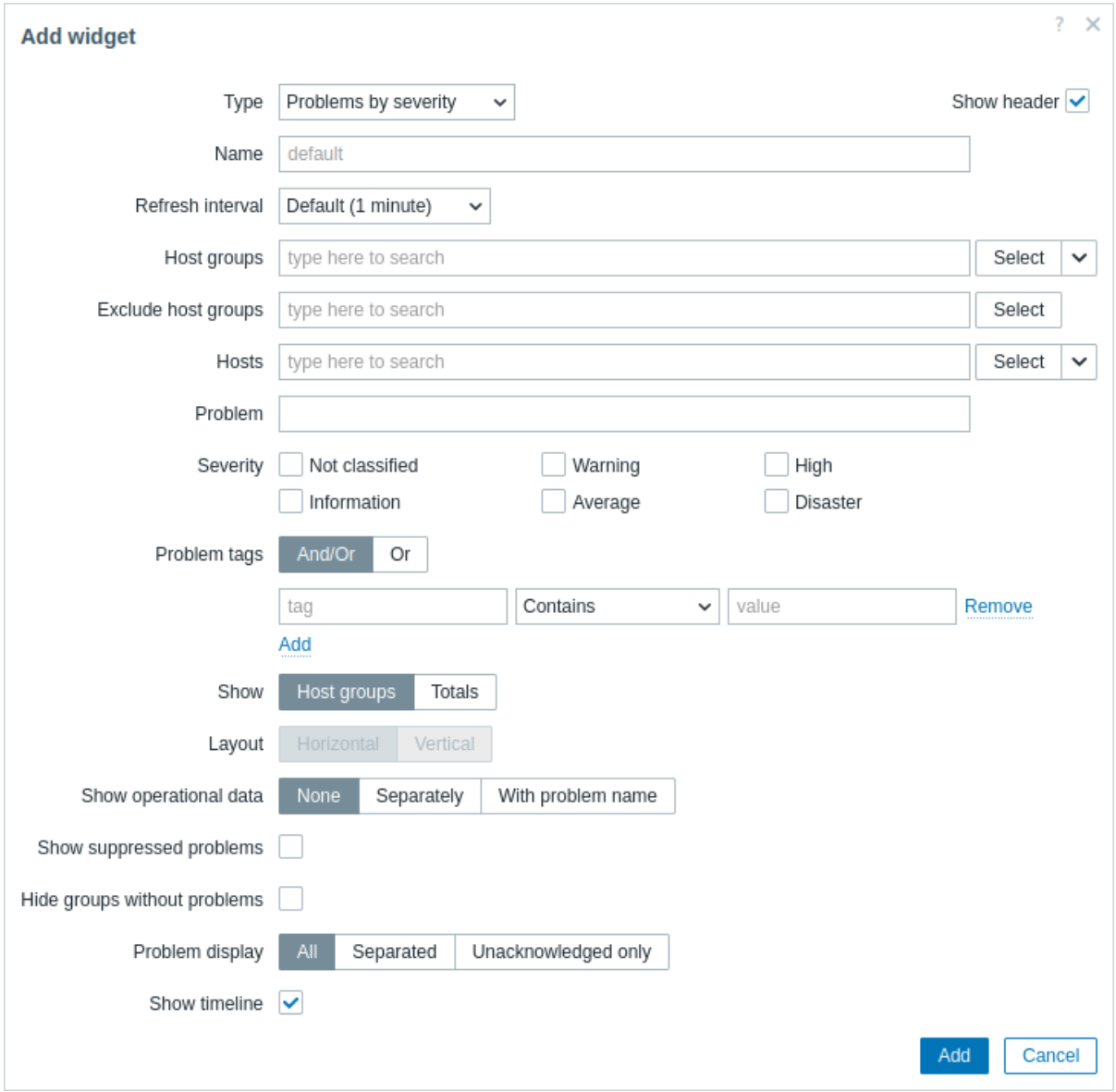


Passando il cursore sul conteggio di un problema viene mostrato l'elenco dei problemi correlati:



Configurazione

Per configurare il widget, selezionare *Problems by severity* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

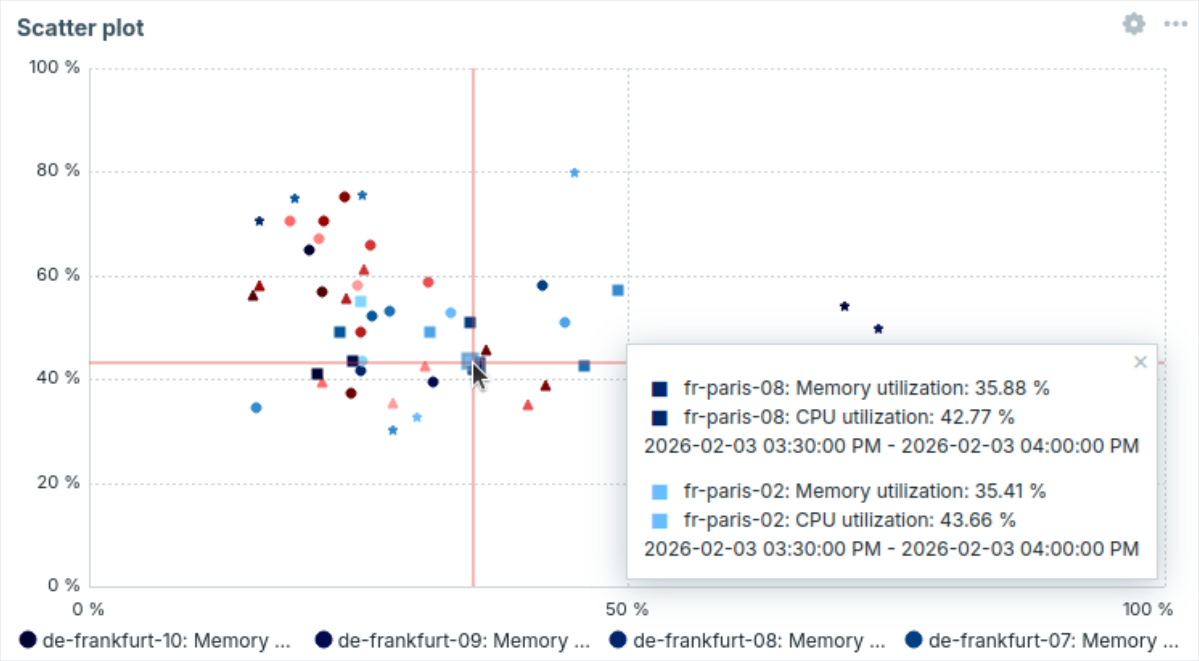
Host groups	Selezionare i gruppi di host da visualizzare nel widget. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà mostrato un elenco a discesa con i gruppi corrispondenti. Specificando un gruppo di host padre vengono selezionati implicitamente anche 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 durante la configurazione del 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 elenco a discesa con i gruppi corrispondenti. Specificando un gruppo di host padre vengono selezionati implicitamente anche 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 durante la configurazione del 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 elenco 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 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 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 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. 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 in cui i valori dei tag contengono la stringa inserita (corrispondenza per 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 in cui i valori dei 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 tag verranno raggruppate dalla condizione *Or*; **Or** - è sufficiente che sia soddisfatta una sola condizione.
Show	Selezionare l'opzione di visualizzazione: **Host groups** - visualizza i problemi per gruppo di host; **Totals** - visualizza un totale dei problemi per tutti i gruppi di host selezionati in blocchi colorati corrispondenti alla severità del problema. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di template 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	Visualizzare i dati operativi (vedere la descrizione di *Operational data* in *Monitoring* > *Problems*) come: **None** - i dati operativi non verranno visualizzati **Separately** - i dati operativi verranno visualizzati su una riga separata **With problem name** - i dati operativi verranno visualizzati insieme al nome del problema.
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 dei gruppi di host senza problemi. Questo parametro non è disponibile durante la configurazione del 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.
Show timeline	Selezionare la casella di controllo per visualizzare una timeline visiva.

25 Grafico a dispersione

Panoramica

Il widget *Scatter plot* visualizza la relazione tra due metriche tracciando singoli punti dati lungo un asse X e un asse Y. Questo aiuta a individuare pattern, cluster, correlazioni e valori anomali nel set di dati.



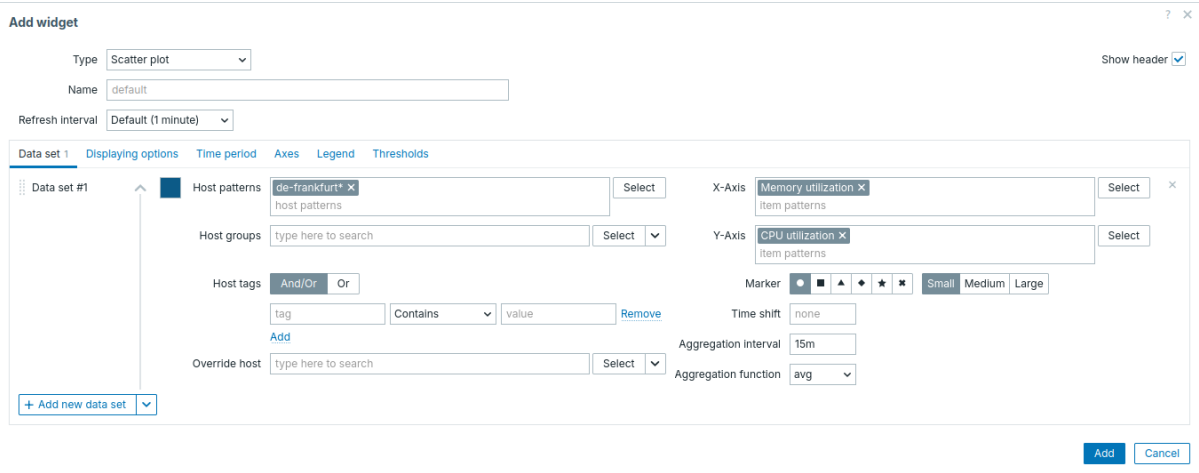
Passando il cursore su un punto dati viene visualizzato un tooltip con un elenco di item di quel punto dati e di altri punti entro un raggio di 6 px; ordinati prima in base ai valori dell'asse X dello scatter plot in ordine crescente e poi in base ai valori dell'asse Y in ordine decrescente.

Selezionando un item nel tooltip, i relativi dati vengono trasmessi ad altri widget.

Le informazioni visualizzate nel widget *Scatter plot* possono essere scaricate come immagine PNG selezionando l'opzione *Download image* nel menu del widget.

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:

- *Pattern 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 palette 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 singolarmente il colore di ciascun set di dati dal selettore.

I punti dati vengono creati solo per quelle coppie host/item che hanno almeno un valore su entrambi gli assi.

Per impostazione predefinita, viene aggiunto un set di dati *Pattern di item*.

Set di dati	Per il set di dati **Pattern di item**: Selezionare o inserire pattern di host e item; i dati degli item che corrispondono a questi pattern verranno visualizzati nel grafico a dispersione; possono essere visualizzati fino a 100 item (50 su ciascun asse per un totale di 50 host). Per la selezione possono essere utilizzati pattern con caratteri jolly (ad esempio, * restituirà risultati che corrispondono a zero o più caratteri). Per specificare un pattern con caratteri jolly, inserire manualmente la stringa e premere *Invio*. Il simbolo jolly viene sempre interpretato, quindi non è possibile aggiungere singolarmente, ad esempio, un item denominato *item** se esistono altri item corrispondenti (ad esempio, *item2*, *item3*). La specifica dei pattern di host e item è obbligatoria per i set di dati "Pattern di item". Vedere anche: Dettagli di 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 anche 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 di configurazione del set di dati. Si noti 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.
Gruppi di host	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 proposto un menu a discesa con i gruppi corrispondenti. La specifica di un gruppo di host padre seleziona implicitamente 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 è disponibile solo per i set di dati *Pattern 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 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 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 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. Questo parametro è disponibile solo per i set di dati *Pattern di item* e non è disponibile quando si configura il widget in una dashboard di template.

Sostituisci 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.
Marcatore	Scegliere il tipo e la dimensione del marcatore per i punti dati degli host nel set di dati.
Spostamento temporale	Specificare uno spostamento temporale, se necessario. In questo campo è possibile utilizzare i suffissi temporali. Sono consentiti valori negativi.
Funzione di aggregazione	Specificare quale funzione di aggregazione utilizzare: **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 conteggio dei valori (se esiste almeno un valore su entrambi gli assi); **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore; **none** - visualizza tutti i valori (nessuna aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo scelto (5 minuti, un'ora, un giorno), invece di tutti i valori. 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 verrà considerato in secondi. Si noti che se il widget è configurato per visualizzare dati storici basati su trends (*Selezione dati storici* è impostata su *Trends* o *Auto*), si consiglia di utilizzare un intervallo di aggregazione che sia un multiplo di 1 ora (ad esempio, 3600, 60m, 1h, 3h, ecc.). I trends memorizzano valori aggregati orari, quindi l'utilizzo di un intervallo di aggregazione che non sia un multiplo di 1 ora (ad esempio, 100s, 7min, 15min, 90min, ecc.) può portare a 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 su X e 1 su Y nell'intervallo specificato.

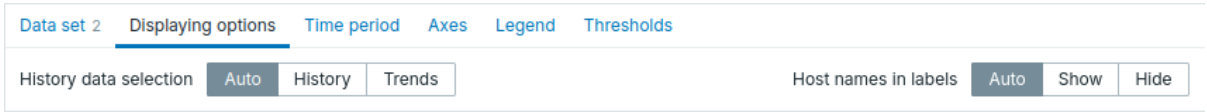
Dettagli della configurazione del set di dati

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 dello storico:

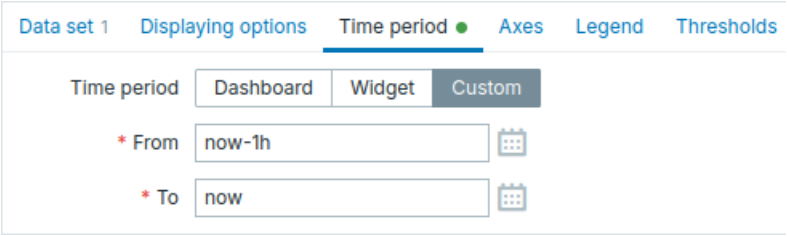


Selezione dei dati dello storico	Imposta la sorgente dei dati del grafico a dispersione: **Auto** - i dati vengono recuperati in base all'algoritmo del grafico classico (predefinito). **Storico** - dati dallo storico. **Trend** - dati dai trend.

Nomi degli host nelle etichette	Scegli se visualizzare i nomi degli host nella legenda del grafico e nel tooltip che appare passando il cursore sui dati del grafico: **Auto** - i nomi degli host vengono visualizzati solo se nei set di dati è presente più di un host (predefinito). **Mostra** - i nomi degli host vengono visualizzati. **Nascondi** - i nomi degli host vengono nascosti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard del template.

Periodo di tempo

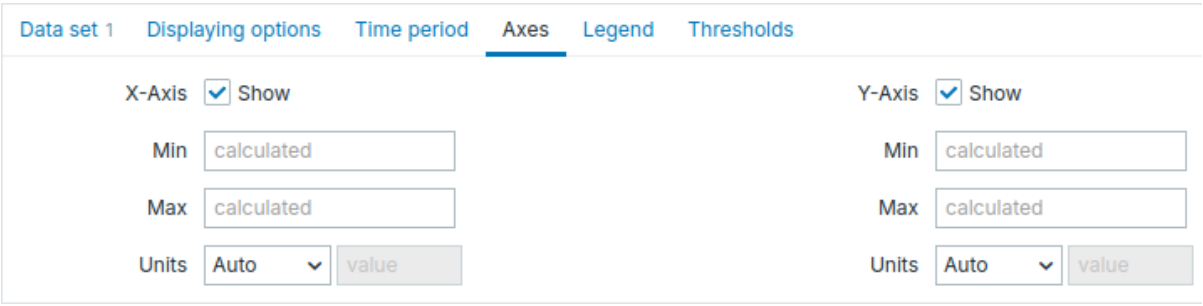
La scheda **Periodo di tempo** consente di impostare un periodo di tempo per il quale visualizzare i dati nel grafico a dispersione:



Periodo di tempo	Selezionare 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 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. 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	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 *Periodo di tempo* è impostato su "Personalizzato".

Assi

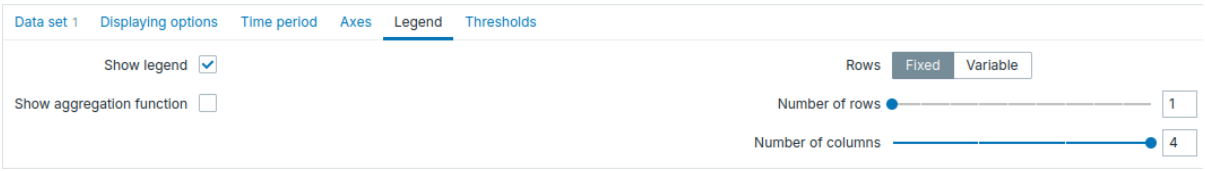
La scheda **Assi** consente di personalizzare la visualizzazione degli assi:



Asse X/Asse Y	Deseleziona questa casella di controllo per nascondere l'asse X/l'asse Y (selezionata per impostazione predefinita).
Min	Imposta il valore minimo dell'asse corrispondente.
Max	Imposta il valore massimo dell'asse corrispondente.
Unità	Scegli l'unità per i valori degli assi del grafico a dispersione dal menu a discesa: **Auto** - 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

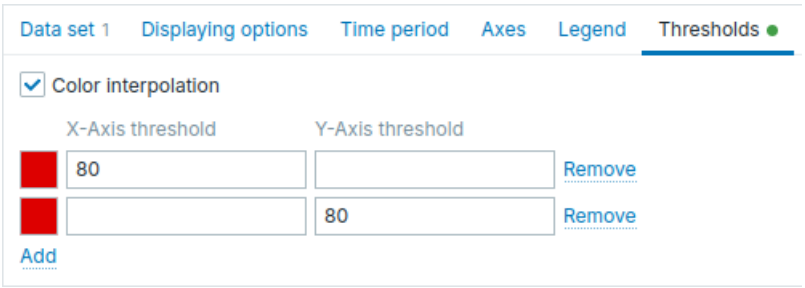
La scheda **Legenda** consente di personalizzare la legenda del grafico a dispersione:



Mostra legenda	Deseleziona questa casella di controllo per nascondere la legenda nel grafico a dispersione (selezionata per impostazione predefinita).
Mostra funzione di aggregazione	Seleziona questa casella di controllo per mostrare la funzione di aggregazione nella legenda.
Righe	Seleziona 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”, imposta il numero di righe della legenda da visualizzare (1-10). Se *Righe* è impostato su ”Variabile”, imposta il numero massimo di righe della legenda da visualizzare (1-10).
Numero di colonne	Imposta 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 dei marcatori degli host:

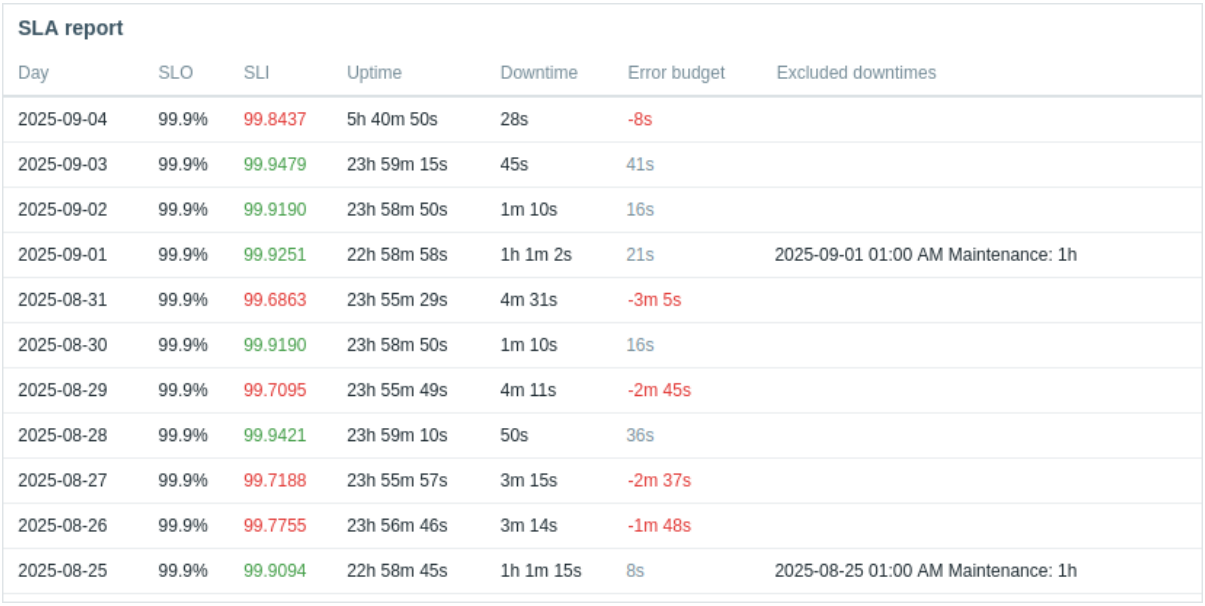


Interpolazione dei colori	Selezionare la casella di controllo per abilitare una transizione fluida tra i colori delle soglie per i marcatori degli host. Questo parametro si applica se sono impostate due o più soglie per l’asse X, per l’asse Y oppure per entrambi gli assi contemporaneamente.
Soglia asse X/asse Y	Fare clic su *Aggiungi* per aggiungere una soglia, selezionare un colore della soglia dal selettore colori e specificare un valore numerico. Le soglie possono essere applicate a un singolo asse, a entrambi gli assi separatamente oppure a entrambi gli assi contemporaneamente. Ad esempio, una singola soglia con X=80 e Y=80 colorerà i marcatori quando entrambe le condizioni sono soddisfatte. L’elenco delle soglie verrà ordinato in ordine crescente al momento del salvataggio. Sono supportati i suffissi (ad esempio, ”1d”, ”2w”, ”4K”, ”8G”). Sono supportate le mappature dei valori.

26 Report SLA

Panoramica

Il widget *Report SLA* visualizza i report SLA, aiutandoti a monitorare le prestazioni del servizio rispetto agli obiettivi definiti.



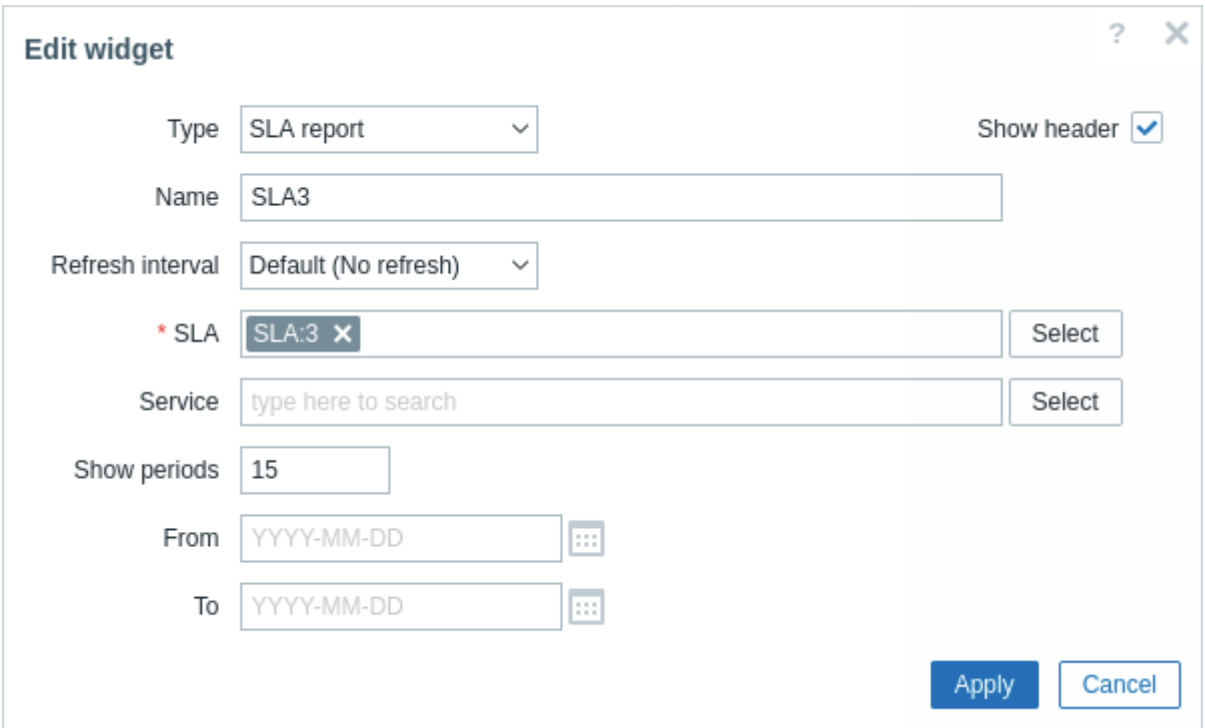
SLA report

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:



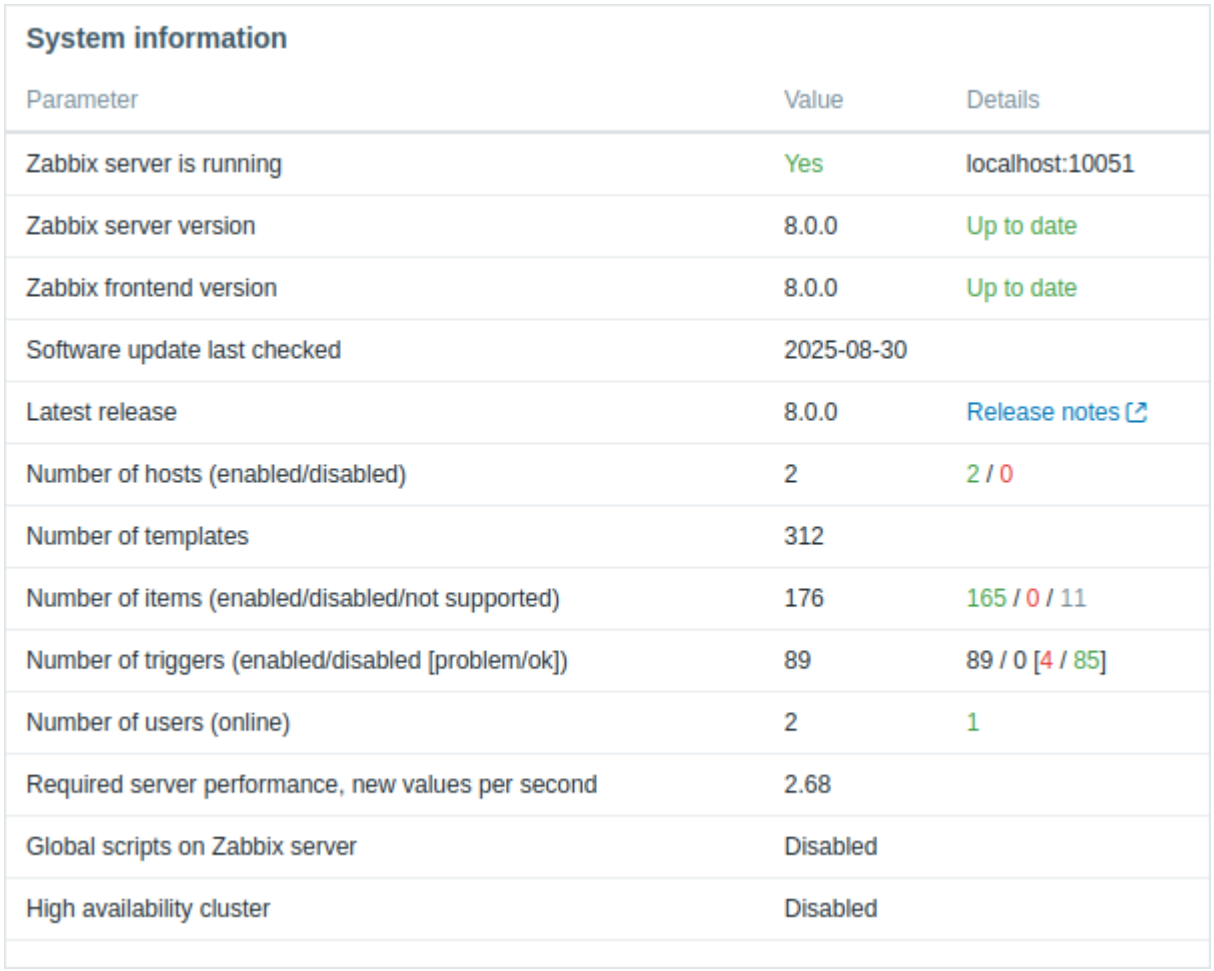
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

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.

27 Informazioni di sistema

Panoramica

Il widget *Informazioni di sistema* visualizza un riepilogo dei principali dati di sistema e del server Zabbix oppure i dettagli dei nodi ad alta disponibilità.



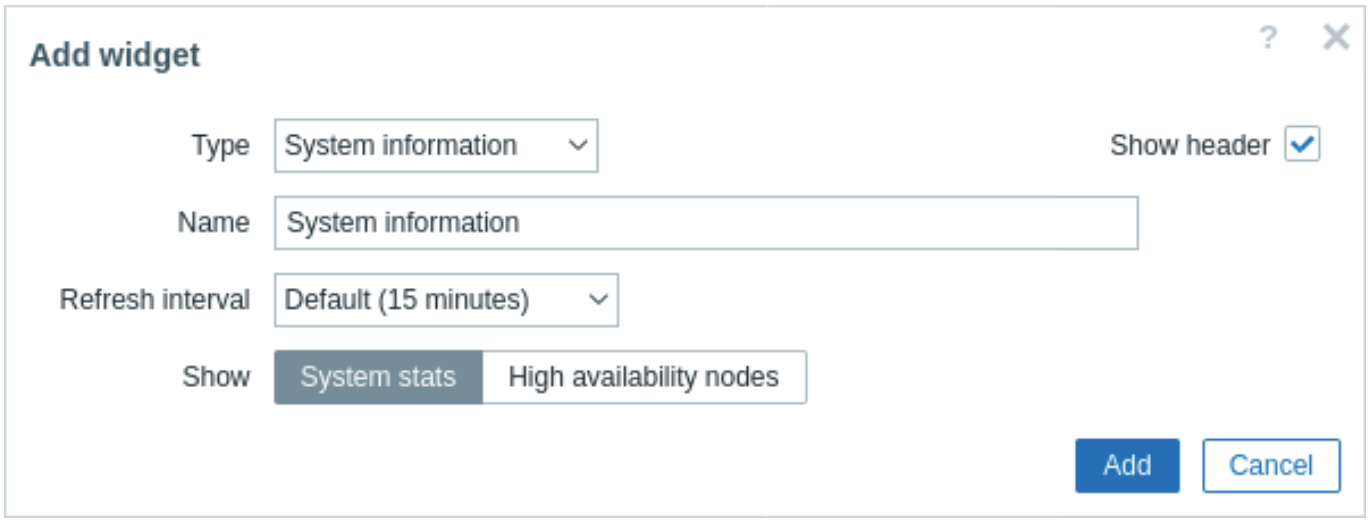
System information

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	

Visualizza gli stessi dati di *Report* > *Informazioni di sistema*.

Configurazione

Per configurare, selezionare *Informazioni di sistema* come tipo:



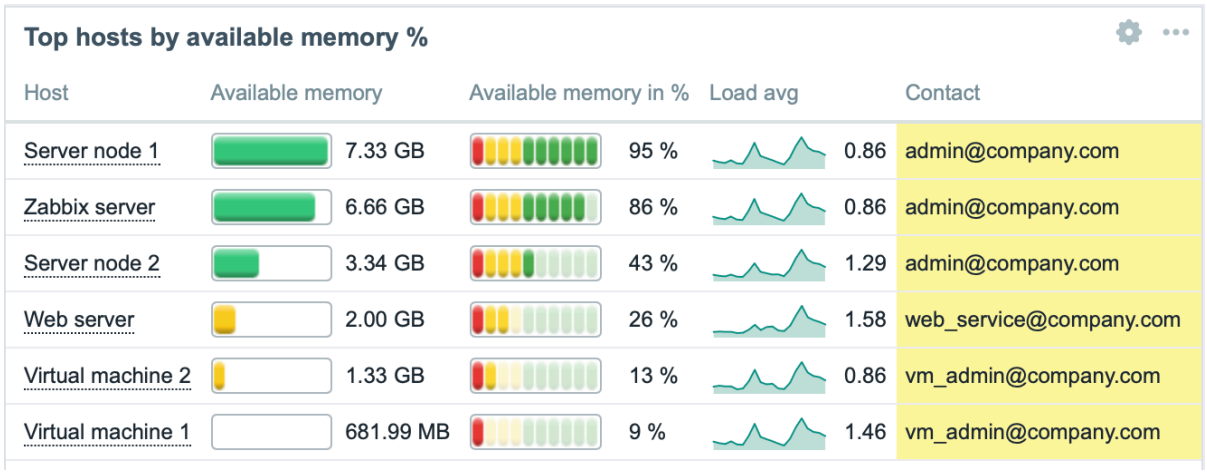
Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

Mostra	Selezionare cosa visualizzare: **Statistiche di sistema** - visualizza un riepilogo dei principali dati 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 software di Zabbix. Questa opzione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione del server Zabbix e nel campo *Mostra* è selezionato "Statistiche di sistema".

28 Host principali

Panoramica

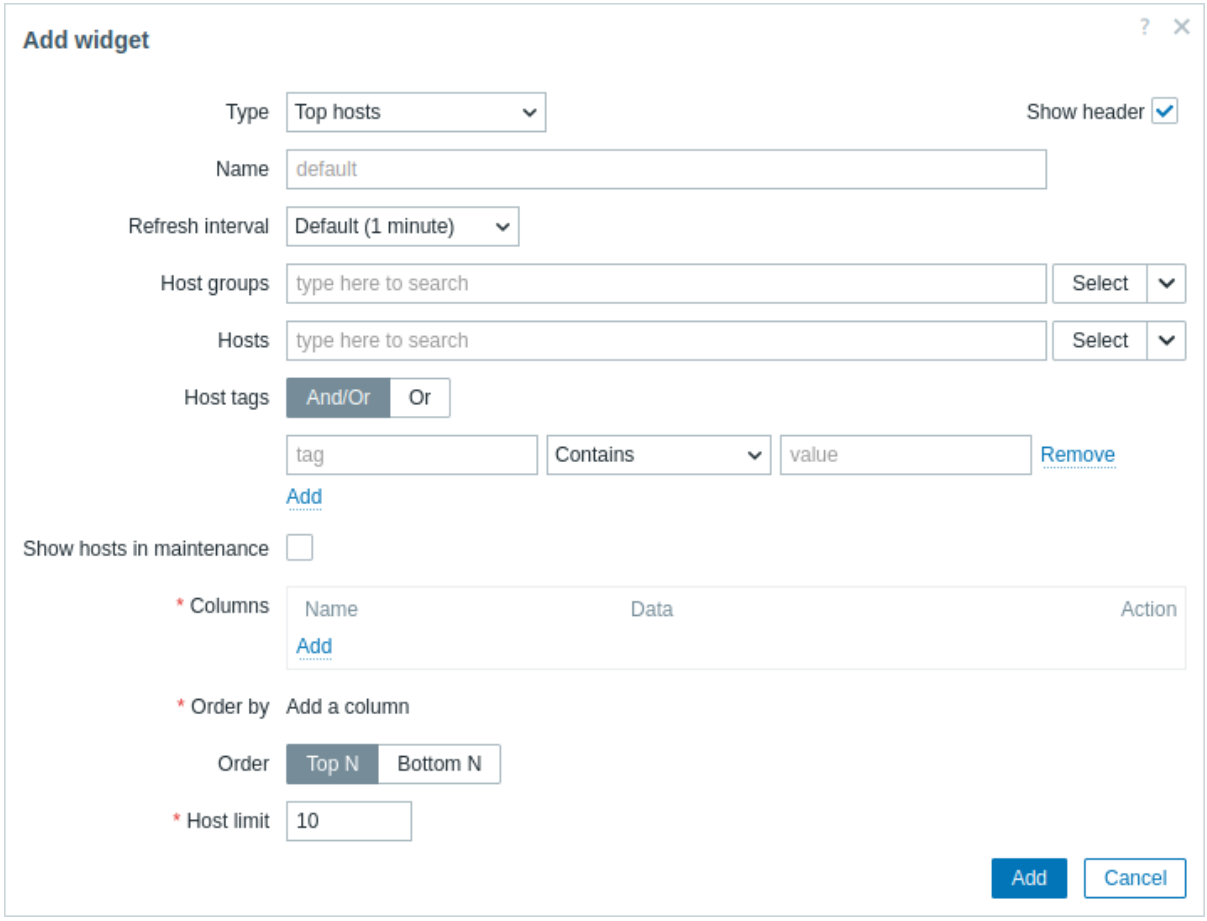
Il widget *Top hosts* visualizza i valori massimi/minimi per gli item selezionati da più host.



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.

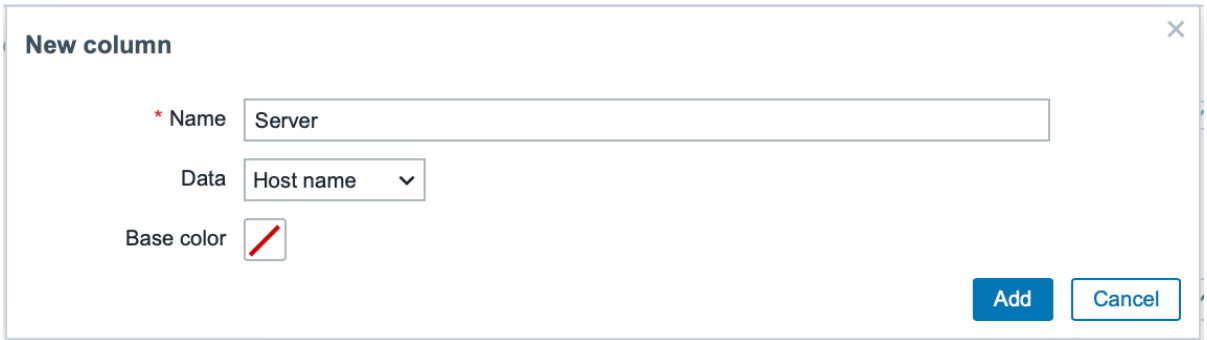
Tag host	Specificare i tag per limitare il numero di host visualizzati nel widget. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi 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.



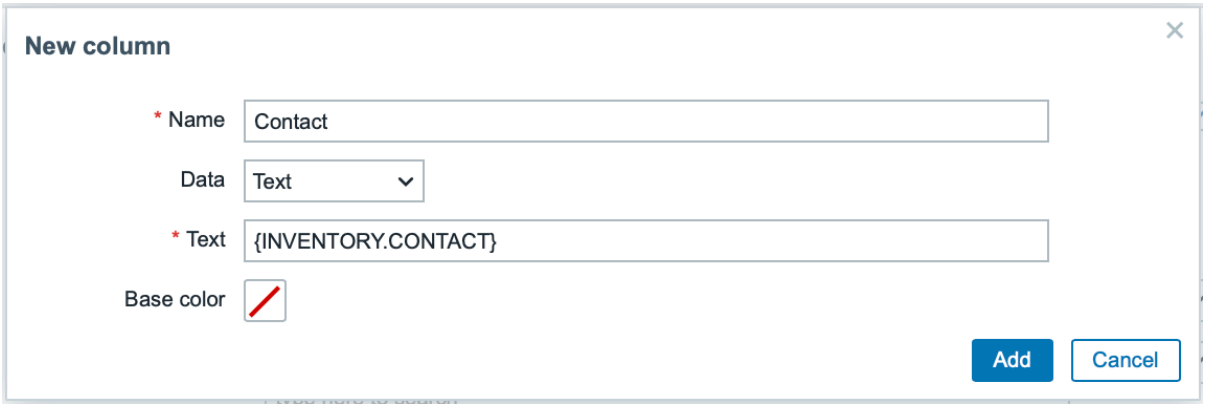
Parametri supportati:

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.

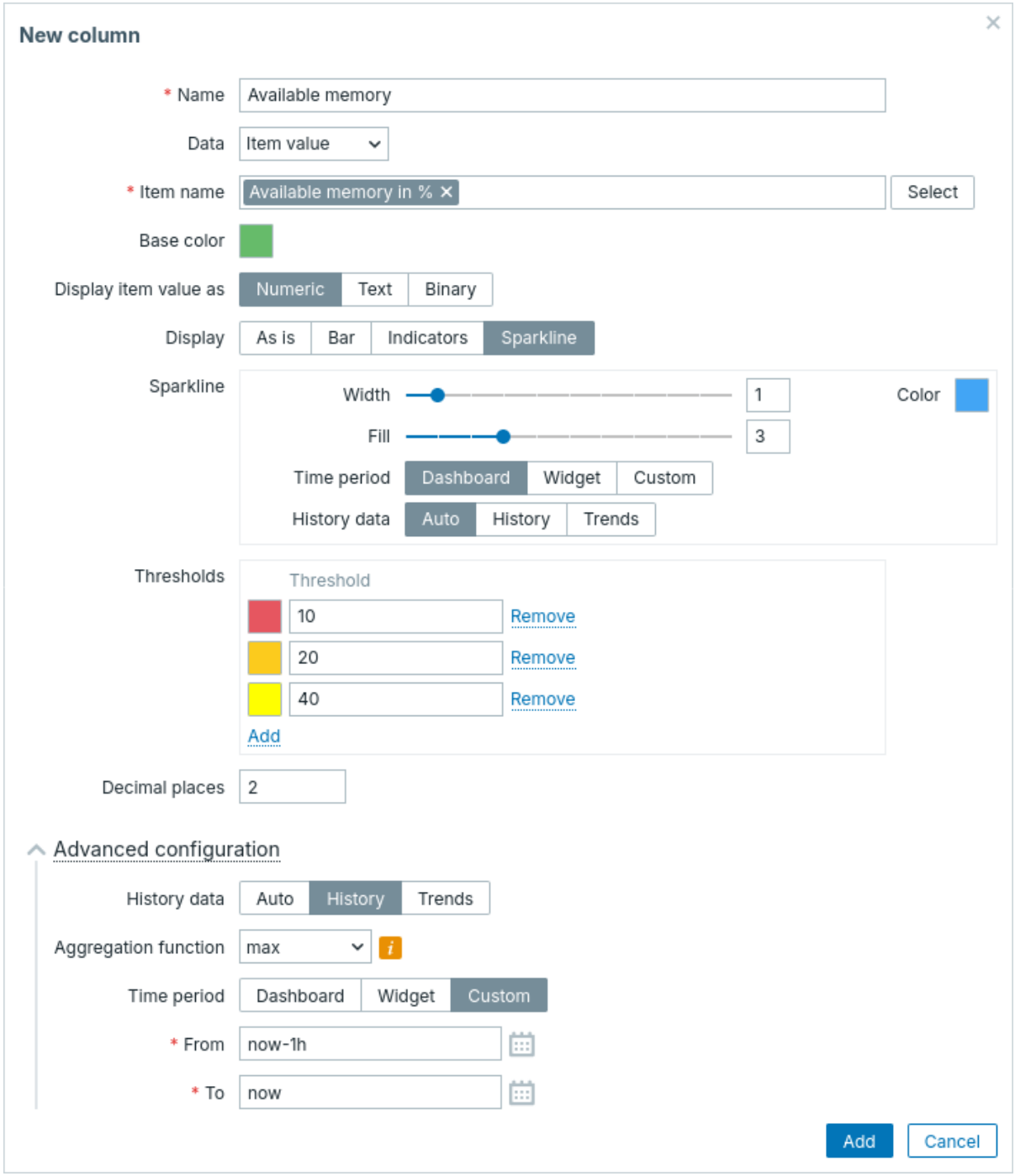


Parametri supportati:

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 valore item

La colonna valore item viene utilizzata per visualizzare il valore dell'item specificato; supporta più formati e opzioni di visualizzazione del valore.



Parametri supportati:

Name	Nome della colonna.
Data	Tipo di dati da visualizzare nella colonna; selezionare **Item value**.
Item name	Selezionare un item; il suo nome viene utilizzato per trovare e visualizzare gli item con lo stesso nome su tutti gli host selezionati. Se un host ha più item di questo tipo, il widget visualizza l'item la cui chiave viene per prima in ordine alfabetico (ad esempio, *CPU utilization* con chiave `proc.cpu.util` invece di `system.cpu.util`). Quando si configura il widget in una dashboard di 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* sono 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 per la visualizzazione del valore dell'item: **Numeric**, **Text** o **Binary**. L'opzione selezionata determina quali parametri aggiuntivi saranno disponibili. Fare riferimento all'elenco dei parametri per ciascun formato riportato di seguito.
Advanced configuration	

History data	Preleva 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 valori item numerici. Per i valori di testo e binari, i dati verranno sempre prelevati dalla cronologia.
Aggregation function	Specificare quale funzione di aggregazione utilizzare: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il conteggio dei valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore; **not used** - visualizza il valore più recente (nessuna aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo scelto (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.
Time period	Specificare il periodo di tempo da utilizzare per l'aggregazione dei valori. Selezionare 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*); **Custom** - usa un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, verrà visualizzata un'icona a forma di orologio nell'angolo superiore destro del widget, che indicherà l'ora impostata al passaggio del mouse. 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. Questo parametro non è disponibile se *Aggregation function* è impostato su "not used".
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. 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	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 *Time period* è impostato su "Custom".

Parametri del valore item numerico:

Display	Definire come deve essere visualizzato il valore: **As is** - come testo normale; **Bar** - come barra piena riempita di colore; **Indicators** - come barra segmentata riempita di colore; **Sparkline** - mini grafico a linee.
Thresholds	Specificare i valori di soglia ai quali il colore di sfondo/riempimento deve cambiare. L'elenco verrà ordinato in ordine crescente al salvataggio. Per gli sparkline, le soglie vengono applicate solo all'ultimo valore dell'item.
Decimal places	Specificare quante cifre decimali devono essere visualizzate con il valore.
Bar/Indicators	
Min	Valore minimo.
Max	Valore massimo.
Sparkline	
Width	Impostare lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.
Color	Selezionare il colore della linea e del riempimento.
Fill	Impostare il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo manualmente un valore nell'intervallo da 0 a 10.

Time period	Specificare il periodo di tempo dei valori da includere nel grafico sparkline. Selezionare 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*); **Custom** - usa un periodo di tempo personalizzato impostato nei parametri *From* e *To*; se impostato, verrà visualizzata un'icona a forma di orologio nell'angolo superiore destro del widget, che indicherà l'ora impostata al passaggio del mouse. 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. 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	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 *Time period* è impostato su "Custom".
History data	Preleva 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 item di testo:

Highlights	Specificare le espressioni regolari alla cui corrispondenza il colore di sfondo/riempimento deve cambiare.

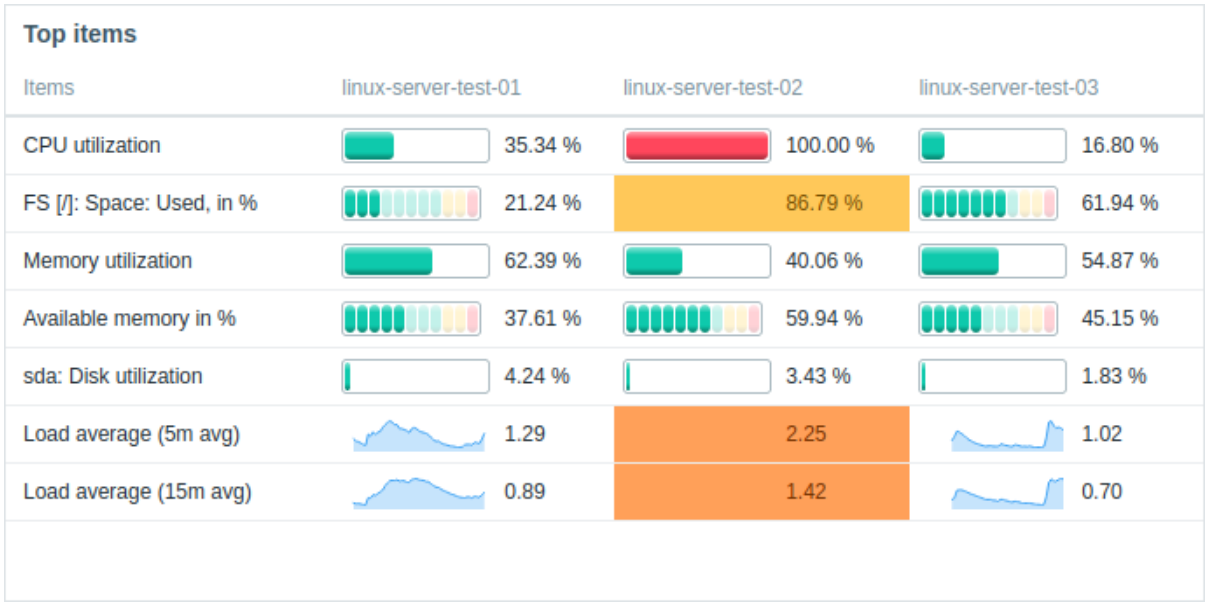
Parametri del valore item binario:

Show thumbnail	Specificare se creare e visualizzare una miniatura per l'immagine contenente dati binari oppure visualizzare un collegamento ipertestuale *Show* che porta all'immagine a dimensione intera nella colonna del valore.

29 Item principali

Panoramica

Il widget *Top items* visualizza i valori massimi/minimi per gli item selezionati, fornendo una rapida panoramica delle loro prestazioni.

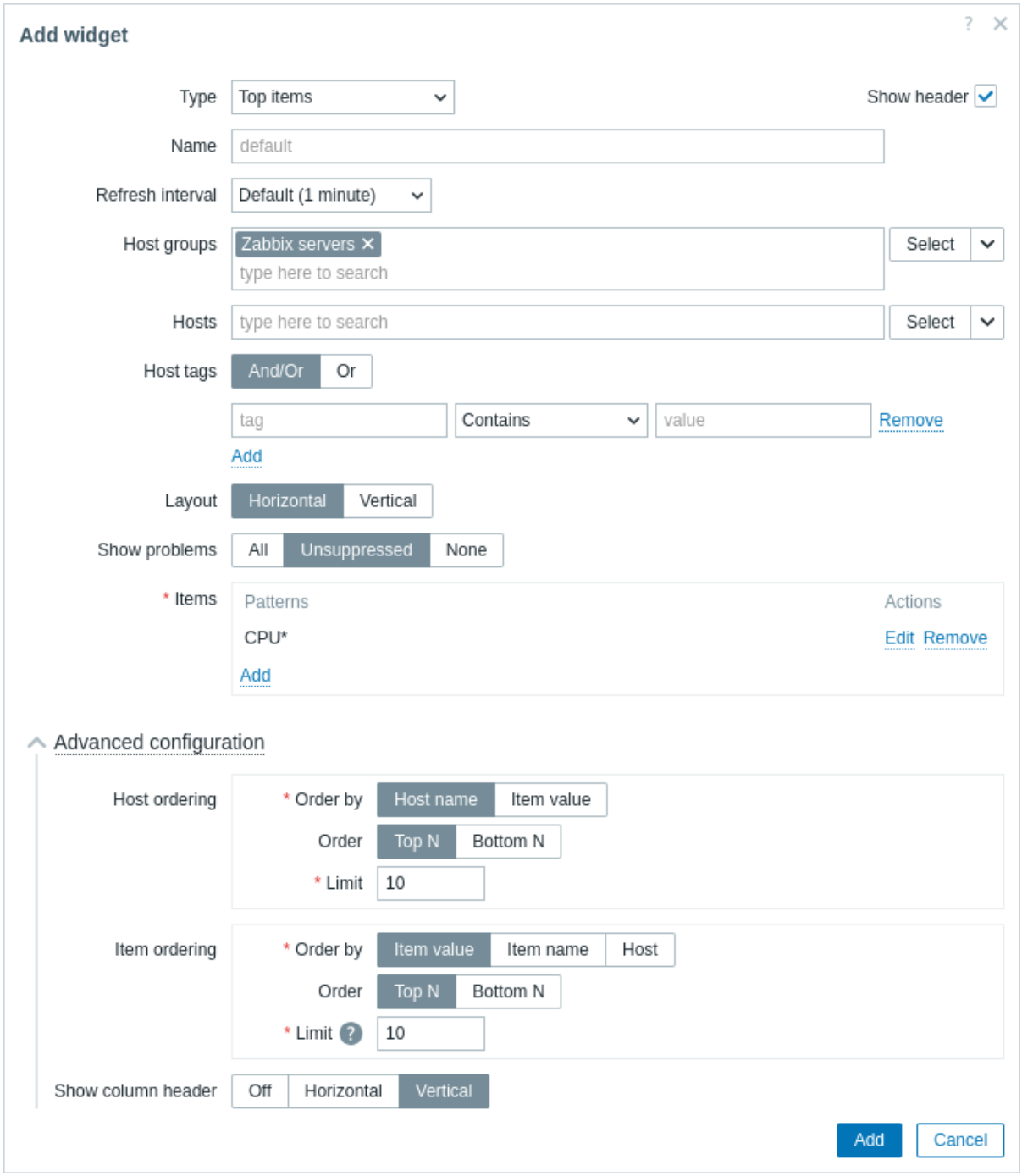


Per impostazione predefinita, il widget visualizza i valori delle ultime 24 ore. Questo periodo può essere modificato in *Administration > General > GUI* (*Max history display period*).

Facendo clic sul valore dell'item si apre il menu dell'item.

Configurazione

Per configurare il widget, selezionare *Top items* come tipo:



Oltre ai parametri comuni a tutti i widget, è possibile impostare le seguenti opzioni specifiche:

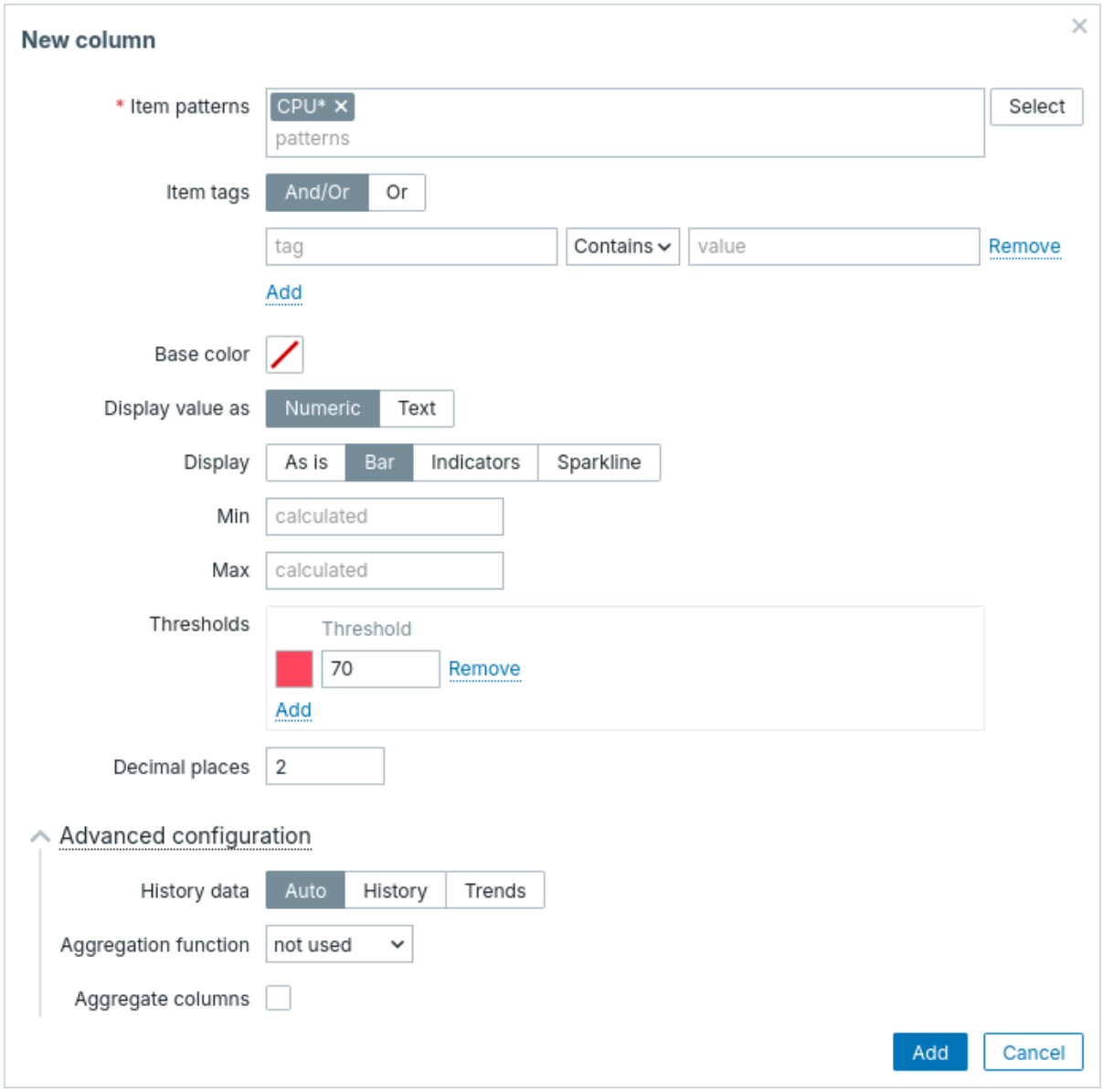
Gruppi di host	Selezionare i gruppi di host. In alternativa, selezionare un widget compatibile come origine dati per i gruppi di host. Questo campo supporta il completamento automatico, quindi iniziando a digitare il nome di un gruppo verrà visualizzato un menu a discesa con i gruppi corrispondenti. Questo parametro non è disponibile durante la configurazione del widget in una dashboard di 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 di template.
Tag host	Specificare i tag per limitare la quantità di dati host 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. Sono disponibili diversi operatori per ciascuna condizione: **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). Sono disponibili due tipi di calcolo per le condizioni: **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 sola condizione.
Layout	Selezionare l'opzione di layout: **Orizzontale** - i nomi degli host verranno visualizzati a sinistra; **Verticale** - i nomi degli host verranno visualizzati in alto.
Mostra problemi	Filtrare quali problemi visualizzare in base al loro stato (tutti, non soppressi, nessuno). I problemi sono indicati con valori degli item colorati, che sovrascrivono le impostazioni di visualizzazione della colonna degli 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.
Item	Aggiungere pattern di item o item specifici da visualizzare (vedere Configurazione delle colonne).
Configurazione avanzata	
Ordinamento host	Selezionare le opzioni di ordinamento per la colonna/riga host.
Ordina per	Ordinare gli host per: **Nome host** - gli host verranno ordinati per nome host; **Valore item** - gli host verranno ordinati in base al valore degli item selezionati.
Ordine	Selezionare 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 *Limite*.
Limite	Immettere il limite per gli host visualizzabili (intervallo 1-1000; predefinito 10). Questo valore diventerà il valore di N nel campo *Ordine*.
Ordinamento item	Selezionare le opzioni di ordinamento per la colonna/riga item.
Ordina per	Ordinare gli item per: **Valore item** - gli item verranno ordinati in base al valore dell'item; **Nome item** - gli item verranno ordinati per nome item; **Host** - gli item verranno ordinati in base al pattern host selezionato.
Ordine	Selezionare 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 *Limite*.

Limite	Immettere il limite per gli item visualizzabili (intervallo 1-1000; predefinito 10). Questo valore diventerà il valore di N nel campo *Ordine*.
Mostra intestazione colonna	Selezionare le opzioni di visualizzazione dell'intestazione della colonna: **Off** - non visualizzare l'intestazione della colonna; **Orizzontale** - visualizzare il testo orizzontalmente nell'intestazione; **Verticale** - 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 di colonna per valori numerici:

Display	Definire come deve essere visualizzato il valore: **As is** - come testo normale; **Bar** - come barra piena colorata; **Indicators** - come barra segmentata colorata; **Sparkline** - mini grafico a linee.
Min	Valore minimo per la visualizzazione barra/indicatori.
Max	Valore massimo per la visualizzazione barra/indicatori.
Width	Impostare lo spessore della linea del grafico utilizzando il cursore o inserendo manualmente un valore nell’intervallo da 0 a 10. Questo parametro è solo per la visualizzazione sparkline.
Fill	Impostare il livello di trasparenza del colore di riempimento utilizzando il cursore o inserendo manualmente un valore nell’intervallo da 0 a 10. Questo parametro è solo per la visualizzazione sparkline.
Color	Selezionare il colore della linea e del riempimento. Questo parametro è solo per la visualizzazione sparkline.
Vedere *Configurazione avanzata* per la descrizione dei campi relativi al periodo di tempo e alla selezione dei dati storici.	
Thresholds	Specificare i valori di soglia ai quali il colore di sfondo/riempimento deve cambiare. L’elenco verrà ordinato in ordine crescente al momento del salvataggio. Per le sparkline, le soglie vengono applicate solo all’ultimo valore dell’item.
Decimal places	Specificare quante cifre decimali verranno visualizzate con il valore.
Configurazione avanzata	

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 di testo, i dati verranno sempre prelevati dalla cronologia.
Aggregation function	Specificare quale funzione di aggregazione utilizzare: **min** - visualizza il valore più piccolo; **max** - visualizza il valore più grande; **avg** - visualizza il valore medio; **count** - visualizza il conteggio dei valori; **sum** - visualizza la somma dei valori; **first** - visualizza il primo valore; **last** - visualizza l'ultimo valore; **not used** - visualizza il valore più recente (nessuna aggregazione). L'aggregazione consente di visualizzare un valore aggregato per l'intervallo scelto (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.
Time period	Specificare il periodo di tempo da utilizzare per l'aggregazione dei valori. Selezionare la 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, verrà visualizzata un'icona dell'orologio nell'angolo superiore destro del widget, che indicherà l'ora impostata al passaggio del mouse. Si noti che, indipendentemente dalla configurazione del *Time period* del widget, i widget compatibili possono comunque utilizzarlo come origine 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 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".
Aggregate columns	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 *Column aggregation function* selezionata e visualizzati sotto *Combined column name*. Predefinito: disattivato (non selezionato). Questo parametro viene mostrato solo quando *Display value as* è **Numeric**.
Column aggregation function	Scegliere come combinare i valori dei pattern degli item raggruppati. Opzioni disponibili: **min**, **max**, **avg**, **count**, **sum** (predefinito). Questo parametro viene mostrato solo quando *Aggregate columns* è selezionato.
Combined column name	Inserire il nome visualizzato per la colonna aggregata. Questo campo è obbligatorio quando *Aggregate columns* è selezionato e diventa l'intestazione della colonna per i valori combinati.

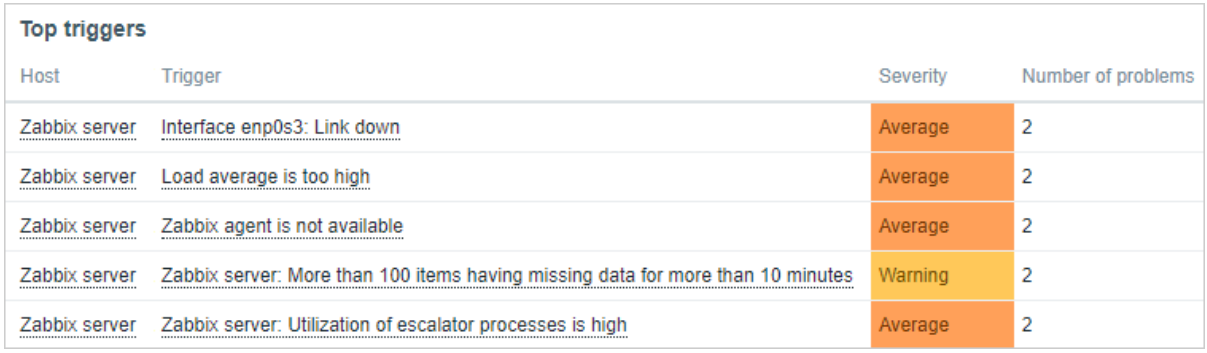
Parametri di colonna per valori di testo:

Highlights	Specificare le espressioni regolari alla cui corrispondenza il colore di sfondo/riempimento deve cambiare.

30 trigger principali

Panoramica

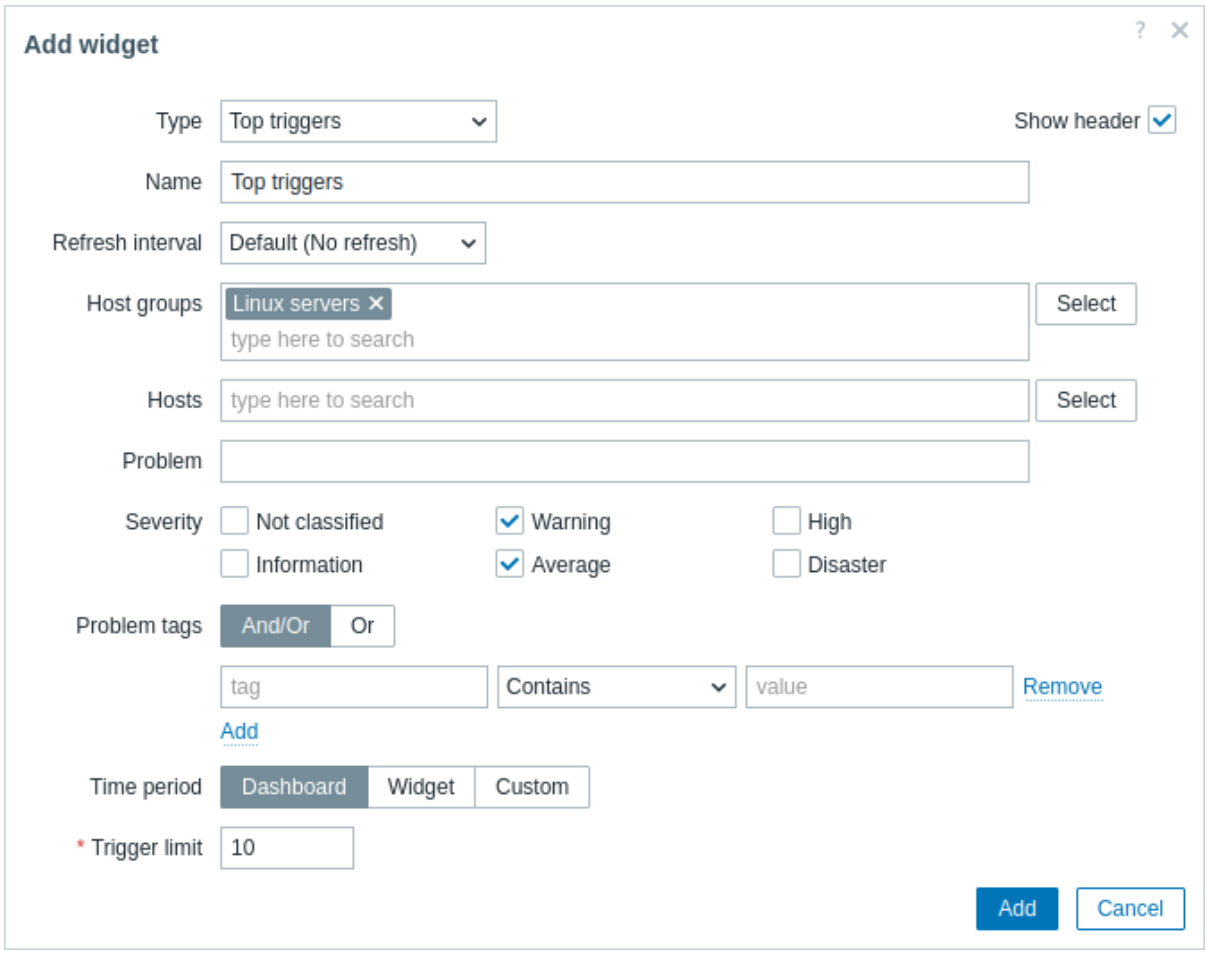
Il widget *Top triggers* visualizza i trigger con il maggior numero di problemi.



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.

31 Panoramica dei trigger

Panoramica

Il widget *Panoramica trigger* visualizza gli stati correnti dei trigger degli host come una tabella di blocchi colorati. Fornisce una rapida panoramica visiva dello stato di salute e dell'attività dei trigger su host diversi.



Il widget utilizza più elementi visivi per trasmettere i dettagli dei trigger:

- I colori dei blocchi per i trigger in stato PROBLEM corrispondono alle severità dei problemi.
- I blocchi lampeggianti indicano modifiche recenti dello stato dei trigger (negli ultimi 2 minuti).
- Le frecce grigie indicano trigger con dipendenze (passaci sopra con il mouse per vedere i dettagli).
- Le icone a forma di casella di controllo indicano problemi confermati; perché questa icona venga visualizzata, tutti i problemi attivi o risolti del trigger devono essere confermati.

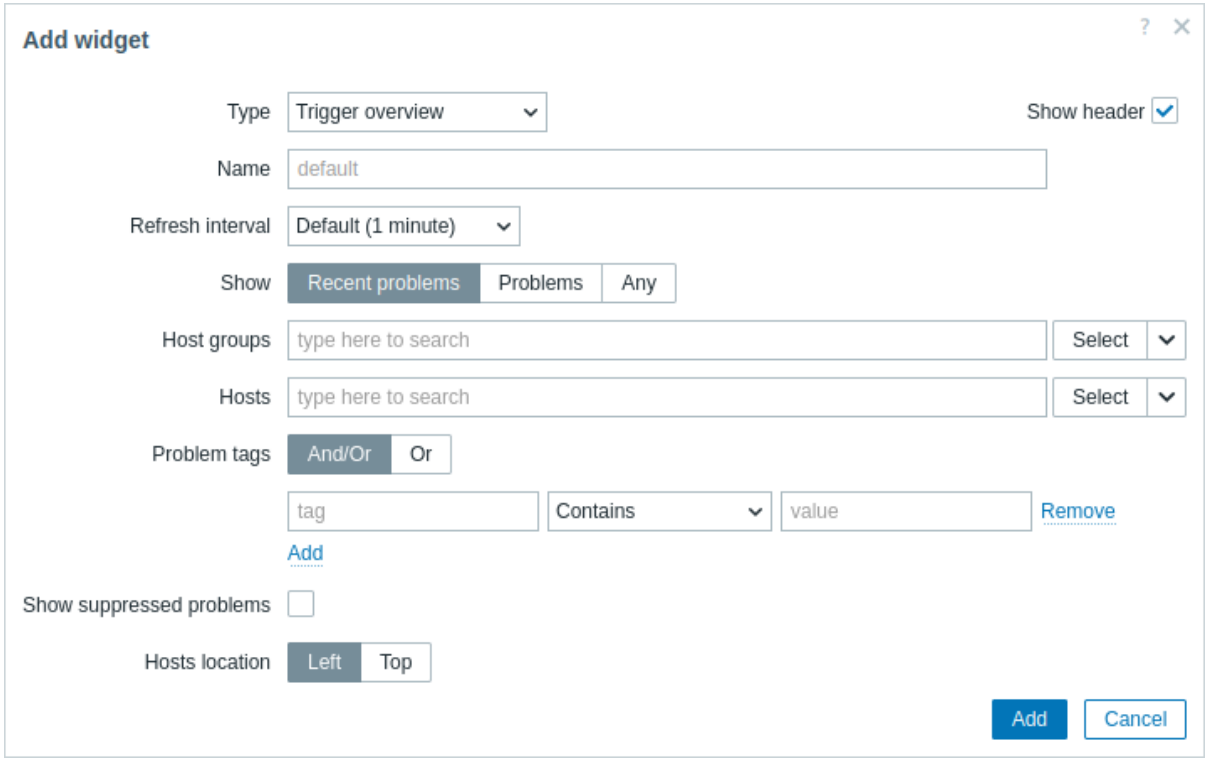
Note:
I colori delle severità dei trigger possono essere regolati in *Administration* > *General*. Le severità dei trigger possono essere modificate durante l'aggiornamento dei problemi.

Facendo clic su un blocco vengono forniti 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* in *Max number of columns and rows in overview tables*). Se esistono più trigger, un messaggio 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:

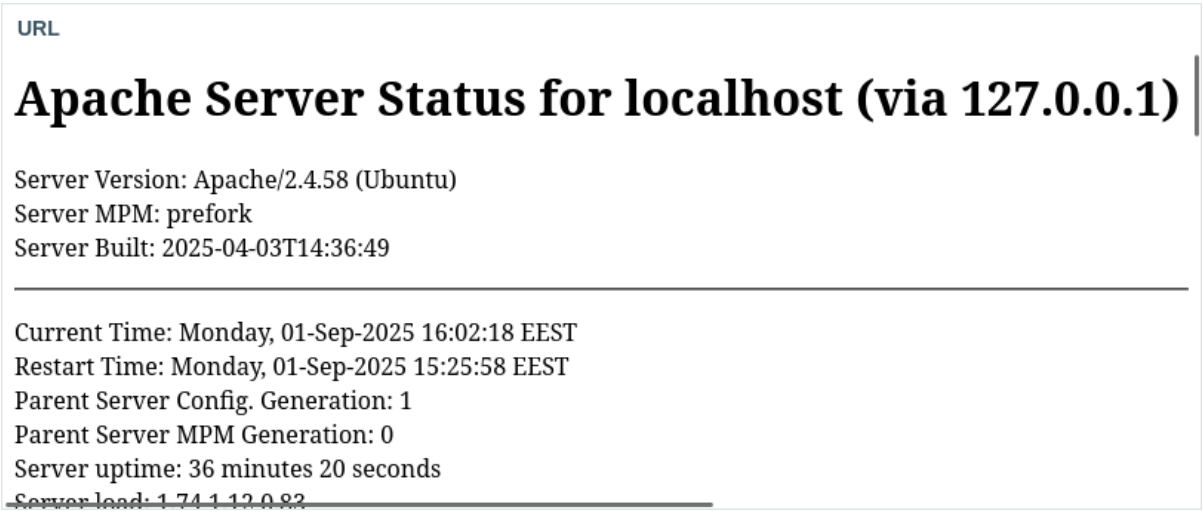
Mostra	Filtra i trigger in base allo stato del trigger: **Problemi recenti** - *(predefinito)* mostra i trigger che recentemente sono stati o sono ancora nello stato PROBLEM (risolti e non risolti); **Problemi** - mostra i trigger che sono nello stato PROBLEM (non risolti); **Qualsiasi** - mostra tutti 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.

32 URL

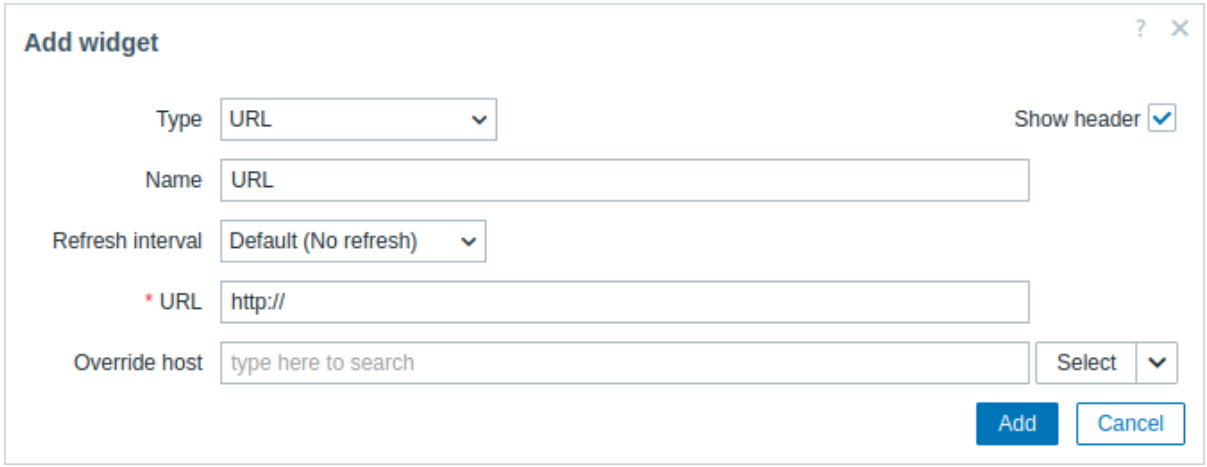
Panoramica

Il widget *URL* visualizza il contenuto recuperato da un URL.



Configurazione

Per configurare, selezionare *URL* come tipo:



Oltre ai parametri comuni per tutti i widget, è possibile impostare le seguenti opzioni specifiche:

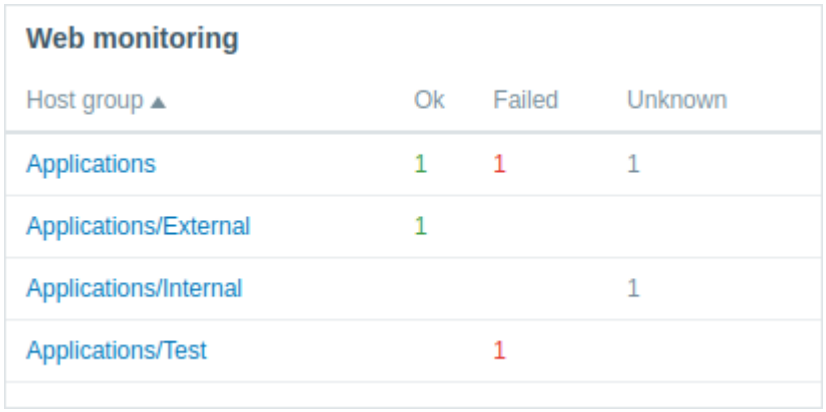
URL	Inserire l'URL da visualizzare (fino a 2048 caratteri). Gli URL esterni devono iniziare con `http://` o `https://`. Gli URL interni supportano percorsi relativi (ad esempio, `zabbix.php?action=report.status`). Le macro {HOST.*} sono supportate.
Override host	Selezionare un widget compatibile oppure 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.

Attention:
I browser potrebbero non caricare una pagina HTTP configurata nel widget se il frontend di Zabbix viene aperto tramite HTTPS. Inoltre, potrebbe essere necessario modificare l'intestazione di risposta HTTP X-Frame-Options per visualizzare contenuti esterni al frontend di Zabbix.

33 Monitoraggio web

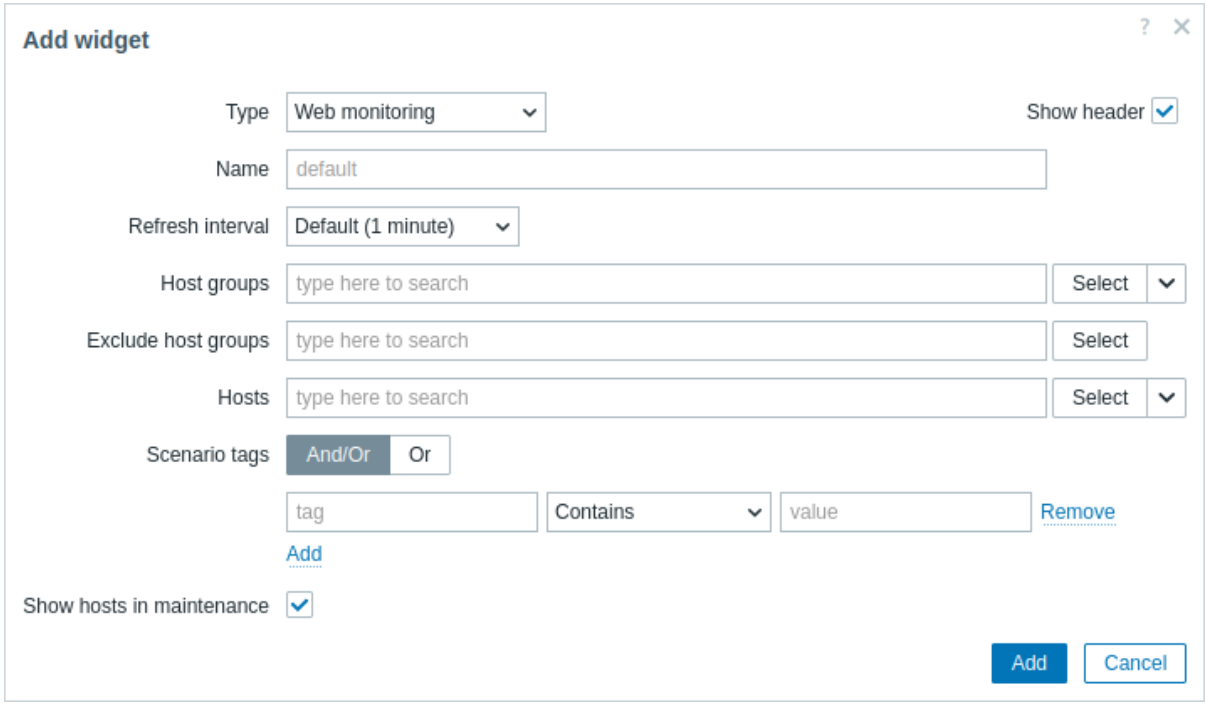
Panoramica

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.

2 Monitoraggio

Panoramica

Il menu Monitoraggio riguarda la visualizzazione dei dati. Qualunque informazione Zabbix è configurato per raccogliere, visualizzare e agire, lo sarà visualizzata nelle varie sezioni del menu Monitoraggio.

Pulsanti della modalità di visualizzazione

I seguenti pulsanti, situati nell'angolo superiore destro, sono comuni a ogni sezione:

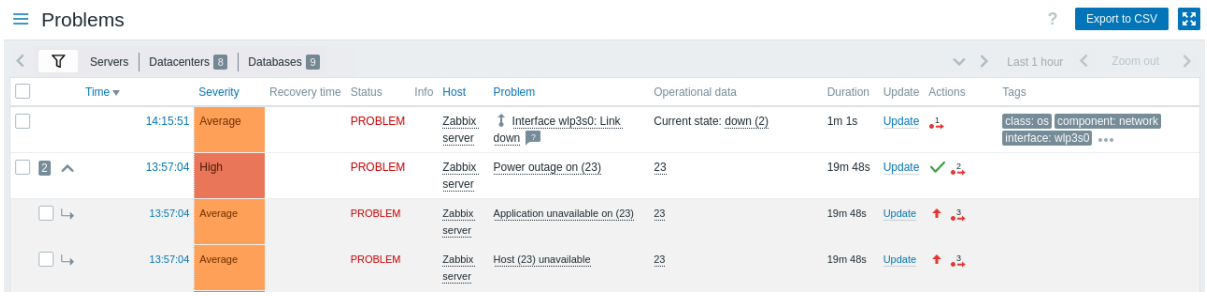
	Visualizza la pagina in modalità kiosk. In questa modalità viene visualizzato solo il contenuto della pagina. Per uscire dalla modalità kiosk, spostare il cursore del mouse finché non appare il pulsante di uscita  , quindi fare clic su di esso. Si tornerà alla modalità normale.

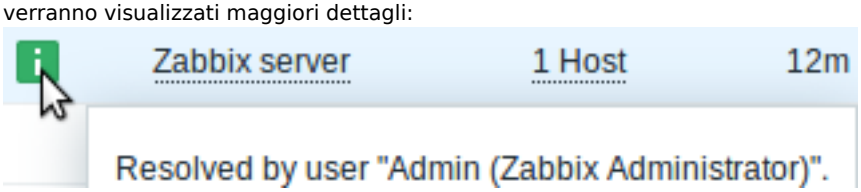
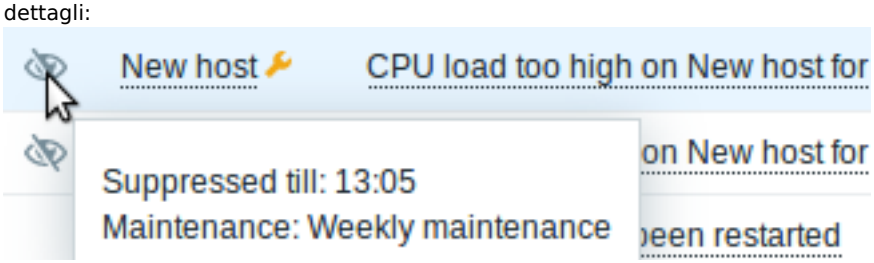
1 Problemi

Panoramica

In *Monitoring* > *Problems* puoi vedere quali problemi sono attualmente presenti. 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 determinati problemi come problemi sintomo del problema causa. Per maggiori dettagli, vedi eventi causa e sintomo.



Colonna	Descrizione
Checkbox	Vengono visualizzate le caselle di controllo per la selezione dei problemi. Le icone accanto alle caselle di controllo hanno il seguente significato:  - il numero di eventi sintomo per il problema causa;  - espande per mostrare gli eventi sintomo;  - comprime per nascondere gli eventi sintomo;  - questo è un evento sintomo.
Time	Viene visualizzata l'ora di inizio del problema.
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 utilizzando la schermata *Update problem*. Il colore della gravità del problema viene utilizzato come sfondo della cella per la durata del problema.
Recovery time	Viene visualizzata l'ora di risoluzione del problema.
Status	Viene visualizzato lo stato del problema: **Problem** - problema non risolto **Resolved** - problema risolto di recente. Puoi nascondere i problemi risolti di recente utilizzando il filtro. I problemi nuovi e quelli risolti di recente lampeggiano per 2 minuti. I problemi risolti vengono visualizzati per un totale di 5 minuti. Entrambi questi valori sono configurabili in *Administration > General > Trigger displaying options*.
Info	Viene visualizzata un'icona informativa verde se un problema viene chiuso dalla correlazione globale o manualmente durante l'aggiornamento del problema. Passando il mouse sopra l'icona verranno visualizzati maggiori dettagli:  La seguente icona viene visualizzata se è mostrato un problema soppresso (vedi l'opzione *Show suppressed problems* nel filtro). Passando il mouse sopra l'icona verranno visualizzati maggiori dettagli: 
Host	Viene visualizzato il host del problema. Facendo clic sul nome del host si apre il menu host.
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. *Nota* che è possibile aggiungere al nome del problema i dati operativi, che mostrano alcuni valori recenti degli item. Facendo clic sul nome del problema si apre il menu evento. Passando il mouse sull'icona  dopo il nome del problema, verrà visualizzata la descrizione del trigger (per quei problemi che la possiedono).
Operational data	Vengono visualizzati i dati operativi, contenenti i valori più recenti degli item. I dati operativi possono essere una combinazione di testo e macro di valori degli item, se configurati a livello di trigger. Se non sono configurati dati operativi a livello di trigger, vengono visualizzati i valori più recenti di tutti gli item dell'espressione. Questa colonna viene visualizzata solo se nel filtro è selezionato *Separately* per *Show operational data*.
Duration	Viene visualizzata la durata del problema. Vedi anche: Durata negativa del problema

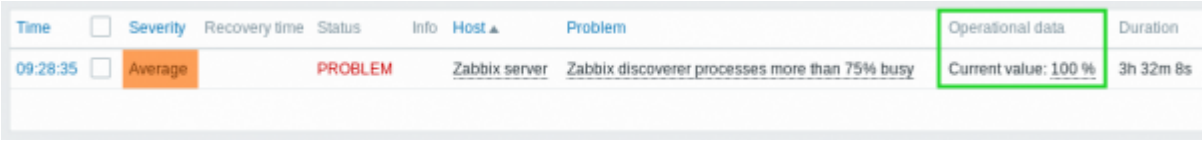
Colonna	Descrizione
Update	Fai clic sul collegamento *Update* per andare alla schermata di aggiornamento del problema, dove è possibile eseguire varie azioni sul problema, inclusi commenti e conferma del problema.
Actions	Viene visualizzata la cronologia delle attività relative al problema tramite icone simboliche:  - il problema è stato confermato. Questa icona viene sempre visualizzata per prima.  - sono stati aggiunti commenti. Viene visualizzato anche il numero di commenti.  - la gravità del problema è stata aumentata (ad es. Information > Warning).  - la gravità del problema è stata diminuita (ad es. Warning > Information).  - la gravità del problema è stata modificata, ma è tornata al livello originale (ad es. Warning > Information > Warning).  - il problema è stato soppresso.  - la soppressione del problema è stata rimossa.  - 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 non è riuscita. Viene visualizzato anche il numero di azioni. Passando il mouse sopra le icone, vengono visualizzati popup con i dettagli dell'attività. Vedi visualizzazione dei dettagli per saperne di più sulle icone utilizzate nel popup per le azioni eseguite.
Tags	Vengono visualizzati i tag (se presenti). Inoltre, possono essere visualizzati anche tag provenienti da un sistema di ticketing esterno (vedi l'opzione *Process tags* durante la configurazione dei webhook).

Dati operativi dei problemi

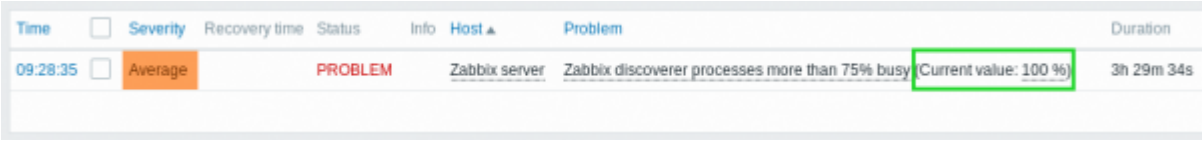
È possibile visualizzare i dati operativi per i problemi correnti, cioè gli ultimi valori degli item invece dei valori degli item al momento del problema.

La visualizzazione dei dati operativi può essere configurata nel filtro di *Monitoring* > *Problems* oppure nella configurazione del rispettivo widget della dashboard, selezionando una delle tre opzioni:

- *None* - non viene visualizzato alcun dato operativo
- *Separately* - i dati operativi vengono visualizzati in una colonna separata



- *With problem name* - i dati operativi vengono aggiunti al nome del problema, tra parentesi. I dati operativi vengono aggiunti al nome del problema solo se il campo *Operational data* non è vuoto nella configurazione del trigger.



Il contenuto dei dati operativi può essere configurato per ciascun trigger, nel campo *Operational data*. Questo campo accetta una stringa arbitraria con macro, in particolare la macro `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` in questo campo verrà sempre risolta negli ultimi valori degli item nell'espressione del trigger. `{ITEM.VALUE<1-9>}` in questo campo verrà risolta nei valori degli item al momento del cambiamento di stato del trigger (cioè il passaggio a `**Problem**`, il passaggio a `**OK**`, la chiusura manuale da parte dell'utente oppure la chiusura tramite correlazione).

Si noti che la chiusura manuale del problema non produce un nuovo valore, quindi il valore risolto di {ITEM.LASTVALUE<1-9>} o {ITEM.VALUE<1-9>} continuerà a mostrare il valore del momento del problema.

{ITEM.LASTVALUE<1-9>} o {ITEM.VALUE<1-9>} verranno risolte come *UNKNOWN* se l'ultimo valore della cronologia è stato raccolto più tempo fa rispetto al periodo *Max history display period* (vedere *Administration > General*).

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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Aggiornamento di massa* - aggiorna i problemi selezionati accedendo alla schermata di aggiornamento dei problemi

Per utilizzare questa opzione, seleziona le caselle di controllo davanti ai rispettivi problemi, quindi fai clic sul pulsante *Aggiornamento di massa*.

Pulsanti

Il pulsante a destra offre la seguente opzione:

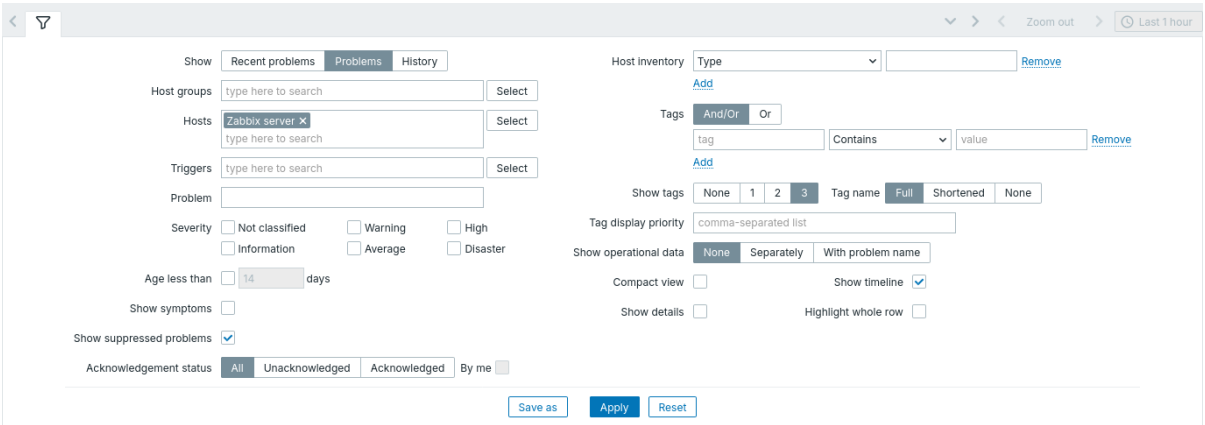
Export to CSV	Esporta il contenuto di tutte le pagine in un file CSV.

I pulsanti della modalità di visualizzazione, essendo comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Uso del filtro

È possibile usare il filtro per visualizzare solo i problemi di interesse. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il filtro si trova sopra la tabella. Le impostazioni di filtro preferite possono essere salvate come schede e poi richiamate rapidamente facendo clic sulle schede sopra il filtro.



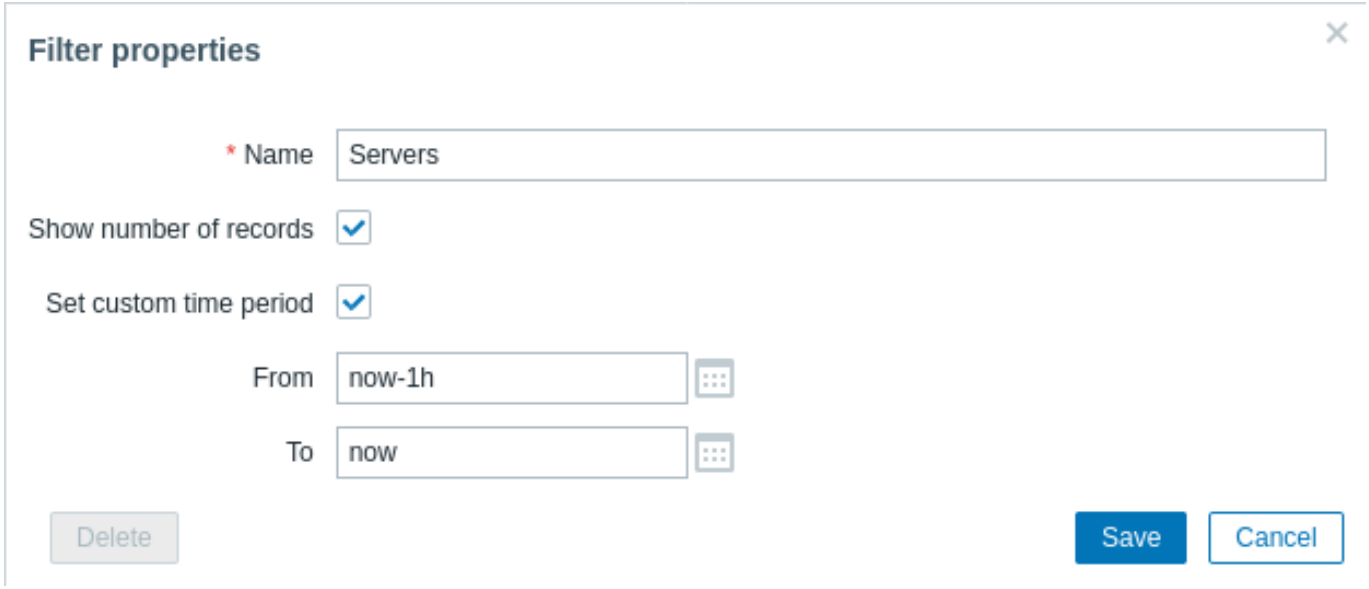
Parametro	Descrizione
Show	Filtra per stato del problema: **Recent problems** - vengono visualizzati i problemi non risolti e quelli risolti di recente (predefinito) **Problems** - vengono visualizzati i problemi non risolti **History** - viene visualizzata la cronologia di tutti gli eventi
Host groups	Filtra per uno o più gruppi di host. Specificando un gruppo di host padre vengono implicitamente selezionati anche tutti i gruppi di host annidati.
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.
Show symptoms	Selezionare la casella per visualizzare su una riga separata i problemi classificati come sintomi.
Show suppressed problems	Selezionare la casella per visualizzare i problemi che altrimenti verrebbero 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 riconosciuti oppure solo i problemi riconosciuti. Selezionare la casella aggiuntiva per escludere i problemi che sono stati riconosciuti almeno una volta dall'utente corrente.
Host inventory	Filtra per tipo e valore dell'inventario.
Tags	Filtra per nome e valore del tag evento. È possibile includere oppure 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 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) 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 sola condizione Quando viene applicato il filtro, i tag specificati qui verranno visualizzati per primi insieme al problema, a meno che non siano sovrascritti dall'elenco *Tag display priority* (vedere sotto).
Show tags	Selezionare il numero di tag visualizzati: **None** - nessuna colonna *Tags* in *Monitoring* > *Problems* **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, passare il mouse sull'icona con i tre puntini.
Tag name	Selezionare 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 vengono abbreviati a 3 simboli; i valori dei tag vengono visualizzati per intero **None** - vengono visualizzati solo i valori dei tag; nessun nome
Tag display priority	Inserire la priorità di visualizzazione dei tag per un problema come elenco di tag separati da virgole (ad esempio: `customer,scope,component`). Devono essere usati solo i nomi dei tag, non i valori. I tag di questo elenco verranno sempre visualizzati per primi, sovrascrivendo l'ordinamento naturale alfabetico.
Show operational data	Selezionare la modalità di visualizzazione dei dati operativi: **None** - non vengono visualizzati dati operativi **Separately** - i dati operativi vengono visualizzati in una colonna separata **With problem name** - aggiunge i dati operativi al nome del problema, usando parentesi per i dati operativi
Compact view	Selezionare la casella per abilitare la vista compatta.
Show details	Selezionare la casella per visualizzare le espressioni trigger sottostanti dei problemi. Disabilitato se la casella *Compact view* è selezionata.

Parametro	Descrizione
Show timeline	Selezionare la casella per visualizzare la timeline visiva e il raggruppamento. Disabilitato se la casella *Compact view* è selezionata.
Highlight whole row	Selezionare la casella 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.

Schede per i filtri preferiti

Gli insiemi di parametri di filtro usati più frequentemente possono essere salvati in schede.

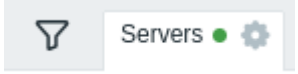
Per salvare un nuovo insieme di parametri di filtro, aprire la scheda principale e configurare le impostazioni del filtro, quindi premere il pulsante *Salva come*. In una nuova finestra popup, definire le *Proprietà del filtro*.



Parametro	Descrizione
Nome	Il nome del filtro da visualizzare nell'elenco delle schede.
Mostra numero di record	Selezionare questa opzione se si desidera che il numero di problemi venga visualizzato accanto al nome della scheda.
Sovrascrivi selettore del periodo di tempo	Selezionare questa opzione per impostare un periodo di tempo predefinito specifico per questo insieme di filtri. Se impostato, sarà possibile 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 di tempo può essere modificato premendo il pulsante del selettore del tempo nell'angolo superiore destro (il nome del pulsante dipende dall'intervallo di tempo selezionato: Questa settimana, Ultimi 30 minuti, Ieri, ecc.). Questa opzione è disponibile solo per i filtri in *Monitoring* > *Problems*.
Da/A	Inizio e fine del periodo di tempo in sintassi temporale assoluta (`Y-m-d H:i:s`) o relativa (`now-1d`). Disponibile se *Imposta periodo di tempo personalizzato* è selezionato.

Una volta salvato, il filtro viene creato come scheda di filtro con nome e attivato immediatamente.

Per modificare le proprietà del filtro di un filtro esistente, premere il simbolo dell'ingranaggio accanto al nome della scheda attiva.



Note:

- Per nascondere l'area del filtro, fare clic sul nome della scheda corrente. Fare nuovamente clic sul nome della scheda attiva per riaprire l'area del filtro.
- La navigazione tramite tastiera è supportata: usare le frecce per passare da una scheda all'altra, premere *Invio* per aprire.
- I pulsanti sinistra/destra sopra il filtro possono essere usati per passare tra i filtri salvati. In alternativa, il pulsante con la freccia verso il basso apre un menu a discesa con tutti i filtri salvati e si può fare clic su quello necessario.
- Le schede dei filtri possono essere riordinate tramite trascinamento.

- 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, fare clic sul pulsante *Aggiorna*, visualizzato al posto del pulsante *Salva come*.
- Le impostazioni correnti del filtro vengono ricordate nel profilo utente. Quando l'utente apre nuovamente la pagina, le impostazioni del filtro rimangono invariate.

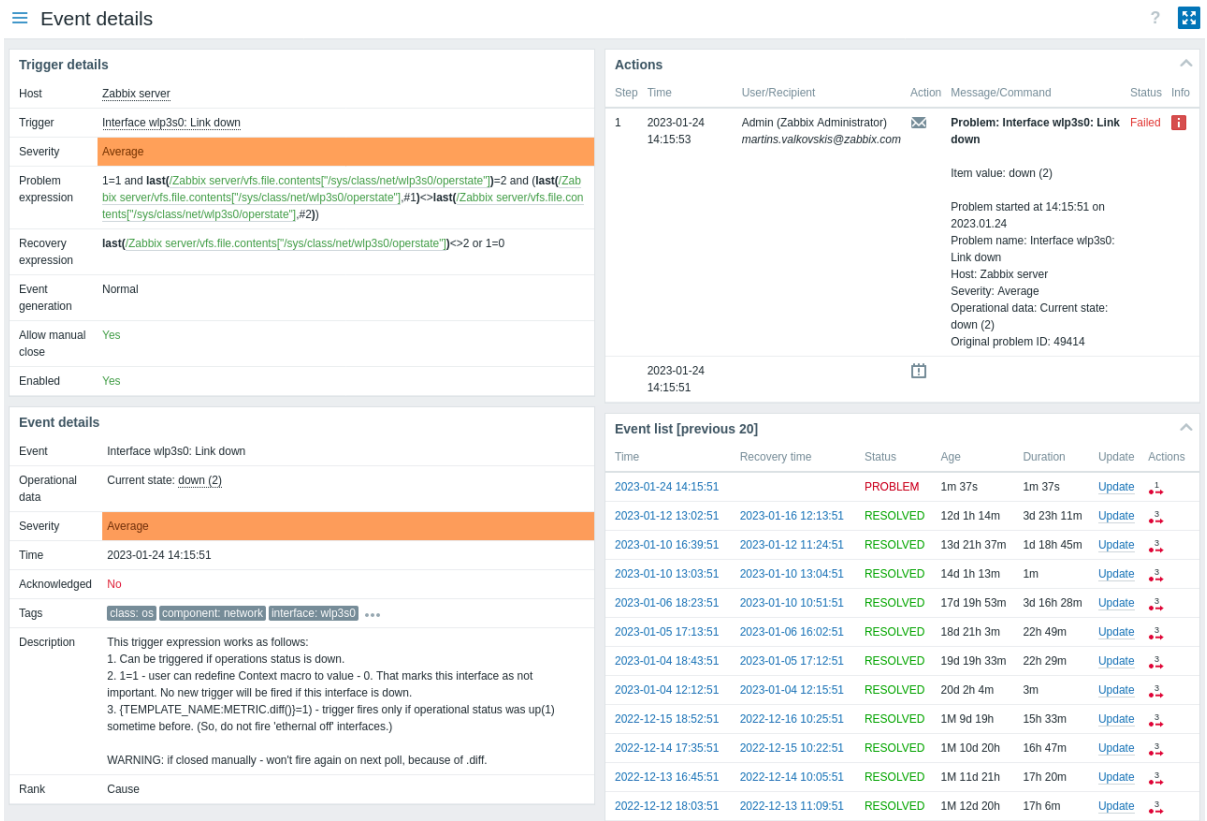
Note:
Per condividere i filtri, copiare e inviare ad altri l'URL di un filtro attivo. Dopo aver aperto questo URL, gli altri utenti potranno salvare questo insieme di parametri come filtro permanente nel proprio account Zabbix.
Vedere anche: Parametri della pagina.

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 e di ripristino del problema in *Monitoring* > *Problems* sono link. Facendo clic su di essi si aprono ulteriori dettagli dell'evento.



Si noti che la gravità del problema può differire tra il trigger e l'evento del problema, se è stata aggiornata per l'evento del problema utilizzando la schermata *Update problem*.

Nell'elenco delle azioni, vengono utilizzate le seguenti icone per indicare il tipo di attività:

- - evento del problema generato.

-  - il messaggio è stato inviato.
-  - evento del problema riconosciuto.
-  - evento del problema non riconosciuto.
-  - è stato aggiunto un commento.
-  - la gravità del problema è stata aumentata (ad es. Information > Warning).
-  - la gravità del problema è stata diminuita (ad es. Warning > Information).
-  - la gravità del problema è stata modificata, ma è tornata al livello originale (ad es. Warning > Information > Warning).
-  - è stato eseguito un comando remoto.
-  - l'evento del 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 convertito in un problema sintomo.
-  - il problema è stato convertito in un problema causa.

1 Problemi di causa e 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.

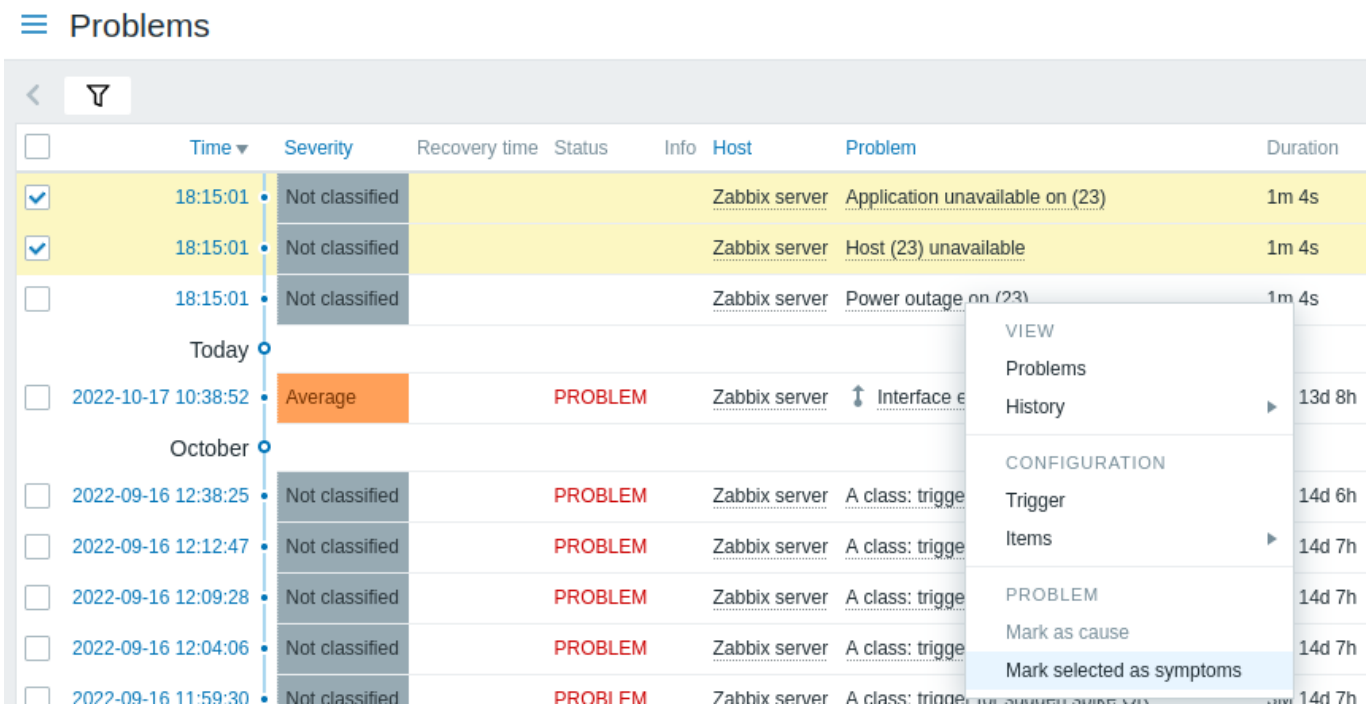
Solo i problemi causa vengono conteggiati nei totali dei problemi nelle mappe, nei widget della dashboard come *Problemi per gravità* o *Host con problemi*, ecc. Tuttavia, la classificazione dei problemi non influisce sui servizi.

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, selezionarlo innanzitutto nell'elenco dei problemi. È possibile selezionare uno o più problemi.

Quindi andare al problema causa e, nel relativo menu contestuale, fare clic sull'opzione *Contrassegna i selezionati come sintomi*.



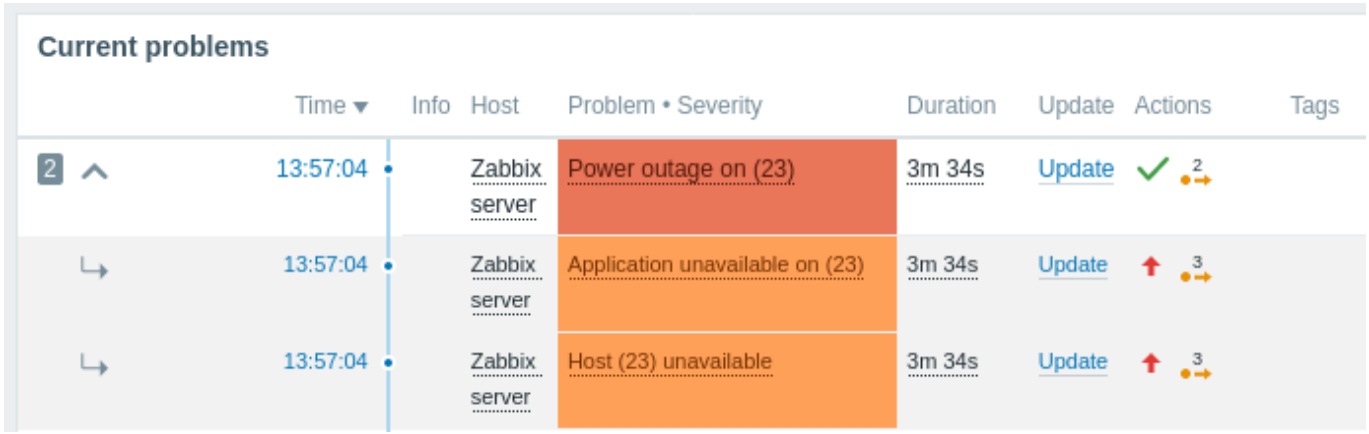
Successivamente, i problemi selezionati verranno aggiornati dal server come problemi sintomo del problema causa.

Mentre lo stato del problema viene aggiornato, viene visualizzato in uno dei due modi seguenti:

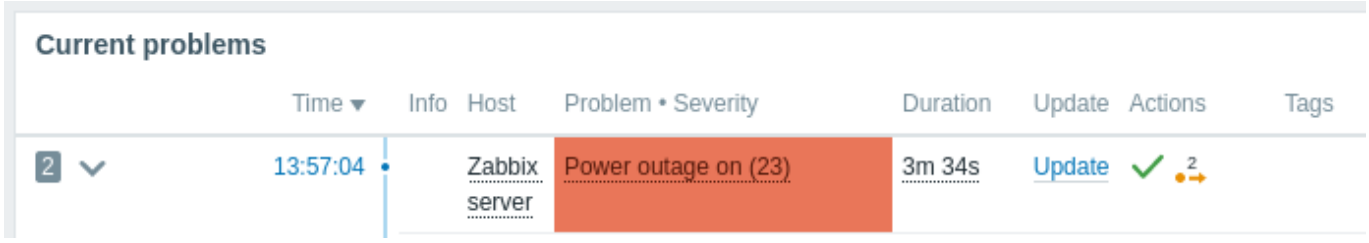
- Nella colonna Stato viene visualizzato uno stato ”UPDATING” lampeggiante;
- Nella colonna Info viene visualizzata un’icona lampeggiante oppure (questo avviene se nel filtro è selezionata solo l’opzione *Problemi* e quindi la colonna Stato non viene mostrata).

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, è visibile solo il problema causa; la presenza di 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 propria riga. Per farlo, selezionare *Show symptoms* nelle impostazioni del filtro o nella configurazione del widget.

Ripristino a problema causa

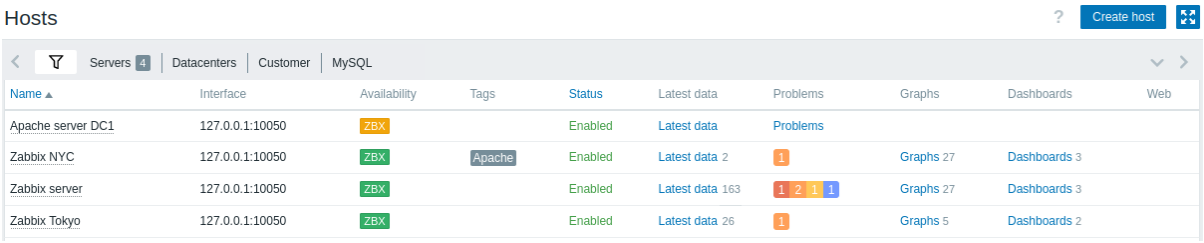
Un problema sintomo può essere ripristinato come problema causa. Per farlo, è possibile:

- fare clic sull'opzione *Mark as cause* nel menu contestuale del problema sintomo;
- selezionare l'opzione *Convert to cause* nella schermata di aggiornamento del problema e fare clic su *Update* (questa opzione funziona anche se sono selezionati più problemi).

2 Host

Panoramica

La sezione *Monitoraggio* → *Host* visualizza un elenco completo degli host monitorati con informazioni dettagliate sull'interfaccia dell'host, disponibilità, tag, problemi correnti, stato (abilitato/disabilitato) e collegamenti per passare facilmente agli ultimi dati dell'host, alla cronologia dei problemi, ai grafici, alle dashboard e agli scenari web.



Colonna	Descrizione
Name	Il nome visibile dell'host. Facendo clic sul nome si apre il menu host. Un'icona a forma di chiave inglese arancione 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.

Colonna	Descrizione
Availability	Viene visualizzata la disponibilità dell'host per ogni interfaccia configurata. Le icone di disponibilità rappresentano lo stato corrente dell'interfaccia dell'host sullo Zabbix server. Pertanto, se si disabilita un host nel frontend, la sua disponibilità verrà aggiornata dopo che lo Zabbix server avrà sincronizzato le modifiche di configurazione. Analogamente, se si abilita un host, la sua disponibilità verrà aggiornata dopo che lo Zabbix server avrà sincronizzato le modifiche di configurazione ed eseguito il polling dell'host. Le icone di disponibilità rappresentano solo i tipi di interfaccia configurati (Agent, SNMP, IPMI, JMX). Passando il puntatore sull'icona viene visualizzato un popup con un elenco di tutte le interfacce dello stesso tipo con dettagli, stato ed errori. Per l'interfaccia Agent, il popup visualizza le interfacce (passive) e i controlli attivi. Se un host ha solo controlli attivi, l'icona dell'interfaccia Agent viene visualizzata anche se l'host non ha un'interfaccia Agent configurata. La colonna è vuota per gli host senza interfacce. Lo stato di una singola interfaccia host è determinato dalla connessione tra un item abilitato che utilizza l'interfaccia e l'host. Lo stato può essere: **Available** - la connessione all'host è riuscita; **Not available** - la connessione all'host non è riuscita (timeout, problemi di firewall, ecc.); **Unknown** - la connessione all'host non è stata tentata oppure il risultato è sconosciuto. Per ulteriori dettagli su come lo Zabbix server determina lo stato dell'interfaccia, vedere Unknown interface status e Unreachable/unavailable host interface settings. Lo stato di tutte le interfacce 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** - nessuna interfaccia è disponibile; **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 totale 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 loro host rimane sconosciuta.
Tags	I tag dell'host e di tutti i template collegati, con le macro non risolte.
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 aprirà la pagina *Monitoraggio* **-** *Ultimi dati* con tutti gli ultimi dati raccolti dall'host. Il numero di item con ultimi dati viene visualizzato in grigio.
Problems	Il numero di problemi aperti dell'host ordinati per gravità. Il colore del quadrato indica la gravità del problema. Il numero nel quadrato indica il numero di problemi per la gravità specificata. Facendo clic sull'icona si aprirà la pagina *Monitoraggio* **-** *Problemi* per l'host corrente. Se un host non ha problemi, un collegamento alla sezione *Problemi* per questo host viene visualizzato come testo. Utilizzare il filtro per selezionare se i problemi soppressi devono essere inclusi (per impostazione predefinita non sono inclusi).
Graphs	Facendo clic sul collegamento verranno visualizzati i grafici configurati per l'host. Il numero di grafici viene visualizzato in grigio. Se un host non ha grafici, non viene visualizzato alcun collegamento.
Dashboards	Facendo clic sul collegamento verranno visualizzate le dashboard configurate per l'host. Il numero di dashboard viene visualizzato in grigio. Se un host non ha dashboard, non viene visualizzato alcun collegamento.

Colonna	Descrizione
Web	Facendo clic sul collegamento verranno visualizzati gli scenari web configurati per l'host. Il numero di scenari web viene visualizzato in grigio. Se un host non ha scenari web, non viene visualizzato alcun collegamento.

Pulsanti

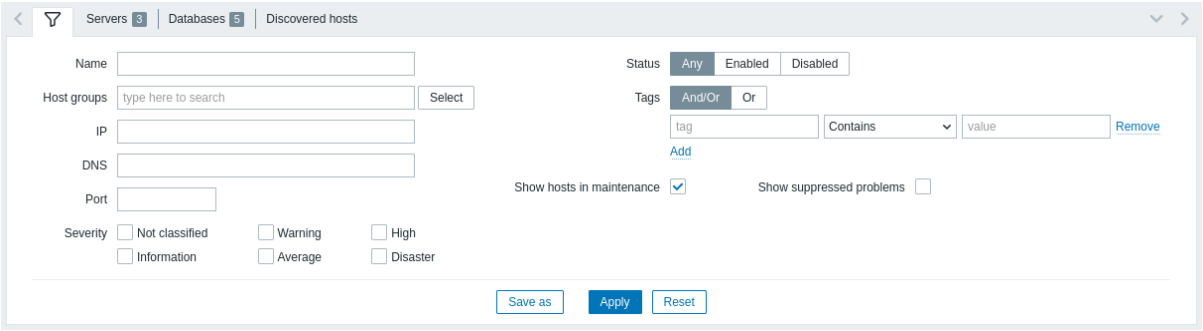
Crea host consente di creare un nuovo host. Questo pulsante è disponibile solo per gli utenti Admin e Super Admin.

I pulsanti della modalità di visualizzazione comuni a tutte le sezioni sono descritti nel Monitoraggio pagina.

Uso del filtro

È possibile usare il filtro per visualizzare solo gli host di interesse. Per migliori 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 selezionare se visualizzare i problemi soppressi e gli host attualmente in manutenzione.



Parametro	Descrizione
Name	Filtra per nome visibile dell'host.
Host groups	Filtra per uno o più gruppi host. Specificando un gruppo host padre vengono implicitamente selezionati anche 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 in base ai tag a livello di host così come ai tag di tutti i template collegati, inclusi i template nidificati. 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
Show hosts in maintenance	Selezionare la casella di controllo per visualizzare gli host in manutenzione (visualizzati per impostazione predefinita).

Parametro	Descrizione
Show suppressed problems	Selezionare la casella di controllo per visualizzare i problemi che altrimenti verrebbero soppressi (non mostrati) a causa della manutenzione dell'host o della singola problem suppression.

Salvataggio del filtro

Le impostazioni di filtro preferite possono essere salvate come schede e quindi richiamate rapidamente facendo clic sulla rispettiva scheda sopra il filtro.

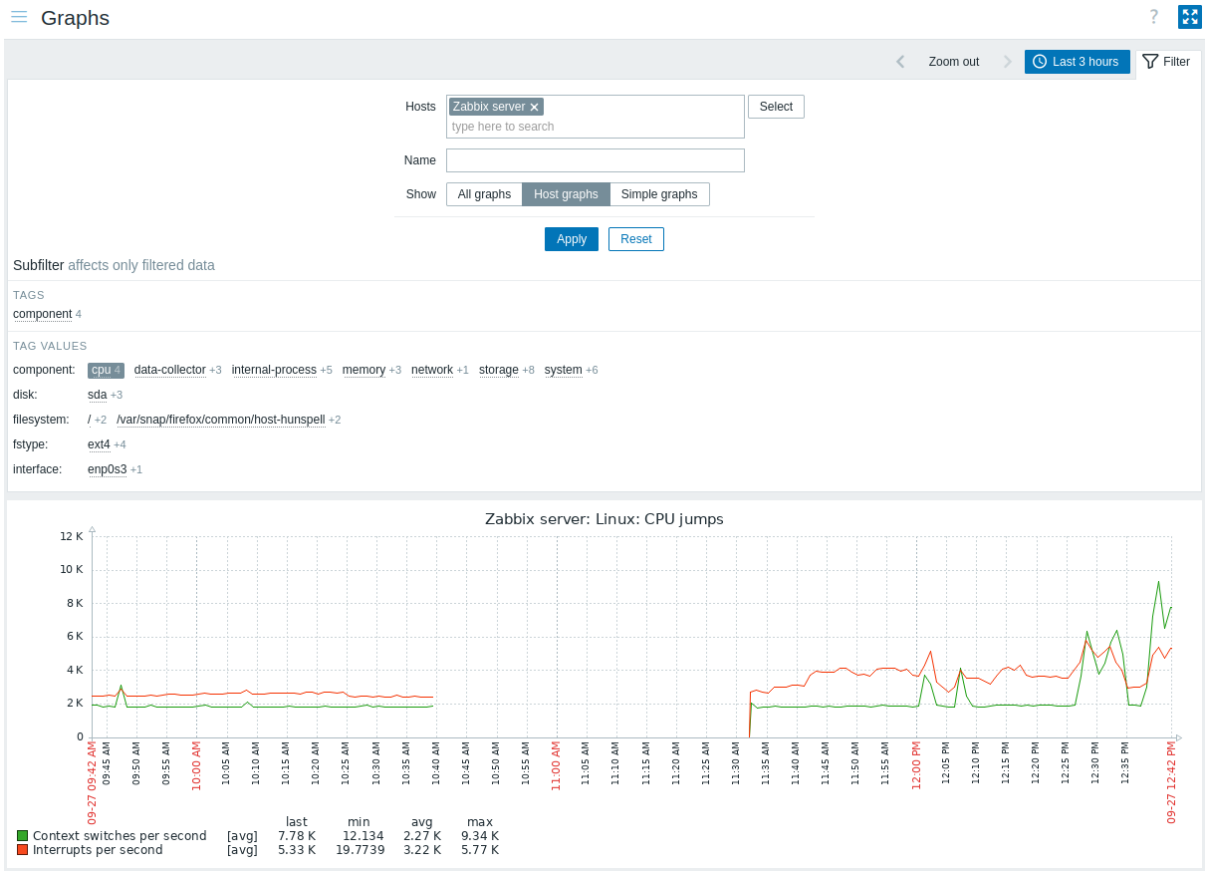
Per maggiori dettagli, vedere saving filters.

1 Grafici

Panoramica

È possibile accedere ai grafici del host da *Monitoring* → *Hosts* facendo clic su Graphs per il rispettivo host.

È possibile visualizzare qualsiasi grafico personalizzato configurato per il host, così come qualsiasi grafico semplice.



I grafici sono ordinati per:

- nome del grafico (grafici personalizzati)
- nome dell'item (grafici semplici)

Anche i grafici dei host disabilitati sono accessibili.

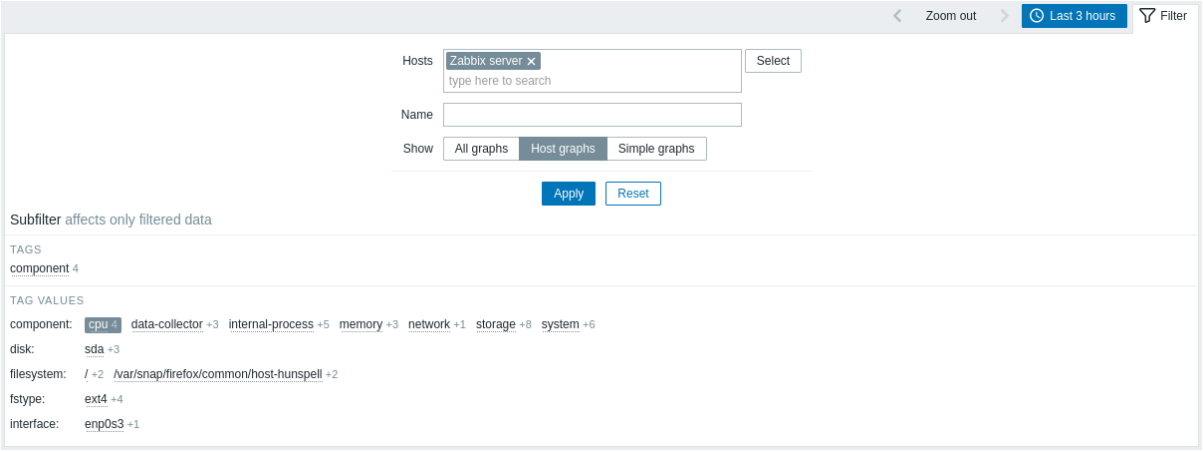
Selettore del periodo di tempo

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, selezionalo 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 sottofiltro

Il sottofiltro consente di modificare ulteriormente il filtraggio del filtro principale.

Contiene link cliccabili per un accesso rapido ai grafici correlati. I grafici sono correlati da un'entità comune: nome o valore del tag. Quando si fa clic su un nome/valore di tag, questo viene evidenziato con uno sfondo grigio e i grafici vengono filtrati immediatamente (non è necessario fare clic su *Apply* nel filtro principale). Facendo clic su un altro nome/valore di tag, questo viene aggiunto ai risultati filtrati. Facendo nuovamente clic sul nome/valore di tag, il filtro viene rimosso.

I sottofiltri vengono generati in base ai dati filtrati, che sono limitati a 1000 record. Se si desidera visualizzare più record nel sottofiltro, è necessario aumentare il valore del parametro *Limit for search and filter results* (in *Administration -> General -> GUI*).

A differenza del filtro principale, il sottofiltro 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 è limitato orizzontalmente a 100. Se ce ne sono di più, alla fine viene visualizzata un'icona con tre puntini; non è cliccabile. Gli elenchi verticali (come i tag con i relativi valori) sono limitati a 20 voci. Se ce ne sono di più, viene visualizzata un'icona con tre puntini; non è cliccabile.

Un numero accanto a ciascuna entità cliccabile indica il numero di grafici presenti nei risultati del filtro principale.

Una volta selezionata 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.

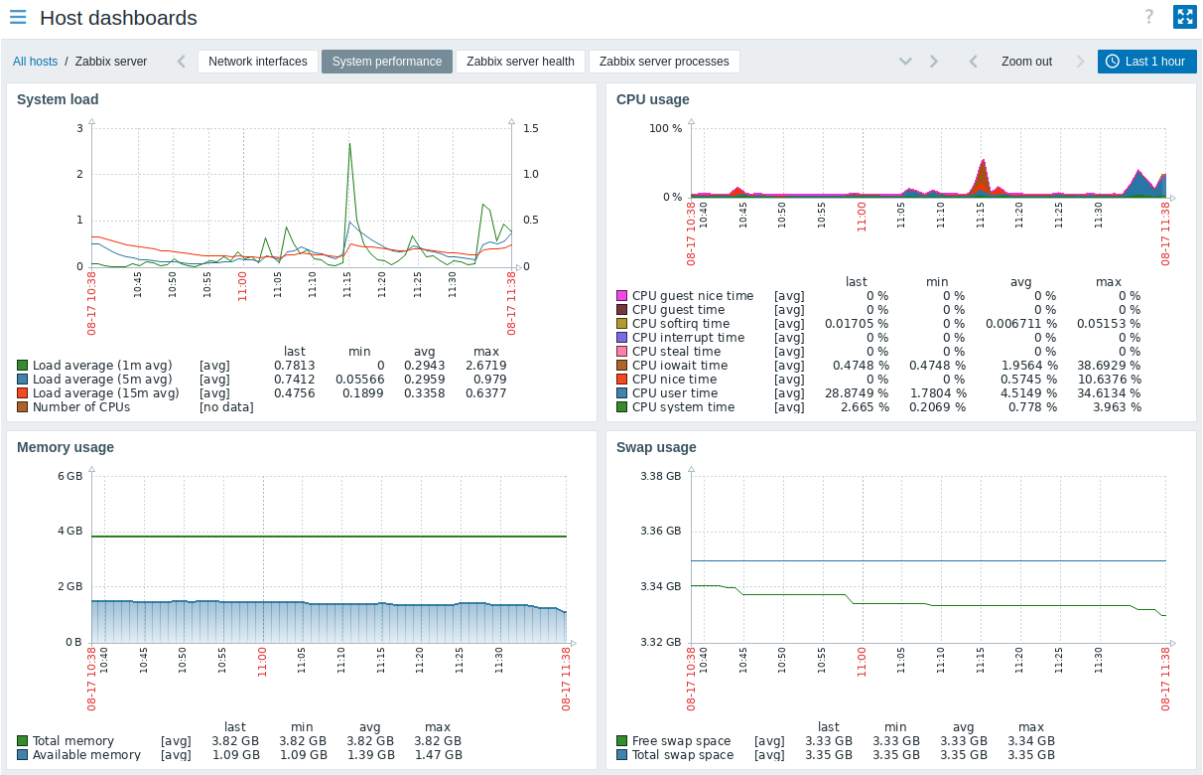
Pulsanti

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

2 Dashboard host

Panoramica

Le dashboard host sono simili alle dashboard globali; tuttavia, le dashboard host non hanno un proprietario e visualizzano i dati solo per l'host selezionato.



Durante la visualizzazione delle dashboard host, è possibile passare da una dashboard configurata all'altra facendo clic su:

- le schede della dashboard;
- i pulsanti freccia sotto l'intestazione;
- il pulsante freccia sotto l'intestazione, che mostrerà l'elenco completo delle dashboard 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.

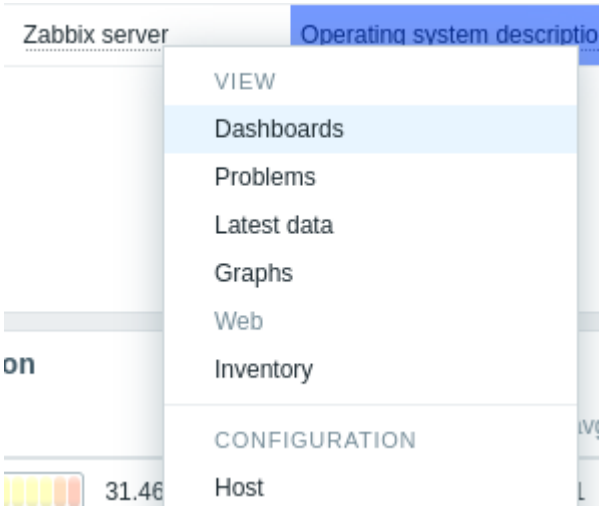
Anche i widget delle dashboard host possono essere configurati solo a livello di template, ad eccezione della modifica dell'intervallo di aggiornamento. Inoltre, i widget delle dashboard host possono essere copiati solo in altre dashboard host all'interno dello stesso template. Si noti che i widget delle 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 del host possono essere aperte:

- dopo aver cercato un nome host nella ricerca globale (fare clic sul collegamento *Dashboard* fornito nei risultati della ricerca);
- dopo aver fatto clic su un nome host in *Inventario* → *Host* (fare clic sul collegamento *Dashboard* fornito nella panoramica del host);
- dal menu host facendo clic su *Dashboard*.

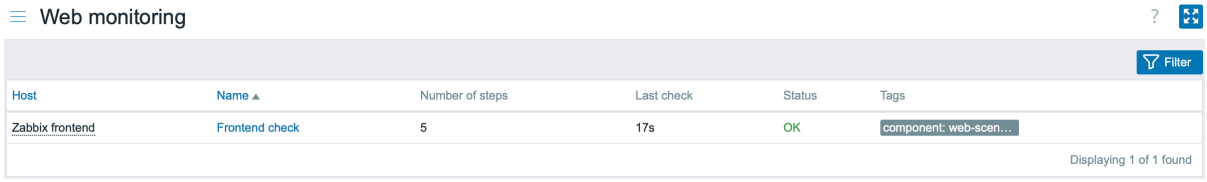


Si noti che le dashboard del host *non possono* essere aperte direttamente nella sezione *Dashboard*, che è riservata alle dashboard globali.

3 Scenari web web

Panoramica

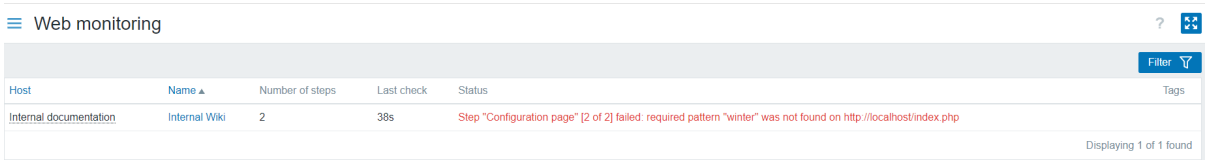
Le informazioni sugli scenari web del host sono accessibili da *Monitoraggio* → *Host* facendo clic su Web per il rispettivo host.



Facendo clic sul nome del host si apre il menu host. I dati dei host disabilitati sono anch'essi accessibili. Il nome di un host disabilitato è visualizzato in rosso.

Lo stato di uno scenario web può essere:

- OK (verde) - tutti i passaggi sono stati completati con successo e Zabbix ha raccolto dati per lo scenario.
- Errore (rosso) - uno o più passaggi non sono riusciti (vedere il messaggio di errore per indicazioni su quali parametri potrebbero richiedere correzione).
- Sconosciuto (grigio) - lo stato dei passaggi non è ancora disponibile oppure non sono stati raccolti dati.



Il numero massimo di scenari visualizzati per pagina dipende dall'impostazione del profilo utente *Righe per pagina* 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 pagine di dati recenti di grandi dimensioni. È possibile estendere questo periodo di tempo modificando il valore del parametro *Periodo massimo di visualizzazione dello storico* nella sezione di menu *Amministrazione* → *Generale* → *GUI*.

Il nome dello scenario è un collegamento a statistiche più dettagliate:



Utilizzo del filtro

La pagina mostra un elenco di tutti gli scenari web dell'host selezionati. Per vedere scenari web per un altro host o gruppo host senza tornare al file Nella pagina *Monitoraggio* → *Host*, seleziona l'host o il gruppo nel filtro. Voi può anche filtrare gli scenari in base ai tag.

Pulsanti

I pulsanti della modalità di visualizzazione comuni a tutte le sezioni sono descritti in Monitoraggio pagina.

3 Dati più recenti

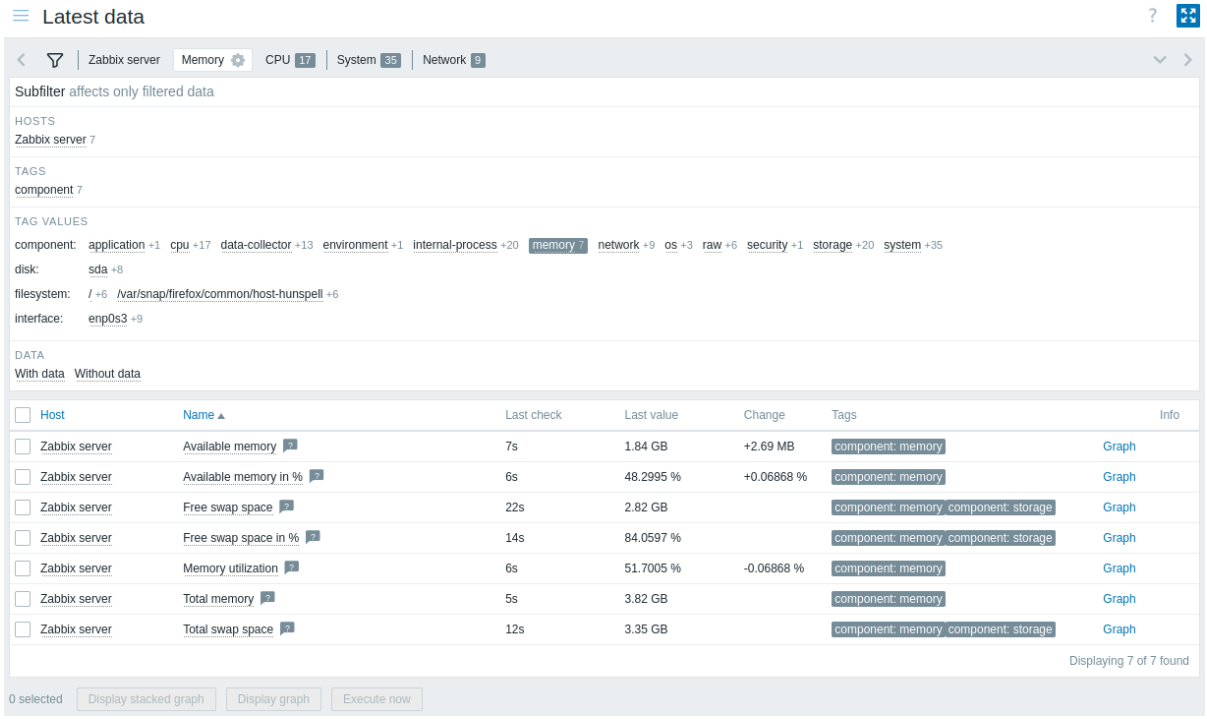
Panoramica

La sezione *Monitoraggio* → *Ultimi dati* visualizza i valori più recenti 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 ci sono risultati da mostrare.



Colonna	Descrizione
Host	Nome dell'host a cui appartiene l'item. Facendo clic sul nome si apre il menu contestuale dell'host. Se un host è in manutenzione, dopo il nome dell'host viene visualizzata un'icona a forma di chiave inglese arancione . Se un host è disabilitato, il nome dell'host viene visualizzato in rosso. Si noti che i dati degli host disabilitati (inclusi i grafici e gli elenchi dei valori degli item) sono accessibili nella sezione *Ultimi dati*.
Nome	Nome dell'item. Facendo clic sul nome si apre 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 cursore sull'icona per visualizzare un suggerimento con la descrizione dell'item.
Ultimo controllo	Tempo trascorso dall'ultimo controllo dell'item.
Ultimo valore	Valore più recente dell'item. I valori vengono visualizzati con la conversione delle unità e la mappatura dei valori applicate. Passare il cursore sul valore per visualizzare i dati grezzi (troncati a 8192 caratteri). I valori binari vengono visualizzati come segnaposto (*valore binario*) invece del valore effettivo. Per impostazione predefinita, vengono visualizzati solo i valori ricevuti nelle ultime 24 ore. Questo limite migliora i tempi di caricamento iniziali per pagine di ultimi dati di grandi dimensioni; per estenderlo, aggiornare il valore del parametro *Max history display period* in *Amministrazione* → *Generale* → *GUI*.
Variazione	Differenza tra il valore precedente e il valore più recente. Per gli item con una frequenza di aggiornamento pari o superiore a 1 giorno, l'entità della variazione non verrà mai visualizzata (con l'impostazione predefinita). In questo caso, l'ultimo valore non verrà visualizzato affatto se è stato ricevuto più di 24 ore fa.
Tag	Tag associati all'item. I tag nell'elenco degli item sono cliccabili. Facendo clic su un tag, questo viene abilitato nel sottofiltro, facendo sì che l'elenco degli item mostri solo gli item che contengono questo tag (e qualsiasi altro tag precedentemente selezionato nel sottofiltro). Si noti che, una volta che gli item sono stati filtrati in questo modo, i tag nell'elenco degli item non sono più cliccabili. Ulteriori modifiche basate sui tag (ad esempio, per rimuovere tag o specificare altri filtri) devono essere effettuate nel sottofiltro.
Grafico/Cronologia	Collegamento al grafico semplice/cronologia dei valori dell'item.

Colonna	Descrizione
Info	Informazioni aggiuntive sull'item. Se un item presenta errori (ad esempio, è diventato non supportato), viene visualizzata un'icona informativa. Passare il cursore sull'icona per i dettagli.

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 per uno o più item selezionati:

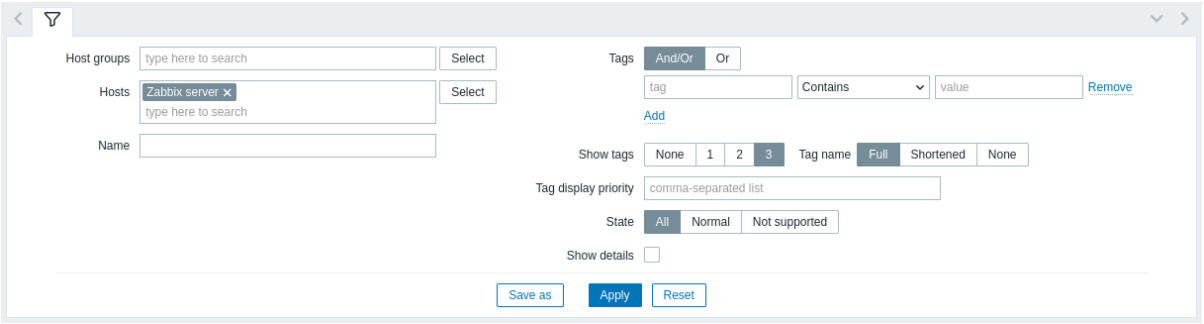
- *Visualizza grafico in pila* - visualizza un grafico ad hoc in pila.
- *Visualizza grafico* - visualizza un semplice grafico ad hoc.
- *Esegui ora* - esegue immediatamente un controllo per ottenere nuovi valori dell'item. Supportato solo per i controlli **passivi** (vedere maggiori dettagli). Questa opzione è disponibile solo per host con accesso in lettura-scrittura. L'accesso a questa opzione per host con permessi di sola lettura dipende dall'opzione del ruolo utente denominata *Richiama "Esegui ora" su host di sola lettura*.

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi item, quindi fare clic sul pulsante richiesto.

Uso del filtro

Puoi usare il filtro per visualizzare solo gli item di tuo interesse. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

L'icona del filtro si trova sopra l'elenco degli item e il sottofiltro. Fai clic su di essa per espandere il filtro.



Il filtro consente di restringere l'elenco degli item in base a gruppo host, host, nome dell'item, tag, stato e altre impostazioni. Specificando un gruppo host padre nel filtro, vengono selezionati implicitamente tutti i gruppi host nidificati. Per i dettagli sul filtraggio per tag, vedi *Monitoring* → *Problems*.

L'opzione di filtro *Show details* consente di ampliare le informazioni visualizzate per gli item. Selezionala per visualizzare dettagli quali l'intervallo di aggiornamento dell'item, le impostazioni di cronologia e trend, il tipo di item e gli errori dell'item (fine/non supportato).

Salvataggio del filtro

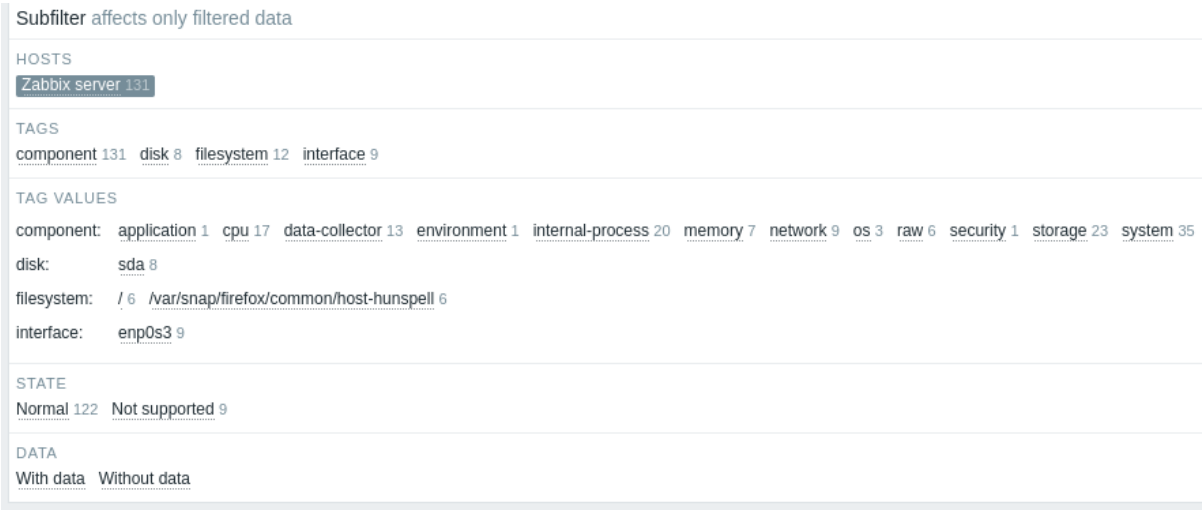
Le impostazioni dei filtri preferiti possono essere salvate come schede e quindi richiamate rapidamente facendo clic sulla rispettiva scheda sopra il filtro.

Per maggiori dettagli sul salvataggio dei filtri.

Utilizzo del sottofiltro

Il sottofiltro consente di modificare ulteriormente il filtraggio del filtro principale.

Contiene link cliccabili per un accesso rapido agli item correlati. Gli item sono correlati da un'entità comune: host, nome o valore del tag, stato dell'item oppure stato dei dati. 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 *Apply* 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 sottofiltri vengono generati in base ai dati filtrati, che sono limitati a 1000 record. Se ci sono 20 host con 100 item ciascuno (quindi 2000 record in totale), nel sottofiltro sarà visibile solo la metà degli host. Se si desidera visualizzare più record nel sottofiltro, è necessario aumentare il valore del parametro *Limit for search and filter results* (in *Administration -> General -> GUI*).

A differenza del filtro principale, il sottofiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da avere sempre informazioni aggiornate sulle opzioni di filtro disponibili e sui relativi contatori.

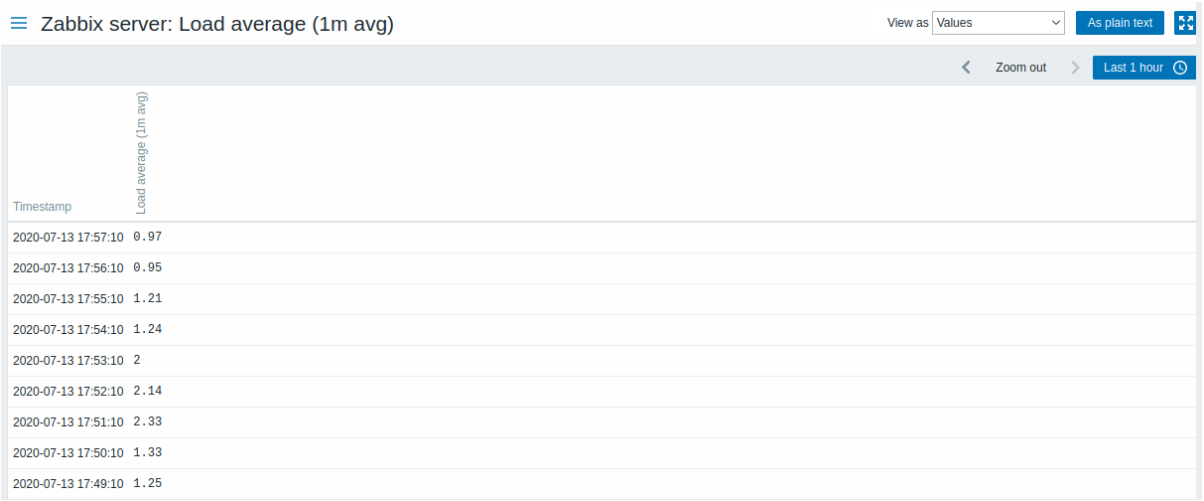
Per ciascun gruppo di entità (host, tag, valori dei tag, ecc.) vengono visualizzate fino a 10 righe di entità. Se sono presenti più entità, questo elenco può essere espanso per visualizzare un massimo di 1000 voci (il valore di SUBFILTER_VALUES_PER_GROUP nelle frontend definitions) facendo clic sull'icona con i tre puntini ••• alla fine dell'elenco. Per i *Tag values*, l'elenco può essere espanso per visualizzare un massimo di 200 nomi di tag con i rispettivi valori. Si noti che, una volta completamente espanso, l'elenco non può essere compresso.

Un numero accanto a ciascuna entità cliccabile indica il numero di item raggruppati in essa (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 state selezionate in precedenza nel sottofiltro.

Grafici e cronologia

La colonna *Grafico/Cronologia* nell'elenco degli item offre i seguenti link:

- **Cronologia** - per tutti gli item testuali, porta agli elenchi (*Valori/ultimi 500 valori*) che mostrano la cronologia dei valori precedenti dell'item.
- **Grafico** - per tutti gli item numerici, porta a un grafico semplice. Si noti che, quando il grafico è visualizzato, un menu a discesa in alto a destra offre anche la possibilità di passare a *Valori/ultimi 500 valori*.



I valori visualizzati in questo elenco sono grezzi, cioè non viene applicata alcuna post-elaborazione.

Note:
La quantità totale di valori visualizzati è definita dal valore del parametro *Limit for search and filter results*, impostato in *Administration → General → GUI*.

4 Mappe

Panoramica

Nella sezione *Monitoraggio* → *Mappe* puoi configurare, gestire e visualizzare mappe di rete.

Quando apri questa sezione, vedrai l'ultima mappa a cui hai accesso o un elenco di tutte le mappe a cui hai accesso.

Tutte le mappe possono essere pubbliche o private. Le mappe pubbliche sono disponibili per tutti gli utenti, mentre le mappe private sono accessibili solo al loro proprietario e al utenti con cui è condivisa la mappa.

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** - impostare le proprietà generali della mappa **Edit** - accedere alla griglia per aggiungere elementi della mappa

Per configurare una nuova mappa, fare clic sul pulsante *Create map* nell'angolo superiore destro. Per importare una mappa da un file YAML, XML o JSON, fare clic sul pulsante *Import* nell'angolo superiore destro. L'utente che importa la mappa verrà impostato come proprietario.

Due pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Export* - esportare le mappe in un file YAML, XML o JSON
- *Delete* - eliminare le mappe

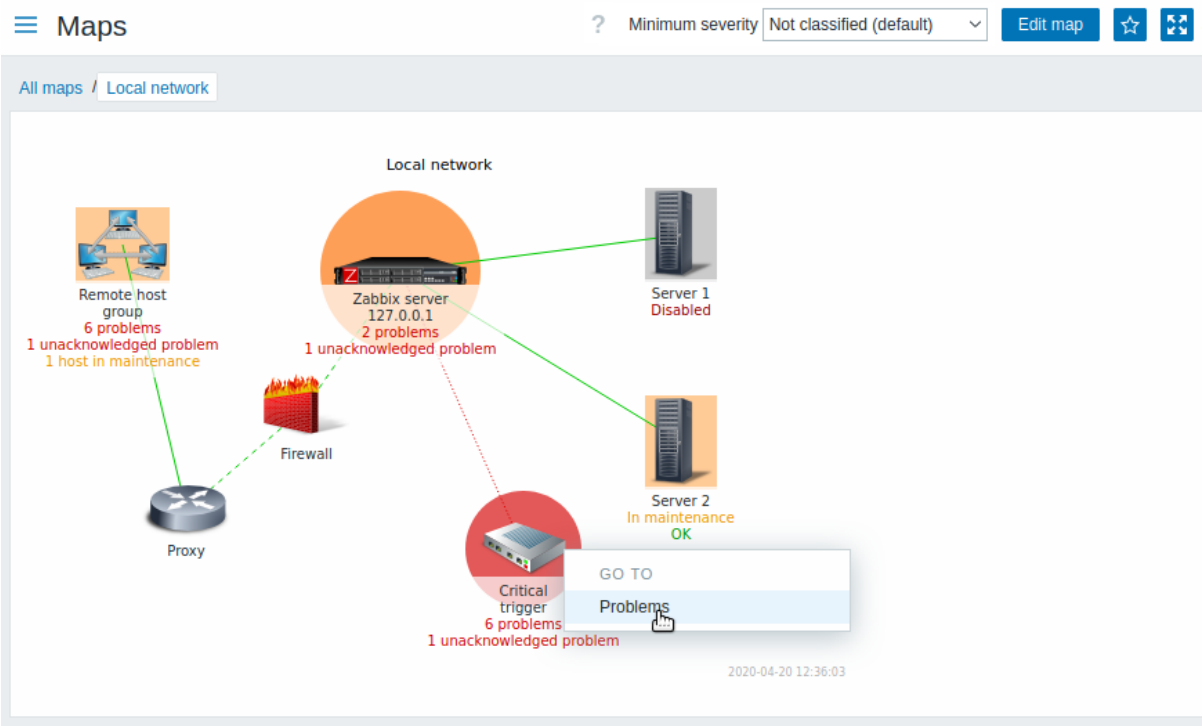
Per utilizzare queste opzioni, selezionare le caselle di controllo davanti alle rispettive mappe, quindi fare clic sul pulsante richiesto.

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, fare clic sul suo nome nell'elenco di tutte le mappe.



È possibile utilizzare l'elenco a comparsa nella barra del titolo della mappa per selezionare il più basso livello di gravità dei trigger del problema da visualizzare. La gravità segnata come *predefinito* è il livello impostato nella configurazione della mappa. Se la mappa contiene una mappa secondaria, la navigazione verso la mappa secondaria manterrà il file gravità della mappa di livello superiore (tranne se è *Non classificato*, in questo caso, non verrà passato alla sottomappa).

Evidenziazione delle icone

Se un elemento della mappa si trova in stato di problema, viene evidenziato con un cerchio rotondo. Il colore di riempimento del cerchio corrisponde al colore di gravità del problema. Con l'elemento verranno visualizzati solo i problemi pari o superiori al livello di gravità selezionato. Se tutti i problemi sono confermati, viene visualizzato un bordo verde spesso attorno al cerchio.

Inoltre:

- un host in manutenzione viene evidenziato con un quadrato arancione pieno. Si noti che l'evidenziazione 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 *Icon highlighting* è selezionata nella configurazione della mappa.

Indicatori di modifiche recenti

I triangoli rossi rivolti verso l'interno attorno a un elemento indicano una recente modifica dello stato del trigger, avvenuta negli ultimi 30 minuti. Questi triangoli vengono mostrati se la casella di controllo *Mark elements on trigger status change* è selezionata nella configurazione della mappa.

Link

Facendo clic su un elemento della mappa si apre un menu con alcuni link disponibili. Facendo clic sul nome del host viene visualizzato il menu del host.

Pulsanti

I pulsanti sulla destra offrono le seguenti opzioni:

Edit map	Vai alla modifica del contenuto della mappa.
	Aggiungi la mappa al widget dei preferiti in Dashboards.
	La mappa è nel widget dei preferiti in Dashboards. Fare clic per rimuovere la mappa dal widget dei preferiti.

I pulsanti della modalità di visualizzazione, comuni a tutte le sezioni, sono descritti nella pagina Monitoring.

Sommario leggibile nelle mappe

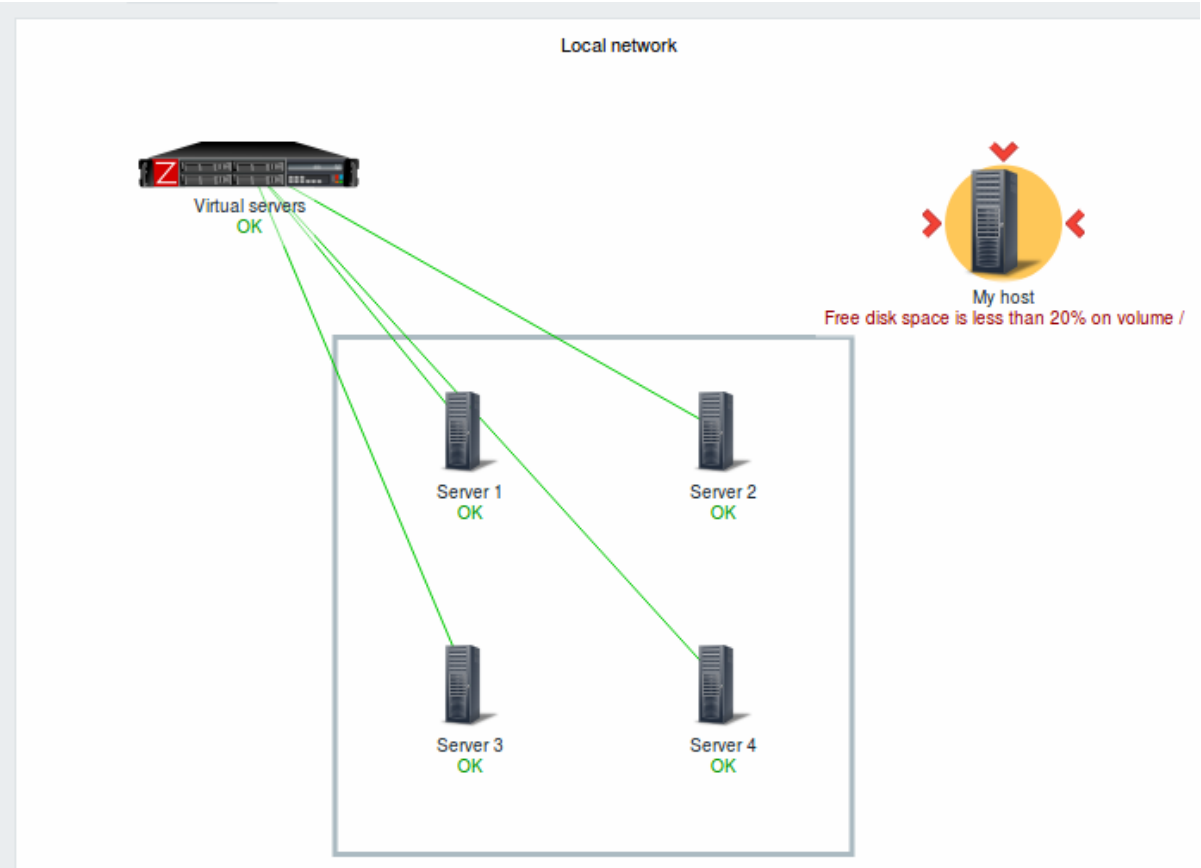
È disponibile una proprietà nascosta "aria-label" che consente l'accesso alle informazioni sulla mappa per essere letto con uno screen reader. Sia la descrizione generale della mappa che la descrizione del singolo elemento sono disponibili, nel seguente formato:

- per la descrizione della mappa: `<Nome mappa>, <* di * elementi in stato problematico>, <* problemi in totale>.`
- per descrivere un elemento con un problema: `<Tipo elemento>, Stato <Stato elemento>, <Nome elemento>, <Descrizione problema>.`
- per descrivere un elemento con più problemi: `<Tipo elemento>, Stato <Stato elemento>, <Nome elemento>, <* problemi>.`
- per descrivere un elemento senza problemi: `<Tipo elemento>, Stato <Stato elemento>, <Nome elemento>.`

Ad esempio, questa descrizione è disponibile:

```
'Rete locale, 1 elemento su 6 in stato problematico, 1 problema in totale. Host, Problema di stato, 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
```

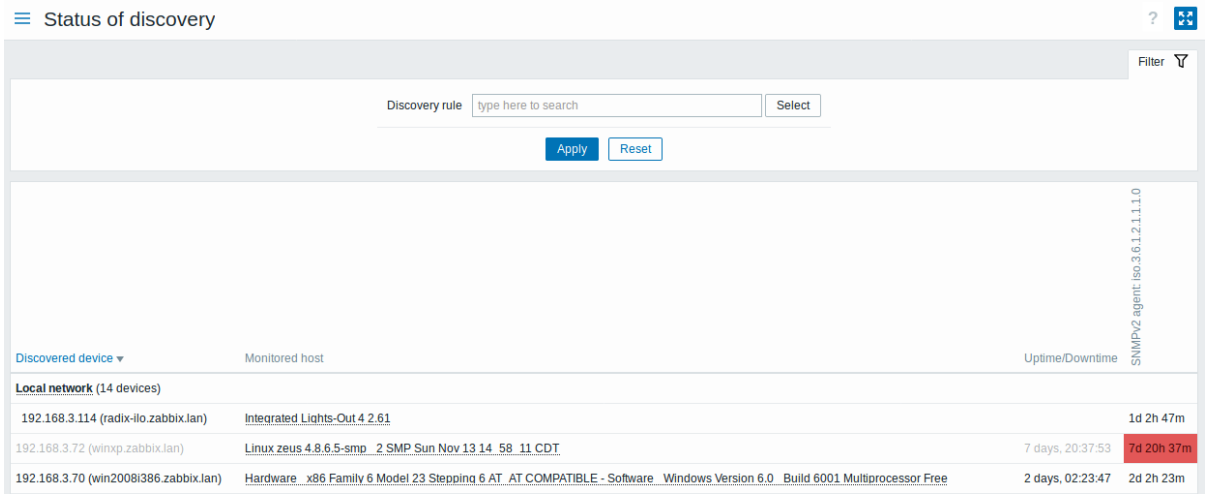
aprirà la mappa con quel nome (Rete locale).

Se sono specificati sia `sysmapid` (ID mappa) che `mapname` (nome mappa), `mapname` ha una priorità maggiore.

5 Individuazione

Panoramica

Nella sezione *Monitoraggio* → *Individuazione* vengono mostrati i risultati della individuazione di rete. I dispositivi individuati sono ordinati in base alla regola di individuazione.



Dati visualizzati:

Colonna	Descrizione
Dispositivo individuato	I dispositivi individuati sono elencati, raggruppati in base alla regola di individuazione. Facendo clic sulla regola di individuazione si apre il menu della regola, che contiene il collegamento al modulo di configurazione della regola di individuazione.
Host monitorato	Se un dispositivo è già monitorato, il nome dell'host verrà elencato in questa colonna. Facendo clic sul nome dell'host si apre il menu host.
Tempo di attività/Tempo di inattività	In questa colonna viene visualizzata la durata durante la quale il dispositivo è stato individuato oppure perso dopo la precedente individuazione.
Controllo di individuazione	Viene visualizzato lo stato del singolo servizio (controllo di individuazione) per ciascun dispositivo individuato. Una cella rossa indica che il servizio non è disponibile. Il tempo di attività o di inattività del servizio è incluso nella cella. Questa colonna viene visualizzata solo se il servizio è stato trovato su almeno un dispositivo individuato.

Pulsanti

I pulsanti della modalità di visualizzazione comuni a tutte le sezioni sono descritti in Monitoraggio pagina.

Utilizzo del filtro

Puoi utilizzare il filtro per visualizzare solo le regole di discovery a cui si è interessati. Per migliorare le prestazioni della ricerca, i dati vengono cercati con macro irrisolte.

Con niente selezionato nel filtro, tutte le regole di individuazione abilitate vengono visualizzate. Per selezionare una regola di individuazione specifica per la visualizzazione, iniziare a digitare il suo nome nel filtro. Tutte le regole di individuazione abilitate corrispondenti saranno elencati per la selezione. È possibile selezionare più di una regola di rilevamento.

3 Servizi

Panoramica

Il menu Servizi è per le funzioni monitoraggio servizi di Zabbix.

1 Servizi

Panoramica

In questa sezione puoi vedere uno stato di alto livello dell'insieme dei servizi che sono stati configurati in Zabbix, in base alla tua infrastruttura.

Un servizio può essere una gerarchia composta da diversi livelli di altri servizi, chiamati servizi "figlio", che contribuiscono allo stato complessivo del servizio (vedi anche una panoramica della funzionalità di monitoraggio dei servizi.)

Le principali categorie dello stato del servizio sono *OK* o *Problema*, dove lo stato *Problema* è espresso dal nome e dal colore della corrispondente severità del problema.

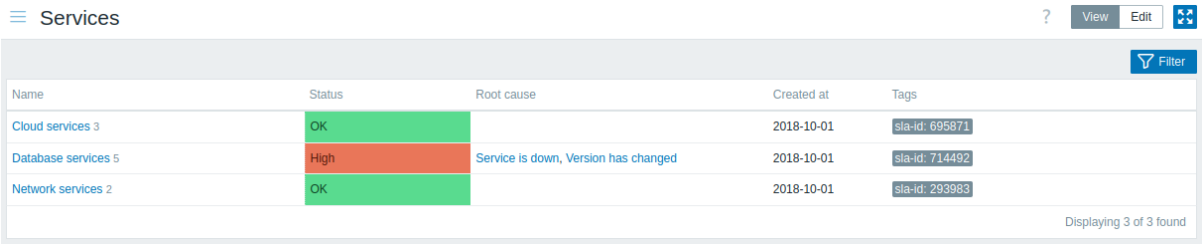
Mentre la modalità di visualizzazione consente di monitorare i servizi con il loro stato e altri dettagli, puoi anche configurare la gerarchia dei servizi in questa sezione (aggiungere/modificare servizi e servizi figlio) passando alla modalità di modifica.

Per passare dalla modalità di visualizzazione a quella di modifica (e viceversa), fai clic sul pulsante corrispondente nell'angolo in alto a destra:

- View - visualizzare i servizi
- Edit - aggiungere/modificare servizi e servizi figlio

Tieni presente che l'accesso alla modifica dipende dalle impostazioni del ruolo utente.

Visualizzazione dei servizi



Viene visualizzato un elenco dei servizi esistenti.

Dati visualizzati:

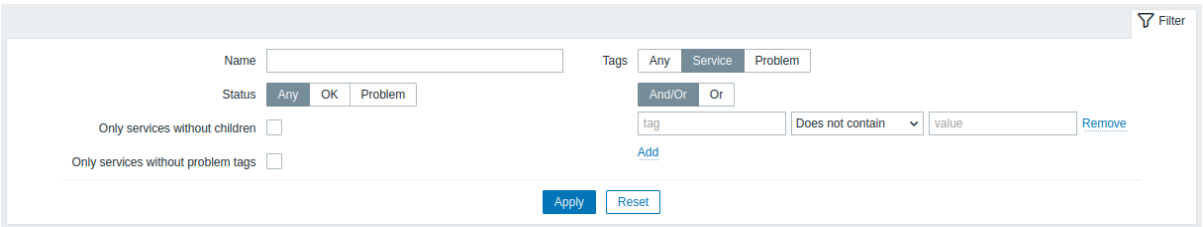
Parametro	Descrizione
Name	Nome del servizio. Il nome del servizio è un collegamento ai dettagli del servizio. Il numero dopo il nome indica quanti servizi figli ha il servizio.
Status	Stato del servizio: **OK** - nessun problema **<colore del problema e gravità>** - indica un problema e la sua gravità. In caso di più problemi, vengono visualizzati il colore e la gravità del problema più critico.
Root cause	Sono elencati i problemi sottostanti che influenzano direttamente o indirettamente lo stato del servizio. Gli stessi problemi sono elencati come restituiti dalla macro {SERVICE.ROOTCAUSE}. Fare clic sul nome del problema per visualizzarne maggiori dettagli in *Monitoring → Problems*. I problemi che non influiscono sullo stato del servizio non sono inclusi nell'elenco.
Created at	Viene visualizzata l'ora in cui il servizio è stato creato.
Tags	Vengono visualizzati i tag del servizio. I tag vengono utilizzati per identificare un servizio nelle azioni del servizio e negli SLA.

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.



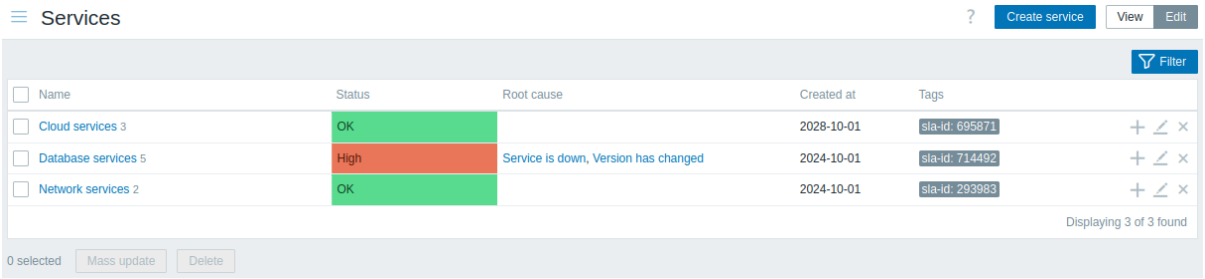
Parametro	Descrizione
Name	Filtra per nome del servizio.
Status	Filtra per stato del servizio.
Only services without children	Selezionare la casella per visualizzare solo i servizi senza servizi figli. Questo parametro è disponibile solo nella modalità di modifica del servizio.

Parametro	Descrizione
Only services without problem tags *Tags*	Selezionare la casella per visualizzare solo i servizi senza tag di problema. Questo parametro è disponibile solo nella modalità di modifica del servizio. 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

Fare clic sul pulsante *Modifica* per accedere alla modalità di modifica. In modalità di modifica, l'elenco viene integrato con caselle di controllo davanti alle voci e anche con queste opzioni aggiuntive:

-  - aggiungere un servizio figlio a questo servizio
-  - modificare questo servizio
-  - eliminare questo servizio

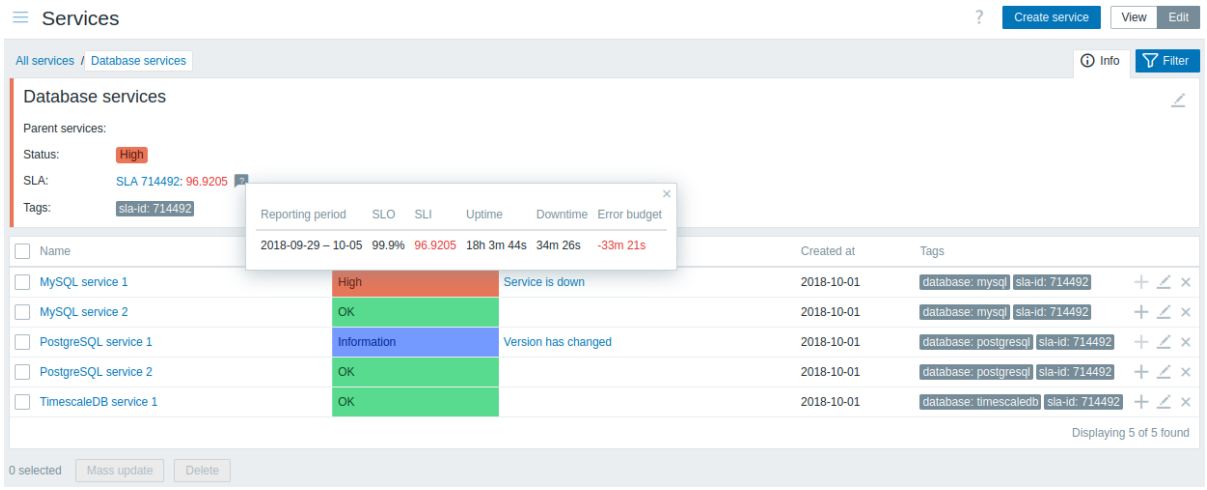


Per configurare un nuovo servizio, fare clic sul pulsante *Crea servizio* nell'angolo superiore destro.

Dettagli del servizio

Per accedere ai dettagli del servizio, fare clic sul nome del servizio. Per tornare all'elenco di tutti i servizi, fare clic su *Tutti i servizi*.

I dettagli del servizio includono il riquadro informativo e l'elenco dei servizi figli.



Per accedere al riquadro informativo, fare clic sulla scheda *Info*. Il riquadro informativo contiene le seguenti voci:

- Nomi dei servizi padre (se presenti)
- Stato corrente di questo servizio
- SLA correnti di questo servizio, nel formato `Nome SLA:indicatore del livello di servizio`. Anche 'Nome SLA' è un collegamento al report SLA per questo servizio. Se si posiziona il puntatore del mouse sul riquadro informativo 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 informativo contiene anche un collegamento alla configurazione del servizio.

Per utilizzare il filtro per i servizi figli, fare clic sulla scheda *Filtro*.

In modalità di modifica, l'elenco dei servizi figli è integrato con ulteriori opzioni di modifica:

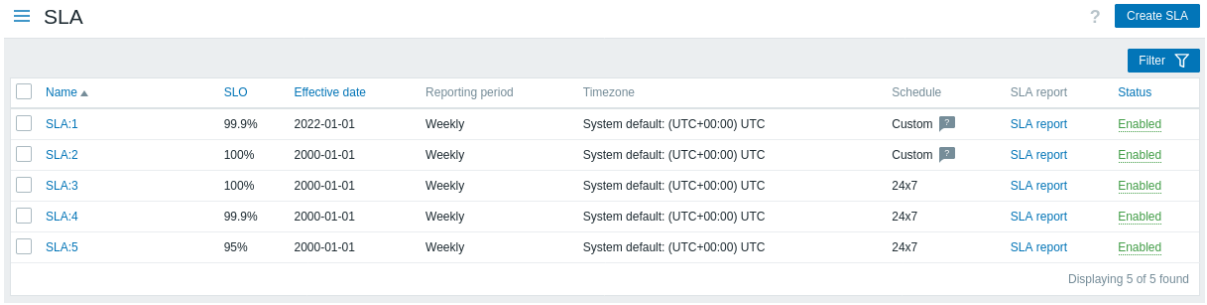
-  - aggiungere un servizio figlio a questo servizio
-  - modificare questo servizio
-  - eliminare questo servizio

2 SLA

Panoramica

Questa sezione permette di visualizzare e configurare gli SLA.

SLA



Viene visualizzato un elenco degli SLA configurati. *Nota* solo gli SLA relativi ai servizi accessibili all'utente verranno visualizzati (in sola lettura, a meno che *Gestisci SLA* non sia abilitato per il ruolo utente).

Dati visualizzati:

Parametro	Descrizione
Nome	Viene visualizzato il nome SLA. Il nome è un collegamento a Configurazione SLA.
SLO	Viene visualizzato l'obiettivo del livello di servizio (SLO).
Data effettiva	Viene visualizzata la data di inizio del calcolo SLA.

Parametro	Descrizione
Periodo del rapporto	Viene visualizzato il periodo utilizzato nel rapporto SLA: *giornaliero*, *settimanale*, *mensile*, *trimestrale* o *annuale*.
Fuso orario	Viene visualizzato il fuso orario SLA.
Pianificazione	Viene visualizzata la pianificazione SLA - 24 ore su 24, 7 giorni su 7 o personalizzata.
Rapporto SLA	Fare clic sul collegamento per visualizzare il rapporto SLA per questo SLA.
Stato	Viene visualizzato lo stato SLA - abilitato o disabilitato.

3 Report SLA

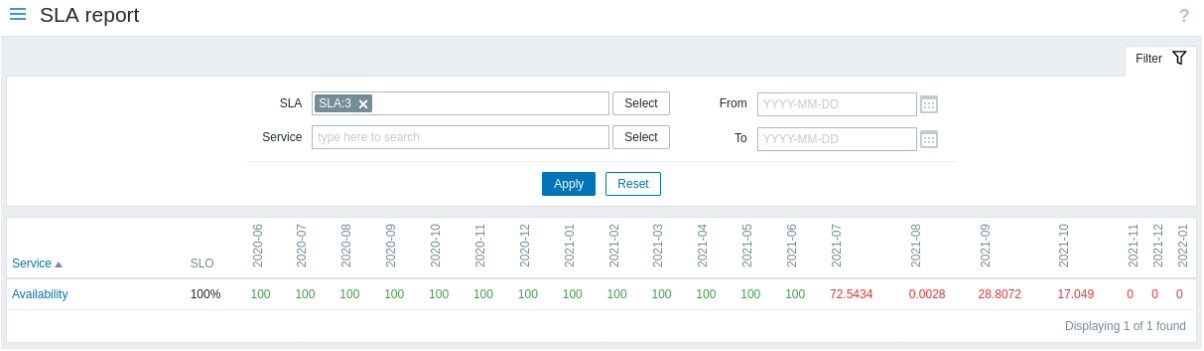
Panoramica

Questa sezione consente di visualizzare i report SLA, in base ai criteri selezionati nel filtro.

I report SLA possono anche essere visualizzati come widget della dashboard.

Report

Il filtro permette di selezionare il report in base al nome dello SLA e al Nome di Servizio. È anche possibile limitare il periodo visualizzato.

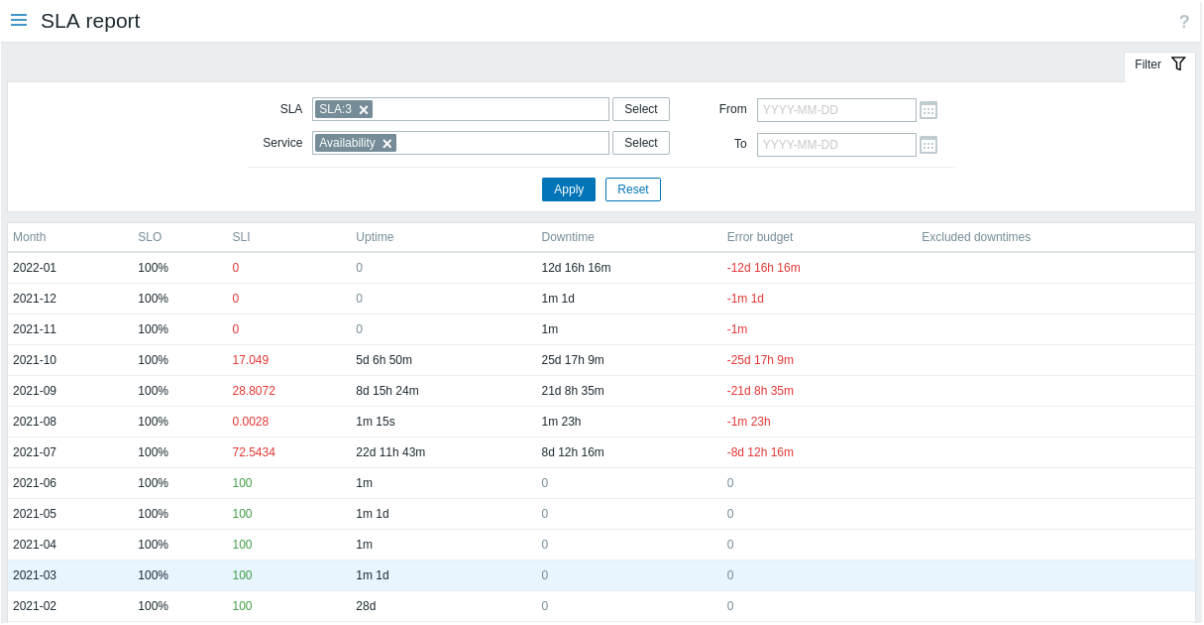


Ciascuna colonna (periodo) visualizza lo SLI per quel periodo. SLI che violano il set Gli SLA sono evidenziati in rosso.

Nel report vengono visualizzati 20 periodi. È possibile visualizzare un massimo di 100 periodi, se sono specificate sia la data *Da* che *A*.

Dettagli del report

Se si fa clic sul nome del servizio nel report, è possibile accedere a un altro report che mostra una vista più dettagliata.



Si noti che la durata negativa del problema non influisce sul calcolo o sulla reportistica dello SLA.

4 Inventario

Panoramica

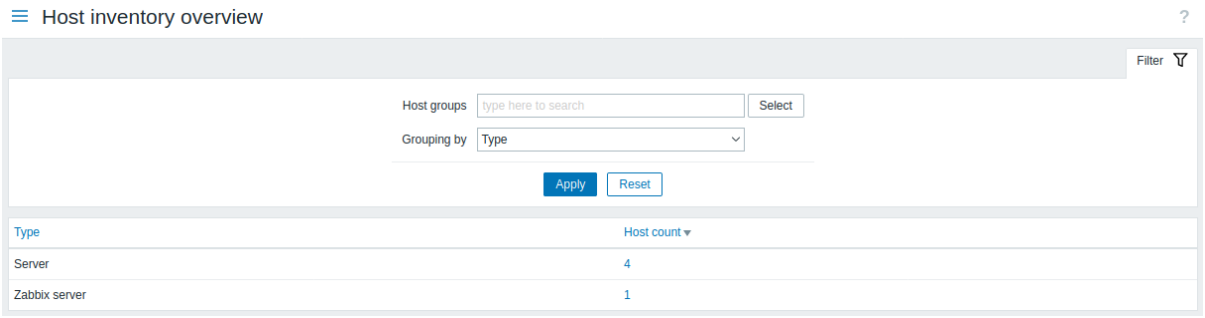
Il menu Inventario presenta sezioni che forniscono una panoramica dei dati dell'host in base a un parametro scelto, nonché la possibilità di visualizzare di dettagli dell'inventario dell'host.

1. Panoramica

Panoramica

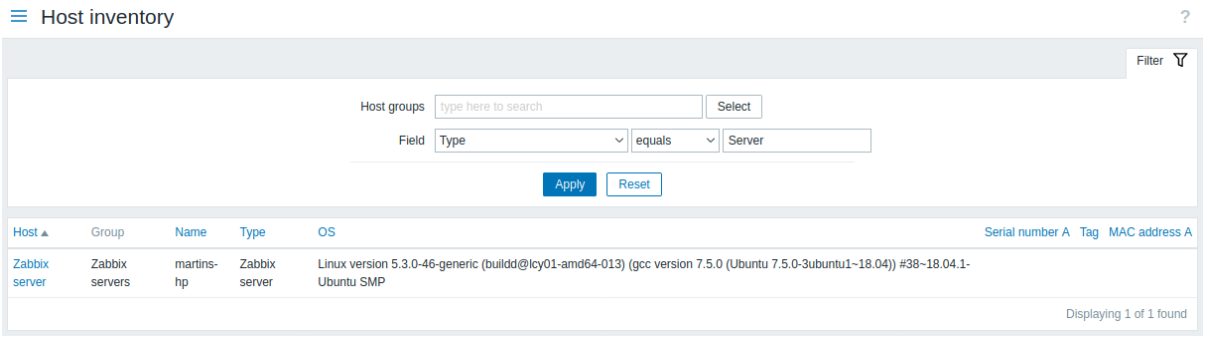
La sezione *Inventario* → *Panoramica* fornisce modi per avere una panoramica dei dati dell'inventario host.

Per visualizzare una panoramica, scegli i gruppi host (o nessuno) e il campo di inventario in base al quale dati vorresti visualizzare. Il numero di host corrispondente a ciascuna voce del campo prescelto verrà visualizzato.



La completezza di una panoramica dipende dalla quantità di inventario le informazioni vengono mantenute con gli host.

I numeri nella colonna *Conteggio host* sono link; conducono a questi host possono essere filtrati nella tabella *Host Inventories*.

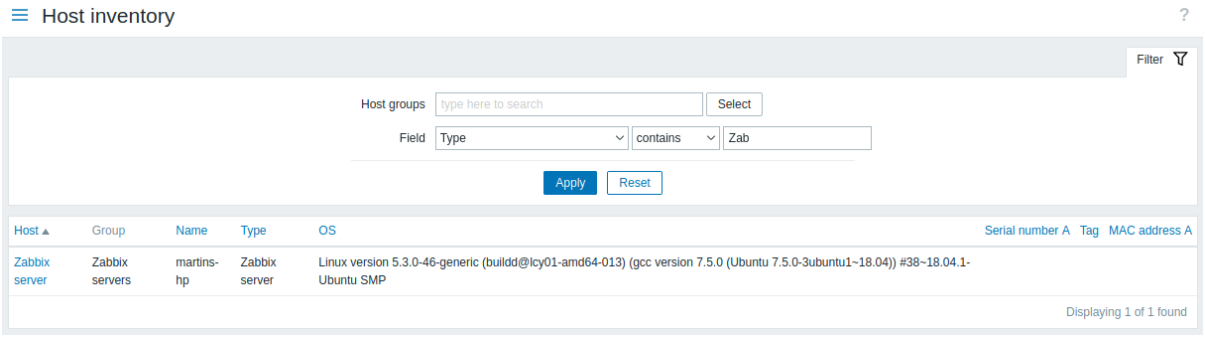


2 Host

Panoramica

Nella sezione *Inventory* → *Hosts* vengono visualizzati i dati di inventario dei host.

Puoi filtrare i host per gruppo/i di host e per qualsiasi campo di inventario, in modo da visualizzare solo i host di tuo interesse.

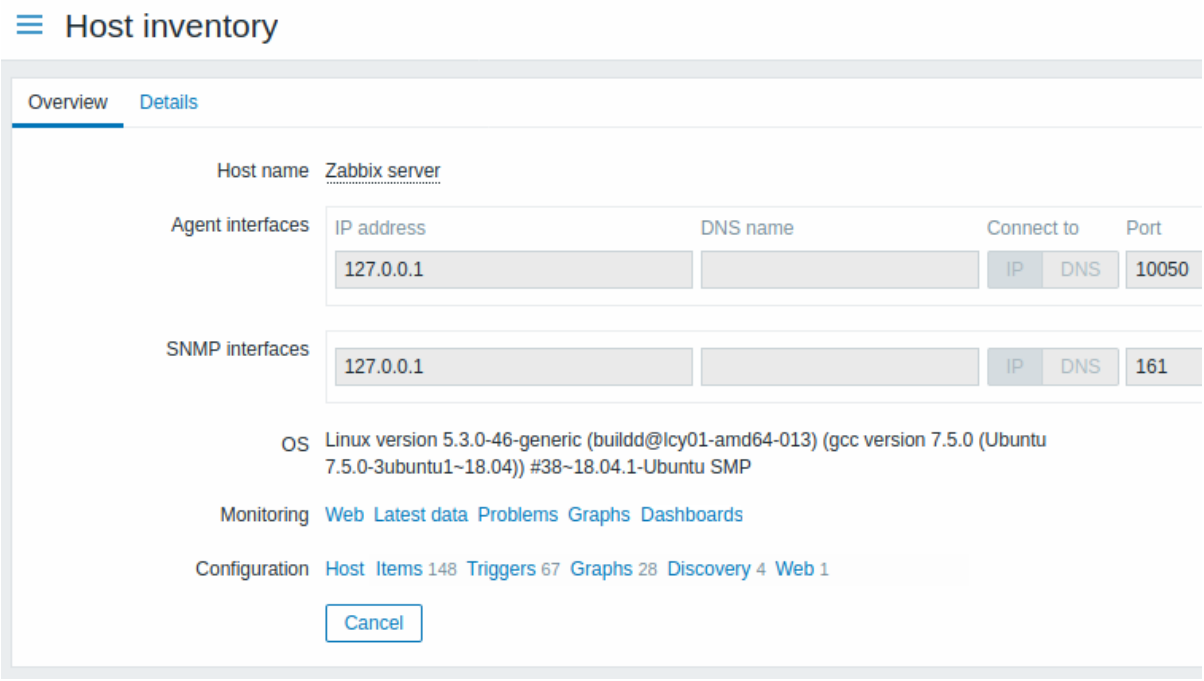


Per visualizzare tutti gli inventari dei host, premi il pulsante "Reset".

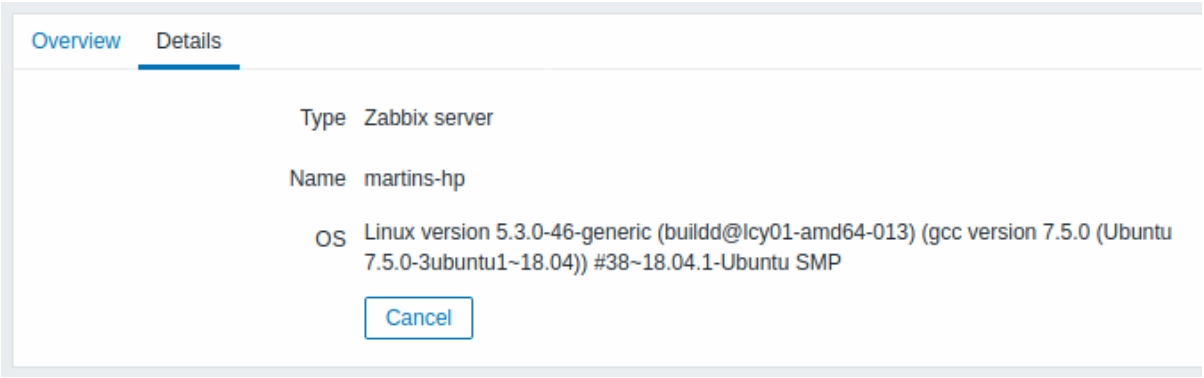
Sebbene nella tabella vengano mostrati solo alcuni campi chiave dell'inventario, puoi anche visualizzare tutte le informazioni di inventario disponibili per quel host. Per farlo, fai clic sul nome del host nella prima colonna.

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.

5 Report

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.

1 Informazioni di sistema

Panoramica

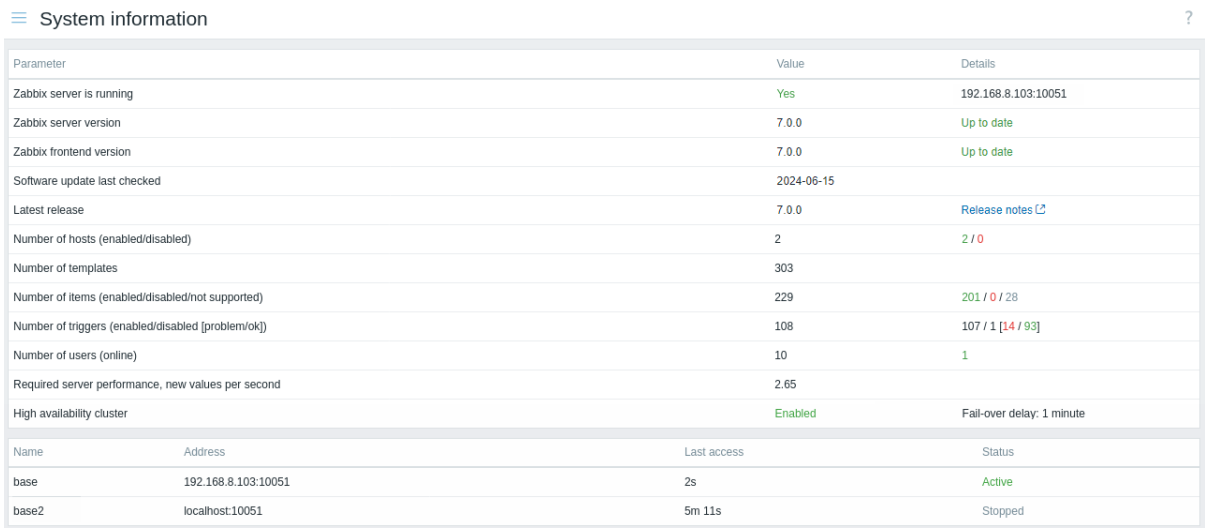
In *Report* → *Informazioni di sistema* viene visualizzato un riepilogo dei principali dati del server Zabbix e del sistema. I dati di sistema vengono raccolti utilizzando item interni.

Si noti che in una configurazione ad alta disponibilità è possibile reindirizzare la sorgente delle informazioni di sistema (istanza del server). Per farlo, modificare il file *zabbix.conf.php*: rimuovere il commento e impostare `$ZBX_SERVER` oppure sia `$ZBX_SERVER` che `$ZBX_SERVER_PORT` su un server diverso da quello mostrato come attivo. Si noti che, se si imposta solo `$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 di alta disponibilità. Questo blocco è visibile solo agli utenti Zabbix *Super Admin*.

Informazioni di sistema è disponibile anche come widget della dashboard.

Statistiche di sistema



Dati visualizzati:

Parametro	Valore	Dettagli
Zabbix server is running	Stato di Zabbix server: **Sì** - il server è in esecuzione **No** - il server non è in esecuzione *Nota:* Per visualizzare il resto delle informazioni, il frontend web richiede che il server sia in esecuzione e che sul server sia stato avviato almeno un processo trapper (parametro StartTrappers nel file zabbix_server.conf > 0).	Posizione e porta di Zabbix server.
Zabbix server version	Viene visualizzato il numero della versione corrente del server. *Nota:* Viene visualizzato solo quando Zabbix server è in esecuzione.	Viene visualizzato lo stato della versione del server: **Aggiornato** - è in uso la versione più recente; **Nuovo aggiornamento disponibile** - è disponibile una versione più recente; **Obsoleto** - il periodo di supporto completo per questa versione è scaduto. Questa informazione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione di Zabbix server. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non esistono dati sulla versione corrente.
Zabbix frontend version	Viene visualizzato il numero della versione di Zabbix frontend.	Viene visualizzato lo stato della versione di Zabbix frontend: **Aggiornato** - è in uso la versione più recente; **Nuovo aggiornamento disponibile** - è disponibile una versione più recente; **Obsoleto** - il periodo di supporto completo per questa versione è scaduto. Questa informazione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione di Zabbix server. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non esistono dati sulla versione corrente.

Parametro	Valore	Dettagli
Software update last checked	Viene visualizzata la data dell'ultimo controllo degli aggiornamenti software di Zabbix. Questa informazione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione di Zabbix server.	
Latest release	Viene visualizzato il numero di una release più recente (se disponibile) per la versione corrente di Zabbix. Questa informazione è disponibile solo se il controllo degli aggiornamenti software è abilitato nella configurazione di Zabbix server. Non viene visualizzato nulla se l'ultimo controllo degli aggiornamenti software è stato eseguito più di una settimana fa o se non esistono dati sulla versione corrente.	Viene visualizzato un collegamento alle note di rilascio dell'ultima release disponibile di Zabbix.
Number of hosts	Viene visualizzato il numero totale di host configurati.	Numero di host monitorati/non monitorati.
Number of templates	Viene visualizzato il numero totale di template.	
Number of items	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.
Number of triggers	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 dipendenti da item disabilitati o assegnati a host disabilitati vengono conteggiati come disabilitati.
Number of users	Viene visualizzato il numero totale di utenti configurati.	Numero di utenti online.
Required server performance, new values per second	Viene visualizzato il numero previsto di nuovi valori elaborati da Zabbix server al secondo.	*Required server performance* è una stima e può essere utile come linea guida. Per numeri precisi dei valori elaborati, utilizzare l'internal item `zabbix[wcache,values,all]`. Nel calcolo sono inclusi gli item abilitati degli host monitorati. Gli item log vengono conteggiati come un valore per intervallo di aggiornamento dell'item. Vengono conteggiati i valori con intervallo regolare; i valori con intervallo flessibile e di pianificazione non vengono conteggiati. Il calcolo non viene adattato durante un periodo di manutenzione "nodata". Gli item trapper non vengono conteggiati.
Global scripts on Zabbix server	In questo campo verrà visualizzato **Disabilitato** se gli script globali sono disabilitati su Zabbix server impostando EnableGlobalScripts=0 nella configurazione del server.	
High availability cluster	Stato del cluster ad alta disponibilità per Zabbix server: **Disabilitato** - server standalone **Abilitato** - esiste almeno un nodo ad alta disponibilità	Se abilitato, viene visualizzato il ritardo di failover.

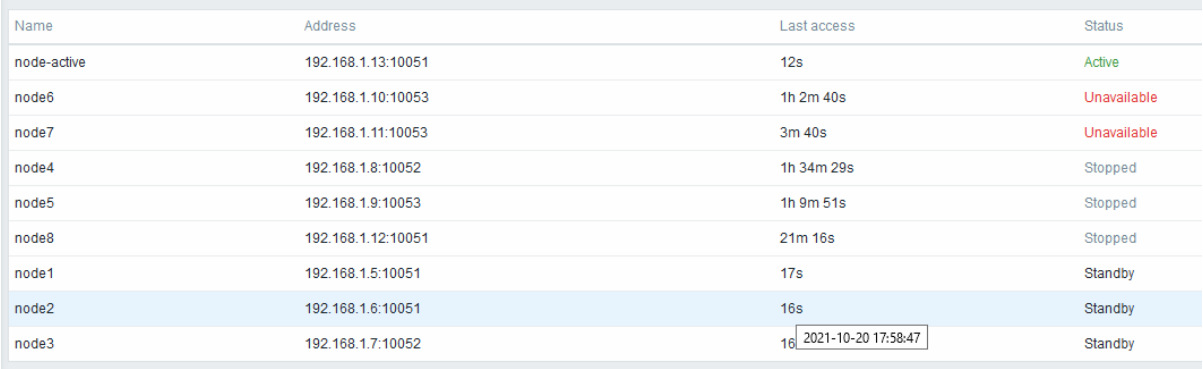
System information visualizzerà anche un messaggio di errore nelle seguenti condizioni:

- 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 tipo di *Super admin role*).
- La housekeeping per TimescaleDB è configurata in modo errato (le tabelle della cronologia o dei trend contengono chunk compressi, ma le opzioni *Override item history period* o *Override item trend period* sono disabilitate).

Nodi ad alta disponibilità

Se il cluster ad alta disponibilità è abilitato, viene visualizzato un altro blocco di dati con 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 2021-10-20 17:58:47	Standby

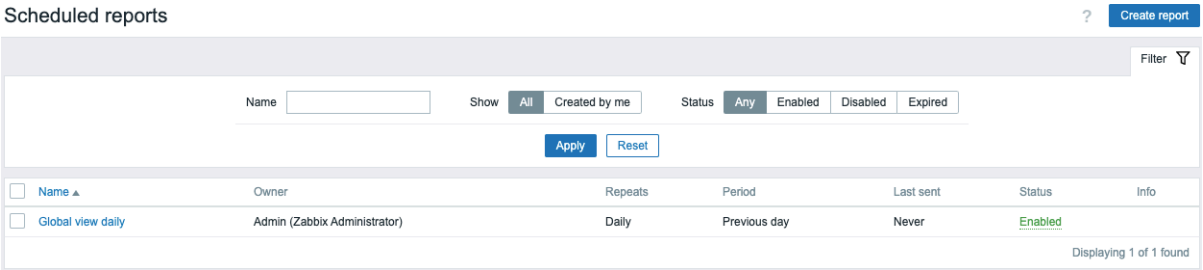
Dati visualizzati:

Colonna	Descrizione
Name	Nome del nodo, come definito nella configurazione del server.
Address	Indirizzo IP e porta del nodo.
Last access	Ora dell'ultimo accesso del nodo. Passando il cursore sulla cella viene mostrato il timestamp dell'ultimo accesso nel formato esteso.
Status	Stato del nodo. Le righe della tabella sono ordinate in base a questi stati secondo il seguente ordine di priorità: **Active** - il nodo è attivo e funzionante **Unavailable** - il nodo non è stato rilevato per un periodo superiore al ritardo di failover (potrebbe essere necessario verificarne il motivo) **Stopped** - il nodo è stato arrestato o non è riuscito ad avviarsi (potrebbe essere necessario avviarlo o eliminarlo) **Standby** - il nodo è attivo e in attesa

2 Report programmati

Panoramica

In *Report* → *Report pianificati*, gli utenti con autorizzazioni sufficienti possono configurare la generazione pianificata di versioni PDF delle dashboard, che verranno inviate tramite email ai destinatari specificati.



La schermata iniziale visualizza informazioni sui report pianificati, che possono essere filtrati per facilitare la navigazione - vedere la sezione Uso del filtro di seguito.

Dati visualizzati:

Column	Description
Name	Nome del report. Facendo clic su di esso si apre il modulo di configurazione del report.
Owner	Utente che ha creato il report.
Repeats	Frequenza di generazione del report (giornaliera/settimanale/mensile/annuale).

Column	Description
Period	Periodo di tempo per il quale il report viene preparato (giorno, settimana, mese o anno precedente).
Last sent	Data e ora di invio dell'ultimo report.
Status	Stato corrente del report (abilitato/disabilitato/scaduto). Gli utenti con autorizzazioni sufficienti possono modificare lo stato facendoci clic sopra - da "Abilitato" a "Disabilitato" (e viceversa); da "Scaduto" a "Disabilitato" (e viceversa). Per gli utenti con diritti insufficienti, lo stato non è cliccabile.
Info	Visualizza icone informative: Un'icona rossa indica che la generazione del report non è riuscita; passando il cursore sopra di essa verrà visualizzato un suggerimento 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 cursore sopra di essa verrà visualizzato un suggerimento 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:

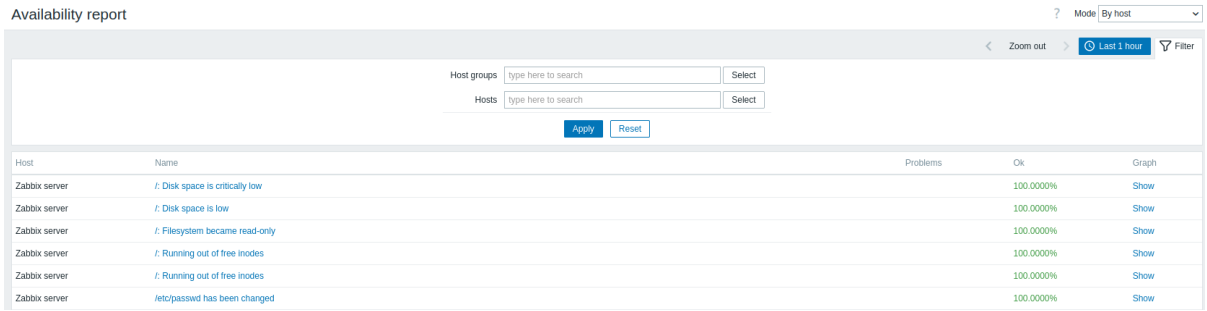
- Seleziona le caselle di controllo dei report da aggiornare nell'elenco
- Fai clic sul pulsante richiesto sotto l'elenco per apportare le modifiche (*Abilita*, *Disabilita* oppure *Elimina*)

3 Rapporto sulla disponibilità

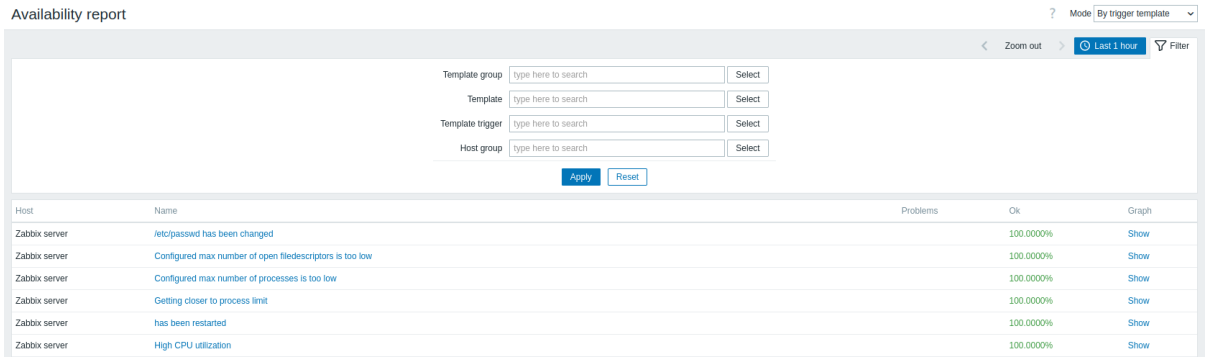
Panoramica

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.

Utilizzo del filtro

Il filtro può aiutare a ridurre il numero di host e/o trigger visualizzati. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il filtro si trova sotto il nome della sezione *Availability report*. Può essere aperto e compresso 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.
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.

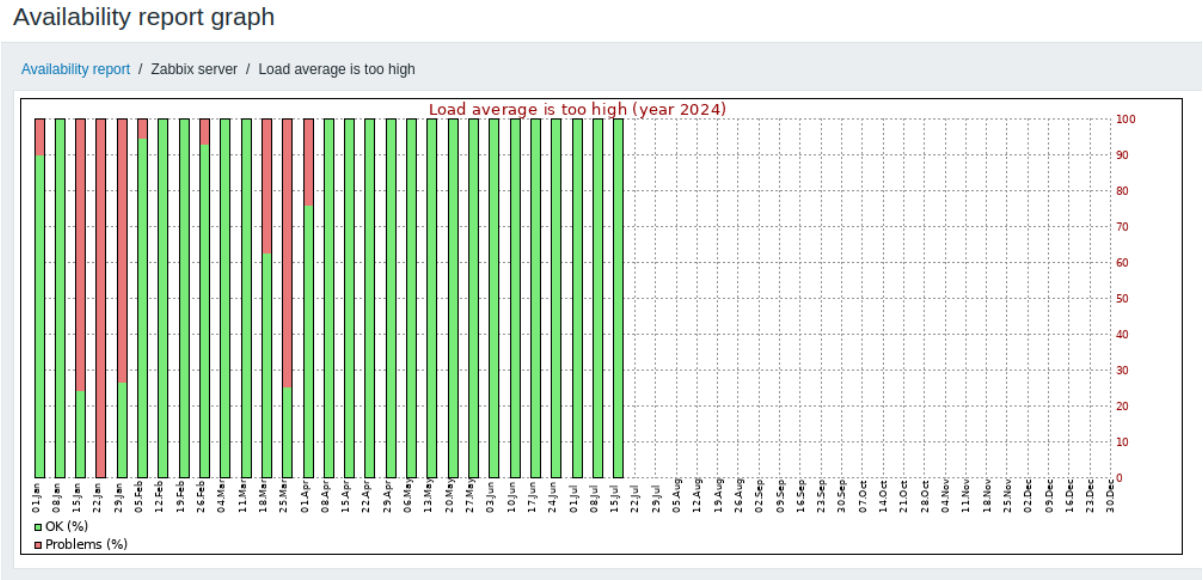
Filtraggio per host

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 usati più comunemente 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.



Il verde di una barra indica il tempo OK e il rosso il tempo di problema.

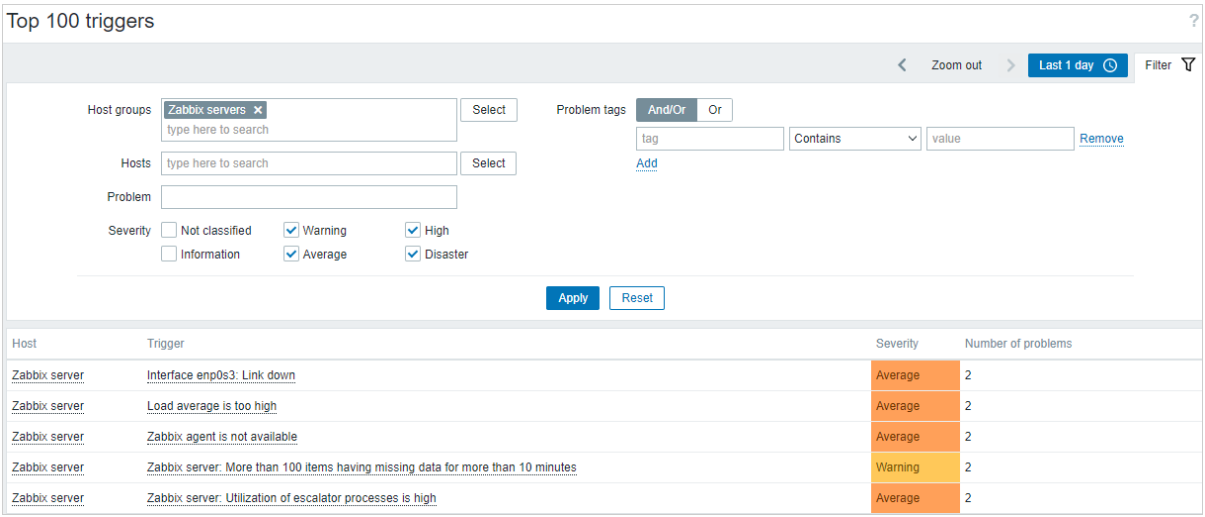
Effetto dei periodi di manutenzione

La manutenzione non esclude automaticamente il tempo dal report di disponibilità. Solo la manutenzione configurata come *Nessuna raccolta dati* interrompe il polling (quindi non vengono generati problemi) e pertanto lascia invariata la percentuale di disponibilità per i trigger interessati.

4 Top 100 trigger

Panoramica

In *Report* → *Top 100 trigger* puoi vedere i trigger con il maggior numero di problemi rilevati durante il periodo selezionato.



Sia le voci nelle colonne host che trigger sono collegamenti che offrono alcune utili opzioni:

- per host - facendo clic sul nome dell'host si apre il menu host
- per trigger - facendo clic sul nome del trigger vengono visualizzati i collegamenti agli ultimi eventi, al grafico semplice per ciascun item del trigger e ai moduli di configurazione del trigger stesso e di ciascun item del trigger

Uso del filtro

Puoi usare il filtro per visualizzare i trigger per gruppo host, host, nome del problema, tag o gravità del trigger. Specificando un gruppo host padre vengono selezionati implicitamente tutti i gruppi host nidificati. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il filtro si trova sotto il nome della sezione *Top 100 trigger*. Può essere aperto e compresso facendo clic sulla scheda *Filtro* a destra.

Selettore del periodo di tempo

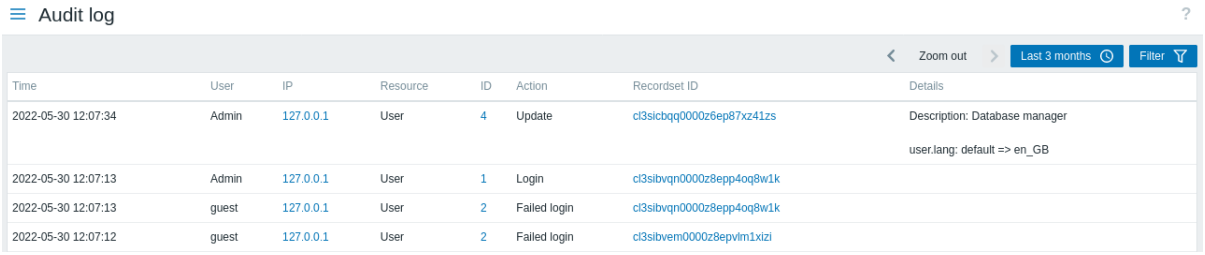
Il selettore del periodo di tempo accanto al filtro consente di selezionare con un clic del mouse i periodi richiesti più di frequente. Per ulteriori informazioni, vedi Selettori del periodo di tempo e host.

5 Log di audit

Panoramica

Nella sezione *Report* → *Log di audit* è possibile visualizzare i record delle attività degli utenti e del sistema.

Note:
Affinché i record di audit vengano raccolti e visualizzati, la casella *Abilita logging di audit* deve essere selezionata nella sezione *Amministrazione* → *Log di audit*. Senza questa impostazione abilitata, la cronologia delle attività non verrà registrata nel database e non sarà mostrata nel log di audit.



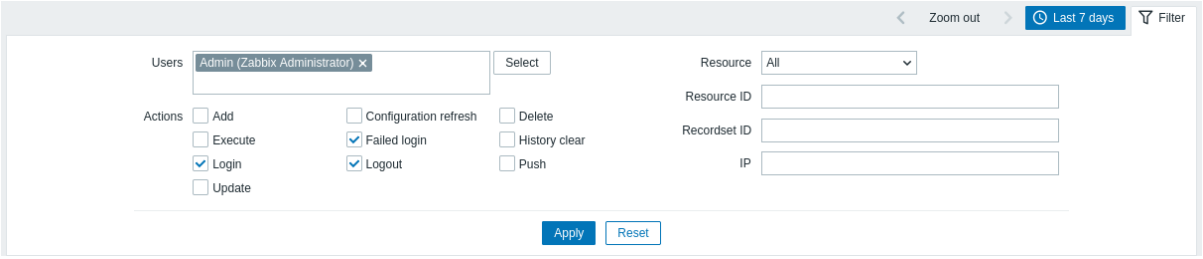
Il log di audit mostra i seguenti dati:

Column	Description
Ora	Timestamp del record di audit.
Utente	Utente che ha eseguito l'attività.
IP	IP da cui è stata avviata l'attività. Facendo clic sul collegamento ipertestuale, i record del log di audit verranno filtrati in base a questo IP.
Risorsa	Tipo della risorsa interessata (*Token API*, *Azione*, *Autenticazione*, *Autoregistrazione*, ecc.).
ID	ID della risorsa interessata. Facendo clic sul collegamento ipertestuale, i record del log di audit verranno filtrati in base a questo ID risorsa.
Azione	Tipo di attività (*Aggiungi*, *Aggiornamento configurazione*, *Elimina*, *Esegui*, *Accesso non riuscito*, *Cancella cronologia*, *Accesso*, *Disconnessione*, *Push*, *Aggiorna*).
ID set di record	ID condiviso per tutti i record del log di audit creati come risultato della stessa operazione. Ad esempio, quando si collega un template a un host, viene creato un record separato nel log di audit per ogni entità di template ereditata (item, trigger, ecc.) - tutti questi record avranno lo stesso *ID set di record*. Facendo clic sul collegamento ipertestuale, i record del log di audit verranno filtrati in base a questo *ID set di record*.
Dettagli	Descrizione della risorsa e informazioni dettagliate sull'attività eseguita. Se un record contiene più di due righe, verrà visualizzato un collegamento aggiuntivo *Dettagli*. Fare clic su questo collegamento per visualizzare l'elenco completo delle modifiche.

Note:
Quando un item trapper o un item agent HTTP (con trapping abilitato) ha ricevuto dei dati, verrà aggiunta una voce nel log di audit solo se i dati sono stati inviati utilizzando il metodo API `history.push`, e non l'utilità Zabbix sender.

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.

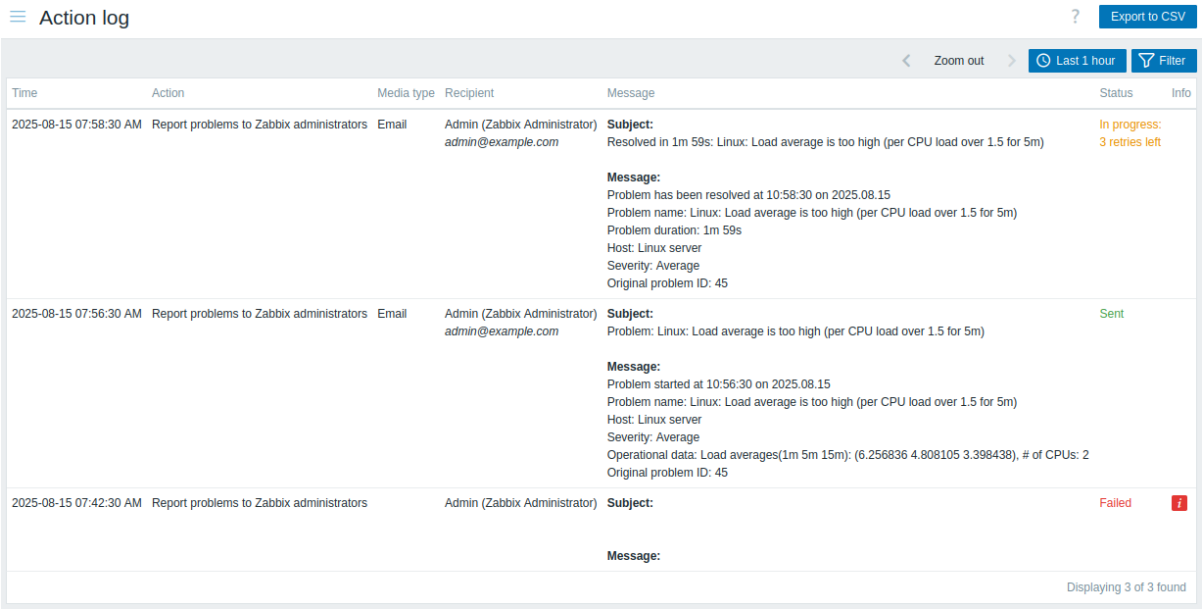
Selettore del periodo di tempo

Il selettore del periodo di tempo accanto al filtro consente di selezionare i periodi richiesti più spesso con un clic del mouse. Per ulteriori informazioni, vedere Time period and host selectors.

6 Registro delle azioni

Panoramica

In *Report* → *Log azioni* è possibile 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 supporto (ad es. Email, Jira, ecc.) utilizzato per l'invio di una notifica. Per le operazioni che hanno eseguito comandi remoti, questa colonna sarà vuota.
Recipient	Informazioni sul destinatario della notifica: nome utente, nome e cognome (tra parentesi), e informazioni aggiuntive a seconda del tipo di supporto (email, nome utente, ecc.). Per le operazioni che hanno eseguito comandi remoti, questa colonna sarà vuota.
Message	Contenuto del messaggio, del comando remoto o nome dello script globale. Un comando remoto è separato dal 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 visualizzato anche 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

Column	Description
Info	Informazioni sull'errore (se presenti) relative all'esecuzione dell'operazione.

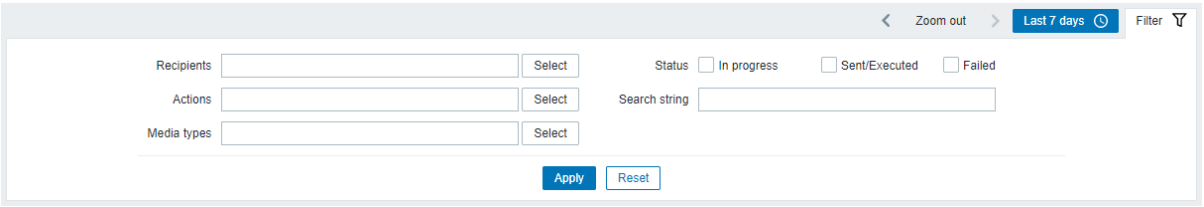
Pulsanti

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 del *Log delle azioni*. Può essere aperto e compresso facendo clic sulla scheda *Filtro* nell'angolo superiore destro della pagina.



È possibile utilizzare il filtro per restringere i record in base ai destinatari delle notifiche, alle azioni, ai tipi di supporto, allo stato oppure al contenuto del messaggio/comando remoto (*Stringa di ricerca*). Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Selettore del periodo di tempo

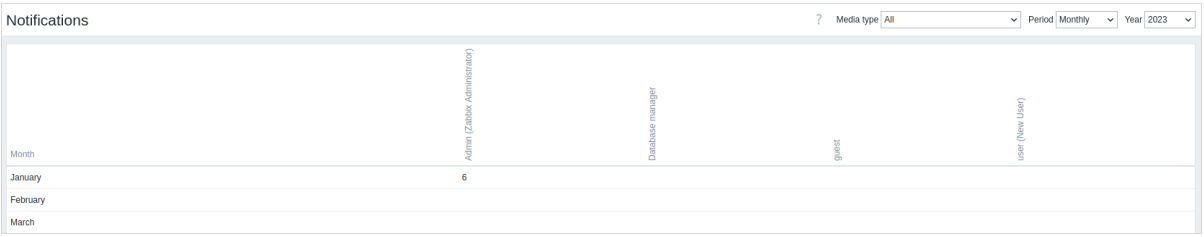
Il selettore del periodo di tempo accanto al filtro consente di selezionare i periodi richiesti più spesso con un clic del mouse. Per ulteriori informazioni, vedere Selettori del periodo di tempo e del host.

7 Notifiche

Panoramica

Nella sezione *Report* → *Notifiche* 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 (o tutti), il periodo (dati per ciascun giorno/settimana/mese/anno) e l'anno delle notifiche inviate.



Ogni colonna mostra i totali per un singolo utente del sistema.

6 Raccolta dei dati

Panoramica

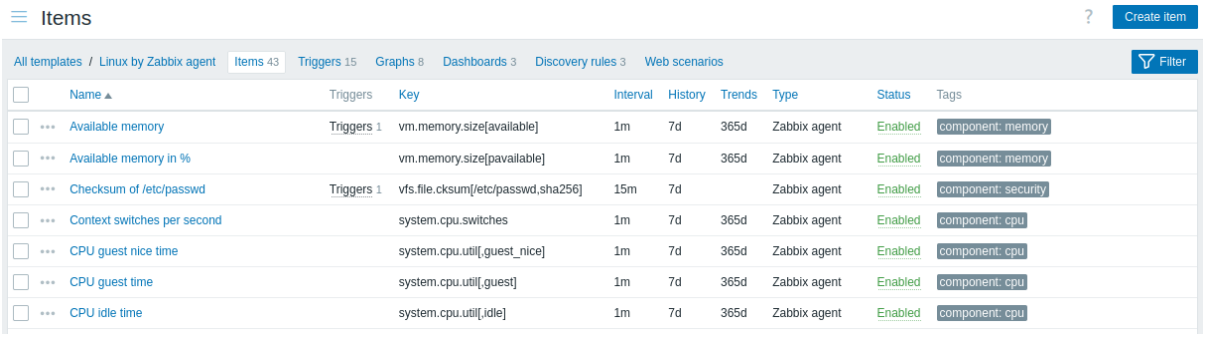
Questo menu include sezioni relative alla configurazione della raccolta dati.

1 Item

Panoramica

L'elenco degli item per un template è accessibile da *Raccolta dati* → *Template* facendo clic su *Item* per il rispettivo template.

Viene visualizzato un elenco degli item esistenti.



Dati visualizzati:

Colonna	Descrizione
Menu item	Fare clic sull'icona con i tre puntini per aprire il menu item.
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 del 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 di template.
Trigger	Passando il mouse su Trigger verrà visualizzato un riquadro informativo che mostra i trigger associati all'item. Il numero dei trigger viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item.
Intervallo	Viene visualizzata la frequenza del controllo.
Storico	Viene visualizzato per quanti giorni verrà conservato lo storico dei dati dell'item.
Trend	Viene visualizzato per quanti giorni verrà conservato lo storico 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. Possono essere visualizzati fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di vedere tutti i tag al passaggio del mouse.

Per configurare un nuovo item, fare clic sul pulsante *Crea item* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

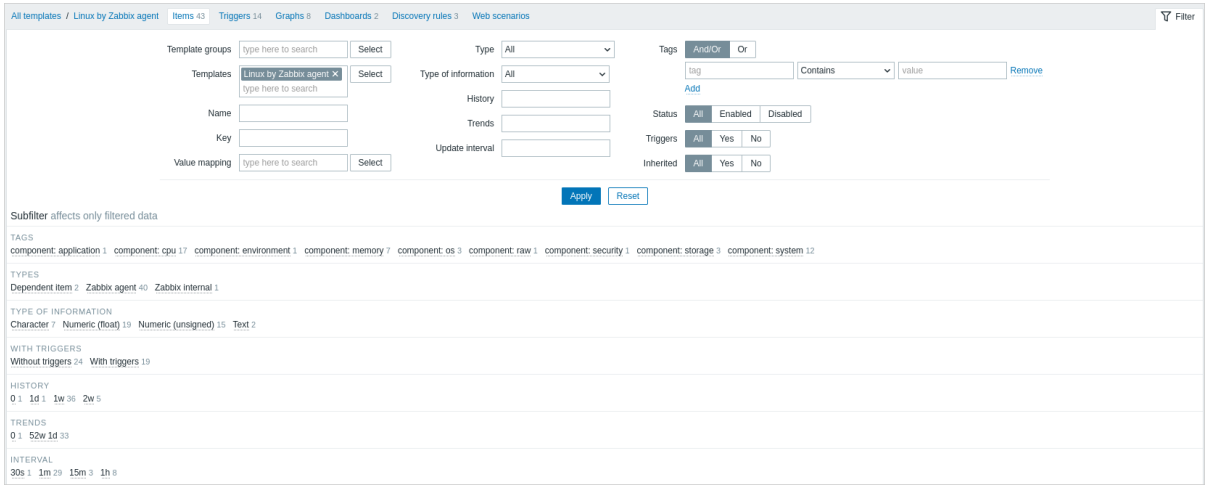
- *Abilita* - cambia lo stato dell'item in *Abilitato*.
- *Disabilita* - cambia lo stato dell'item in *Disabilitato*.
- *Copia* - copia gli item in altri host o template.
- *Aggiornamento di massa* - aggiorna diverse proprietà per più item contemporaneamente.
- *Elimina* - elimina gli item.

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi item, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

L'elenco degli item può contenere molti item. Utilizzando il filtro, è possibile filtrarne alcuni per individuare rapidamente gli item che si stanno cercando. Per prestazioni di ricerca migliori, 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 filtro desiderati.

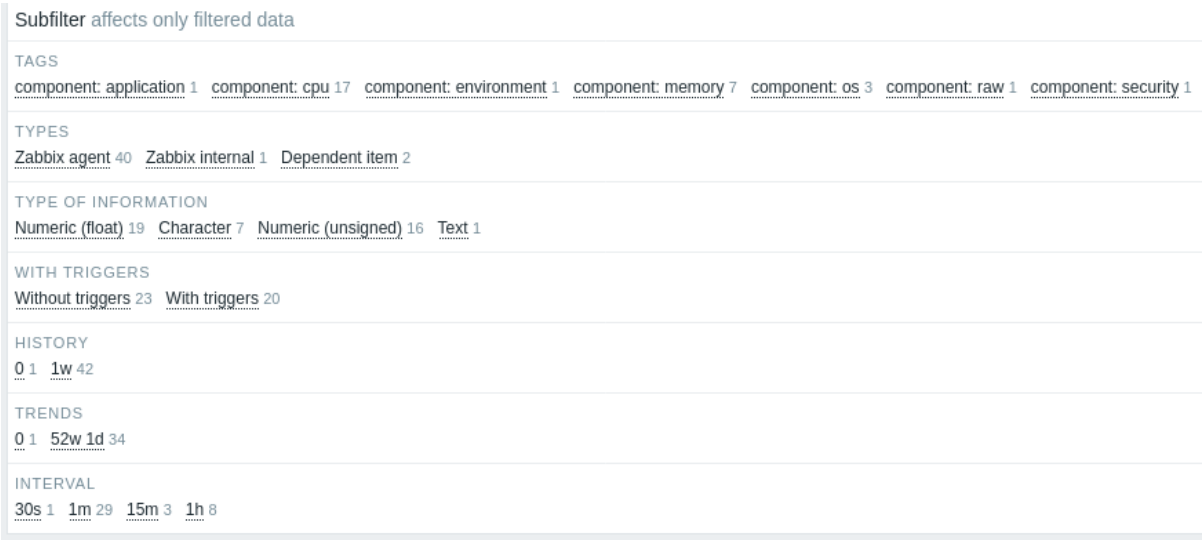


Parametro	Descrizione
Gruppi di template	Filtra per uno o più gruppi di template. Specificando un gruppo di template padre vengono implicitamente selezionati tutti i gruppi nidificati.
Template	Filtra per uno o più template.
Nome	Filtra per nome dell'item.
Chiave	Filtra per chiave dell'item.
Mappatura valori	Filtra per la mappa dei valori utilizzata. Questo parametro non viene visualizzato se l'opzione *Template* è vuota.
Tipo	Filtra per tipo di item (Zabbix agent, SNMP agent, ecc.).
Tipo di informazione	Filtra per tipo di informazione (Numerico senza segno, virgola mobile, ecc.).
Storico	Filtra in base a per quanto tempo viene conservato lo storico dell'item.
Trend	Filtra in base a per quanto tempo vengono conservati i trend dell'item.
Intervallo di aggiornamento	Filtra per intervallo di aggiornamento dell'item.
Tag	Specificare i tag per limitare il numero di item visualizzati. 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: **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) 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
Stato	Filtra per stato dell'item - *Abilitato* o *Disabilitato*.
Trigger	Filtra gli item con trigger (o senza trigger).
Ereditato	Filtra gli item ereditati (o non ereditati) dai template collegati.

Uso del sottofiltro

Il sottofiltro consente di modificare ulteriormente il filtraggio del filtro principale.

Contiene collegamenti cliccabili per un accesso rapido agli item correlati. Gli item sono correlati da un'entità comune: tag, tipo di item, intervallo di aggiornamento dell'item, ecc. 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 sottofiltri vengono generati in base ai dati filtrati, che sono limitati a 1000 record. Se si desidera visualizzare più record nel sottofiltro, è necessario aumentare il valore del parametro *Limit for search and filter results* (in *Administration -> General -> GUI*).

A differenza del filtro principale, il sottofiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da avere sempre informazioni aggiornate sulle opzioni di filtro disponibili e sui relativi contatori.

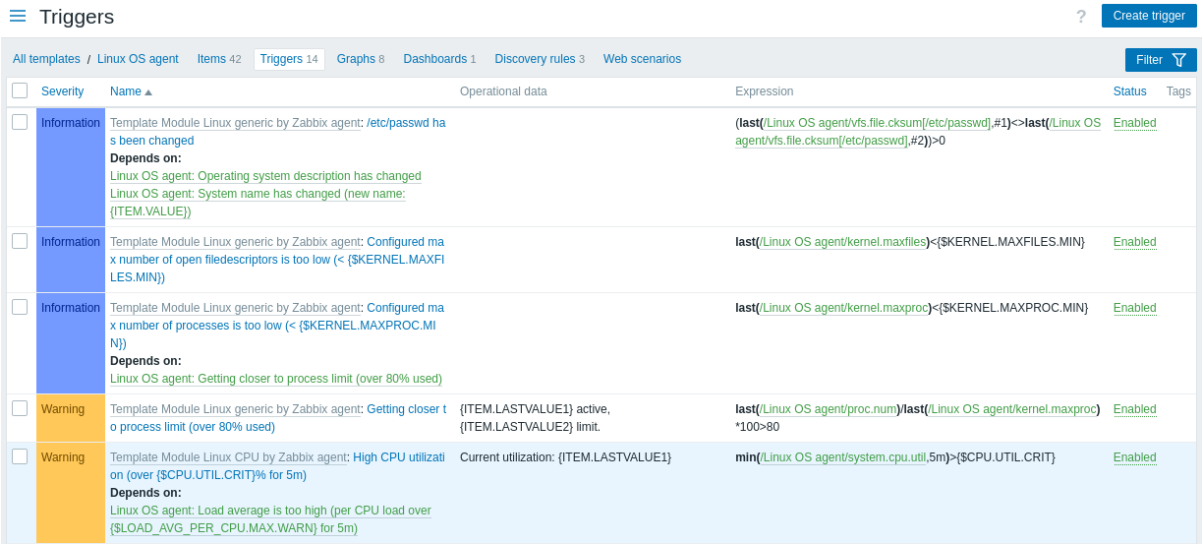
Il numero di entità visualizzate è limitato orizzontalmente a 100. Se sono di più, alla fine viene visualizzata un'icona con tre puntini; non è cliccabile.

Un numero accanto a ciascuna entità cliccabile indica il numero di item raggruppati in essa (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 state selezionate in precedenza nel sottofiltro.

2 Trigger

Panoramica

L'elenco dei trigger per un template è accessibile da *Raccolta dati* → *Template* facendo clic su *Trigger* per il template corrispondente.



Dati visualizzati:

Colonna	Descrizione
Gravità	La gravità del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.

Colonna	Descrizione
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 oppure 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 di 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 porta 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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

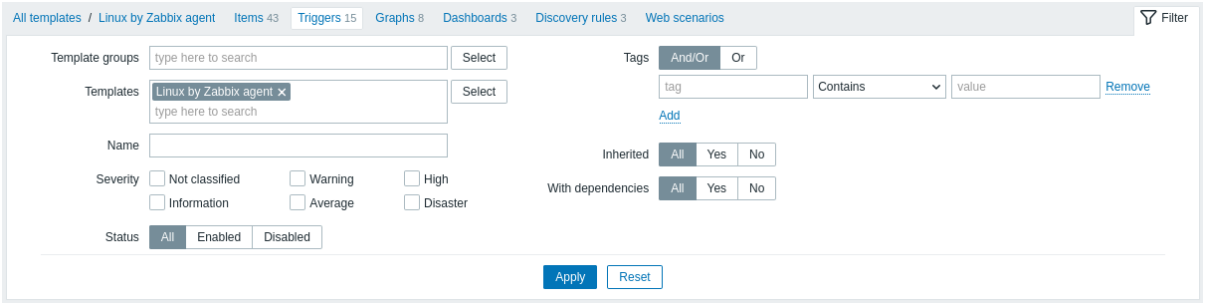
- *Abilita* - cambia lo stato del trigger in *Abilitato*
- *Disabilita* - cambia lo stato del trigger in *Disabilitato*
- *Copia* - copia i trigger in altri host o template
- *Aggiornamento di massa* - aggiorna contemporaneamente diverse proprietà per un certo numero di trigger
- *Elimina* - elimina i trigger

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi trigger, quindi fare 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 le macro non risolte.

L'icona *Filtro* è disponibile nell'angolo superiore destro. Facendo clic su di essa si aprirà un filtro in cui puoi 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 selezionati implicitamente anche tutti i gruppi nidificati.
Templates	Filtra per uno o più template. Se sopra sono già stati selezionati dei gruppi di template, la selezione dei template è limitata a tali gruppi.
Name	Filtra per nome del trigger.
Severity	Seleziona per filtrare in base a una o più severità del trigger.
Status	Filtra per stato di attivazione del trigger (Abilitato/Disabilitato).

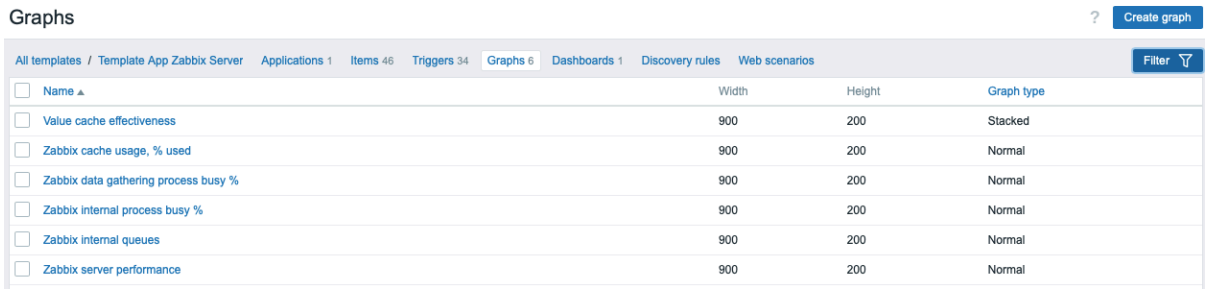
Parameter	Description
Tags	Filtra per nome e valore del tag del trigger. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori 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 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 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 Le macro e le funzioni macro sono supportate nei campi nome del tag e valore del tag.
Inherited	Filtra i trigger ereditati (o non ereditati) dai template collegati.
With dependencies	Filtra i trigger con (o senza) dipendenze.

3 Grafici

Panoramica

L'elenco dei grafici personalizzati per un template è accessibile da *Raccolta dati* → *Template* facendo clic su *Grafici* per il template corrispondente.

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.
Name	Nome del grafico personalizzato, visualizzato come collegamento blu ai dettagli del grafico. Facendo clic sul collegamento del 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 di template.
Width	Viene visualizzata la larghezza del grafico.
Height	Viene visualizzata l'altezza del grafico.
Graph type	Viene visualizzato il tipo di grafico: *Normale*, *Impilato*, *Torta* o *Esploso*.

Per configurare un nuovo grafico, fare clic sul pulsante *Crea grafico* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Copia* - copia i grafici su altri host o modelli
- *Cancella* - cancella i grafici

Per utilizzare queste opzioni, contrassegnare le caselle prima dei rispettivi grafici, quindi fare clic sul pulsante richiesto.

Uso del filtro

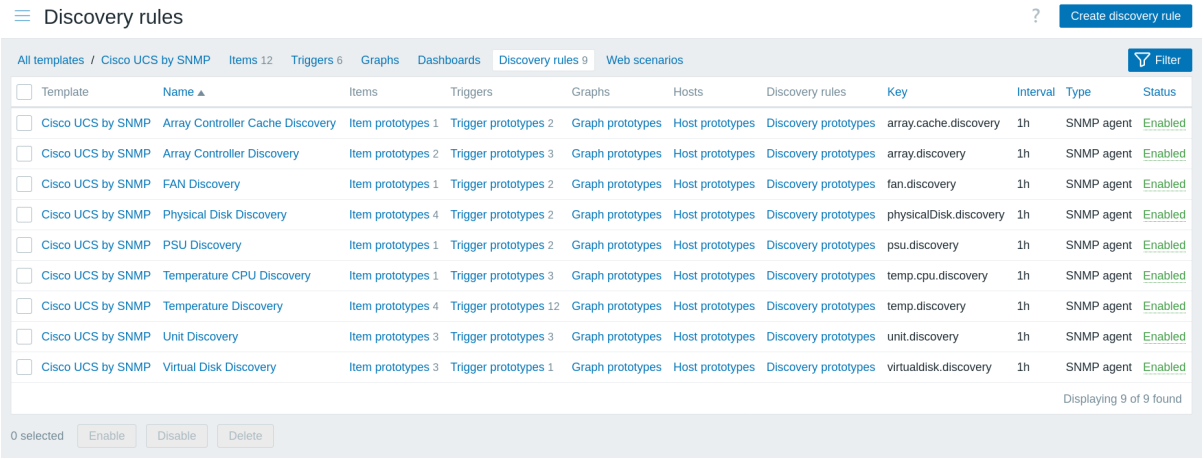
È possibile filtrare i grafici per gruppo di template e template. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

4 Regole di Discovery

Panoramica

L'elenco delle regole di low-level discovery per un template è accessibile da *Raccolta dati* → *Template* facendo clic su *Discovery* per il template corrispondente.

Viene visualizzato un elenco delle regole di low-level discovery 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.
Name	Nome della regola, visualizzato come link blu. Facendo clic sul nome della regola si apre il modulo di configurazione della regola di low-level discovery. Se la regola di discovery è ereditata da un altro template, il nome del template viene visualizzato prima del nome della regola come link grigio. Facendo clic sul link del template si apre l'elenco delle regole di discovery a quel livello del template.
Items	Viene visualizzato un link all'elenco dei prototipi di item. Il numero di prototipi di item esistenti è visualizzato in grigio.
Triggers	Viene visualizzato un link all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti è visualizzato in grigio.
Graphs	Viene visualizzato un link all'elenco dei prototipi di grafico. Il numero di prototipi di grafico esistenti è visualizzato in grigio.
Hosts	Viene visualizzato un link all'elenco dei prototipi di host. Il numero di prototipi di host esistenti è 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.).
Status	Viene visualizzato lo stato della regola di discovery: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo, da Abilitato a Disabilitato (e viceversa).

Per configurare una nuova regola di low-level 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:

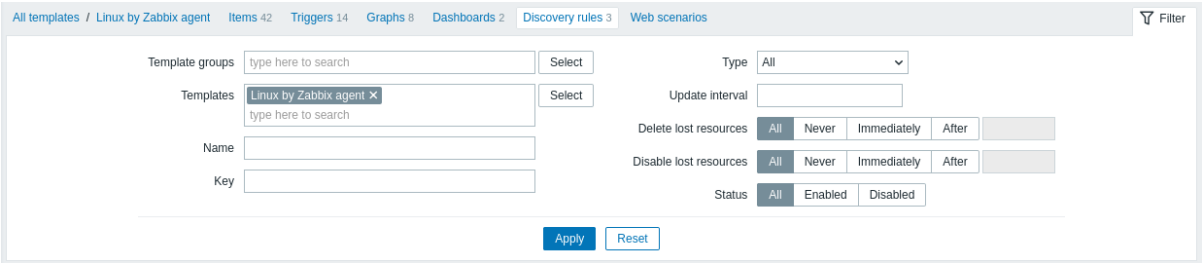
- *Abilita* - modifica lo stato della regola di rilevamento di basso livello in *Abilitato*
- *Disabilita* - modifica lo stato della regola di rilevamento di basso livello in *Disabilitata*
- *Elimina* - elimina le regole di rilevamento di basso livello

Per utilizzare queste opzioni, contrassegnare le caselle prima delle rispettive regole di rilevamento, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo le regole di discovery di tuo interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filtro* è disponibile nell'angolo superiore destro. Facendo clic su di essa si aprirà un filtro in cui puoi specificare i criteri di filtraggio desiderati, come template, nome della regola di discovery, chiave item, tipo di item, ecc.



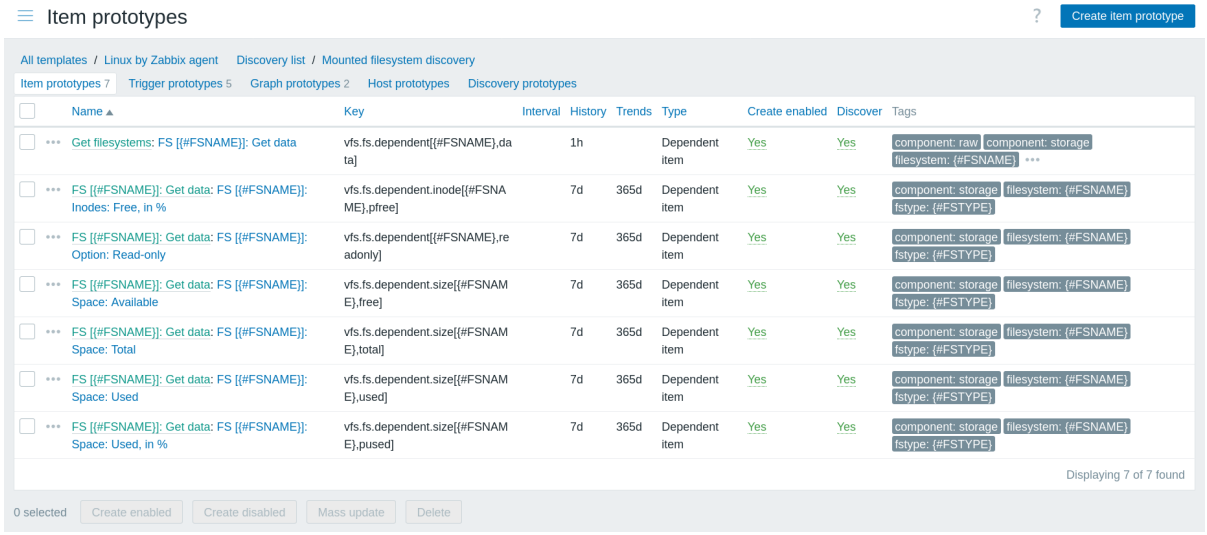
Parametro	Descrizione
Gruppi di template	Filtra per uno o più gruppi di template. Specificando un gruppo di template padre vengono selezionati implicitamente tutti i gruppi nidificati.
Template	Filtra per uno o più template.
Nome	Filtra per nome della regola di discovery.
Chiave	Filtra per chiave item di discovery.
Tipo	Filtra per tipo di item di discovery.
Intervallo di aggiornamento	Filtra per intervallo di aggiornamento. Non disponibile per Zabbix trapper e item dipendenti.
Elimina risorse perse	Filtra per periodo di *Elimina risorse perse*.
Disabilita risorse perse	Filtra per periodo di *Disabilita risorse perse*.
Stato	Filtra per stato della regola di discovery (Tutti/Abilitato/Disabilitato).

1 Prototipi di item

Panoramica

In questa sezione vengono visualizzati gli item prototype configurati di una regola di low-level discovery nel template.

Se il template è collegato al host, gli item prototype diventeranno la base per la creazione di veri items del host durante la low-level discovery.



Dati visualizzati:

Column	Description
Name	Viene visualizzato il nome dell'item prototype come link blu. Facendo clic sul nome si apre il modulo di configurazione dell'item prototype. Se l'item prototype appartiene a un template collegato, il nome del template viene visualizzato prima del nome dell'item prototype come link grigio. Facendo clic sul link del template si apre l'elenco degli item prototype a livello del template collegato. Se l'item prototype è del tipo "Dependent item", il nome del master item viene visualizzato prima del nome dell'item prototype in verde (come nel caso degli item prototype a livello host).
Key	Viene visualizzata la key dell'item prototype.
Interval	Viene visualizzata la frequenza del controllo.
History	Viene visualizzato per quanti giorni mantenere la cronologia dei dati dell'item.
Trends	Viene visualizzato per quanti giorni mantenere la cronologia dei trend dell'item.
Type	Viene visualizzato il tipo di item prototype (Zabbix agent, SNMP agent, simple check, ecc.).
Create enabled	Crea l'item basato su questo prototype come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Discover	Rileva l'item basato su questo prototype: **Yes** - rileva **No** - non rilevare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Tags	Vengono visualizzati i tag dell'item prototype.

Per configurare un nuovo item prototype, 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

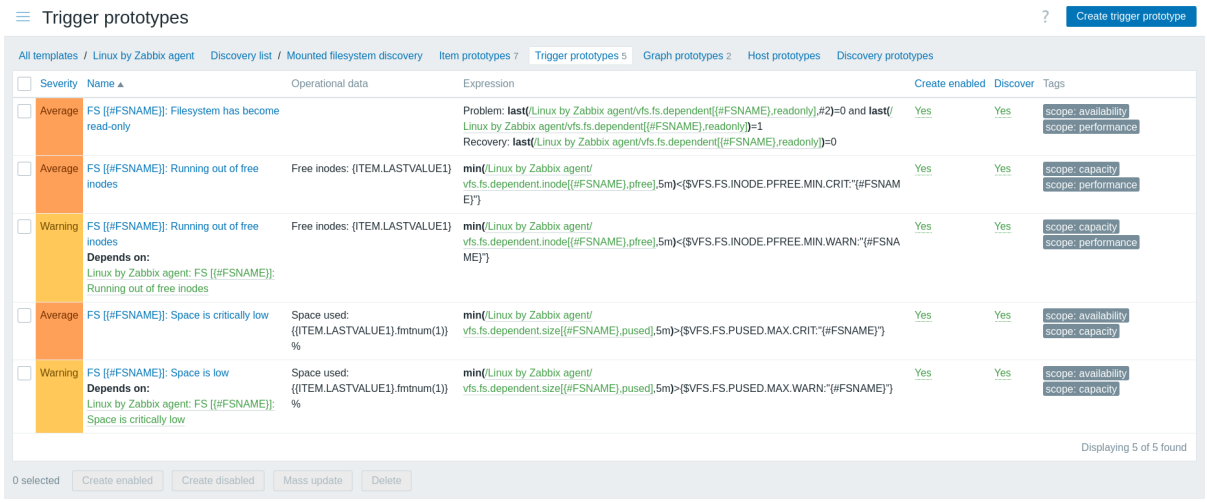
Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di item, quindi fai clic sul pulsante richiesto.

2 Prototipi di trigger

Panoramica

In questa sezione vengono visualizzati i trigger prototype configurati di una regola di low-level discovery sul template.

Se il template è collegato al host, i trigger prototype diventeranno la base per la creazione dei trigger reali del host durante la low-level discovery.



Dati visualizzati:

Column	Description
Severity	La severity del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.
Name	Nome del trigger prototype, visualizzato come link blu. Facendo clic sul nome si apre il modulo di configurazione del trigger prototype. Se il trigger prototype appartiene a un template collegato, il nome del template viene visualizzato prima del nome del trigger come link grigio. Facendo clic sul link del template si apre l'elenco dei trigger prototype al livello del template collegato.
Expression	Viene visualizzata l'espressione del trigger. La parte template-item dell'espressione viene visualizzata come link che porta al modulo di configurazione dell'item.
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 basato su questo prototype come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Discover	Rileva il trigger basato su questo prototype: **Yes** - rileva **No** - non rilevare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Tags	Vengono visualizzati i tag del trigger prototype.

Per configurare un nuovo trigger prototype, fare clic sul pulsante *Create trigger 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 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

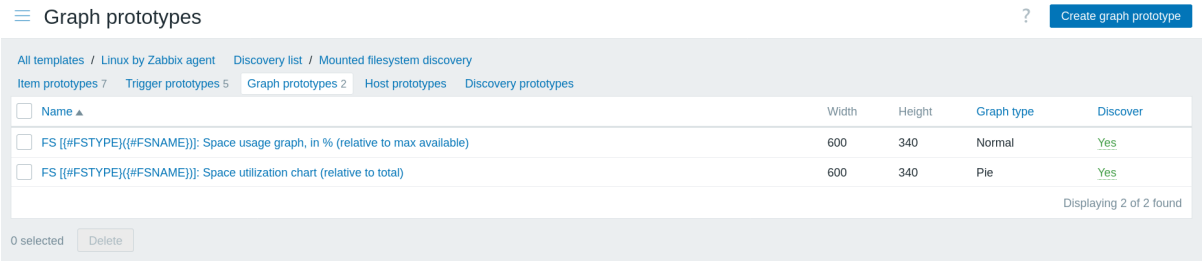
Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di trigger, quindi fai clic sul pulsante richiesto.

3 Prototipi di grafico

Panoramica

In questa sezione vengono visualizzati i prototipi di grafico configurati di una regola di low-level discovery nel template.

Se il template è collegato al host, i prototipi di grafico diventeranno la base per la creazione di veri grafici del host durante la low-level discovery.



Dati visualizzati:

Column	Description
Name	Viene visualizzato il nome del prototipo di grafico come link 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 link grigio. Facendo clic sul link 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 del prototipo di grafico - *Normale*, *In pila*, *Torta* oppure *Esploso*.
Discover	Consente di individuare il grafico in base a questo prototipo: **Yes** - individua **No** - non individuare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.

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:

- *Delete* - elimina questi prototipi di grafico

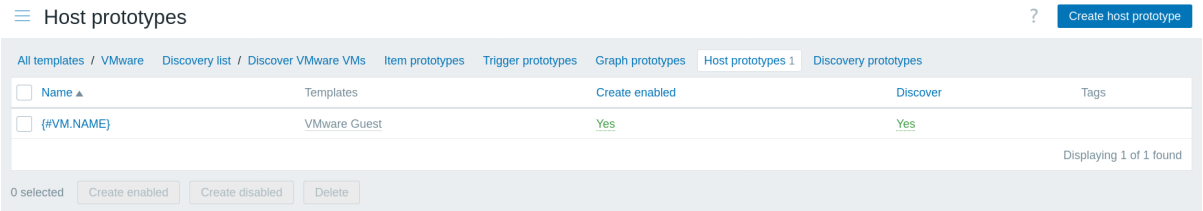
Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di grafico, quindi fai clic sul pulsante richiesto.

4 Prototipi degli host

Panoramica

In questa sezione vengono visualizzati i prototipi di host configurati di una regola di low-level discovery nel template.

Se il template è collegato all'host, i prototipi di host diventeranno la base per la creazione di host reali durante la low-level discovery.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di host, visualizzato come link 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 link grigio. Facendo clic sul link del template si apre l'elenco dei prototipi di host al livello del template collegato.
Templates	Vengono visualizzati i template del prototipo di host.
Create enabled	Crea l'host in base a questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.

Column	Description
Discover	Rileva l'host in base a questo prototipo: **Yes** - rileva **No** - non rilevare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Tags	Vengono visualizzati i tag del prototipo di host.

Per configurare un nuovo prototipo di host, fare clic sul pulsante *Create host prototype* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Crea abilitato* - crea questi host come *Abilitato*
- *Crea disabilitato* - crea questi host come *Disabilitato*
- *Elimina* - elimina questi prototipi di host

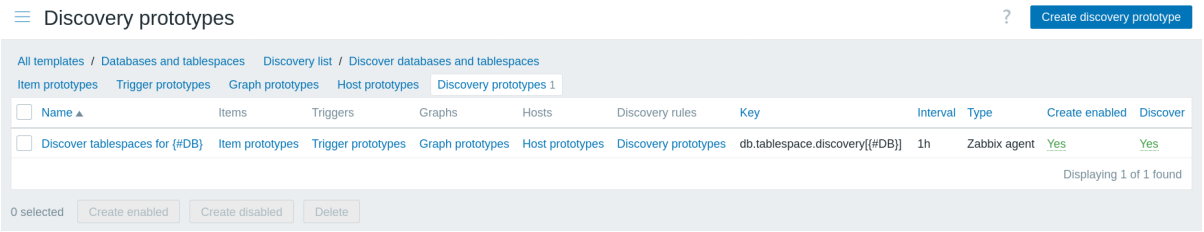
Per utilizzare queste opzioni, contrassegnare le caselle prima delle rispettive host prototipi, quindi fare clic sul pulsante richiesto.

5 Prototipi di discovery

Panoramica

In questa sezione vengono visualizzati i prototipi di discovery configurati di una regola di low-level discovery nel template.

Se il template è collegato al host, i prototipi di discovery diventeranno la base per la creazione dei reali prototipi di discovery della regola di discovery padre.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di discovery, visualizzato come link 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 link grigio. Facendo clic sul link del template si apre l'elenco dei prototipi di discovery a livello del template collegato.
Items	Viene visualizzato un link all'elenco dei prototipi di item. Il numero di prototipi di item esistenti è visualizzato in grigio.
Triggers	Viene visualizzato un link all'elenco dei prototipi di trigger. Il numero di prototipi di trigger esistenti è visualizzato in grigio.
Graphs	Viene visualizzato un link all'elenco dei prototipi di grafico. Il numero di prototipi di grafico esistenti è visualizzato in grigio.
Hosts	Viene visualizzato un link all'elenco dei prototipi di host. Il numero di prototipi di host esistenti è visualizzato in grigio.
Discovery rules	Viene visualizzato un link all'elenco dei prototipi di discovery. Il numero di prototipi di discovery esistenti è visualizzato in grigio.
Key	Viene visualizzata la key 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: discover (*Yes*) o non discover (*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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di 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

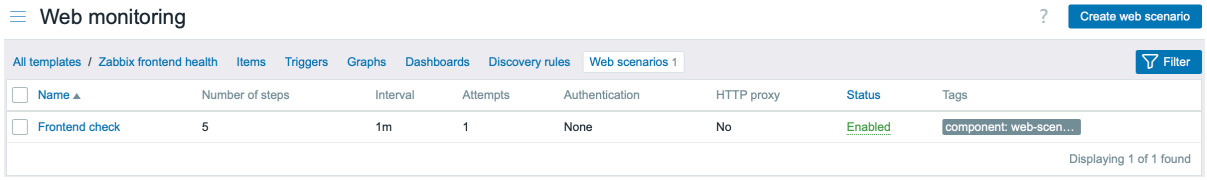
Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di discovery, quindi fai clic sul pulsante richiesto.

5 Scenari web

Panoramica

L'elenco degli scenari web per un template è accessibile da *Raccolta dati* → *Template* facendo clic su *Web* per il rispettivo template.

Viene visualizzato un elenco degli scenari web esistenti.



Dati visualizzati:

Column	Description
Name	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 di template.
Number of steps	Il numero di passaggi contenuti nello scenario.
Update interval	La frequenza con cui viene eseguito lo scenario.
Attempts	Quanti tentativi vengono effettuati per eseguire i passaggi dello scenario web.
Authentication	Viene visualizzato il metodo di autenticazione: Basic, NTLM, Kerberos, Digest oppure None.
HTTP proxy	Visualizza l'HTTP proxy oppure 'No' se non viene utilizzato.
Status	Viene visualizzato lo stato dello scenario web: *Enabled* oppure *Disabled*. Facendo clic sullo stato è possibile modificarlo.
Tags	Vengono visualizzati i tag dello scenario web. Possono essere visualizzati fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di vedere tutti i tag al passaggio del mouse.

Per configurare un nuovo scenario web, fare clic sul pulsante *Create web scenario* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

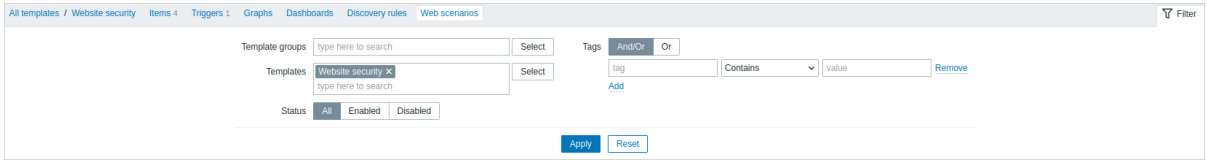
- *Abilita* - cambia lo stato dello scenario in *Abilitato*
- *Disabilita* - modifica lo stato dello scenario in *Disabilitato*
- *Cancella* - cancella gli scenari web

Per utilizzare queste opzioni, contrassegnare le caselle prima del rispettivo file web scenari, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo gli scenari di interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il collegamento *Filtro* è disponibile sopra l'elenco degli scenari web. Facendo clic su di esso, viene reso disponibile un filtro con cui è possibile filtrare gli scenari per gruppo di template, template, stato e tag.

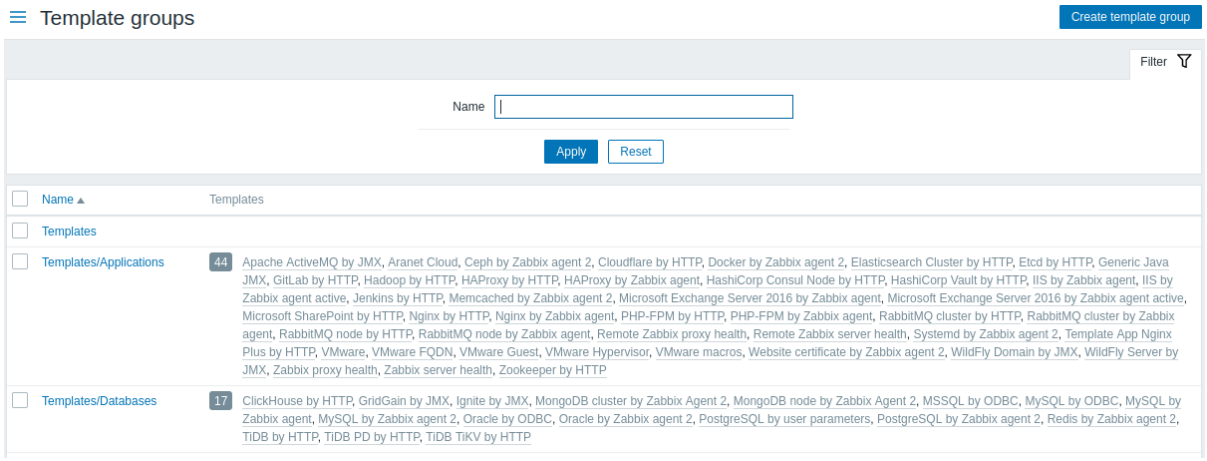


1 Gruppi di template

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 per 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.
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 in questo gruppo.

Opzioni di modifica di massa

Per eliminare contemporaneamente più gruppi di template, seleziona le caselle di controllo davanti ai rispettivi gruppi, quindi fai clic sul pulsante Elimina sotto l'elenco.

Utilizzo del filtro

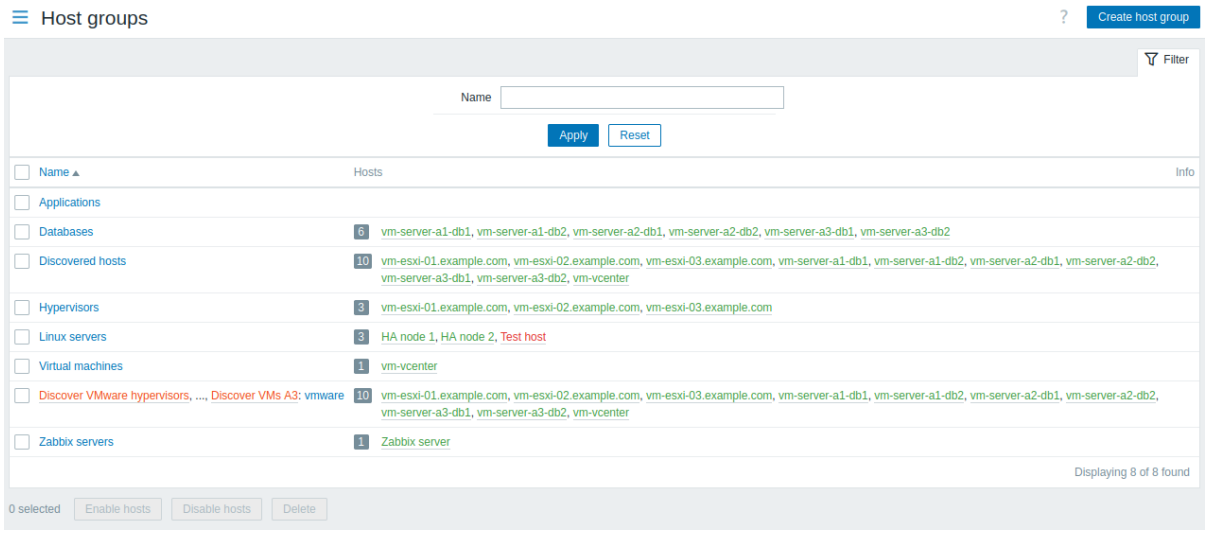
È possibile utilizzare il filtro per visualizzare solo i gruppi di template di proprio interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

2 Gruppi di host

Panoramica

Nella sezione *Raccolta dati* → *Gruppi host* gli utenti possono configurare e gestire i gruppi host.

Viene visualizzato un elenco dei gruppi host esistenti con i relativi dettagli. È possibile cercare e filtrare i gruppi host per nome.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del gruppo host. Facendo clic sul nome del gruppo si apre il modulo di configurazione del gruppo. I gruppi host rilevati vengono visualizzati con i nomi delle regole di low-level discovery come prefisso. Facendo clic sul nome della regola LLD si apre il modulo di configurazione del prototipo host. Si noti che i gruppi 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'intero elenco degli host verranno filtrati quelli che appartengono al gruppo.
Info	Vengono visualizzate le informazioni sugli errori (se presenti) relative al gruppo host.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Abilita host* - modifica lo stato di tutti gli host nel gruppo in "Monitorato"
- *Disabilita host* - cambia lo stato di tutti gli host nel gruppo in "Non monitorato"
- *Elimina* - elimina i gruppi host

Per utilizzare queste opzioni, contrassegnare le caselle di controllo prima del rispettivo host gruppi, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

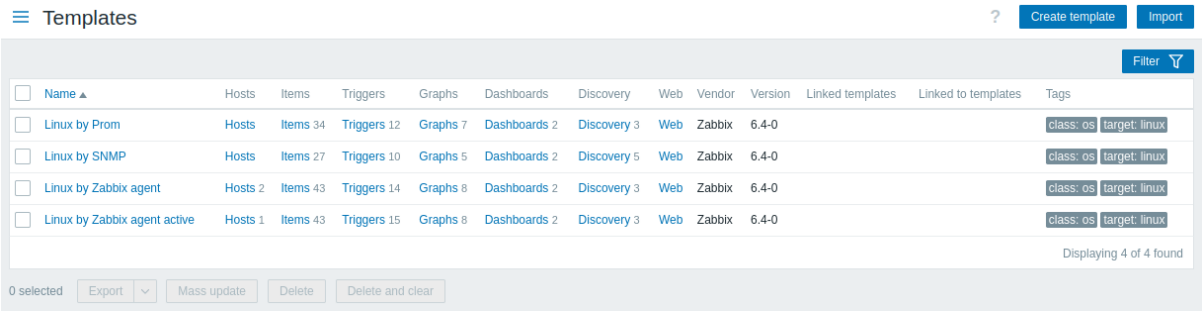
Puoi utilizzare il filtro per visualizzare solo i gruppi host che sei interessati. Per migliorare le prestazioni della ricerca, i dati vengono cercati con macro irrisolte.

3 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.



Dati visualizzati:

Column	Description
Name	Nome del template. Facendo clic sul nome del template si apre il modulo di configurazione del template.
Hosts	Numero di host modificabili a cui il template è collegato; gli host in sola lettura non sono inclusi. Facendo clic su *Hosts* si apre l'elenco degli host con il filtro applicato solo agli host collegati al template.
Entities (Items, Triggers, Graphs, Dashboards, Discovery, Web)	Numero delle rispettive entità nel template (visualizzato in grigio). Facendo clic sul nome dell'entità, nell'elenco completo di tale entità verranno filtrate quelle che appartengono al template.
Linked templates	Template collegati al template.
Linked to templates	Template a cui il template è collegato.
Vendor, *Version*	Fornitore e versione del template; visualizzati se la configurazione del template contiene tali informazioni, e solo per i template predefiniti, i template importati o i template modificati tramite la Template API. 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 nuova versione principale di Zabbix). Ad esempio, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Tags	Tag del template, con le macro non risolte.

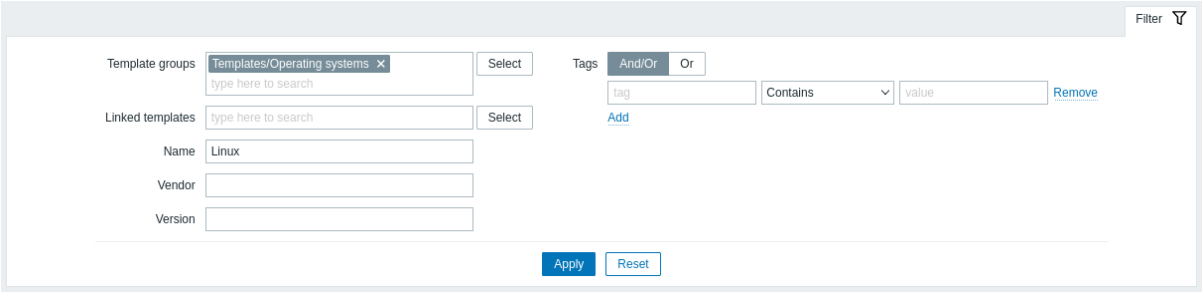
Per configurare un nuovo template, fare clic sul pulsante *Create template* nell'angolo superiore destro.

Per importare un template da un file YAML, XML o JSON, fare clic sul pulsante *Import* nell'angolo superiore destro.

Uso del filtro

È possibile utilizzare 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 collegamento *Filtro* è disponibile sotto i pulsanti *Crea template* e *Importa*. Facendo clic su di esso, diventa disponibile un filtro con cui è possibile filtrare i template per gruppo di template, template collegati, nome e tag.



Parametro	Descrizione
Gruppi di template	Filtra per uno o più gruppi di template. Specificando un gruppo di template padre vengono selezionati implicitamente anche tutti i gruppi nidificati.
Template collegati	Filtra per template collegati direttamente.
Nome	Filtra per nome del template.
Vendor	Filtra per vendor del template.
Versione	Filtra per versione del template.

Parametro	Descrizione
Tag	Filtra per nome e valore del tag del template. Il filtraggio è possibile solo per i tag a livello di template (non quelli ereditati). 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). 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 sola condizione.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *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 sui host le entità collegate (item, trigger ecc.);
- *Delete and clear* - elimina il template e le entità collegate dai host.

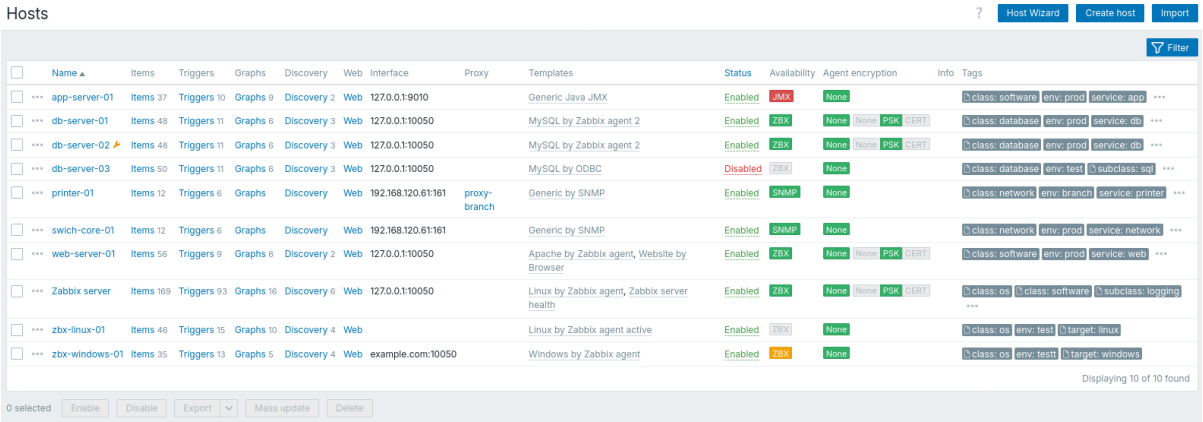
Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi template, quindi fare clic sul pulsante richiesto.

4 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.



Dati visualizzati:

Colonna	Descrizione
Menu host	Fare clic sull'icona con i tre puntini per aprire il menu host.

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 questo host è in manutenzione. I dettagli della manutenzione vengono visualizzati quando il puntatore del mouse viene posizionato sull'icona.
Entità (Item, Trigger, Grafici, Discovery, Web)	Facendo clic sul nome dell'entità verranno visualizzati item, trigger, ecc. dell'host. Il numero delle rispettive entità è visualizzato in grigio.
Interfaccia	Viene visualizzata l'interfaccia principale dell'host.
Proxy	I proxy assegnati vengono visualizzati in questa colonna: <Nome proxy> - l'host è monitorato da un proxy standalone (anche se il proxy fa parte di un gruppo di proxy); <Nome gruppo proxy: nome proxy> - l'host è monitorato da un gruppo di proxy e Zabbix server ha assegnato un proxy per monitorare l'host; <Nome gruppo proxy> - l'host è monitorato da un gruppo di proxy senza alcun proxy, oppure se Zabbix server 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 *Monitored by* è impostata su 'Any', 'Proxy' o 'Proxy group'.
Template	Vengono visualizzati i template collegati all'host. Se il template collegato contiene altri template, questi vengono visualizzati tra parentesi, separati da una virgola. Facendo clic sul nome di un template si aprirà 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 stati persi sono contrassegnati con un'icona informativa. Il testo del suggerimento fornisce dettagli sul loro stato.

Colonna	Descrizione
Disponibilità	Viene visualizzata la disponibilità dell'host per ogni interfaccia configurata. Le icone di disponibilità rappresentano lo stato corrente dell'interfaccia 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. Analogamente, se si abilita un host, la sua disponibilità verrà aggiornata dopo che Zabbix server avrà sincronizzato le modifiche di configurazione ed effettuato il polling dell'host. Le icone di disponibilità rappresentano solo i tipi di interfaccia (Agent, SNMP, IPMI, JMX) che sono configurati. Passando il puntatore 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 visualizza le interfacce (passive) e i controlli attivi. Se un host ha solo controlli attivi, l'icona dell'interfaccia Agent viene visualizzata anche se l'host non ha un'interfaccia Agent configurata. La colonna è vuota per gli host senza interfacce. Lo stato di una singola interfaccia 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 di firewall, ecc.); **Sconosciuto** - la connessione all'host non è stata tentata oppure il risultato è sconosciuto. Per ulteriori dettagli su come Zabbix server determina lo stato dell'interfaccia, vedere Stato dell'interfaccia sconosciuto e Impostazioni delle interfacce host irraggiungibili/non disponibili. Lo stato di tutte le interfacce 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: **Verde** - tutte le interfacce sono disponibili; **Giallo** - almeno un'interfaccia non è disponibile e almeno una è disponibile o sconosciuta; **Rosso** - tutte le interfacce non sono disponibili; **Grigio** - 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 totale 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 loro host rimane sconosciuta.
Crittografia agent	Viene visualizzato lo stato della crittografia per le connessioni verso l'host e dall'host: **Nessuna** - nessuna crittografia; **PSK** - utilizzo di chiave precondivisa; **Cert** - utilizzo di certificato.
Info	Vengono visualizzate le informazioni sugli errori (se presenti) relative all'host.
Tag	I tag dell'host con macro non risolte.

Per configurare un nuovo host, fare clic sul pulsante *Host Wizard* o *Create host* nell'angolo superiore destro. Per importare un host da un file YAML, XML o JSON, fare clic sul pulsante *Import* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

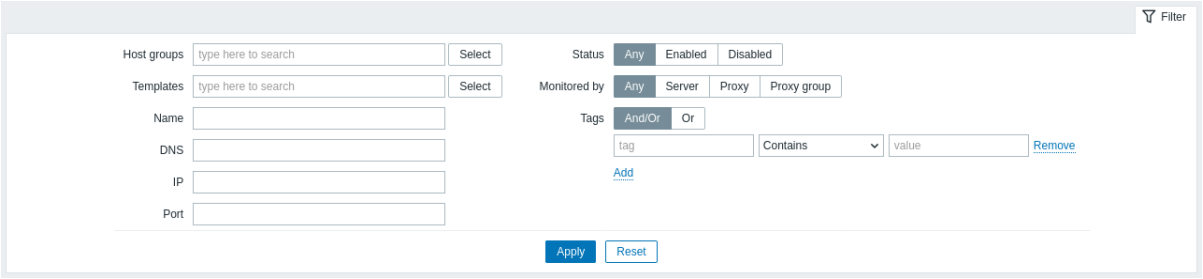
- *Abilita* - cambia lo stato del host in *Monitorato*;
- *Disabilita* - cambia lo stato del host in *Non monitorato*;
- *Esporta* - esporta i host in un file YAML, XML o JSON;
- *Aggiornamento di massa* - aggiorna diverse proprietà per più host contemporaneamente;
- *Elimina* - elimina i host.

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi host, quindi fai clic sul pulsante richiesto.

Uso del filtro

È possibile utilizzare il filtro per visualizzare solo gli host di interesse. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filtro* è disponibile nell'angolo superiore destro. Facendo clic su di essa si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



Parametro	Descrizione
Gruppi di host	Filtra per uno o più gruppi di host. Specificando un gruppo di host padre vengono selezionati implicitamente tutti i gruppi di host nidificati.
Template	Filtra per template collegati.
Nome	Filtra per nome host visibile.
DNS	Filtra per nome DNS.
IP	Filtra per indirizzo IP.
Porta	Filtra per numero di porta.
Stato	Filtra per stato dell'host.
Monitorato da	Filtra gli host monitorati da Zabbix server, proxy o gruppo di proxy.
Proxy	Filtra gli host monitorati dai proxy specificati qui. Questo campo è disponibile solo se nel campo *Monitorato da* è selezionato "Proxy".
Gruppi di proxy	Filtra gli host monitorati dai gruppi di proxy specificati qui. Questo campo è disponibile solo se nel campo *Monitorato da* è selezionato "Gruppo di proxy".
Tag	Filtra per nome e valore del tag dell'host. 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). 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 sola condizione.

Stato sconosciuto dell'interfaccia

Zabbix server visualizza lo stato "Sconosciuto" per un'interfaccia del host (Agent, SNMP, IPMI, JMX) nei seguenti casi:

- Il host è disabilitato.
- Il host è impostato per essere monitorato da un proxy, da un proxy diverso oppure dal server se in precedenza era monitorato da un proxy.
- Il host è monitorato da un proxy che risulta offline (nessun aggiornamento ricevuto dal proxy durante l'intervallo massimo di heartbeat - 1 ora).
- Tutti gli item del 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 "Sconosciuto" dopo la sincronizzazione della cache di configurazione di Zabbix server.

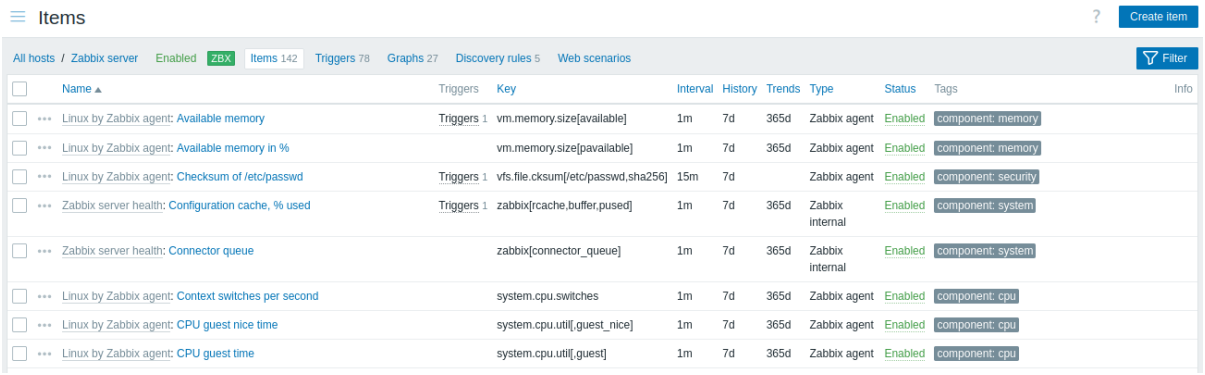
La disponibilità dell'interfaccia (Disponibile/Non disponibile) sui host monitorati dai proxy viene ripristinata dopo la sincronizzazione della cache di configurazione del proxy.

1 Item

Panoramica

L'elenco degli item per un host è accessibile da *Raccolta dati* → *Host* facendo clic su *Item* per il rispettivo host.

Viene visualizzato un elenco degli item esistenti.



Dati visualizzati:

Colonna	Descrizione
Menu item	Fare clic sull'icona con i 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 oppure 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 aprirà 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 low-level discovery, in arancione. Facendo clic sul nome della regola di discovery si aprirà l'elenco dei prototipi di item.
Trigger	Passando il mouse su Trigger verrà visualizzato un riquadro informativo con i trigger associati all'item. Il numero dei trigger viene visualizzato in grigio.
Chiave	Viene visualizzata la chiave dell'item.
Intervallo	Viene visualizzata la frequenza del controllo. *Si noti* che anche gli item passivi possono essere controllati immediatamente premendo il pulsante *Esegui ora*.
Storico	Viene visualizzato per quanti giorni sarà conservato lo storico dei dati dell'item.
Trend	Viene visualizzato per quanti giorni sarà conservato lo storico dei trend dell'item.
Tipo	Viene visualizzato il tipo di item (Zabbix agent, SNMP agent, simple check, ecc.).
Stato	Viene visualizzato lo stato dell'item: *Abilitato*, *Disabilitato* oppure *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 sono contrassegnati con un'icona informativa. Il testo del tooltip fornisce dettagli sul loro stato.
Tag	Vengono visualizzati i tag dell'item. Possono essere visualizzati fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di vedere tutti i tag al passaggio del mouse.

Colonna	Descrizione
Info	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 tooltip con la descrizione dell'errore.

Per configurare un nuovo item, fare clic sul pulsante *Crea item* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

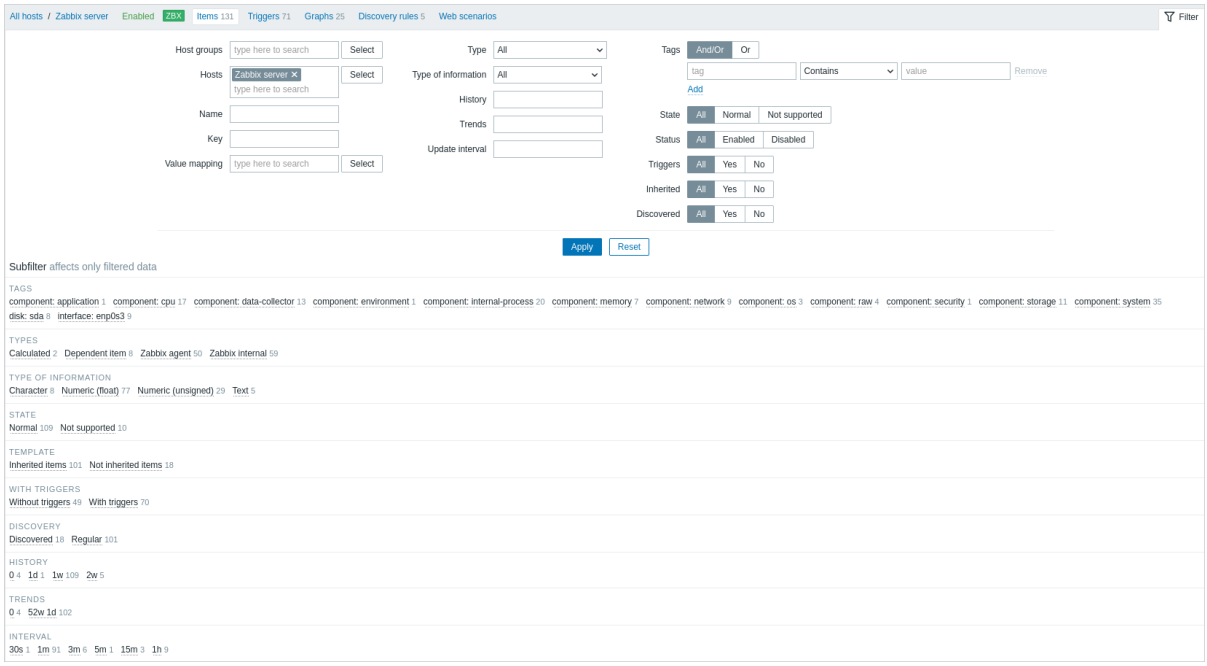
- *Abilita* - cambia lo stato dell'item in *Abilitato*
- *Disabilita* - cambia lo stato dell'item in *Disabilitato*
- *Esegui ora* - esegue immediatamente un controllo per ottenere nuovi valori dell'item. Supportato solo per i controlli **passivi** (vedi maggiori dettagli). Si noti che, quando si controllano immediatamente i valori, la cache di configurazione non viene aggiornata; pertanto, i valori non rifletteranno modifiche molto recenti alla configurazione dell'item.
- *Cancella cronologia e trend* - elimina i dati di cronologia e trend degli item.
- *Copia* - copia gli item in altri host o template.
- *Aggiornamento di massa* - aggiorna diverse proprietà per più item contemporaneamente.
- *Elimina* - elimina gli item.

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi item, quindi fare clic sul pulsante richiesto.

Uso 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 *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.



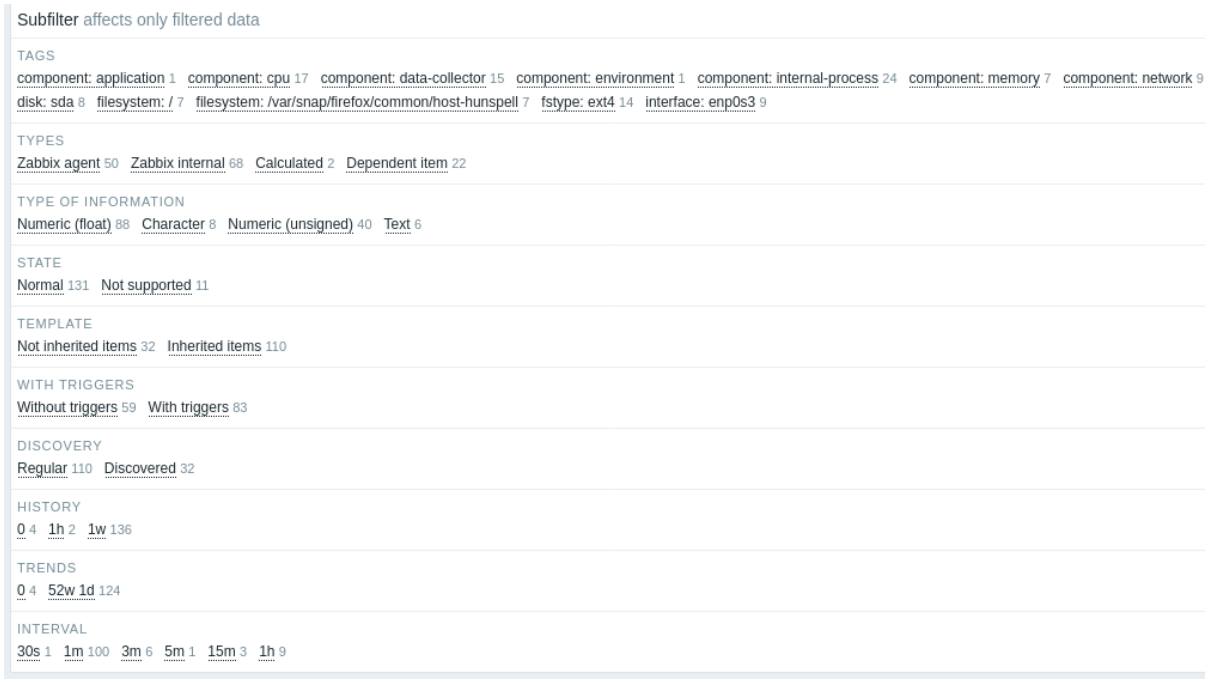
Parametro	Descrizione
Gruppi host	Filtra per uno o più gruppi host. Specificando un gruppo host padre vengono implicitamente selezionati tutti i gruppi host nidificati. Non è possibile selezionare gruppi host che contengono solo template.
Host	Filtra per uno o più host.
Nome	Filtra per nome dell'item.
Chiave	Filtra per chiave dell'item.
Mappatura valori	Filtra per la mappa valori utilizzata. Questo parametro non viene visualizzato se l'opzione *Host* è vuota.
Tipo	Filtra per tipo di item (Zabbix agent, SNMP agent, ecc.).
Tipo di informazione	Filtra per tipo di informazione (Numerico senza segno, virgola mobile, ecc.).

Parametro	Descrizione
Storico	Filtra in base al tempo di conservazione dello storico dell'item.
Trend	Filtra in base al tempo di conservazione dei trend dell'item.
Intervallo di aggiornamento	Filtra per intervallo di aggiornamento dell'item.
Tag	Specifica i tag per limitare il numero di item visualizzati. 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: **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 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 in cui i valori del tag contengono la stringa immessa (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
Stato	Filtra per stato dell'item - *Normale* o *Non supportato*.
Status	Filtra per stato dell'item - *Abilitato* o *Disabilitato*.
Trigger	Filtra gli item con trigger (o senza trigger).
Ereditato	Filtra gli item ereditati (o non ereditati) da un template.
Discovery	Filtra gli item rilevati (o non rilevati) dal low-level discovery.

Utilizzo del sottofiltro

Il sottofiltro consente di modificare ulteriormente il filtraggio del filtro principale.

Contiene collegamenti cliccabili per un accesso rapido agli item correlati. Gli item sono correlati da un'entità comune: tag, tipo di item, stato dell'item, intervallo di aggiornamento dell'item, ecc. 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 *Apply* 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 sottofiltri vengono generati in base ai dati filtrati, che sono limitati a 1000 record. Se si desidera visualizzare più record nel sottofiltro, è necessario aumentare il valore del parametro *Limit for search and filter results* (in *Administration -> General -> GUI*).

A differenza del filtro principale, il sottofiltro viene aggiornato a ogni richiesta di aggiornamento della tabella, in modo da avere sempre informazioni aggiornate sulle opzioni di filtro disponibili e sui relativi contatori.

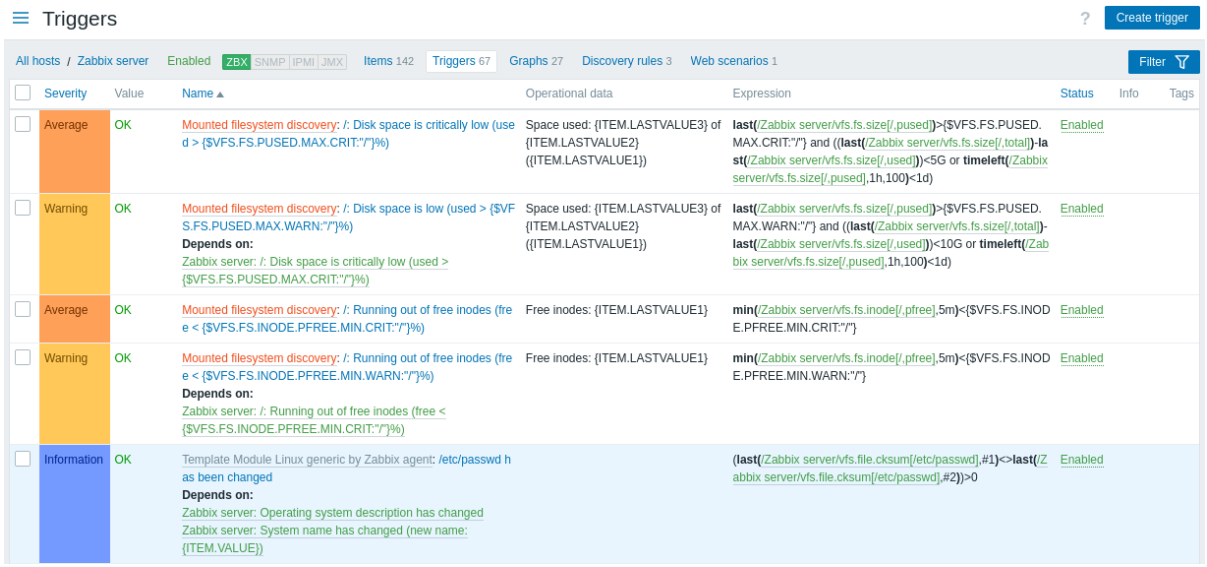
Il numero di entità visualizzate è limitato orizzontalmente a 100. Se ce ne sono di più, alla fine viene visualizzata un'icona con tre puntini; non è cliccabile.

Un numero accanto a ciascuna entità cliccabile indica il numero di item raggruppati in essa (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 state selezionate in precedenza nel sottofiltro.

2 Trigger

Panoramica

L'elenco dei trigger per un host è accessibile da *Raccolta dati* → *Host* facendo clic su *Trigger* per il rispettivo host.



Dati visualizzati:

Colonna	Descrizione
Gravità	La gravità del trigger viene visualizzata sia tramite il nome sia tramite il colore di sfondo della cella.
Valore	Lo stato del trigger viene visualizzato come segue: **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 oppure nessun host.
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 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 aprirà 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 low-level discovery, 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 porta al modulo di configurazione dell'item.
Stato	Viene visualizzato lo stato del trigger: *Abilitato*, *Disabilitato* oppure *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 sono contrassegnati con un'icona informativa. Il testo del suggerimento fornisce dettagli sul loro stato.

Colonna	Descrizione
Info	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 cursore sull'icona per visualizzare un suggerimento con la descrizione dell'errore.
Tag	Se un trigger contiene tag, il nome e il valore del tag vengono visualizzati in questa colonna.

Per configurare un nuovo trigger, fare clic sul pulsante *Crea trigger* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

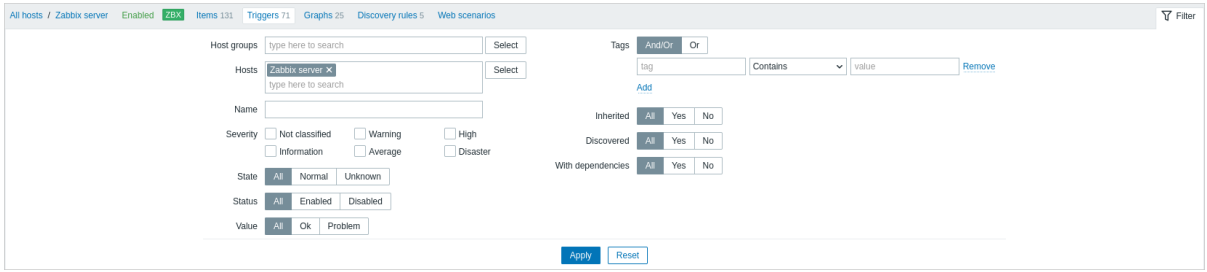
- *Abilita* - cambia lo stato del trigger in *Abilitato*.
- *Disabilita* - cambia lo stato del trigger in *Disabilitato*.
- *Copia* - copia i trigger in altri host o template.
- *Aggiornamento di massa* - aggiorna contemporaneamente diverse proprietà per un certo numero di trigger.
- *Elimina* - elimina i trigger.

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi trigger, quindi fai clic sul pulsante richiesto.

Uso del filtro

È possibile utilizzare il filtro per visualizzare solo i trigger di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

L'icona *Filtro* è disponibile nell'angolo superiore destro. Facendo clic su di essa si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



Parametro	Descrizione
Gruppi di host	Filtra per uno o più gruppi di host. Specificando un gruppo di host padre vengono implicitamente selezionati anche tutti i gruppi di host nidificati. Non è possibile selezionare gruppi di host che contengono solo template.
Host	Filtra per uno o più host. Se sopra sono già stati selezionati gruppi di host, la selezione degli host è limitata a tali gruppi.
Nome	Filtra per nome del trigger.
Gravità	Selezionare per filtrare in base a una o più gravità del trigger.
Stato	Filtra per stato dell'espressione del trigger (Normale/Sconosciuto).
Status	Filtra per stato di attivazione del trigger (Abilitato/Disabilitato).
Valore	Filtra per valore del trigger.

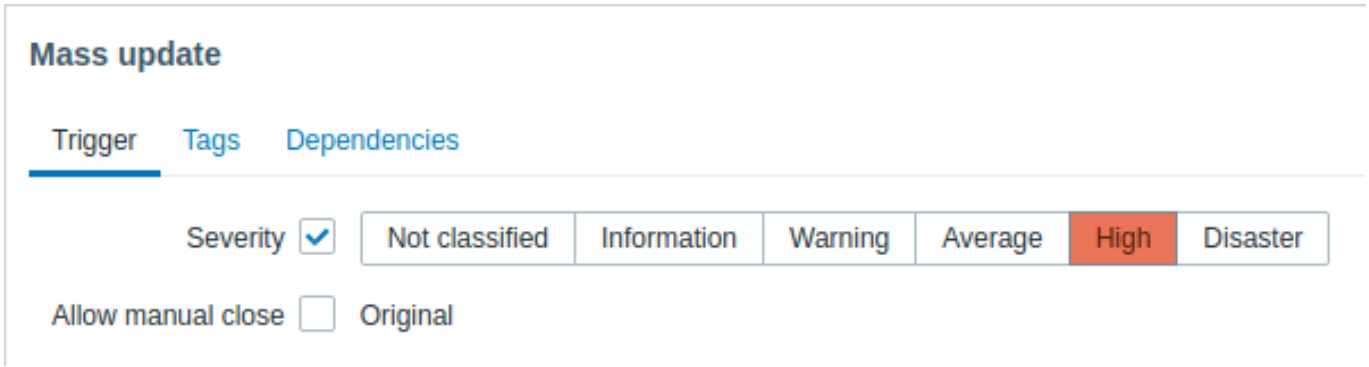
Parametro	Descrizione
Tag	Filtra per nome e valore del tag del trigger. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag distingue sempre tra maiuscole e minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Esiste** - include i nomi di tag specificati **Uguale a** - include i nomi di tag e i valori specificati (distinzione tra maiuscole e minuscole) **Contiene** - include i nomi di tag per i quali 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 (distinzione tra maiuscole e minuscole) **Non contiene** - esclude i nomi di tag per i quali i valori del tag contengono la stringa immessa (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 Sono supportate sia le macro sia le funzioni macro nei campi nome del tag e valore del tag.
Ereditato	Filtra i trigger ereditati (o non ereditati) da un template.
Rilevato	Filtra i trigger rilevati (o non rilevati) dal low-level discovery.
Con dipendenze	Filtra i trigger con (o senza) dipendenze.

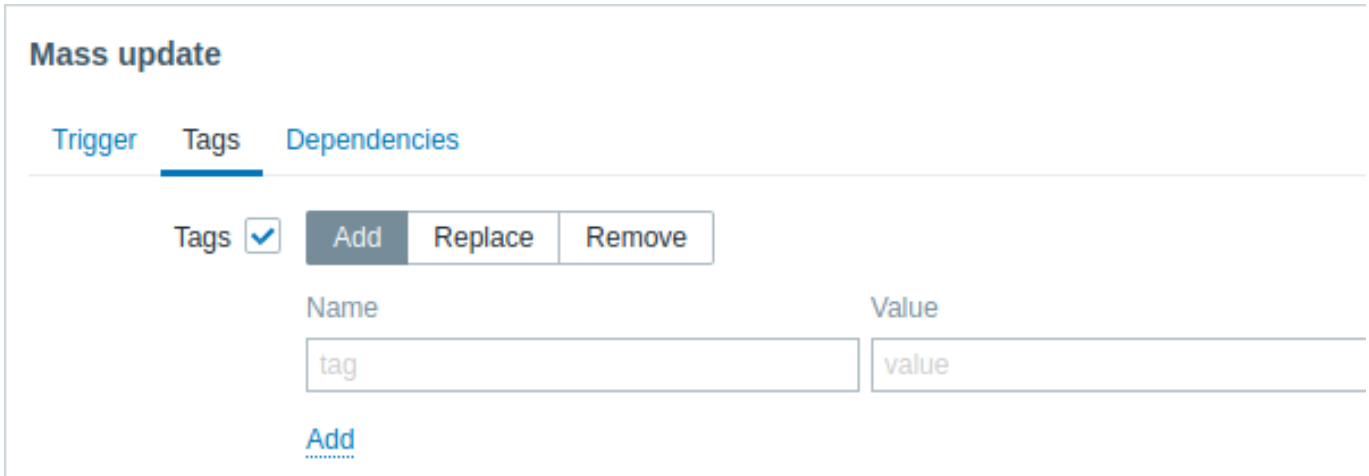
Utilizzo dell'aggiornamento di massa

Con l'aggiornamento di massa è possibile modificare alcuni attributi per più trigger contemporaneamente, evitando così di dover aprire ogni singolo trigger per la modifica.

Per aggiornare in massa alcuni trigger, procedere come segue:

- Selezionare le caselle di controllo dei trigger che si desidera aggiornare nell'elenco
- Fare clic su *Mass update* sotto l'elenco
- Passare alla scheda con gli attributi richiesti (*Trigger*, *Tags* o *Dependencies*)
- Selezionare le caselle di controllo di qualsiasi attributo da aggiornare



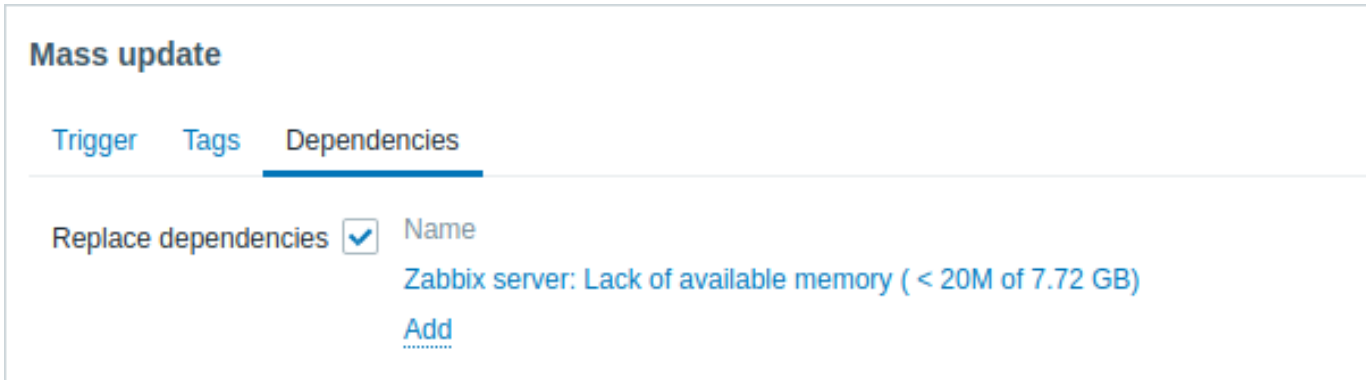


Sono disponibili le seguenti opzioni quando si seleziona il rispettivo pulsante per l'aggiornamento dei tag:

- *Add* - consente di aggiungere nuovi tag ai trigger;

- *Replace* - rimuoverà tutti i tag esistenti dal trigger e li sostituirà con quello/quelli specificati di seguito;
- *Remove* - rimuoverà i tag specificati dai trigger.

Si noti che i tag con lo stesso nome ma valori diversi non sono considerati "duplicati" e possono essere aggiunti allo stesso trigger.



Replace dependencies - rimuoverà tutte le dipendenze esistenti dal trigger e le sostituirà con quella/e specificata/e.

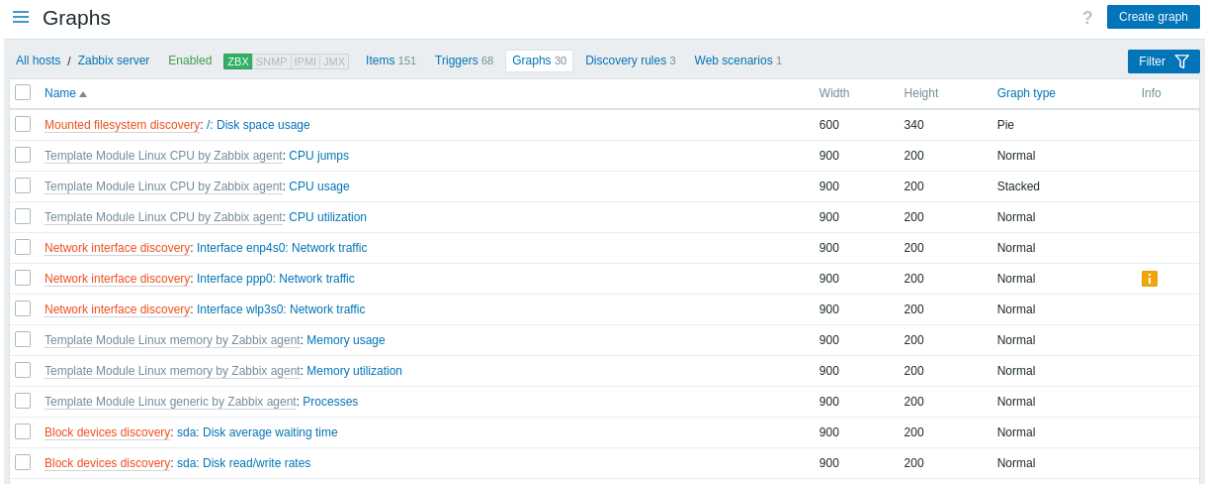
Fare clic su *Update* per applicare le modifiche.

3 Grafici

Panoramica

L'elenco dei grafici personalizzati per un host è accessibile da *Raccolta dati* → *Host* facendo clic su *Grafici* per il rispettivo host.

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 del 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 aprirà 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*, *Torta* o *Esploso*.

Colonna	Descrizione
Info	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 puntatore sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare un nuovo grafico, fare clic sul pulsante *Crea grafico* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Copia* - copia i grafici su altri host o modelli
- *Cancella* - cancella i grafici

Per utilizzare queste opzioni, contrassegnare le caselle prima dei rispettivi grafici, quindi fare clic sul pulsante richiesto.

Uso del filtro

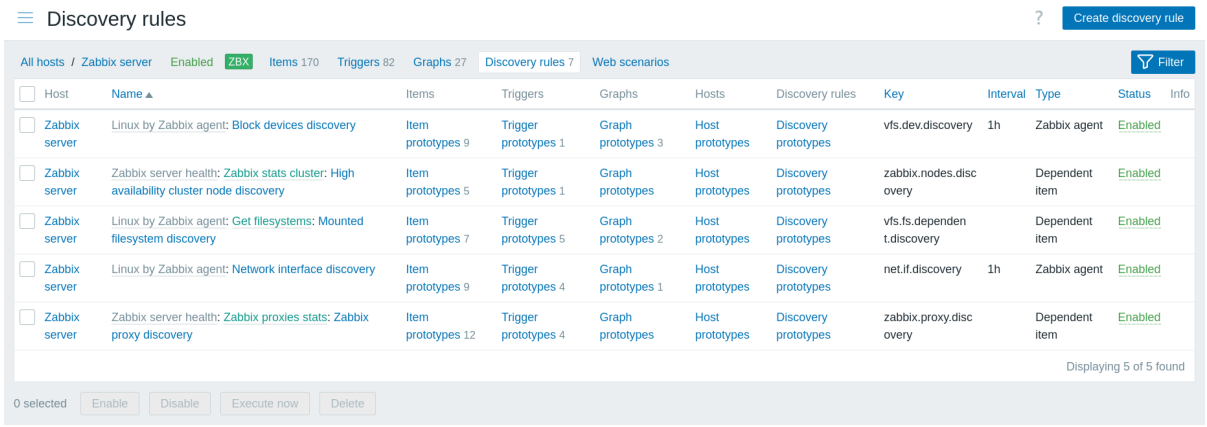
È possibile filtrare i grafici per gruppo host e host. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

4 Regole di Discovery

Panoramica

L'elenco delle regole di low-level discovery per un host è accessibile da *Raccolta dati* → *Host* facendo clic su *Discovery* per il rispettivo host.

Viene visualizzato un elenco delle regole di low-level discovery 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:

Column	Description
Host	Viene visualizzato il nome host visibile. Facendo clic sul nome host si apre il modulo di configurazione dell'host. In assenza di un nome host visibile, viene visualizzato il nome host tecnico.
Name	Nome della regola, visualizzato come collegamento blu. Facendo clic sul nome della regola si apre il modulo di configurazione della regola di low-level discovery. 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.
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 grafico. Il numero di prototipi di grafico esistenti viene visualizzato in grigio.

Column	Description
Hosts	Viene visualizzato un collegamento all'elenco dei prototipi di host. Il numero di prototipi di host esistenti viene visualizzato in grigio.
Key	Viene visualizzata la chiave item utilizzata per la discovery.
Interval	Viene visualizzata la frequenza di esecuzione della discovery. Si noti che la discovery può anche essere eseguita immediatamente premendo il pulsante *Execute now* sotto l'elenco.
Type	Viene visualizzato il tipo di item utilizzato per la discovery (Zabbix agent, SNMP agent, ecc.).
Status	Viene visualizzato lo stato della regola di discovery: *Enabled*, *Disabled* oppure *Not supported*. Facendo clic sullo stato è possibile modificarlo: da Enabled a Disabled (e viceversa); da Not supported a Disabled (e viceversa).
Info	Se tutto è corretto, in questa colonna non viene visualizzata alcuna icona. In caso di errori, viene visualizzata un'icona quadrata con la lettera "i". Passare il puntatore sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare una nuova regola di low-level discovery, fare clic sul pulsante *Create discovery rule* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

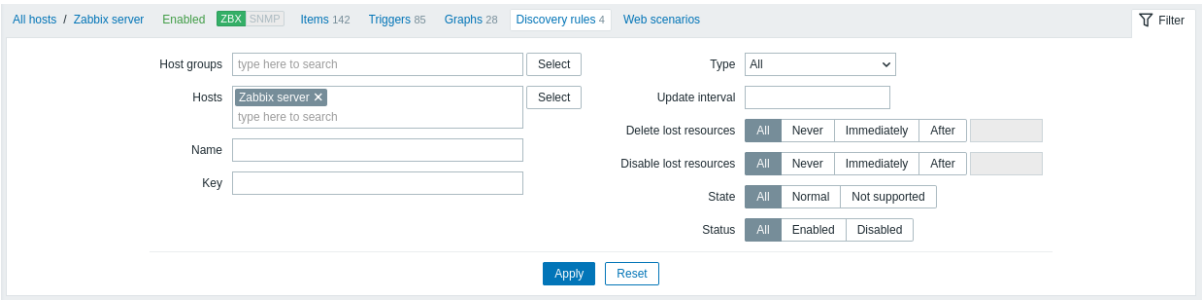
- *Abilita* - cambia lo stato della regola di low-level discovery in *Abilitato*.
- *Disabilita* - cambia lo stato della regola di low-level discovery in *Disabilitato*.
- *Esegui ora* - esegue immediatamente la discovery in base alle regole di discovery. Vedi maggiori dettagli. Nota che, quando si esegue la discovery immediatamente, la cache di configurazione non viene aggiornata, pertanto il risultato non rifletterà modifiche molto recenti alla configurazione della regola di discovery.
- *Elimina* - elimina le regole di low-level discovery.

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti alle rispettive regole di discovery, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo le regole di discovery che ti interessano. Per migliori prestazioni di ricerca, i dati vengono cercati con le macro non risolte.

Il link *Filtro* 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 di host, host, nome, chiave item, tipo di item e altri parametri.



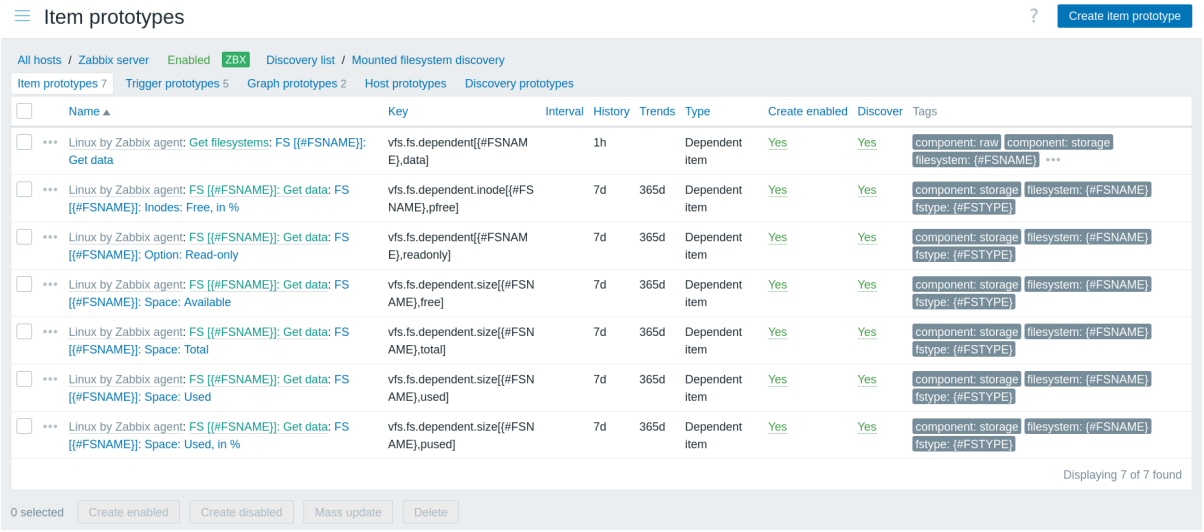
Parametro	Descrizione
Gruppi di host	Filtra per uno o più gruppi di host. Specificando un gruppo di host padre vengono implicitamente selezionati tutti i gruppi di host nidificati.
Host	Filtra per uno o più host.
Nome	Filtra per nome della regola di discovery.
Chiave	Filtra per chiave dell'item di discovery.
Tipo	Filtra per tipo di item di discovery.
Intervallo di aggiornamento	Filtra per intervallo di aggiornamento. Non disponibile per gli item Zabbix trapper e gli item dipendenti.
Elimina risorse perse	Filtra per periodo di *Elimina risorse perse*.
Disabilita risorse perse	Filtra per periodo di *Disabilita risorse perse*.
SNMP OID	Filtra per SNMP OID. Disponibile solo se *SNMP agent* è selezionato come tipo.
Stato	Filtra per stato della regola di discovery (Tutti/Normale/Non supportato).

Parametro	Descrizione
Status	Filtra per status della regola di discovery (Tutti/Abilitato/Disabilitato).

1 Prototipi di item

Panoramica

In questa sezione vengono visualizzati i prototipi di item di una regola di low-level discovery sul host.
I prototipi di item sono la base dei veri items del host che vengono creati durante la low-level discovery.



Dati visualizzati:

Column	Description
Name	Nome del prototipo di item, visualizzato come link 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 link grigio. Facendo clic sul link del template si apre l'elenco dei prototipi di item a livello di template. Se il prototipo di item è del tipo "Dependent item", il nome del master item viene visualizzato in verde prima del nome del prototipo di item.
Key	Viene visualizzata la chiave del prototipo di item.
Interval	Viene visualizzata la frequenza del controllo.
History	Viene visualizzato per quanti giorni mantenere lo storico dei dati dell'item.
Trends	Viene visualizzato per quanti giorni mantenere lo storico dei trend dell'item.
Type	Viene visualizzato il tipo del prototipo di item (Zabbix agent, SNMP agent, simple check, ecc.).
Create enabled	Crea l'item basato su questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Discover	Rileva l'item basato su questo prototipo: **Yes** - rileva **No** - non rilevare. 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:

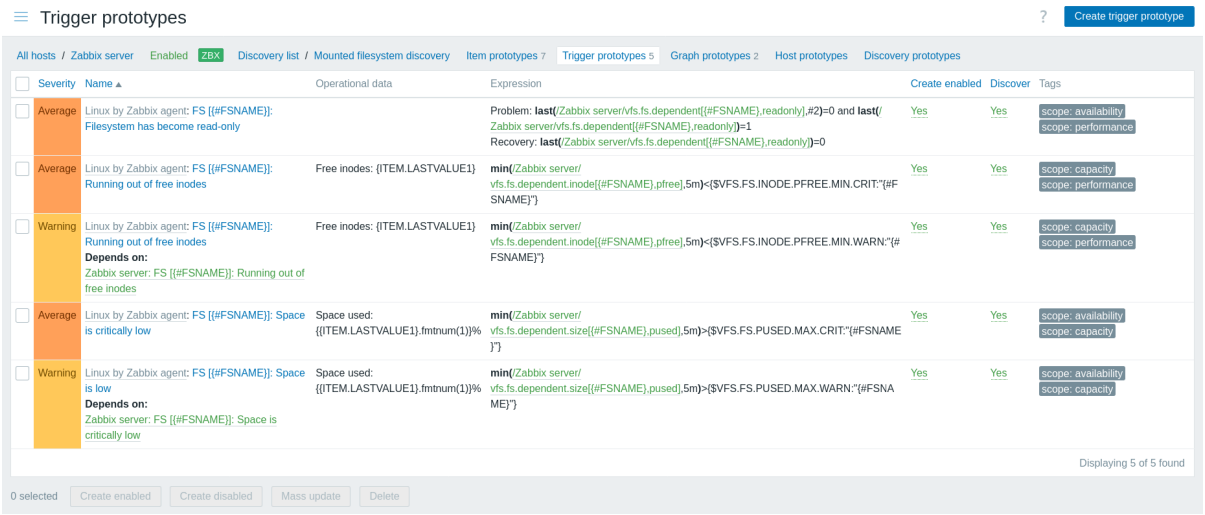
- *Crea abilitati* - crea questi item come *Abilitati*
- *Crea disabilitati* - crea questi item come *Disabilitati*
- *Aggiornamento di massa* - aggiorna in massa questi prototipi di item
- *Elimina* - elimina questi prototipi di item

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di item, quindi fai clic sul pulsante richiesto.

2 Prototipi di trigger

Panoramica

In questa sezione vengono visualizzati i prototipi di trigger di una regola di low-level discovery sul host.
I prototipi di trigger costituiscono la base dei trigger reali del host, che vengono creati durante la low-level discovery.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome del prototipo di trigger, visualizzato come link 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 link grigio. Facendo clic sul link 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 *Monitoring* → *Problems*.
Crea abilitato	Crea il trigger basato su questo prototipo come: **Sì** - abilitato **No** - disabilitato. È possibile passare da 'Sì' a 'No' facendo clic su di essi.
Rileva	Rileva il trigger basato su questo prototipo: **Sì** - rileva **No** - non rilevare. È possibile passare da 'Sì' a 'No' facendo clic su di essi.
Tag	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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

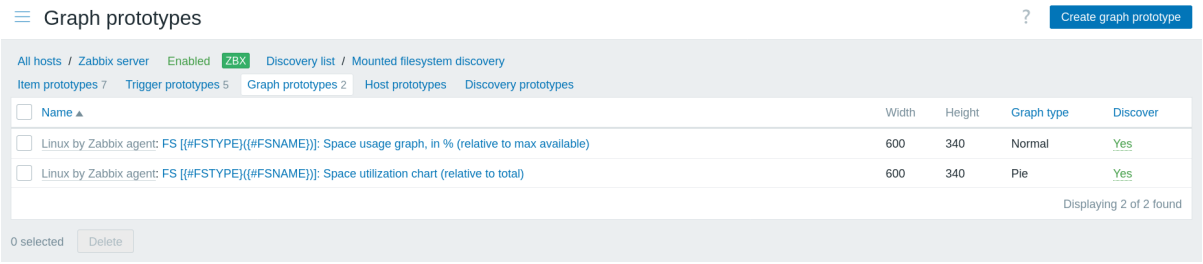
- *Crea abilitati* - crea questi trigger come *Abilitati*
- *Crea disabilitati* - crea questi trigger come *Disabilitati*
- *Aggiornamento di massa* - aggiorna in massa questi prototipi di trigger
- *Elimina* - elimina questi prototipi di trigger

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi prototipi di trigger, quindi fare clic sul pulsante richiesto.

3 Prototipi di grafico

Panoramica

In questa sezione vengono visualizzati i prototipi di grafico di una regola di low-level discovery sull'host. I prototipi di grafico costituiscono la base dei grafici reali dell'host che vengono creati durante la low-level discovery.



Dati visualizzati:

Colonna	Descrizione
Name	Viene visualizzato il nome del prototipo di grafico come link 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 link grigio. Facendo clic sul link 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 del prototipo di grafico: *Normal*, *Stacked*, *Pie* oppure *Exploded*.
Discover	Consente di individuare il grafico in base a questo prototipo: **Yes** - individua **No** - non individuare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.

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:

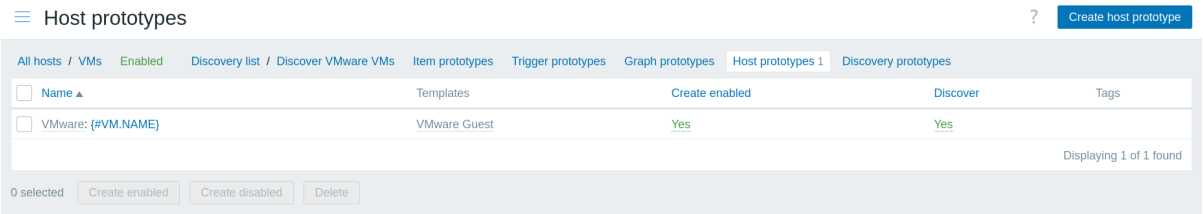
- *Delete* - elimina questi prototipi di grafico

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di grafico, quindi fai clic sul pulsante richiesto.

4 Prototipi degli host

Panoramica

In questa sezione vengono visualizzati i prototipi di host di una regola di low-level discovery sull'host. I prototipi di host costituiscono la base dei reali host che vengono creati durante la low-level discovery.



Dati visualizzati:

Colonna	Descrizione
Name	Nome del prototipo di host, visualizzato come link 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 link grigio. Facendo clic sul link del template si apre l'elenco dei prototipi di host al livello del template collegato.
Templates	Vengono visualizzati i template del prototipo di host.

Colonna	Descrizione
Create enabled	Crea l'host in base a questo prototipo come: **Yes** - abilitato **No** - disabilitato. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Discover	Rileva l'host in base a questo prototipo: **Yes** - rileva **No** - non rilevare. È possibile passare da 'Yes' a 'No' facendo clic su di essi.
Tags	Vengono visualizzati i tag del prototipo di host.

Per configurare un nuovo prototipo di host, fare clic sul pulsante *Create host prototype* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

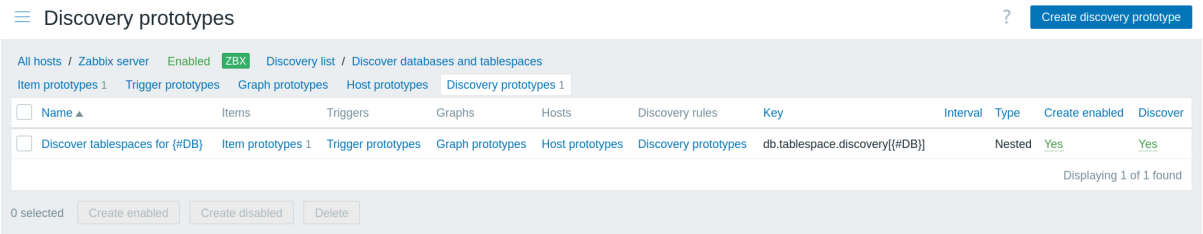
- *Crea abilitati* - crea questi host come *Abilitati*
- *Crea disabilitati* - crea questi host come *Disabilitati*
- *Elimina* - elimina questi prototipi di host

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti ai rispettivi prototipi di host, quindi fai clic sul pulsante richiesto.

5 Prototipi di discovery

Panoramica

In questa sezione vengono visualizzati i prototipi di individuazione di una regola di low-level discovery sul host. I prototipi di individuazione sono regole di individuazione annidate della regola di individuazione padre.



Dati visualizzati:

Colonna	Descrizione
Name	Nome del prototipo di individuazione, visualizzato come collegamento blu. Facendo clic sul nome si apre il modulo di configurazione del prototipo di individuazione. Se il prototipo di individuazione 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 individuazione al 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 grafico. Il numero di prototipi di grafico 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 individuazione. Il numero di prototipi di individuazione esistenti viene visualizzato in grigio.
Key	Viene visualizzata la chiave item utilizzata per l'individuazione.
Interval	Viene visualizzata la frequenza di esecuzione dell'individuazione. Si noti che l'individuazione può essere eseguita anche immediatamente premendo il pulsante *Execute now* sotto l'elenco.
Type	Viene visualizzato il tipo di item utilizzato per l'individuazione (Zabbix agent, SNMP agent, ecc.).

Colonna	Descrizione
Create enabled	Viene visualizzato lo stato di creazione del prototipo di individuazione: 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 individuazione del prototipo di individuazione: individua (*Yes*) o non individuare (*No*). Facendo clic sullo stato è possibile modificarlo, da Yes a No (e viceversa).

Per configurare un nuovo prototipo di individuazione, fare clic sul pulsante *Create discovery prototype* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Create enabled* - crea questi prototipi di discovery come *Abilitati*
- *Create disabled* - crea questi prototipi di discovery come *Disabilitati*
- *Delete* - elimina questi prototipi di discovery

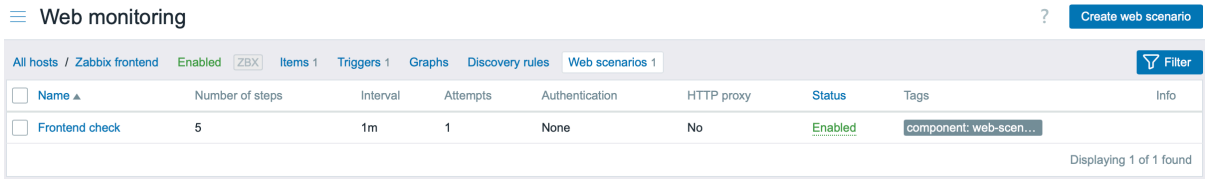
Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi prototipi di discovery, quindi fare clic sul pulsante richiesto.

5 Scenari web

Panoramica

L'elenco degli scenari web per un host è accessibile da *Raccolta dati* → *Host* facendo clic su *Web* per il rispettivo host.

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 aprirà l'elenco degli scenari web a livello di template.
Numero di passaggi	Il numero di passaggi contenuti nello scenario.
Intervallo di aggiornamento	Con quale frequenza viene eseguito lo scenario.
Tentativi	Quanti tentativi vengono effettuati per eseguire i passaggi dello scenario web.
Autenticazione	Viene visualizzato il metodo di autenticazione: Basic, NTLM, Kerberos, Digest oppure None.
Proxy HTTP	Visualizza il proxy HTTP oppure 'No' se non utilizzato.
Stato	Viene visualizzato lo stato dello scenario web: *Abilitato* oppure *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Tag	Vengono visualizzati i tag dello scenario web. Possono essere visualizzati fino a tre tag (coppie nome:valore). Se sono presenti più tag, viene visualizzato un collegamento "..." che consente di vedere tutti i tag al passaggio del mouse.
Info	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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

- *Enable* - cambia lo stato dello scenario in *Enabled*

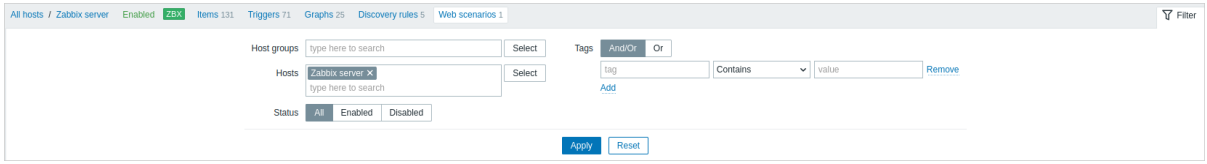
- *Disable* - cambia lo stato dello scenario in *Disabled*
- *Clear history and trends* - cancella i dati di cronologia e di trend degli scenari
- *Delete* - elimina gli scenari web

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi scenari web, quindi fare clic sul pulsante richiesto.

Uso del filtro

È possibile utilizzare il filtro per visualizzare solo gli scenari di interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Il collegamento *Filtro* è disponibile sopra l'elenco degli scenari web. Facendo clic su di esso si aprirà un filtro in cui è possibile specificare i criteri di filtraggio desiderati.



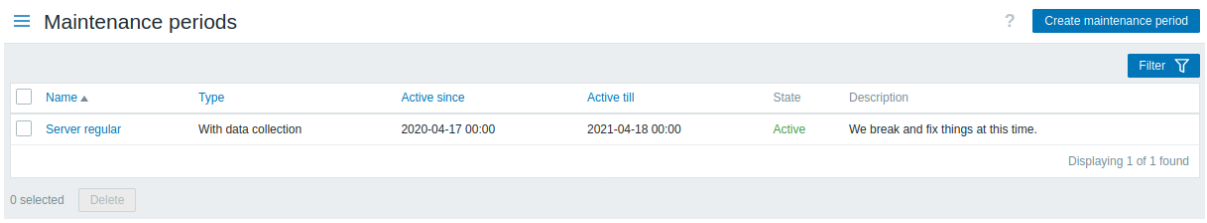
Parametro	Descrizione
Gruppi host	Filtra per uno o più gruppi host. Specificando un gruppo host padre vengono implicitamente selezionati tutti i gruppi host nidificati.
Host	Filtra per uno o più host. Se sopra sono già selezionati dei gruppi host, la selezione degli host è limitata a tali gruppi.
Stato	Filtra per stato dello scenario web.
Tag	Filtra per nome e valore del tag dello scenario web. È possibile includere così come escludere tag e valori di tag specifici. È possibile impostare diverse condizioni. La corrispondenza del nome del tag è sempre sensibile alle maiuscole/minuscole. Sono disponibili diversi operatori per ciascuna condizione: **Esiste** - include i nomi di tag specificati; **Uguale a** - include i nomi e i valori di tag specificati (sensibile alle maiuscole/minuscole); **Contiene** - include i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, non sensibile alle maiuscole/minuscole); **Non esiste** - esclude i nomi di tag specificati; **Diverso da** - esclude i nomi e i valori di tag specificati (sensibile alle maiuscole/minuscole); **Non contiene** - esclude i nomi di tag specificati in cui i valori del tag contengono la stringa immessa (corrispondenza per sottostringa, non sensibile alle maiuscole/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.

5 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* oppure *Senza raccolta dati*
Attivo da	La data e l'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 a	La data e l'ora in cui l'esecuzione dei periodi di manutenzione cessa di essere attiva.
Stato	Lo stato del periodo di manutenzione: **In avvicinamento** - 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 di massa

Un pulsante sotto l'elenco offre un'opzione di modifica di massa:

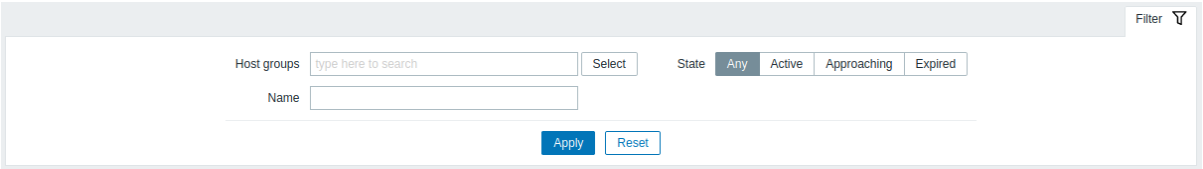
- *Delete* - elimina i periodi di manutenzione

Per utilizzare questa opzione, seleziona le caselle di controllo davanti ai rispettivi periodi di manutenzione e fai clic su *Delete*.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i periodi di manutenzione di proprio interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il link *Filtro* è disponibile sopra l'elenco dei periodi di manutenzione. Facendo clic su di esso, viene visualizzato un filtro che consente di filtrare i periodi di manutenzione per gruppo host, nome e stato.



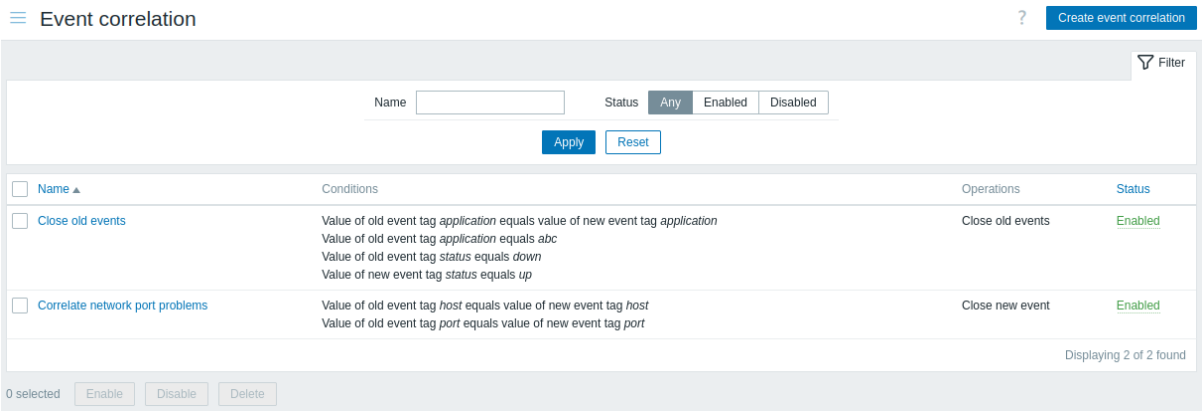
Calcolo delle code durante la manutenzione

Attention:
Il proxy Zabbix non è a conoscenza dei periodi di manutenzione; vedere Calculation of queues during maintenance per i dettagli.

6 Correlazione degli eventi

Panoramica

Nella sezione *Raccolta dati* → *Correlazione eventi* gli utenti possono configurare e gestire le regole di correlazione globale per gli eventi di Zabbix.



Dati visualizzati:

Colonna	Descrizione
Nome	Nome della regola di correlazione. Facendo clic sul nome della regola di correlazione si apre il modulo di configurazione della regola.
Condizioni	Vengono visualizzate le condizioni della regola di correlazione.
Operazioni	Vengono visualizzate le operazioni della regola di correlazione.
Stato	Viene visualizzato lo stato della regola di correlazione: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.

Per configurare una nuova regola di correlazione, fare clic sul pulsante *Crea correlazione eventi* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

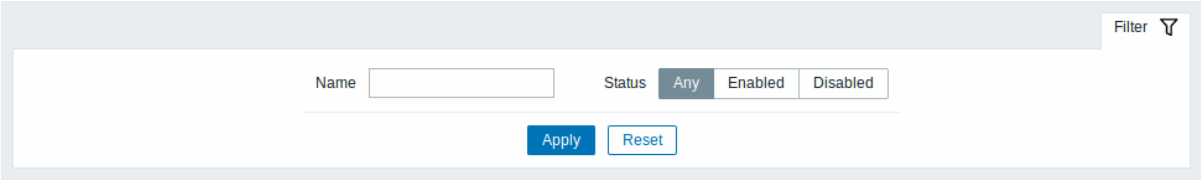
- *Abilita* - cambia lo stato della regola di correlazione in *Abilitato*
- *Disabilita* - cambia lo stato della regola di correlazione in *Disabilitato*
- *Elimina* - elimina le regole di correlazione

Per utilizzare queste opzioni, selezionare le caselle di controllo accanto alle rispettive regole di correlazione, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo le regole di correlazione di proprio interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il link *Filtro* è disponibile sopra l'elenco delle regole di correlazione. Facendo clic su di esso, diventa disponibile un filtro in cui è possibile filtrare le regole di correlazione per nome e stato.

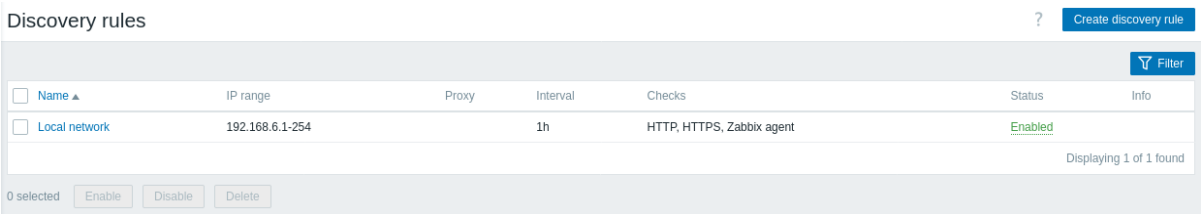


7 Individuazione

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:

Column	Description
Name	Nome della regola di discovery. Facendo clic sul nome della regola di discovery si apre il modulo di configurazione della regola di discovery.
IP range	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.
Interval	Viene visualizzata la frequenza con cui viene eseguita la discovery.
Checks	Vengono visualizzati i tipi di controlli utilizzati per la discovery.
Status	Viene visualizzato lo stato della regola di discovery: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Info	Se tutto funziona correttamente, in questa colonna non viene visualizzato nulla. In caso di errori, viene visualizzata un'icona informativa rossa con la lettera "i". Passare il cursore sull'icona per visualizzare un suggerimento con la descrizione dell'errore.

Per configurare una nuova regola di discovery, fare clic sul pulsante *Create discovery rule* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

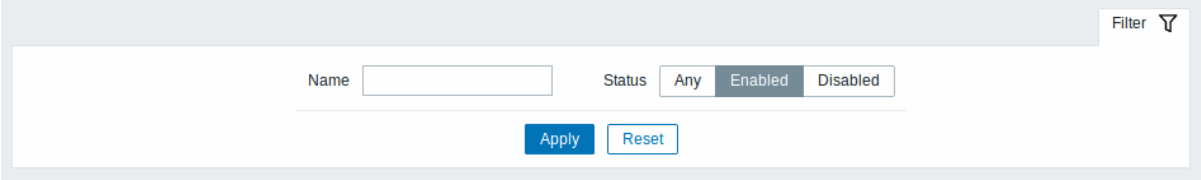
- *Enable* - cambia lo stato della regola di individuazione in *Enabled*
- *Disable* - cambia lo stato della regola di individuazione in *Disabled*
- *Delete* - elimina le regole di individuazione

Per utilizzare queste opzioni, seleziona le caselle di controllo davanti alle rispettive regole di individuazione, quindi fai clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo le regole di discovery di proprio interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il link *Filtro* è disponibile sopra l'elenco delle regole di discovery. Facendo clic su di esso, diventa disponibile un filtro con cui è possibile filtrare le regole di discovery per nome e stato.



7 Avvisi

Panoramica

Questo menu presenta sezioni relative alla configurazione avvisi in Zabbix.

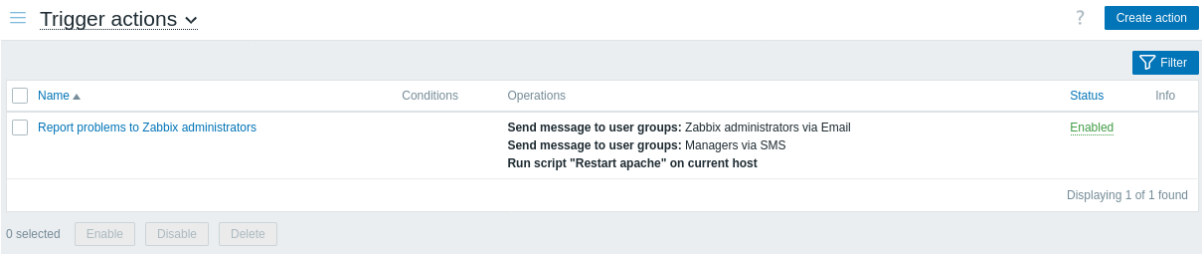
1 Azioni

Panoramica

Nella sezione *Avvisi* → *Azioni* gli utenti possono configurare e gestire le azioni.

Le azioni visualizzate sono quelle assegnate alla sorgente evento selezionata (trigger, servizi, discovery, autoregistrazione, azioni interne).

Per passare a una sorgente evento diversa, fare clic su *Azioni* nella sezione del menu *Avvisi*. È anche possibile passare da una sorgente all'altra usando 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 (email, 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 sia una notifica sia un comando remoto, a seconda del tipo di operazione selezionato.

Colonna	Descrizione
Stato	Viene visualizzato lo stato dell'azione: *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo. Per maggiori dettagli su cosa accade se un'azione viene disabilitata durante un'escalation in corso, vedere la sezione Escalation.
Info	Se tutto funziona correttamente, in questa colonna non viene visualizzata alcuna icona. In caso di errori, come operazioni o condizioni dell'azione mancanti dopo un aggiornamento/discovery di basso livello, viene visualizzata un'icona di avviso. Passare il puntatore 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 dei permessi. Ciò significa che in alcuni casi un utente senza diritti di Super admin non è in grado di visualizzare l'elenco completo delle azioni a causa di determinate restrizioni dei permessi. Un'azione viene visualizzata all'utente senza diritti di Super admin se sono soddisfatte le seguenti condizioni:

- L'utente dispone di accesso in lettura-scrittura ai gruppi host, agli host, ai template e ai trigger nelle condizioni dell'azione
- L'utente dispone di accesso in lettura-scrittura ai gruppi host, agli host e ai template nelle operazioni dell'azione, nelle operazioni di ripristino e nelle operazioni di aggiornamento
- L'utente dispone di accesso in lettura ai gruppi utenti e agli utenti nelle operazioni dell'azione, nelle operazioni di ripristino e nelle operazioni di aggiornamento

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

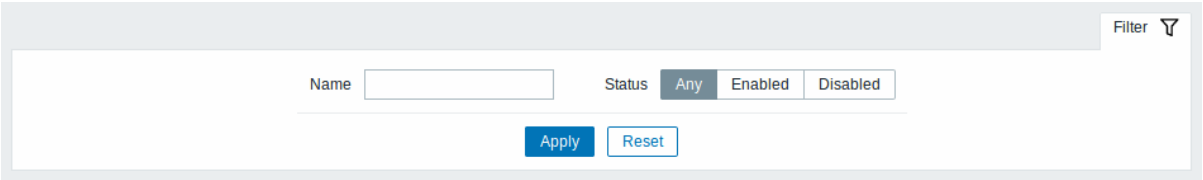
- *Abilita* - cambia lo stato dell'azione in *Abilitato*
- *Disabilita* - modifica lo stato dell'azione in *Disabilitata*
- *Elimina* - elimina le azioni

Per utilizzare queste opzioni, contrassegnare le caselle prima delle rispettive azioni, quindi fare 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.



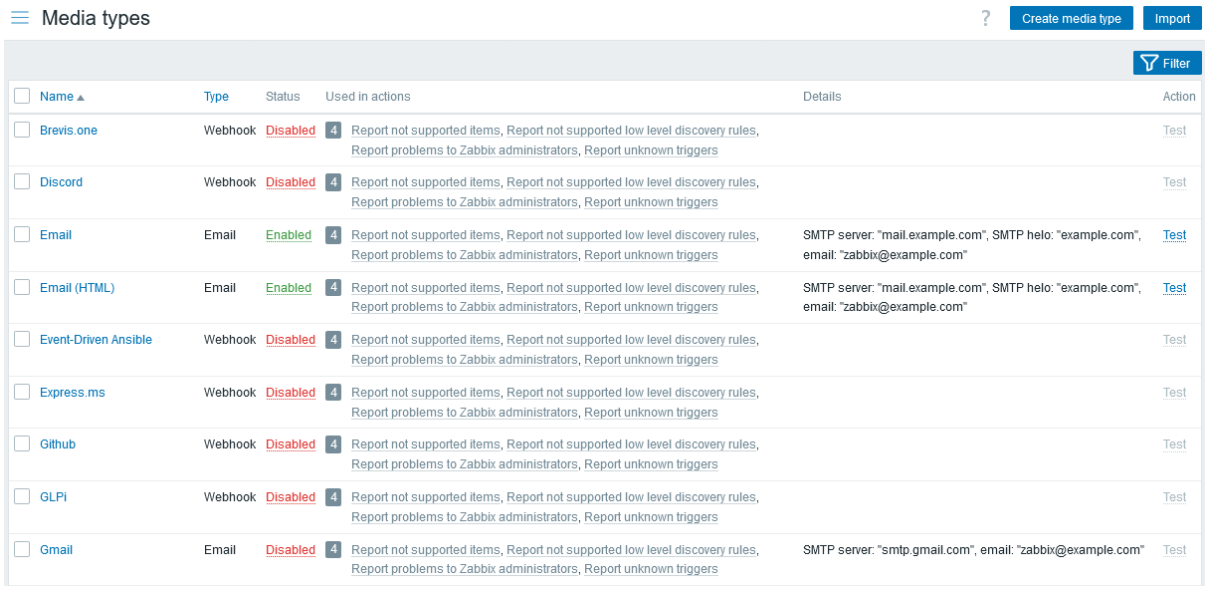
2 tipi di supporto

Panoramica

Nella sezione *Avvisi* → *Tipi di supporto* gli utenti possono configurare e gestire le informazioni sui tipi di supporto.

Le informazioni sul tipo di supporto contengono istruzioni generali per l'uso di un mezzo come canale di consegna per le notifiche. I dettagli specifici, come i singoli indirizzi email a cui inviare una notifica, sono mantenuti per i singoli utenti.

Viene visualizzato un elenco dei tipi di supporto esistenti con i relativi dettagli.



Dati visualizzati:

Column	Description
Name	Nome del tipo di supporto. Facendo clic sul nome si apre il modulo di configurazione del tipo di supporto.
Type	Viene visualizzato il tipo di supporto (email, SMS, ecc.).
Status	Viene visualizzato lo stato del tipo di supporto - *Abilitato* o *Disabilitato*. Facendo clic sullo stato è possibile modificarlo.
Used in actions	Vengono visualizzate le azioni in cui è 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 cliccabile.
Details	Vengono visualizzate informazioni dettagliate sul tipo di supporto.
Actions	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 messaggio di prova) e inviare un messaggio di prova 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:

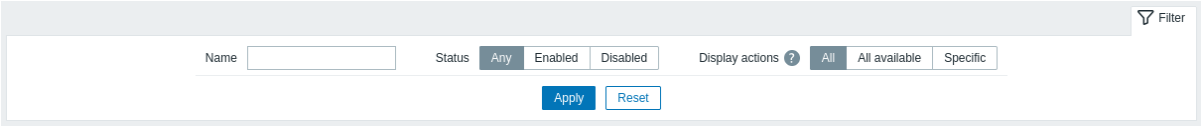
- *Abilita* - modifica lo stato del tipo di supporto in *Abilitato*
- *Disabilita* - modifica lo stato del tipo di supporto in *Disabilitato*
- *Esporta*: esporta i tipi di media in un file YAML, XML o JSON
- *Elimina* - elimina i tipi di media

Per utilizzare queste opzioni, contrassegnare le caselle di controllo prima del rispettivo supporto tipi, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i tipi di supporto di proprio interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

Il collegamento *Filtro* è disponibile sopra l'elenco dei tipi di supporto. Facendo clic su di esso, diventa disponibile un filtro con cui è possibile filtrare i tipi di supporto per nome e stato. Inoltre, è possibile utilizzare il filtro per visualizzare le azioni nella colonna *Utilizzato nelle azioni* in base all'ambito del loro utilizzo del tipo di supporto (definito dal parametro *Send to media type* nei dettagli dell'operazione dell'azione).



3 Script

Panoramica

Nella sezione *Avvisi* > *Script* è possibile configurare e gestire script globali definiti dall'utente.

Gli script globali, a seconda dell'ambito configurato e anche dei permessi utente, sono disponibili per l'esecuzione:

- dal menu host in varie posizioni del frontend (*Dashboard*, *Problemi*, *Ultimi dati*, *Mappe*, ecc.)
- dal menu evento
- possono essere eseguiti come operazione di un'azione

Gli script vengono eseguiti su Zabbix agent, Zabbix server (proxy) oppure solo su Zabbix server. Vedere anche Esecuzione dei comandi.

Sia su Zabbix agent che su Zabbix proxy gli script remoti sono disabilitati per impostazione predefinita. Possono essere abilitati come segue:

- Per i comandi remoti eseguiti su Zabbix agent:
  - aggiungendo un parametro AllowKey=system.run[<command>,*] per ciascun comando consentito nella configurazione dell'agent, dove * indica la modalità wait e nowait;
- Per i comandi remoti eseguiti su Zabbix proxy:
  - **Avviso: non è necessario abilitare i comandi remoti su Zabbix proxy se i comandi remoti vengono eseguiti su Zabbix agent monitorato da Zabbix proxy.** Tuttavia, se è necessario eseguire comandi remoti su Zabbix proxy, impostare il parametro *EnableRemoteCommands* a '1' nella configurazione del proxy.

L'esecuzione degli script globali su Zabbix server può essere disabilitata impostando `EnableGlobalScripts=0` nella configurazione del server. Per le nuove installazioni, 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:

Column	Description
Name	Nome dello script. Facendo clic sul nome dello script si apre il modulo di configurazione dello script.
Scope	Ambito dello script: operazione di un'azione, azione manuale sull'host o azione manuale sull'evento. Questa impostazione determina dove lo script è disponibile.
Used in actions	Vengono visualizzate tutte le azioni in cui lo script è utilizzato, 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 dei permessi per l'azione, il nome non è cliccabile.
Type	Viene visualizzato il tipo di script: comando *URL*, *Webhook*, *Script*, *SSH*, *Telnet* o *IPMI*.
Execute on	Viene indicato se lo script sarà eseguito su Zabbix agent, Zabbix proxy o server, oppure solo su Zabbix server.
Commands	Vengono visualizzati tutti i comandi da eseguire all'interno dello script. Per i webhook qui non viene visualizzato nulla.

Column	Description
User group	Viene visualizzato il gruppo utenti per cui lo script è disponibile (oppure *All* per tutti i gruppi utenti).
Host group	Viene visualizzato il gruppo host per cui lo script è disponibile (oppure *All* per tutti i gruppi host).
Host access	Viene visualizzato il livello di permesso per il gruppo host: *Read* o *Write*. Solo gli utenti con il livello di permesso richiesto avranno accesso all'esecuzione dello script.

Per configurare un nuovo script, fare clic sul pulsante *Create script* nell'angolo superiore destro.

Opzioni di modifica di massa

Un pulsante sotto l'elenco offre un'opzione di modifica di massa:

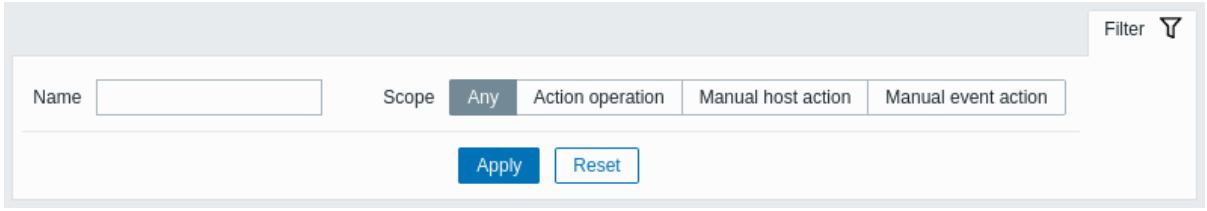
- *Delete* - elimina gli script

Per utilizzare questa opzione, seleziona le caselle di controllo davanti ai rispettivi script e fai clic su *Delete*.

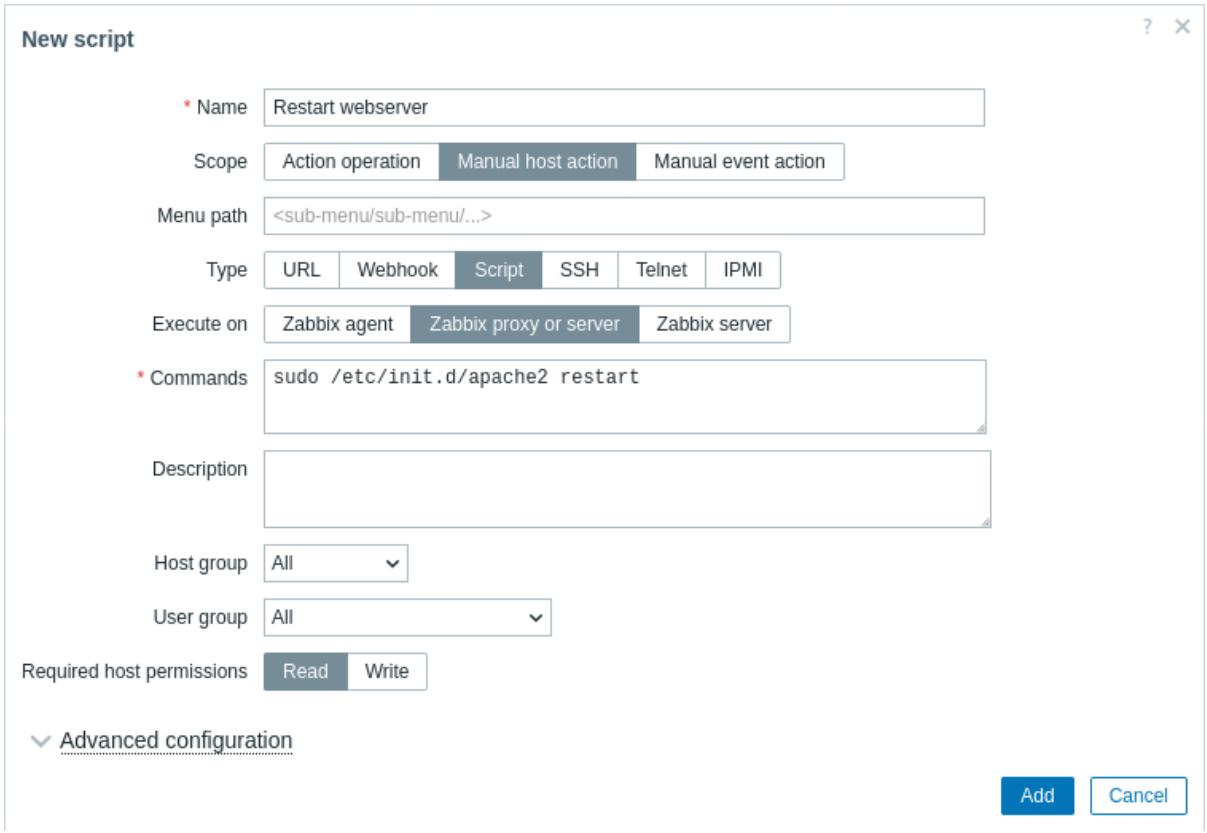
Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo gli script di interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

Il link *Filtro* è disponibile sopra l'elenco degli script. Facendo clic su di esso, viene visualizzato un filtro che consente di 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 all'escaping.

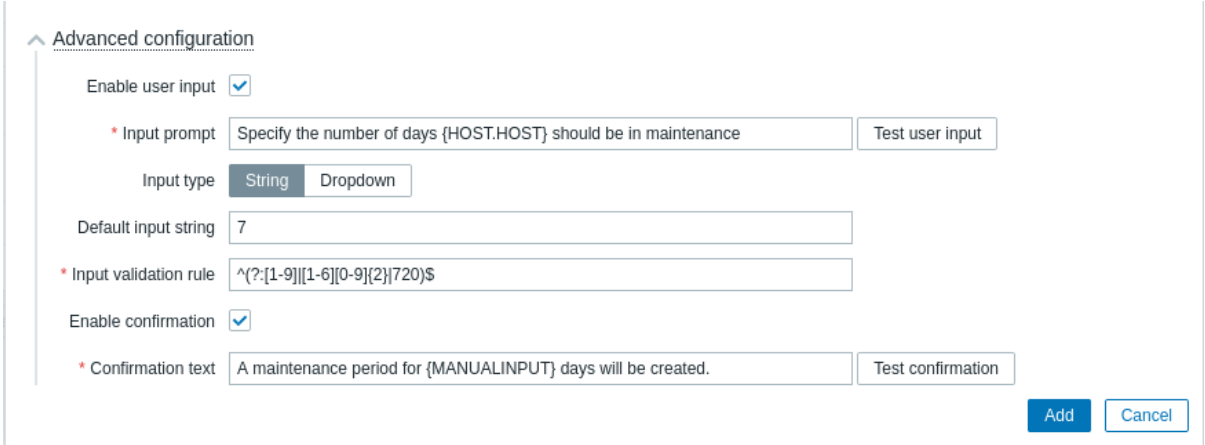
Attributi dello script:

Parametro	Descrizione
Name	Nome univoco dello script. Ad es. `Clear /tmp filesystem`
Scope	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 di azione, dal menu host o dal menu evento rispettivamente. Impostando l'ambito su 'Action operation', lo script sarà disponibile per tutti gli utenti con accesso a *Alerts* > *Actions*. Se uno script è effettivamente utilizzato in un'azione, il suo ambito non può essere modificato da 'action operation' a un altro valore. **Supporto macro** L'ambito influisce sull'intervallo 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 action operation, 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 per gli script di azione manuale sull'host e di azione manuale sull'evento. Per sapere quali altre macro sono supportate, cerca 'Trigger-based notifications and commands/Trigger-based commands', 'Manual host action scripts' e 'Manual event action scripts' nella tabella delle macro supportate. Nota che se una macro può risolversi in un valore contenente spazi (ad esempio, il nome host), non dimenticare di racchiuderla tra virgolette se necessario.
Menu path	Percorso di menu desiderato per lo script. Ad esempio, `Default` o `Default/` visualizzerà lo script nella directory corrispondente. I menu possono essere nidificati, ad es. `Main menu/Sub menu1/Sub menu2`. Quando si accede agli script tramite il menu host/evento nelle sezioni di monitoraggio, essi saranno organizzati in base alle directory specificate. Questo campo viene visualizzato solo se come *Scope* è selezionato 'Manual host action' o 'Manual event action'.
Type	Fai clic sul pulsante corrispondente per selezionare il tipo di script: comando **URL**, **Webhook**, **Script**, **SSH**, **Telnet** o **IPMI**. Il tipo **URL** è disponibile solo quando come *Scope* è selezionato 'Manual host action' o 'Manual event action'.
Tipo di script: **URL**	
URL	Specifica l'URL per l'accesso rapido dal menu host o dal menu evento. Sono supportate le macro e le macro utente personalizzate. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Usa 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 come URL.
Open in new window	Determina se l'URL deve essere aperto in una nuova scheda del browser o nella stessa.
Tipo di script: **Webhook**	
Parameters	Specifica le variabili del webhook come coppie attributo-valore. Vedi anche: configurazione del tipo di supporto Webhook. Le macro e le macro utente personalizzate sono supportate nei valori dei parametri. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra).
Script	Inserisci il codice JavaScript nell'editor modale che si apre facendo clic nel campo del parametro o sull'icona della matita accanto ad esso. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Vedi anche: configurazione del tipo di supporto Webhook, Oggetti Javascript aggiuntivi.
Timeout	Timeout di esecuzione JavaScript (1-60s, predefinito 30s). I suffissi temporali sono supportati, ad es. 30s, 1m.
Tipo di script: **Script**	

Parametro	Descrizione
Execute on	Fai clic sul pulsante corrispondente per eseguire lo script shell su: **Zabbix agent** - lo script verrà eseguito da Zabbix agent (se il item system.run è consentito) sull'host **Zabbix proxy or server** - lo script verrà eseguito da Zabbix proxy o server, a seconda che l'host sia monitorato dal proxy o dal server. Verrà eseguito sul proxy se abilitato da EnableRemoteCommands. Verrà eseguito sul server se gli script globali sono abilitati dal parametro 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 server EnableGlobalScripts.
Commands	Inserisci il percorso completo dei comandi da eseguire all'interno dello script. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **SSH**	
Authentication method	Seleziona il metodo di autenticazione: password o chiave pubblica.
Username	Inserisci il nome utente.
Password	Inserisci la password. Questo campo è disponibile se come metodo di autenticazione è selezionato 'Password'.
Public key file	Inserisci il percorso del file della chiave pubblica. Questo campo è disponibile se come metodo di autenticazione è selezionato 'Public key'.
Private key file	Inserisci il percorso del file della chiave privata. Questo campo è disponibile se come metodo di autenticazione è selezionato 'Public key'.
Passphrase	Inserisci la passphrase. Questo campo è disponibile se come metodo di autenticazione è selezionato 'Public key'.
Port	Inserisci la porta del servizio SSH remoto sull'host di destinazione a cui Zabbix si connetterà.
Commands	Inserisci i comandi. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **Telnet**	
Username	Inserisci il nome utente.
Password	Inserisci la password.
Port	Inserisci la porta del servizio Telnet remoto sull'host di destinazione a cui Zabbix si connetterà.
Commands	Inserisci i comandi. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Sono supportate le macro utente personalizzate.
Tipo di script: **IPMI**	
Command	Inserisci il comando IPMI. Il supporto macro dipende dall'ambito dello script (vedi *Scope* sopra). Sono supportate le macro utente personalizzate.
Description	Inserisci una descrizione per lo script.
Host group	Seleziona il gruppo host per il quale lo script sarà disponibile (oppure *All* per tutti i gruppi host).
User group	Seleziona il gruppo utenti a cui lo script sarà disponibile (oppure *All* per tutti i gruppi utenti). Questo campo viene visualizzato solo se come *Scope* è selezionato 'Manual host action' o 'Manual event action'.
Required host permissions	Seleziona il livello di autorizzazione per il gruppo host: *Read* o *Write*. Solo gli utenti con il livello di autorizzazione richiesto avranno accesso all'esecuzione dello script. Questo campo viene visualizzato solo se come *Scope* è selezionato 'Manual host action' o 'Manual event action'.
Advanced configuration	Fai clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata. Questo campo viene visualizzato solo se come *Scope* è selezionato 'Manual host action' o 'Manual event action'.

Configurazione avanzata

Le opzioni di configurazione avanzata sono disponibili in una sezione comprimibile *Configurazione avanzata*:



Parametro	Descrizione
Abilita input utente	Selezionare la casella per abilitare l'input utente manuale prima dell'esecuzione dello script. L'input utente manuale sostituirà il valore della macro {MANUALINPUT} nello script. Vedere anche: Input utente manuale.
Prompt di input	Inserire un testo personalizzato che richieda un input utente personalizzato. Questo testo verrà visualizzato sopra il campo di input nel popup *Input manuale*. Per vedere un'anteprima del popup *Input manuale*, fare clic su *Test input utente*. L'anteprima consente anche di verificare se la stringa di input è conforme alla regola di validazione dell'input (vedere i parametri sotto). 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; **Menu a discesa** - il valore viene selezionato da più opzioni del menu a discesa.
Opzioni del menu a discesa	Inserire valori univoci per il menu a discesa dell'input utente in un elenco delimitato da virgole. Per includere un'opzione vuota nel menu a discesa, aggiungere una virgola aggiuntiva all'inizio, al centro o alla fine dell'elenco. Questo campo viene visualizzato solo se come *Tipo di input* è selezionato 'Menu a discesa'.
Stringa di input predefinita	Inserire la stringa predefinita per l'input utente (oppure nessuna). Questo campo verrà convalidato rispetto all'espressione regolare fornita nel campo *Regola di validazione dell'input*. Il valore inserito qui verrà visualizzato per impostazione predefinita nel popup *Input manuale*. Questo campo viene visualizzato solo se come *Tipo di input* è selezionato 'Stringa'.
Regola di validazione dell'input	Inserire un'espressione regolare per convalidare la stringa di input dell'utente. Le espressioni regolari globali sono supportate. Questo campo viene visualizzato solo se come *Tipo di input* è selezionato 'Stringa'.
Abilita conferma	Selezionare la casella per visualizzare un messaggio di conferma prima dell'esecuzione dello script. Questa funzionalità può essere particolarmente utile con operazioni potenzialmente pericolose (come uno script di riavvio) o che potrebbero richiedere molto tempo.
Testo di conferma	Inserire un testo di conferma personalizzato per il popup di conferma abilitato con la casella sopra (ad esempio, *Il sistema remoto verrà riavviato. Continuare?*). Per vedere come apparirà il testo, fare clic su *Test conferma* accanto al campo. Sono supportate le macro e le macro utente personalizzate. *Nota:* le macro non verranno espanse durante il test del messaggio di conferma.

Se sono configurati sia l'input utente manuale sia un messaggio di conferma, verranno visualizzati in finestre popup consecutive.

Input utente manuale

L'input utente manuale consente di fornire un parametro personalizzato a ogni esecuzione dello script. In questo modo si evita la necessità di creare più script utente simili che differiscono solo per un singolo parametro.

Ad esempio, durante l'esecuzione potresti voler fornire allo script un numero intero diverso o un indirizzo URL diverso.

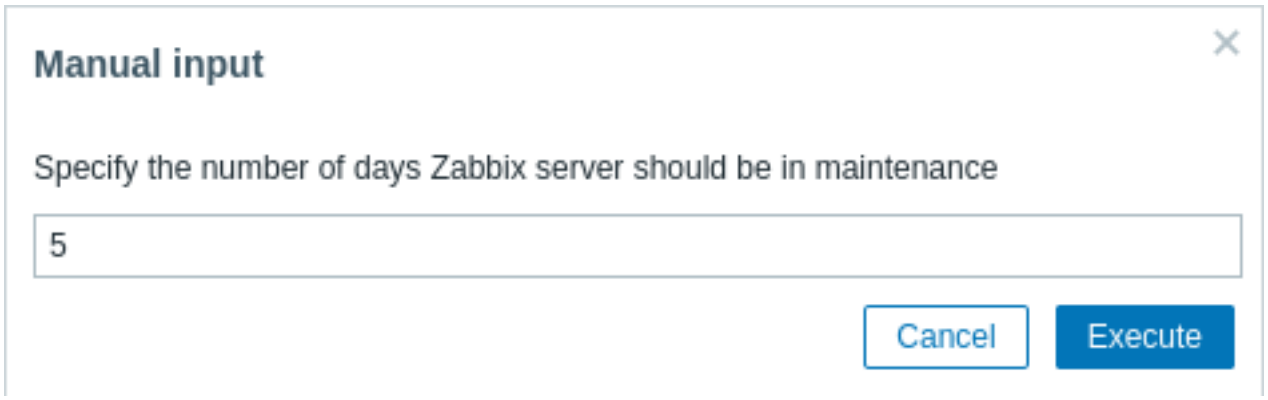
Per abilitare l'input utente manuale:

- utilizzare la macro {MANUALINPUT} nello script (comandi, script, parametro dello script) dove richiesto; oppure nel campo URL degli script URL;

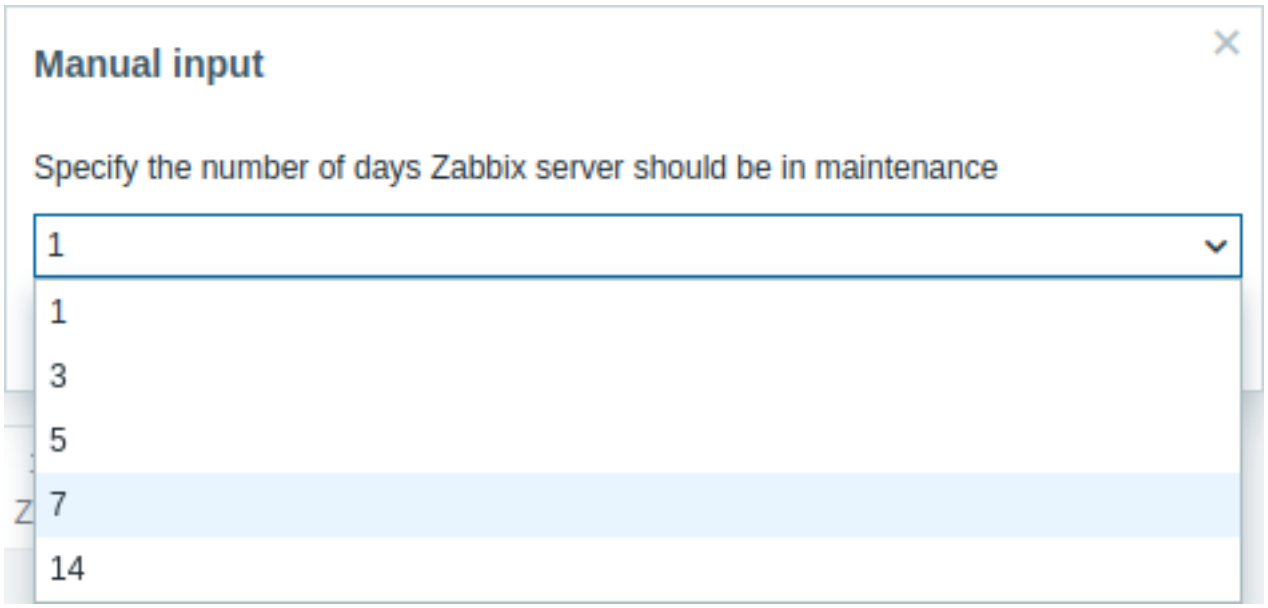
- nella configurazione avanzata dello script, abilitare l'input utente manuale e configurare le opzioni di input.

Con l'input utente abilitato, prima dell'esecuzione dello script, all'utente verrà mostrato un popup *Input manuale* che richiederà 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 menu a discesa con opzioni predefinite:



L'input utente manuale è disponibile solo per gli script in cui l'ambito è 'Manual host action' o 'Manual event action'.

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:

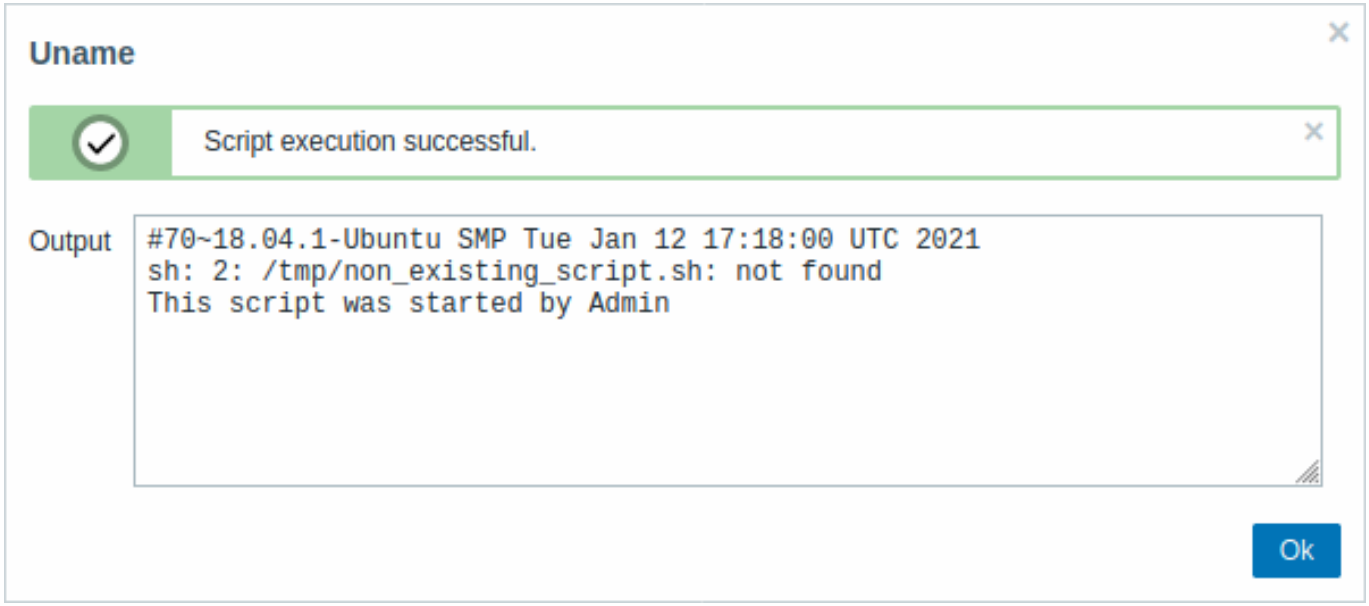
- Se lo script termina correttamente (codice di uscita 0), il valore restituito è limitato a 16 MB (inclusi gli spazi finali che vengono troncati).
- Se lo script termina con un errore (codice di uscita diverso da zero), il valore restituito è l'errore standard limitato a 2 KB.

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.

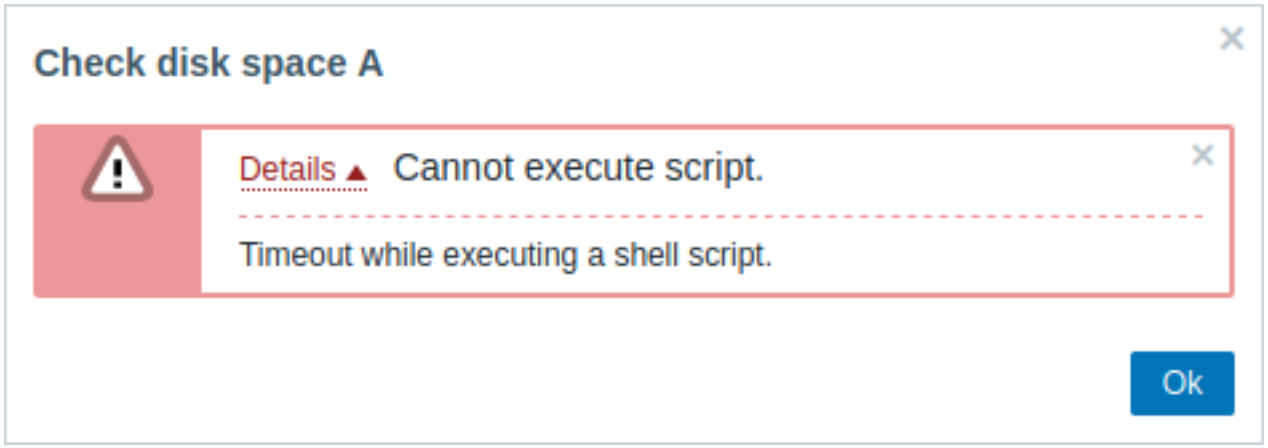
Timeout dello script

Zabbix agent

Potresti riscontrare una situazione in cui si verifica un timeout durante l'esecuzione di uno script.

Di seguito è riportato un esempio di script in esecuzione su Zabbix agent e la relativa finestra dei risultati:

```
sleep 5
df -h
```



Il messaggio di errore, in questo caso, è il seguente:

```
Timeout while executing a shell script.
```

Per evitare tali situazioni, si consiglia di ottimizzare lo script stesso (nell'esempio sopra, "5") invece di modificare il parametro `Timeout` nella configurazione di Zabbix agent e nella configurazione di Zabbix server. Tuttavia, per Zabbix agent in modalità attiva, il parametro `Timeout` nella configurazione di Zabbix server deve essere di almeno alcuni secondi superiore al parametro `RefreshActiveChecks` nella configurazione di Zabbix agent. Questo garantisce che il server abbia tempo sufficiente per ricevere dall'agent i risultati dei controlli attivi. Si noti che l'esecuzione di script su un agent attivo è supportata a partire da Zabbix agent 7.0.

Nel caso in cui il parametro `Timeout` sia stato modificato nella configurazione di Zabbix agent, verrà visualizzato il seguente messaggio di errore:

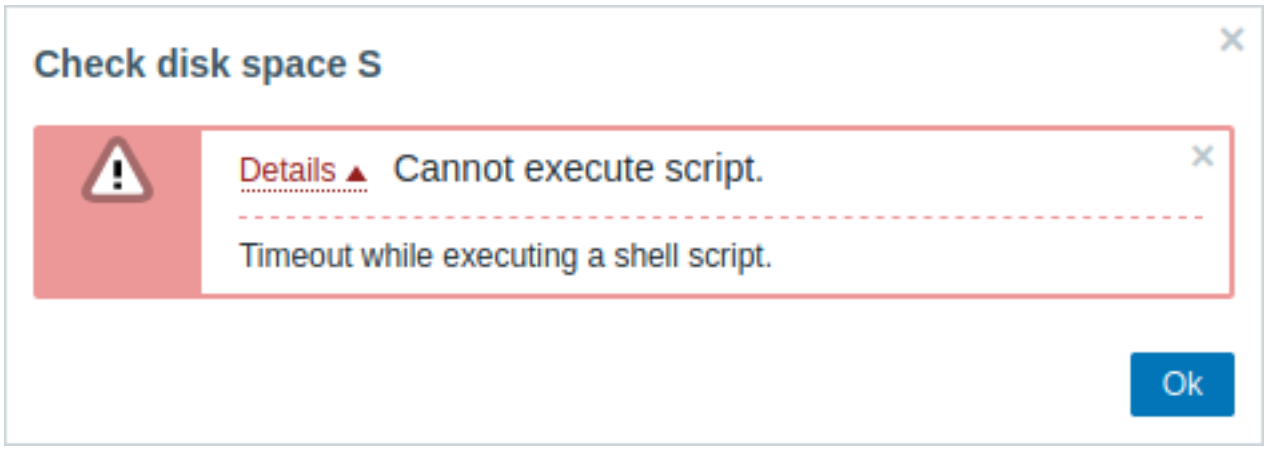
```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Ciò significa che la modifica è stata effettuata nella configurazione di Zabbix agent, ma è necessario modificare anche il parametro `Timeout` nella configurazione di Zabbix server.

Zabbix server/proxy

Di seguito è riportato un esempio di script in esecuzione su Zabbix server e la finestra del risultato:

```
sleep 11
df -h
```



Si consiglia inoltre di ottimizzare lo script stesso (invece di regolare il parametro `TrapperTimeout` su un valore corrispondente (nel nostro caso, > 11)) modificando la configurazione di Zabbix server.

8 Utenti

Panoramica

Questo menu include sezioni relative alla configurazione degli utenti in Zabbix. Questo menu è disponibile solo per gli utenti del tipo SuperAdmin.

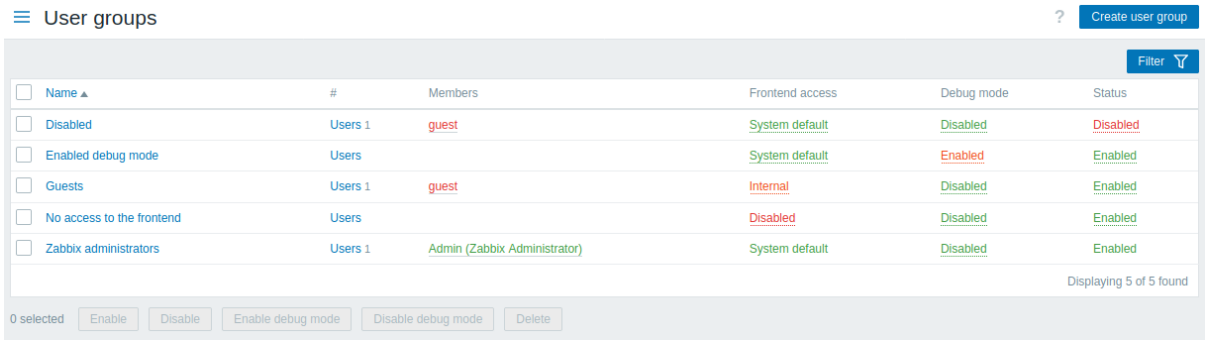
1 Gruppi di utenti

Panoramica

Nella sezione *Utenti* → *Gruppi di utenti* 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.
#	Il numero di utenti nel gruppo. Facendo clic su *Utenti* verranno visualizzati i rispettivi utenti filtrati nell'elenco 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 dei gruppi disabilitati sono 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 quella predefinita globale; **LDAP** - gli utenti vengono autenticati da LDAP; ignorato se l'autenticazione HTTP è quella predefinita globale; **Disabilitato** - l'accesso al frontend di Zabbix è vietato per questo gruppo. Facendo clic sul livello corrente, è possibile modificarlo.
Modalità di debug	Viene visualizzato lo stato della modalità di 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 di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

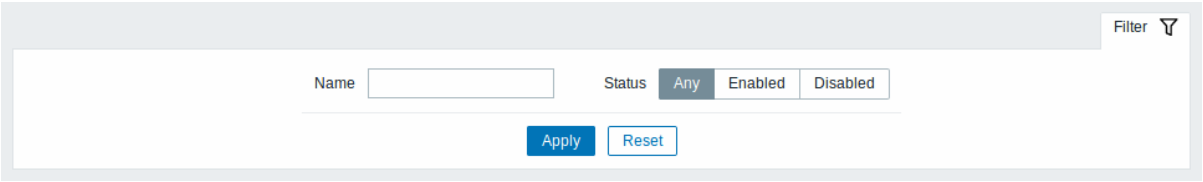
- *Abilita* - modifica lo stato del gruppo utenti in *Abilitato*
- *Disabilita* - modifica lo stato del gruppo utenti in *Disabilitato*
- *Abilita modalità di debug* - abilita la modalità di debug per i gruppi di utenti
- *Disabilita modalità di debug* - disabilita la modalità di debug per i gruppi di utenti
- *Elimina* - elimina i gruppi di utenti

Per utilizzare queste opzioni, contrassegnare le caselle prima del rispettivo gruppo di utenti, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

Puoi usare il filtro per visualizzare solo i gruppi di utenti a cui si è interessati. Per migliorare le prestazioni della ricerca, i dati vengono cercati con macro irrisolte.

Il collegamento *Filtro* è disponibile sopra l'elenco dei gruppi di utenti. Se tu facendo clic su di esso, diventa disponibile un filtro in cui è possibile filtrare i gruppi degli utenti per nome e stato.



2 Ruoli utente

Panoramica

Nella sezione *Utenti* > *Ruoli utente* è possibile creare ruoli utente.

I ruoli utente consentono di creare autorizzazioni granulari in base al tipo di utente inizialmente selezionato (*User*, *Admin*, *Super admin*).

Quando si seleziona un tipo di utente, vengono concesse tutte le autorizzazioni disponibili per quel 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 quanto disponibile per quel tipo di utente.

Le caselle di controllo relative alle autorizzazioni non disponibili sono disattivate; gli utenti non potranno accedere all'elemento neppure inserendo nel browser un URL diretto a tale elemento.

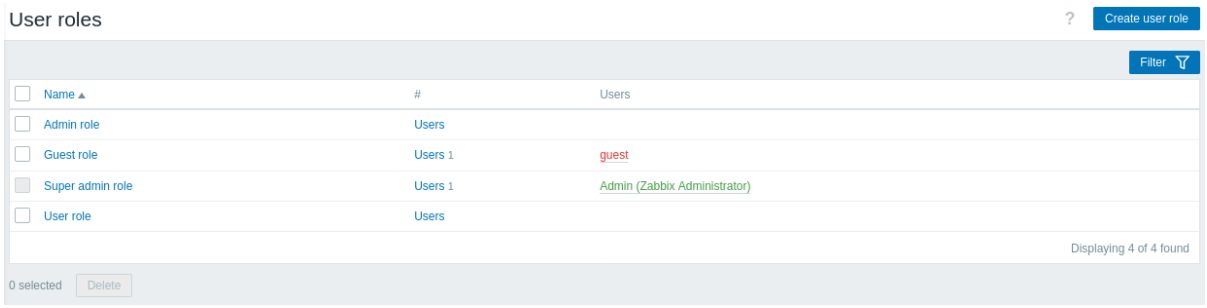
Note:
La limitazione dell'accesso ad alcuni elementi dell'interfaccia utente impedisce solo l'apertura di quella pagina, ma non elimina la possibilità di accedere ai dati sottostanti in altre parti dell'interfaccia.

I ruoli utente possono essere assegnati agli utenti del sistema. A ciascun 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

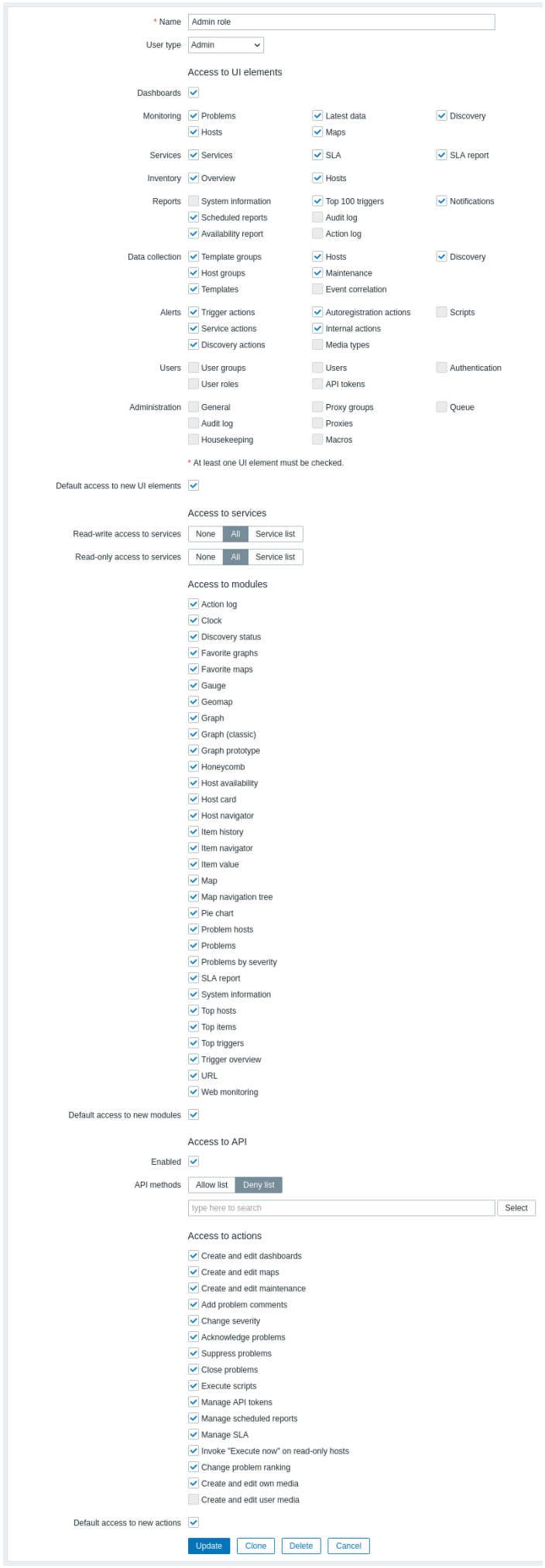


Questi si basano sui principali tipi di utente in Zabbix. Viene visualizzato l'elenco di tutti gli utenti assegnati al rispettivo ruolo. Gli utenti inclusi in gruppi disabilitati sono indicati in rosso. Il *ruolo Guest* è un ruolo di tipo utente con le sole autorizzazioni per visualizzare alcune sezioni 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.


* Name
Admin role
User type
Admin
Access to UI elements
Dashboards
Monitoring
Problems
Latest data
Discovery
Hosts
Maps
Services
Services
SLA
SLA report
Inventory
Overview
Hosts
Reports
System information
Top 100 triggers
Notifications
Scheduled reports
Audit log
Availability report
Action log
Data collection
Template groups
Hosts
Discovery
Host groups
Maintenance
Templates
Event correlation
Alerts
Trigger actions
Autoregistration actions
Scripts
Service actions
Internal actions
Discovery actions
Media types
Users
User groups
Users
Authentication
User roles
API tokens
Administration
General
Proxy groups
Queue
Audit log
Proxies
Housekeeping
Macros
* At least one UI element must be checked.
Default access to new UI elements
Access to services
Read-write access to services
None
All
Service list
Read-only access to services
None
All
Service list
Access to modules
Action log
Clock
Discovery status
Favorite graphs
Favorite maps
Gauge
Geomap
Graph
Graph (classic)
Graph prototype
Honeycomb
Host availability
Host card
Host navigator
Item history
Item navigator
Item value
Map
Map navigation tree
Pie chart
Problem hosts
Problems
Problems by severity
SLA report
System information
Top hosts
Top items
Top triggers
Trigger overview
URL
Web monitoring
Default access to new modules
Access to API
Enabled
API methods
Allow list
Deny list
type here to search
Select
Access to actions
Create and edit dashboards
Create and edit maps
Create and edit maintenance
Add problem comments
Change severity
Acknowledge problems
Suppress problems
Close problems
Execute scripts
Manage API tokens
Manage scheduled reports
Manage SLA
Invoke "Execute now" on read-only hosts
Change problem ranking
Create and edit own media
Create and edit user media
Default access to new actions
Update
Clone
Delete
Cancel

Vengono visualizzate le autorizzazioni disponibili. Per revocare una determinata autorizzazione, deselezionare la relativa casella di controllo.

Le autorizzazioni disponibili, insieme ai valori predefiniti per ciascun ruolo utente preconfigurato in Zabbix, sono descritte di seguito.

Permessi predefiniti

Accesso agli elementi dell'interfaccia utente

L'accesso predefinito alle sezioni del menu dipende dal tipo di utente. Vedere la pagina Permessi per i dettagli.

Accesso ad altre opzioni

Parametro	**Descrizione**	**Ruoli utente predefiniti**			
		Ruolo Super admin	*Ruolo Admin*	*Ruolo User*	*Ruolo Guest*
Accesso predefinito ai nuovi elementi dell'interfaccia utente	Questa opzione specifica come saranno accessibili le nuove sezioni del menu dopo un aggiornamento di Zabbix. Le sezioni di menu esistenti dei moduli rimangono invariate.	Sì	Sì	Sì	Sì
Accesso ai servizi					
Accesso in lettura-scrittura ai servizi	Selezionare l'accesso in lettura-scrittura ai servizi: **Nessuno** - nessun accesso **Tutti** - l'accesso a tutti i servizi è in lettura-scrittura **Elenco 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 ed è ereditato dinamicamente dai servizi figli.	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 corrispondenti al tag. Questa opzione è disponibile se è selezionato 'Elenco servizi' nel parametro *Accesso in lettura-scrittura ai servizi*. L'accesso in lettura-scrittura, se concesso, ha la precedenza sulle impostazioni di accesso in sola lettura ed è ereditato dinamicamente dai servizi figli.				
Accesso in sola lettura ai servizi	Selezionare l'accesso in sola lettura ai servizi: **Nessuno** - nessun accesso **Tutti** - l'accesso a tutti i servizi è in sola lettura **Elenco servizi** - selezionare i servizi per l'accesso in sola lettura L'accesso in sola lettura non ha la precedenza sull'accesso in lettura-scrittura ed è ereditato dinamicamente dai servizi figli.			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 corrispondenti al tag. Questa opzione è disponibile se è selezionato 'Elenco servizi' nel parametro *Accesso in sola lettura ai servizi*. L'accesso in sola lettura non ha la precedenza sull'accesso in lettura-scrittura ed è ereditato dinamicamente dai servizi figli.				
Accesso ai moduli					
<Nome modulo>	Consentire/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 che è 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 *Administration > General > Modules*.				
Accesso all'API					
Abilitato	Abilitare/disabilitare l'accesso all'API.	Sì	Sì	Sì	No
Metodi API	Selezionare *Elenco consentiti* per consentire, oppure *Elenco negati* per negare i metodi API specificati nel campo di ricerca. Si noti che 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 premere il pulsante Select e selezionare i metodi dall'elenco completo disponibile per questo tipo di utente. Si noti che se una determinata azione del blocco Accesso alle azioni è deselezionata, gli utenti non potranno utilizzare i metodi API relativi a tale azione. I caratteri jolly sono supportati. Esempi: `dashboard.*` (tutti i metodi del servizio API 'dashboard.') `*` (qualsiasi metodo), `*.export` (metodi con nome '.export' di tutti i servizi API). Se non è stato specificato alcun metodo, la regola *Elenco consentiti/negati* verrà ignorata.				
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 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	

Cambiare gravità Riconoscere 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 per gestire SLA.		
Richiamare "Execute now" su host in sola lettura	Consentire l'uso dell'opzione "Execute now" nei dati più recenti per gli item degli host in sola lettura.		Sì
Modificare la classificazione del problema	Consentire di modificare la classificazione del problema da causa a sintomo e viceversa.		
Creare e modificare i propri supporti di notifica	Consentire di creare/modificare i propri supporti di notifica.		
Creare e modificare i supporti di notifica degli utenti	Consentire di creare/modificare i supporti di notifica per gli utenti. Questa opzione è disponibile solo per gli utenti Super admin.	No	No
Accesso predefinito alle nuove azioni	Questa opzione specifica come saranno accessibili le nuove azioni dopo un aggiornamento di Zabbix.	Sì	Sì

Vedere anche:

- Configurazione di un utente

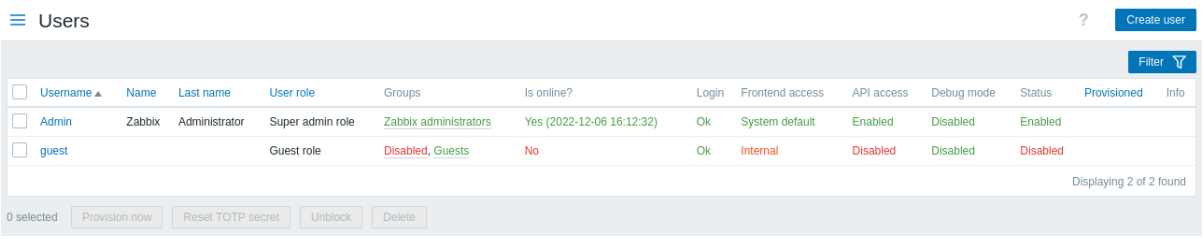
3 Utenti

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:

Column	Description
Username	Nome utente per accedere a Zabbix. Facendo clic sul nome utente si apre il modulo di configurazione dell'utente.
Name	Nome dell'utente.
Last name	Cognome dell'utente.
User role	Viene visualizzato il ruolo utente.
Groups	Sono elencati i gruppi di cui l'utente è membro. Facendo clic sul nome del gruppo utenti si apre il modulo di configurazione del gruppo utenti. I gruppi disabilitati sono visualizzati in rosso.
Is online?	Viene visualizzato lo stato online dell'utente: *Yes* oppure *No*. L'ora dell'ultima attività dell'utente viene visualizzata tra parentesi.
Login	Viene visualizzato lo stato di accesso dell'utente: *Ok* oppure *Blocked*. Un utente può essere temporaneamente bloccato se supera il numero di tentativi di accesso non riusciti impostato nella sezione *Administration → General → Other* (cinque per impostazione predefinita). Facendo clic su *Blocked* è possibile sbloccare l'utente.
Frontend access	Viene visualizzato il livello di accesso al frontend: *System default*, *Internal*, *LDAP* oppure *Disabled*, in base a quello impostato per l'intero gruppo utenti.
API access	Viene visualizzato lo stato dell'accesso API: *Enabled* oppure *Disabled*, in base a quello impostato per il ruolo utente.
Debug mode	Viene visualizzato lo stato della modalità di debug: *Enabled* oppure *Disabled*, in base a quello impostato per l'intero gruppo utenti.
Status	Viene visualizzato lo stato dell'utente: *Enabled* oppure *Disabled*, in base a quello impostato per l'intero gruppo utenti.
Provisioned	Viene visualizzata la data dell'ultimo provisioning dell'utente. Utilizzato per gli utenti creati tramite provisioning JIT da LDAP/SAML.
Info	Vengono visualizzate informazioni sugli errori. Per gli utenti senza gruppi utenti viene visualizzato un avviso giallo. Per gli utenti senza ruoli e per gli utenti senza ruoli e gruppi utenti viene visualizzato un avviso rosso.

Per configurare un nuovo utente, fare clic sul pulsante *Create user* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

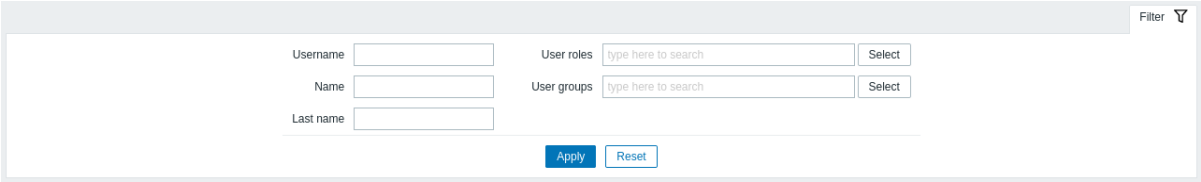
- *Provision now* - aggiorna le informazioni dell'utente da LDAP (questa opzione è abilitata solo se è selezionato un utente LDAP)
- *Reset TOTP secret* - reimposta i segreti TOTP dell'utente per tutti i metodi TOTP ed elimina la sessione dell'utente (questa opzione è abilitata solo se MFA è abilitato; per gli utenti senza segreti TOTP, la loro sessione non verrà eliminata)
- *Unblock* - riabilita l'accesso al sistema per gli utenti bloccati
- *Delete* - elimina gli utenti

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi utenti, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo gli utenti di proprio interesse. Per migliorare le prestazioni della ricerca, i dati vengono cercati con le macro non risolte.

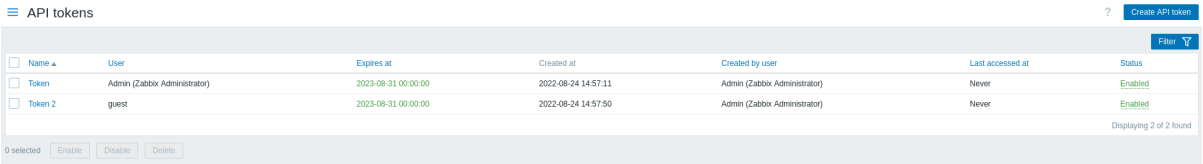
Il link *Filtro* è disponibile sopra l'elenco degli utenti. Facendo clic su di esso, diventa disponibile un filtro con cui è possibile filtrare gli utenti per nome utente, nome, cognome, ruolo utente e gruppo utenti.



4 Token API

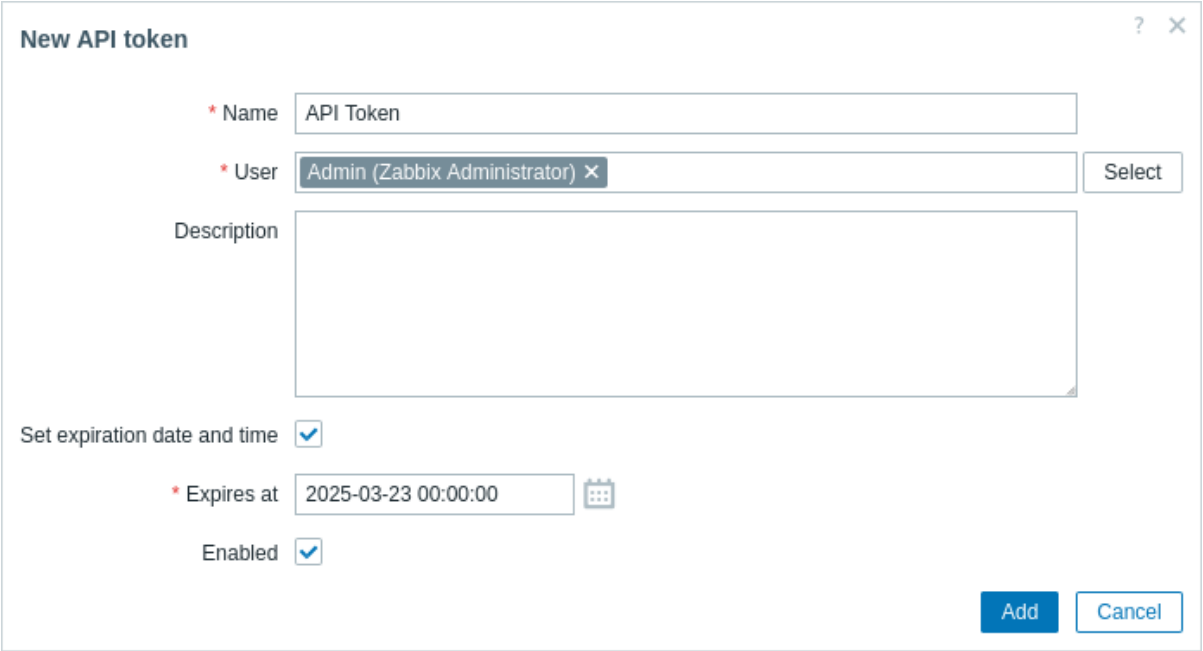
Panoramica

Questa sezione consente di creare e gestire i token API.



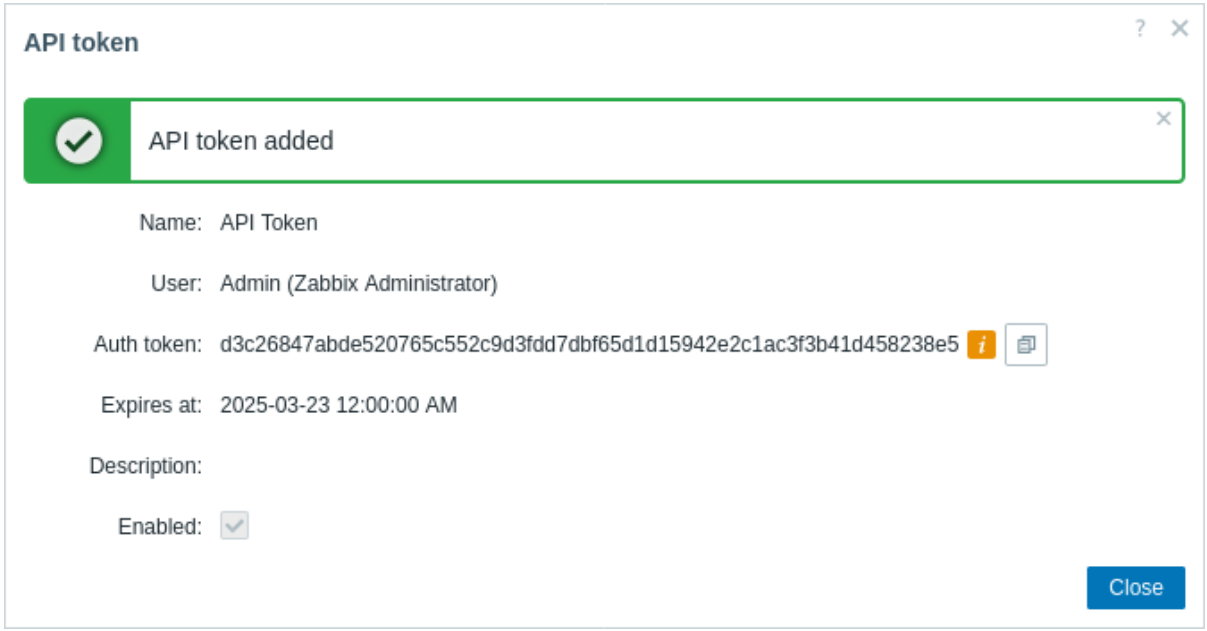
Puoi filtrare i token API per nome, utenti a cui i token sono assegnati, data di scadenza, utenti che hanno creato i token o stato (abilitato/disabilitato). Fai clic sullo stato del token nell'elenco per abilitare/disabilitare rapidamente un token. Puoi anche abilitare/disabilitare più token contemporaneamente selezionandoli nell'elenco e facendo quindi clic sui pulsanti *Enable/Disable* sotto l'elenco.

Per creare un nuovo token, premi il pulsante *Create API token* nell'angolo in alto a destra, quindi compila i campi richiesti nella schermata di configurazione del token:



Parameter	Description
Name	Nome visibile del token.
User	Utente a cui il token deve essere assegnato. Per selezionare rapidamente un utente, inizia a digitare il nome utente, il nome o il cognome, quindi seleziona l'utente richiesto dall'elenco di completamento automatico. In alternativa, puoi premere il 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	Deseleziona questa casella se un token non deve avere una data di scadenza.
Expiry date	Fai clic sull'icona del calendario per selezionare la data di scadenza del token oppure inserisci manualmente la data nel formato YYYY-MM-DD hh:mm:ss
Enabled	Deseleziona questa casella se devi creare un token in stato disabilitato.

Premi *Add* per creare un token.



Copia il valore di *Auth token* e salvalo in un luogo sicuro **prima di chiudere la pagina**, quindi premi Close. Il token apparirà nell'elenco.

Warning:
Il valore di *Auth token* non può essere visualizzato di nuovo in seguito. È disponibile solo immediatamente dopo la creazione di un token. Se perdi un token salvato, dovrai rigenerarlo e così facendo verrà creata una nuova stringa di autorizzazione.

Fai clic sul nome del token per modificare il nome, la descrizione, le impostazioni della data di scadenza o lo stato del token. Tieni presente che non è possibile cambiare l'utente a cui il token è assegnato. Premi il pulsante *Update* per salvare le modifiche. Se un token è stato perso o esposto, puoi premere il pulsante *Regenerate* per generare un nuovo valore del token. Verrà visualizzata una finestra di dialogo di conferma che ti chiederà di confermare questa operazione, poiché dopo aver proceduto il token generato in precedenza diventerà non valido.

Gli utenti senza accesso alla sezione del menu *Administration* possono vedere e modificare i dettagli dei token assegnati a loro nella sezione *User profile* → *API tokens* solo se *Manage API tokens* è consentito nelle autorizzazioni del loro ruolo utente.

5 Autenticazione

Panoramica

La sezione *Utenti* → *Autenticazione* consente di specificare il metodo di autenticazione utente per Zabbix e i requisiti della password interna.

I metodi di autenticazione disponibili sono autenticazione interna, HTTP, LDAP, SAML e MFA.

Autenticazione predefinita

Per impostazione predefinita, Zabbix utilizza l'autenticazione Zabbix **interna** per tutti gli utenti.

È possibile modificare il metodo di autenticazione predefinito a livello di sistema in LDAP. Per farlo, aprire la scheda *LDAP* e configurare i parametri LDAP, quindi tornare alla scheda *Authentication* e impostare il selettore *Default authentication* su LDAP.

Si noti che il metodo di autenticazione può essere regolato in modo più preciso a livello di gruppo utenti. Anche se l'autenticazione LDAP è impostata globalmente, alcuni gruppi utenti possono comunque essere autenticati da Zabbix. Per questi gruppi, l'accesso al frontend deve essere impostato su Internal.

È anche possibile abilitare l'autenticazione LDAP solo per specifici gruppi utenti, se a livello globale viene utilizzata l'autenticazione interna. In questo caso, i dettagli dell'autenticazione LDAP possono essere specificati e utilizzati per determinati gruppi utenti, per i quali l'accesso al frontend deve quindi essere impostato su LDAP. Se un utente appartiene ad almeno un gruppo utenti con autenticazione LDAP, questo utente non potrà utilizzare il metodo di autenticazione interna.

I metodi di autenticazione HTTP, SAML 2.0 e MFA possono essere utilizzati in aggiunta al metodo di autenticazione predefinito.

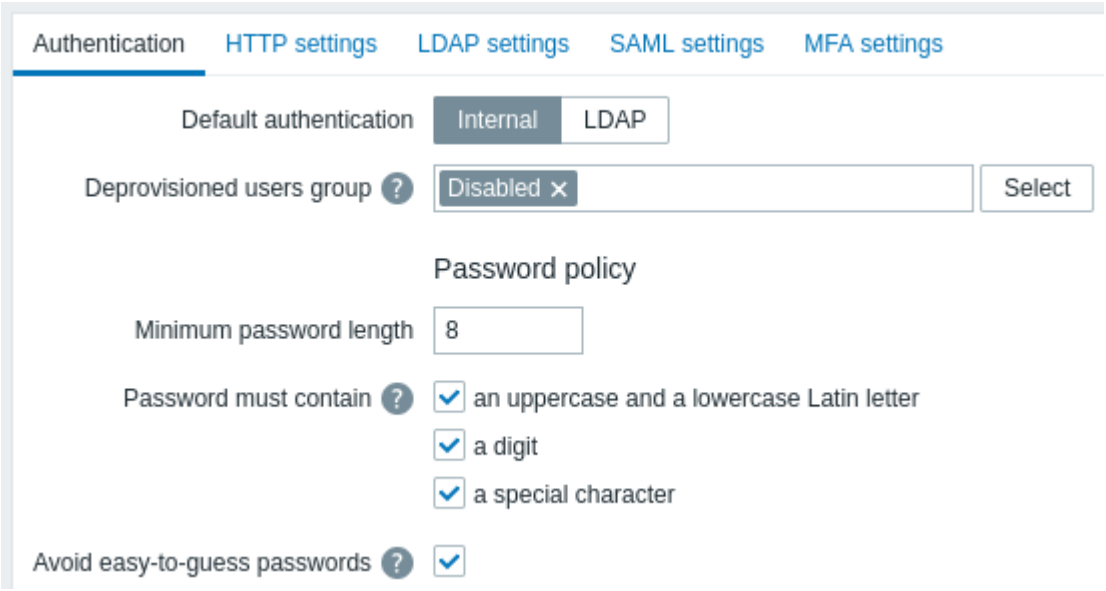
Zabbix supporta il provisioning just-in-time (JIT), che consente di creare account utente in Zabbix al primo accesso di autenticazione di un utente esterno e di effettuare il provisioning di tali account. Il provisioning JIT è supportato per LDAP e SAML.

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 deprovisionati e definire i requisiti di complessità della password per gli utenti di Zabbix.



Parametri di configurazione:

Parameter	Description
Autenticazione predefinita	Selezionare il metodo di autenticazione predefinito per Zabbix: *Interna* oppure *LDAP*.
Gruppo utenti deprovisionati	Specificare un gruppo utenti per gli utenti deprovisionati. Questa impostazione è richiesta solo per il provisioning JIT, relativamente agli utenti creati in Zabbix da sistemi LDAP o SAML, ma che non devono più essere provisionati. Deve essere specificato un gruppo utenti disabilitato.
Lunghezza minima della password	Per impostazione predefinita, la lunghezza minima della password è 8. Intervallo supportato: 1-70. Si noti che 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 in una password: - una lettera latina maiuscola e una minuscola - una cifra - un carattere speciale Passare il puntatore sul punto interrogativo per visualizzare un suggerimento con l'elenco dei caratteri per ciascuna opzione.
Evitare password facili da indovinare	Se selezionato, 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 a partire dall'elenco NCSC "Top 100k passwords", dall'elenco SecLists "Top 1M passwords" 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 della password non influiranno sulle password esistenti degli utenti, ma se un utente esistente sceglie di cambiare password, la nuova password dovrà soddisfare i requisiti correnti. Un suggerimento con l'elenco dei requisiti verrà visualizzato accanto al campo *Password* nel user profile e nel modulo di configurazione utente accessibile dal menu *Utenti* → *Utenti*.

1 HTTP

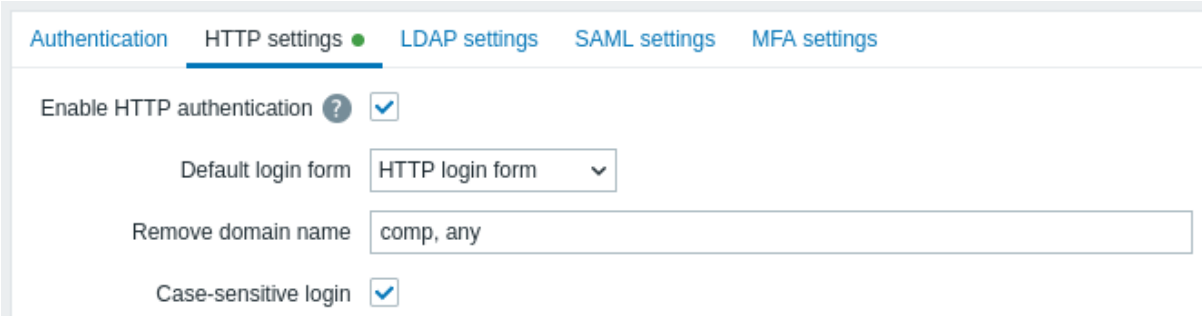
Panoramica

L'autenticazione HTTP o basata su server web (ad esempio: BasicAuthentication, NTLM/Kerberos) può essere utilizzata per verificare nomi utente e password. Si noti che un utente deve esistere anche in Zabbix, tuttavia la sua password Zabbix non verrà utilizzata.

Attention:
Attenzione! Assicurarsi che l'autenticazione tramite server web sia configurata e funzioni correttamente prima di attivarla.

L'autenticazione HTTP può essere disabilitata nel frontend configurando l'opzione corrispondente nella scheda *Impostazioni HTTP* nella sezione *Utenti* > *Autenticazione*. Quando l'autenticazione HTTP è disabilitata, la scheda con le opzioni di autenticazione HTTP non verrà visualizzata nel frontend. Tieni presente che reinstallando il frontend (eseguendo setup.php) verranno reimpostate le impostazioni di autenticazione, inclusa la configurazione dell'autenticazione HTTP.

Configurazione



Parametri di configurazione:

Parametro	Descrizione
Abilita autenticazione HTTP	Selezionare la casella per abilitare l'autenticazione HTTP. Passando il mouse su ? verrà visualizzato un suggerimento che avvisa che, nel caso di autenticazione del web server, tutti gli utenti (anche con accesso frontend impostato su LDAP/Internal) verranno autenticati dal web server, non da Zabbix.
Modulo di accesso predefinito	Specificare se indirizzare gli utenti non autenticati a: **Modulo di accesso Zabbix** - pagina di accesso standard di Zabbix. **Modulo di accesso HTTP** - pagina di accesso HTTP. Si consiglia di abilitare l'autenticazione basata su 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 login utente valido nella variabile `$_SERVER`. Le chiavi `$_SERVER` supportate sono `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Rimuovi nome di dominio	Un elenco di nomi di dominio separati da virgole che devono essere rimossi dal nome utente. Ad es. `comp,any` - se il nome utente è 'Admin@any', '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 per disabilitare l'accesso con distinzione tra maiuscole e minuscole per i nomi utente (abilitato per impostazione predefinita). La disabilitazione dell'accesso con distinzione tra maiuscole e minuscole consente, ad esempio, di accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Si noti che se l'accesso con distinzione tra maiuscole e minuscole è disabilitato 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: "Authentication failed: supplied credentials are not unique."

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, potrebbe essere utile aggiungere una riga `ErrorDocument 401 /index.php?form=default` alle direttive di autenticazione di base, che reindirizzerà al normale modulo di accesso di Zabbix.

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 utenti

È possibile configurare il **provisioning** JIT (just-in-time) degli 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 LDAP e la password, Zabbix controlla se questo utente esiste nel server LDAP *predefinito*. Se l'utente esiste e non ha ancora un account in Zabbix, viene creato un nuovo utente in Zabbix e l'utente può accedere.

Un utente creato tramite provisioning JIT viene associato al server LDAP (directory) impostato come predefinito al momento della creazione. La modifica successiva del server LDAP predefinito non cambia né aggiorna il server LDAP collegato agli utenti che sono già stati sottoposti a provisioning.

Attention:
Se il provisioning JIT è abilitato, nella scheda *Authentication* deve essere specificato un gruppo utenti per gli utenti deprovisionati.

Il provisioning JIT consente anche di aggiornare gli account utente sottoposti a provisioning in base alle modifiche in LDAP. Ad esempio, se un utente viene spostato da un gruppo LDAP a un altro, anche l'utente verrà spostato da un gruppo a un altro in Zabbix; se un utente viene rimosso da un gruppo LDAP, 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. Gli account utente sottoposti a provisioning vengono aggiornati in base al periodo di provisioning configurato oppure 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 oppure ha almeno una pagina del frontend aperta nel browser. Non esistono processi in background dedicati al provisioning degli utenti.

LDAP supporta tre modalità di bind alla directory per l'autenticazione e le ricerche:

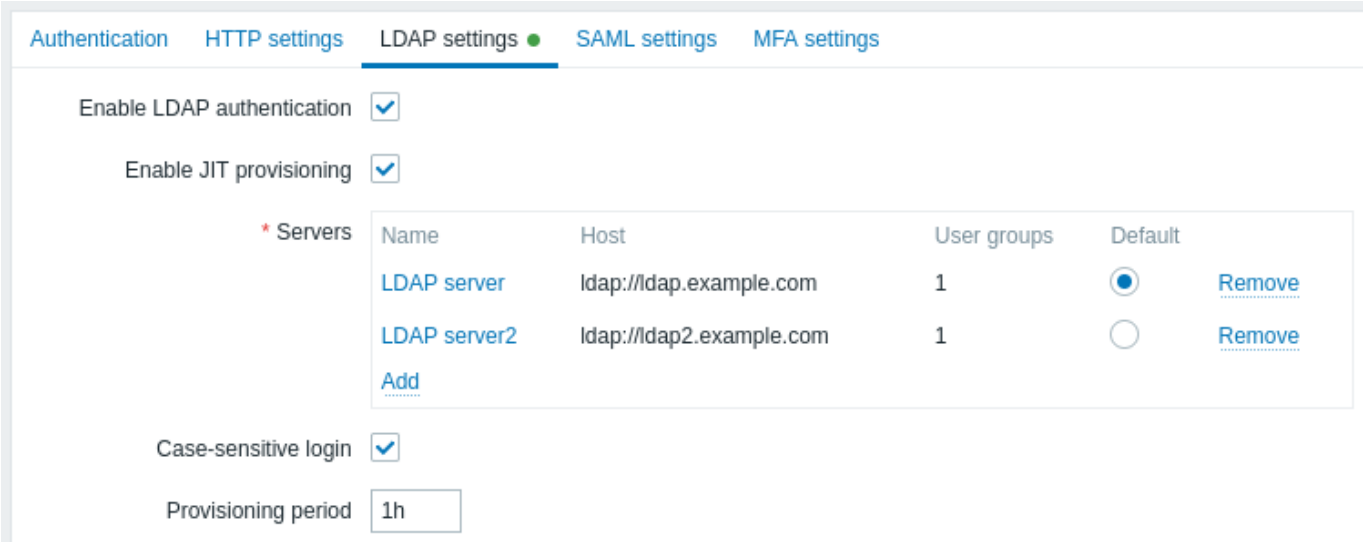
- Bind anonimo — non viene fornito alcun *Bind DN* / *Bind password* e il server LDAP consente query anonime.
- Utente di bind dedicato (account di servizio) — un account LDAP specifico viene impostato in *Bind DN* / *Bind password* e viene utilizzato da Zabbix per la ricerca e il provisioning. Questa è l'opzione consigliata e più flessibile, perché Zabbix può eseguire ricerche e provisioning in background senza le credenziali dell'utente finale.
- Bind diretto dell'utente — Zabbix esegue il bind utilizzando le credenziali inserite dall'utente al momento dell'accesso (senza *Bind DN* / *Bind password* configurati); 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 il login interattivo. Di conseguenza, le azioni di provisioning che richiedono l'autenticazione a LDAP al di fuori della sessione di login dell'utente (ad esempio, l'uso del pulsante *Provision now* o l'esecuzione del provisioning in background quando l'utente non ha effettuato attivamente l'accesso) non possono autenticarsi e quindi non funzioneranno. Il provisioning e gli aggiornamenti per il bind diretto dell'utente avvengono 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.

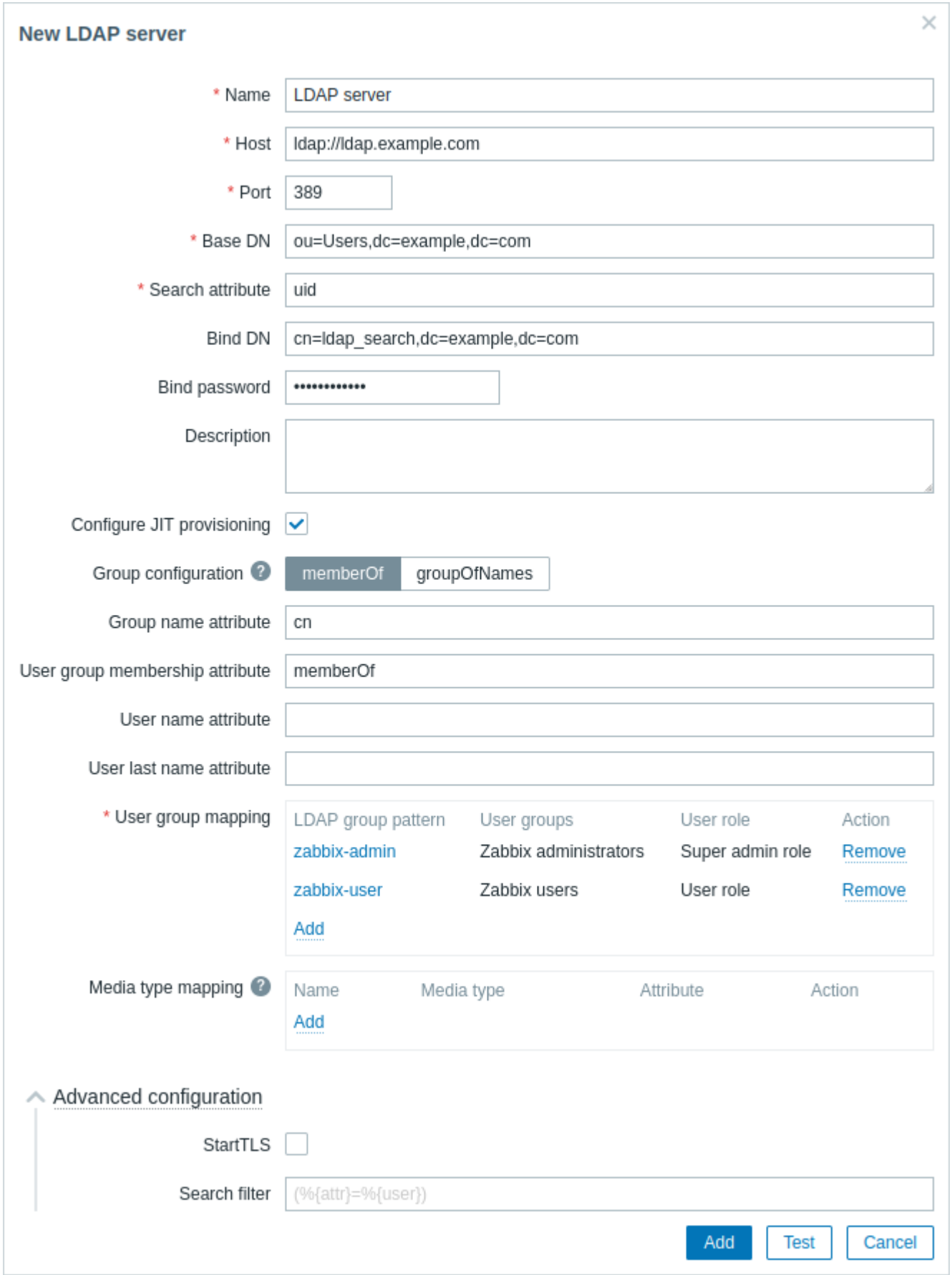
Configurazione



Parametri di configurazione:

Parametro	Descrizione
Abilita autenticazione LDAP	Selezionare la casella per abilitare l'autenticazione LDAP.
Abilita provisioning JIT	Selezionare la casella 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 per disabilitare l'accesso con distinzione tra maiuscole e minuscole per i nomi utente (abilitato per impostazione predefinita). La disabilitazione dell'accesso con distinzione tra maiuscole e minuscole consente, ad esempio, di accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Si noti che, se l'accesso con distinzione tra maiuscole e minuscole è disabilitato 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, cioè la frequenza con cui l'utente che ha effettuato l'accesso verrà sottoposto a provisioning mentre lavora con il frontend.

Configurazione del server LDAP



Parametri di configurazione del server LDAP:

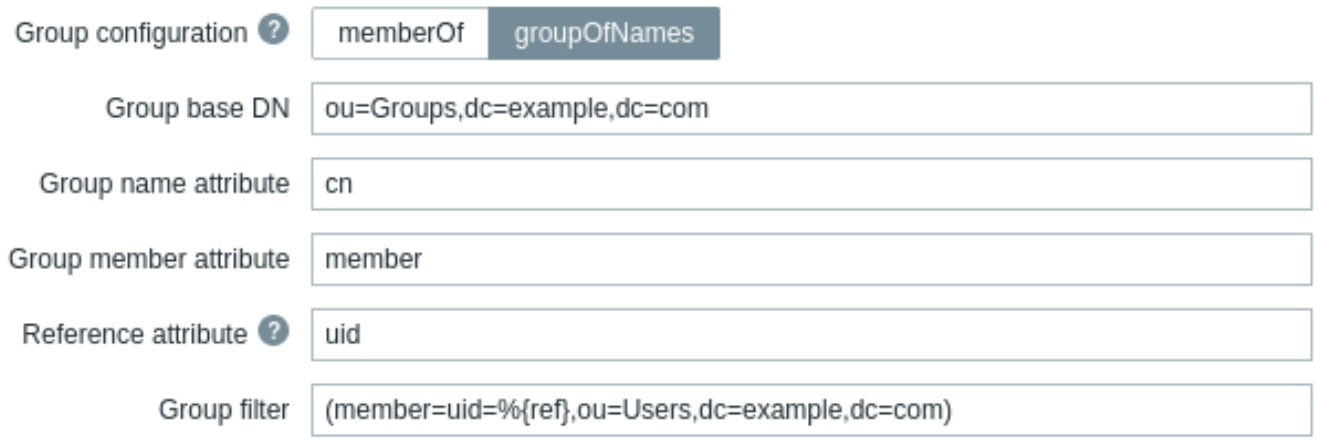
Parametro	Descrizione
Name	Nome del server LDAP nella configurazione di Zabbix.

Parametro	Descrizione
Host	Hostname, IP o URI del server LDAP. Esempi: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Per un server LDAP sicuro, utilizzare il protocollo *ldaps* e l'hostname. Esempio: ldaps://ldap.example.com Con OpenLDAP 2.x.x e versioni successive, può essere utilizzato un URI LDAP completo nel formato ldap://hostname:port o 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 utilizzata quando si usano URI LDAP completi.
Base DN	Percorso di base agli 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 utilizzato per la ricerca: uid (per OpenLDAP), sAMAccountName (per Microsoft Active Directory)
Bind DN	Account LDAP per il binding e la ricerca nel 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 può potenzialmente esporre la configurazione del dominio a utenti non autorizzati (informazioni su utenti, computer, server, gruppi, servizi, ecc.). Per motivi di sicurezza, disabilitare i binding anonimi sugli host LDAP e utilizzare invece l'accesso autenticato.
Bind password	Password LDAP dell'account per il binding e la ricerca nel server LDAP.
Description	Descrizione del server LDAP.
Configure JIT provisioning	Selezionare questa casella per mostrare le opzioni relative al provisioning JIT.
Group configuration	Selezionare il metodo di configurazione del gruppo: **memberOf** - tramite la ricerca degli utenti e del loro attributo di appartenenza al gruppo **groupOfNames** - tramite la ricerca dei gruppi attraverso 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 le informazioni sui gruppi a cui appartiene l'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	Mappare un pattern di gruppo utente LDAP a un gruppo utenti Zabbix e a un ruolo utente. Questo è necessario per determinare quale gruppo utenti/ruolo riceverà in Zabbix l'utente sottoposto a provisioning. 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 utenti Zabbix, l'utente diventa membro di tutti. Se un utente corrisponde a più ruoli utente Zabbix, riceverà quello con il livello di autorizzazione più alto tra essi.
Media type mapping	Mappare gli attributi LDAP media dell'utente (ad esempio email) ai supporti utente Zabbix per l'invio delle notifiche (il valore dell'attributo viene utilizzato come campo *Send to* del supporto).
Advanced configuration	Fare clic sull'etichetta *Advanced configuration* per visualizzare le opzioni di configurazione avanzata (vedere sotto).
StartTLS	Selezionare la casella per utilizzare l'operazione StartTLS durante la connessione al server LDAP. La connessione fallirà se il server non supporta StartTLS. StartTLS non può essere utilizzato con server che usano il protocollo *ldaps*.

Parametro	Descrizione
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 è personalizzato, LDAP utilizzerà 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* (per esempio,*uid=%{user},dc=example,dc=com*) e lasciare vuoti i parametri *BindDN* e *Bind password*. Durante l'autenticazione, il segnaposto %{user} verrà sostituito con il nome utente inserito durante il login.
 Con il binding diretto dell'utente Zabbix ha accesso alle credenziali dell'utente solo durante il login interattivo. Pertanto, le attività di provisioning che operano senza un login interattivo (per esempio, il pulsante *Provision now* o il metodo API `user.provision`) ignoreranno gli utenti che si autenticano usando il binding diretto dell'utente perché Zabbix non può effettuare il binding per loro conto. Utilizzare 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 che l'utente finale abbia effettuato l'accesso.

I seguenti campi sono specifici di "groupOfNames" come metodo *Group configuration*:



Parametro	Descrizione
Group base DN	Percorso di base ai gruppi nel server LDAP.
Group name attribute	Specificare l'attributo da cui ottenere il nome del gruppo nel percorso di base ai gruppi specificato. Il nome del gruppo è necessario per la mappatura dei gruppi utente.
Group member attribute	Specificare l'attributo che contiene le 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 utilizzare `%{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 nel server LDAP con privilegi minimi in LDAP invece di utilizzare 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 questo è il nome dell'account *ldap_search*.

Verifica dell'accesso

Il pulsante *Test* consente di verificare l'accesso dell'utente:

Parametro	Descrizione
Login	Nome utente LDAP da verificare (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 prova.
User password	Password dell'utente LDAP da verificare.

3 SAML

Panoramica

L'autenticazione SAML 2.0 può essere utilizzata per accedere a Zabbix.

Se è configurato solo l'accesso tramite SAML, l'utente deve comunque esistere anche in Zabbix, tuttavia la sua password di Zabbix non verrà utilizzata. Se l'autenticazione ha esito positivo, Zabbix confronterà un nome utente locale con l'attributo nome utente restituito da SAML.

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

Configurazione di Zabbix

Attention:
È necessario installare php-openssl se si desidera utilizzare l'autenticazione SAML nel frontend.

Per utilizzare l'autenticazione SAML, Zabbix deve essere configurato nel seguente modo:

1. Fornire la chiave privata SP e i certificati. Per attivare SAML tramite il frontend, è necessario fornire un certificato IdP (incollando il testo o selezionando un file). Se sono selezionate le opzioni *Sign* o *Encrypt*, è necessario fornire un certificato SP e una chiave privata SP.

Se in zabbix.conf.php è impostato `$SSO['CERT_STORAGE'] = 'database'`, durante la configurazione SAML è possibile incollare il testo del certificato o caricare il file del certificato nel frontend — non sono richiesti file nel filesystem.

Se in zabbix.conf.php è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere reso disponibile nel filesystem (per impostazione predefinita in ui/conf/certs o nel percorso configurato in zabbix.conf.php) e il frontend non memorizzerà i certificati nel database. Si noti che se `$SSO['CERT_STORAGE']` non è impostato o è commentato, si presume l'archiviazione su file.

Per impostazione predefinita, Zabbix cercherà nelle seguenti posizioni:

- 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 file. Sebbene in alcuni ambienti siano consentite importazioni non crittografate per comodità, è fortemente consigliato utilizzare un trasporto crittografato (ad esempio, abilitare HTTPS/TLS per il frontend di Zabbix). Quando i certificati/chiavi SAML sono memorizzati nel database, abilitare TLS/SSL per le connessioni al database per proteggere il traffico tra l'applicazione e il database.
Crittografare i backup del database che possono 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 di Zabbix. Tuttavia, è possibile specificare impostazioni aggiuntive nel file di configurazione.



Parametri di configurazione disponibili nel frontend di Zabbix:

Parameter	Description
Enable SAML authentication	Selezionare la casella di controllo per abilitare l'autenticazione SAML.
Enable JIT provisioning	Selezionare la casella di controllo per abilitare il provisioning utenti JIT.
IDP entity ID	L'identificatore univoco dell'entità all'interno dell'identity provider SAML.
SSO service URL	L'URL a cui gli utenti verranno reindirizzati durante l'accesso.
SLO service URL	L'URL a cui gli utenti verranno reindirizzati durante la disconnessione. Se lasciato vuoto, il servizio SLO non verrà utilizzato.

Parameter	Description
Username attribute	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
SP entity ID	L'identificatore univoco del service provider (se non corrisponde, l'operazione verrà rifiutata). È possibile specificare un URL o qualsiasi stringa di dati.
SP name ID format	Richiede un particolare formato name ID 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
IdP certificate	Contenuto del certificato del service provider (SP) per la configurazione del servizio SAML single sign-on (SSO).
SP private key	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).
SP certificate	Contenuto del certificato del service provider (SP) per la configurazione del servizio SAML single sign-on (SSO).
Sign	Selezionare le caselle di controllo per scegliere le entità per cui abilitare la firma SAML: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Selezionare le caselle di controllo per scegliere le entità per cui abilitare la crittografia SAML: *Name ID* *Assertions*
Case-sensitive login	Deselezionare la casella di controllo per disabilitare l'accesso con distinzione tra maiuscole e minuscole per i nomi utente (abilitato per impostazione predefinita). Disabilitando l'accesso con distinzione tra maiuscole e minuscole, è possibile, ad esempio, accedere come "admin" anche se l'utente Zabbix è "Admin" o "ADMIN". Si noti che se l'accesso con distinzione tra maiuscole e minuscole è disabilitato 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: "Authentication failed: supplied credentials are not unique."
Configure JIT provisioning	Selezionare questa casella di controllo per mostrare le opzioni relative al provisioning utenti JIT.
Group name attribute	Specificare l'attributo del nome del gruppo per il provisioning utenti JIT.
User name attribute	Specificare l'attributo del nome dell'utente per il provisioning utenti JIT.
User last name attribute	Specificare l'attributo del cognome dell'utente per il provisioning utenti JIT.
User group mapping	Associare un pattern di gruppo utente SAML a un gruppo utenti Zabbix e a un ruolo utente. Questo è necessario per determinare quale gruppo utenti/ruolo verrà assegnato all'utente sottoposto a provisioning in Zabbix. Fare clic su *Add* per aggiungere un'associazione. Il campo *SAML group pattern* supporta i caratteri jolly. Il nome del gruppo deve corrispondere a un gruppo esistente. Se un utente SAML corrisponde a più gruppi utenti Zabbix, l'utente diventa membro di tutti essi. Se un utente corrisponde a più ruoli utente Zabbix, all'utente verrà assegnato il livello di autorizzazione più elevato tra essi.
Media type mapping	Associare gli attributi media SAML dell'utente (ad esempio email) ai media utente di Zabbix per l'invio di notifiche.
Enable SCIM provisioning	Selezionare questa casella di controllo per abilitare il provisioning SCIM 2.0.

Vedere esempi di configurazione degli identity provider SAML per l'accesso e il provisioning utenti in Zabbix con:

- Microsoft Azure AD
- Okta
- Onelogin

Note sul provisioning SCIM

Per il provisioning SCIM, specificare il percorso del frontend di Zabbix e aggiungervi api_scim.php dal lato del provider di identità, ad esempio:

`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 essere uguale allo schema utente: `urn:ietf:params:scim:schemas:core:2.0:User`.

Impostazioni avanzate

Parametri SAML aggiuntivi possono essere configurati nel file di configurazione del frontend di Zabbix (*zabbix.conf.php*):

- $SSO['SP_KEY'] = '<percorso del file della chiave privata SP>';
- $SSO['SP_CERT'] = '<percorso del file del certificato SP>';
- $SSO['IDP_CERT'] = '<percorso del file del certificato IDP>';
- $SSO['SETTINGS']

Note:
L'array `$SSO['SETTINGS']` deve seguire la stessa struttura prevista dalla libreria *SAML PHP Toolkit* (fornita con Zabbix). Per una descrizione completa delle opzioni di configurazione disponibili, fare riferimento alla documentazione ufficiale della libreria.

Solo le seguenti opzioni possono essere impostate come parte di $SSO['SETTINGS']:

- *strict*
- *baseurl*
- *compress*
- *contactPerson*
- *organization*
- *sp* (solo le opzioni specificate in questo elenco)
  - *attributeConsumingService*
  - *x509certNew*
- *idp* (solo le opzioni specificate in questo elenco)
  - *singleLogoutService* (solo un'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 verranno prese dal database e non potranno essere sovrascritte. L'opzione *debug* verrà ignorata.

Inoltre, se la UI di Zabbix si trova dietro un proxy o un load balancer, può essere utilizzata l'opzione personalizzata *use_proxy_headers*:

- *false* (predefinito) - ignora l'opzione;
- *true* - utilizza gli header HTTP X-Forwarded-* per costruire l'URL di base.

Se si utilizza un load balancer per connettersi all'istanza Zabbix, dove il load balancer usa TLS/SSL e Zabbix no, è necessario indicare 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 di configurazione:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Configurazione del frontend con Kerberos/ADFS

Il file di configurazione del frontend di Zabbix (*zabbix.conf.php*) può essere utilizzato per configurare l'SSO con autenticazione Kerberos e 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
```

4 MFA

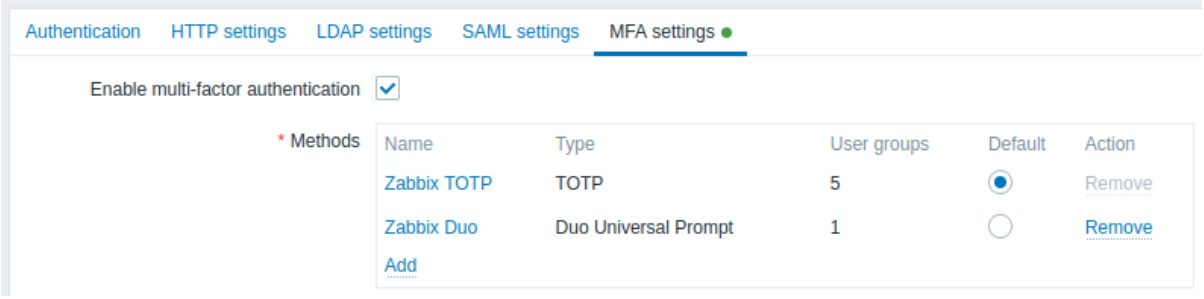
Panoramica

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.

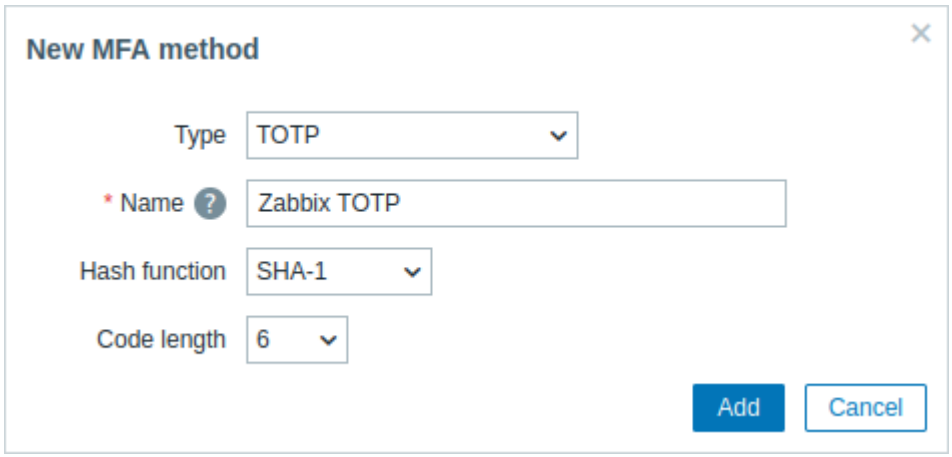
Configurazione



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.
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. Accedere alle impostazioni MFA in Zabbix in *Users* → *Authentication* e abilitare l'autenticazione a più fattori.
2. Aggiungere un nuovo metodo MFA con la seguente configurazione:

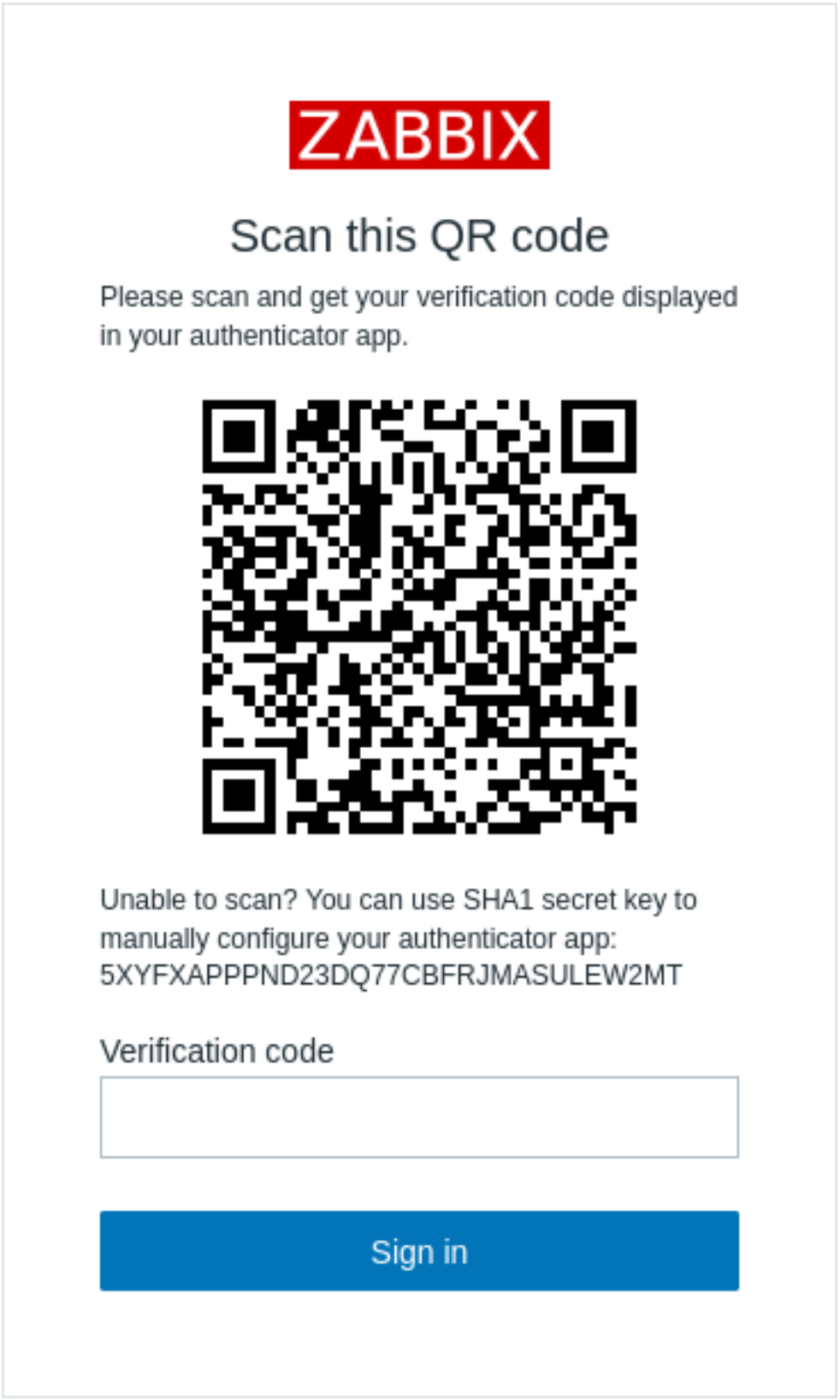
- Tipo: TOTP
- Nome: Zabbix TOTP
- Funzione hash: SHA-1
- Lunghezza del codice: 6

3. Fare clic sul pulsante *Add* e quindi sul pulsante *Update*.

4. Andare in *Users* → *User groups* e creare un nuovo gruppo utenti con la seguente configurazione:

- Nome del gruppo: Gruppo TOTP
- Utenti: Admin
- Autenticazione a più fattori: Predefinita (oppure "Zabbix TOTP" se non è impostato come predefinito)

5. Disconnettersi da Zabbix e accedere nuovamente utilizzando le proprie credenziali. Dopo un accesso riuscito, verrà richiesto di effettuare l'iscrizione all'MFA, con la visualizzazione di un codice QR e di una chiave segreta.



6. Scansionare il codice QR oppure inserire la chiave segreta nell'app Google Authenticator. L'app genererà un codice di verifica da inserire per completare il processo di accesso.

7. Per gli accessi successivi, recuperare il codice di verifica dall'app Google Authenticator e inserirlo durante l'accesso.

Prompt universale Duo

Per il Prompt universale Duo, gli utenti devono verificare la propria identità utilizzando l'app di autenticazione Duo Mobile.

> **Attention:**
> Il metodo MFA Prompt universale Duo 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 server web, 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 in *Users* → *Authentication* e abilita l'autenticazione a più fattori.

5. Aggiungi un nuovo metodo MFA con la seguente configurazione:

   - Tipo: Prompt universale Duo
   - Nome: 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 utenti con la seguente configurazione:

   - Nome gruppo: Gruppo Duo
   - Utenti: Admin
   - Autenticazione a più fattori: Predefinita (oppure "Zabbix Duo" se non è impostato come predefinito)

8. Esci da Zabbix e accedi di nuovo utilizzando le tue credenziali. Dopo un accesso riuscito, ti verrà chiesto di effettuare l'iscrizione all'MFA e verrai reindirizzato a Duo. Completa la configurazione di Duo e verifica il tuo utente con l'app Duo del telefono per accedere.

9. Per gli accessi successivi, utilizza il metodo MFA appropriato fornito dall'app Duo (ad esempio recuperando un codice di verifica, rispondendo alle notifiche push o utilizzando chiavi hardware) e inserisci le informazioni richieste durante l'accesso.

9 Amministrazione

Panoramica

Il menu Administration è dedicato alle funzioni amministrative di Zabbix. Questo menu è disponibile solo per gli utenti del tipo SuperAdmin.

1. Generale

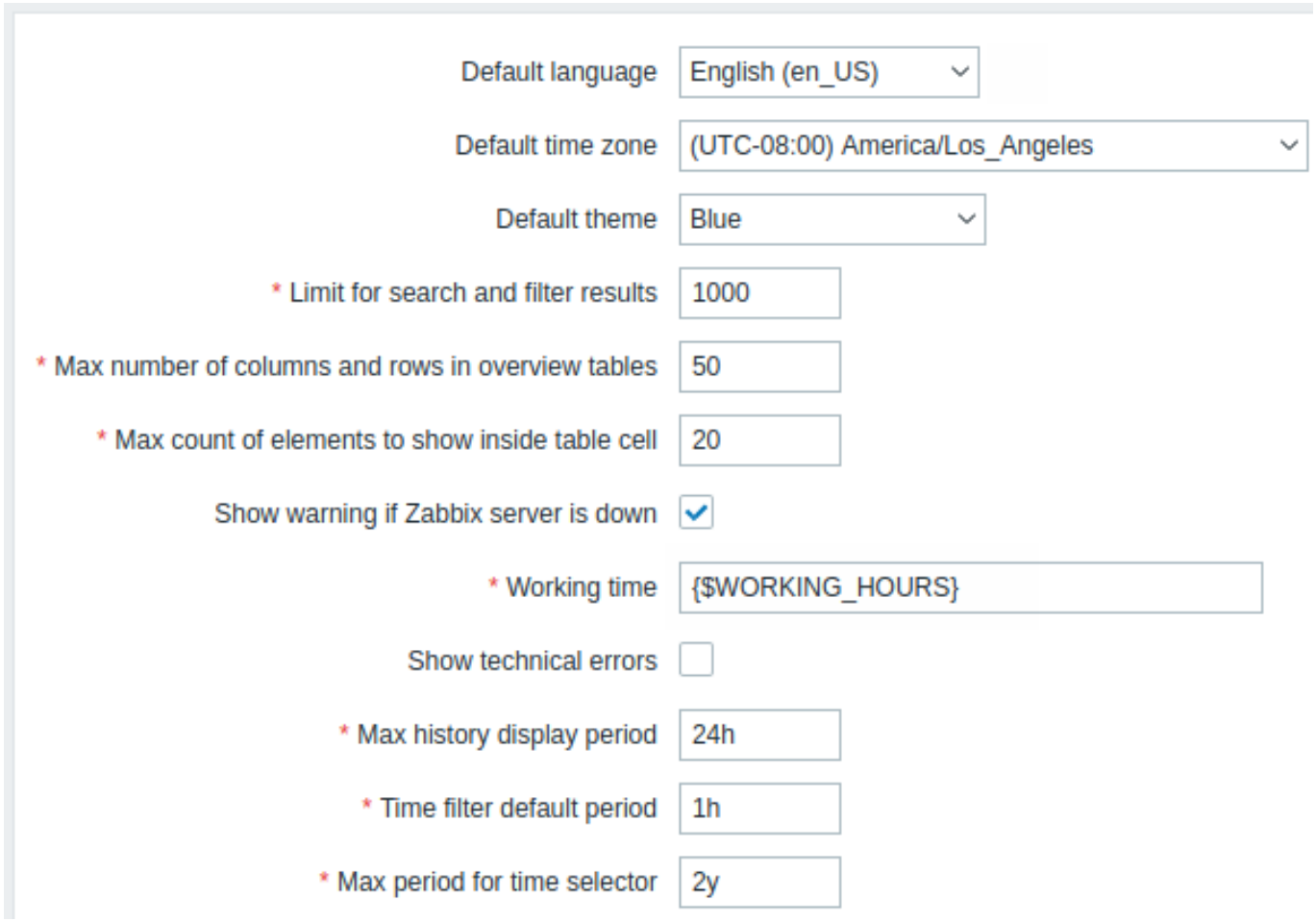
Panoramica

La sezione *Amministrazione* → *Generale* contiene diverse sottosezioni per impostare i valori predefiniti relativi al frontend e personalizzare Zabbix.

L'elenco delle sottosezioni disponibili viene visualizzato facendo clic su *Generale* nella sezione di menu *Amministrazione*. È inoltre possibile passare da una sottosezione all'altra utilizzando il menu a discesa del titolo nell'angolo superiore sinistro.

GUI

Questa sezione consente di personalizzare diversi valori predefiniti relativi al frontend.



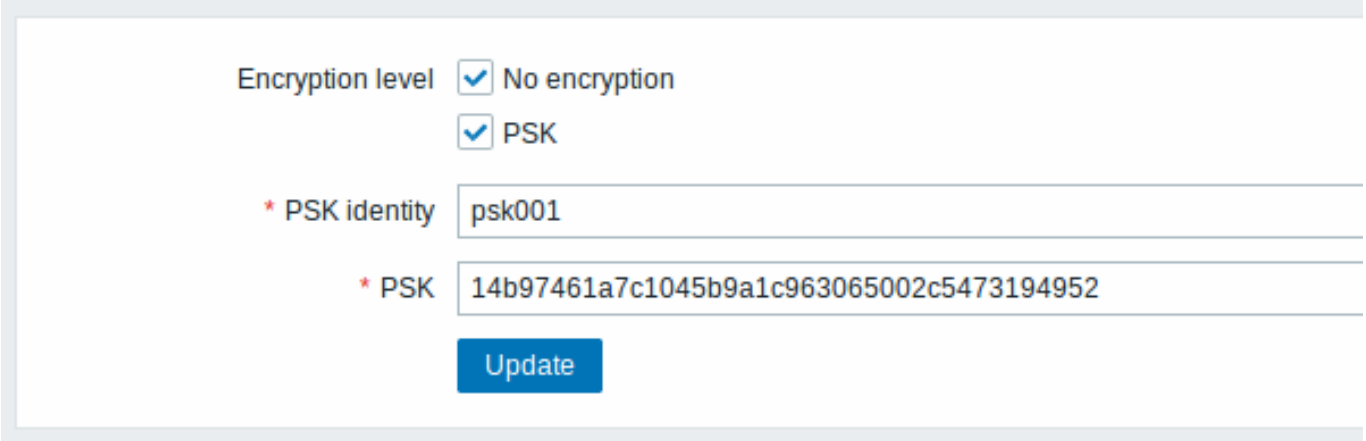
Parametri di configurazione:

Parameter	Description
Lingua predefinita	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.
Fuso orario predefinito	Fuso orario predefinito per gli utenti che non hanno specificato un fuso orario nei propri profili e per gli utenti guest.
Tema predefinito	Tema predefinito per gli utenti che non hanno specificato un tema nei propri profili e per gli utenti guest.
Limite per i risultati di ricerca e filtro	Numero massimo di elementi (righe) che verranno visualizzati in un elenco dell'interfaccia web, ad esempio in *Raccolta dati > Host*. *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 ci sono comunque più di 50 corrispondenze, verranno visualizzate solo le prime 50. Si noti che l'aumento del valore di questo parametro può comportare una riduzione delle prestazioni e un aumento del consumo di memoria lato web server.
Numero massimo di colonne e righe nelle tabelle di panoramica	Numero massimo di colonne e righe da visualizzare nel widget dashboard *Panoramica trigger*. Lo stesso limite si applica sia alle colonne che alle righe. Se esistono più righe e/o colonne di quelle mostrate, il sistema visualizzerà un avviso nella parte inferiore della tabella: "Not all results are displayed. Please provide more specific search criteria."
Numero massimo di elementi da mostrare all'interno della cella della tabella	Per le voci visualizzate in una singola cella della tabella, non ne verranno mostrate più di quelle configurate qui.
Mostra avviso se Zabbix server non è raggiungibile	Questo parametro abilita la visualizzazione di un messaggio di avviso in una finestra del browser se Zabbix server non può essere raggiunto (potrebbe essere non attivo). Il messaggio rimane visibile anche se l'utente scorre la pagina verso il basso. Passandoci sopra con il puntatore, il messaggio viene temporaneamente nascosto per rivelare il contenuto sottostante.

Parameter	Description
Orario di lavoro	Questo parametro a livello di sistema definisce l'orario di lavoro. Nei grafici, l'orario di lavoro viene 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.
Mostra errori tecnici	Se selezionato, tutti gli utenti registrati potranno vedere gli errori tecnici (PHP/SQL). Se non selezionato, le informazioni saranno disponibili solo per gli Zabbix Super Admin e per gli utenti appartenenti ai gruppi utenti con modalità di debug abilitata.
Periodo massimo di visualizzazione dello storico	Periodo di tempo massimo per il quale visualizzare i dati storici in *Monitoraggio* > *Ultimi dati*, le informazioni sullo scenario web dell'host in *Monitoraggio* > *Host* e nel widget dashboard *Top items*. Intervallo consentito: da 24 ore (predefinito) a 1 settimana. Sono supportati i suffissi temporali, ad esempio 1w (una settimana), 36h (36 ore).
Periodo predefinito del filtro temporale	Periodo di tempo da utilizzare per impostazione predefinita nei grafici e nelle dashboard. Intervallo consentito: da 1 minuto a 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 utente, sostituendo il valore predefinito globale o una precedente selezione dell'utente.
Periodo massimo per il selettore temporale	Periodo di tempo massimo disponibile per grafici e dashboard. Gli utenti non potranno visualizzare dati meno recenti. Intervallo consentito: da 1 anno a 10 anni (predefinito: 2 anni). Sono supportati i suffissi temporali, ad esempio 1y (un anno), 365w (365 settimane).

Registrazione automatica

In questa sezione è possibile configurare il livello di crittografia per la registrazione automatica dell'agent attivo.



I parametri contrassegnati con un asterisco sono obbligatori.

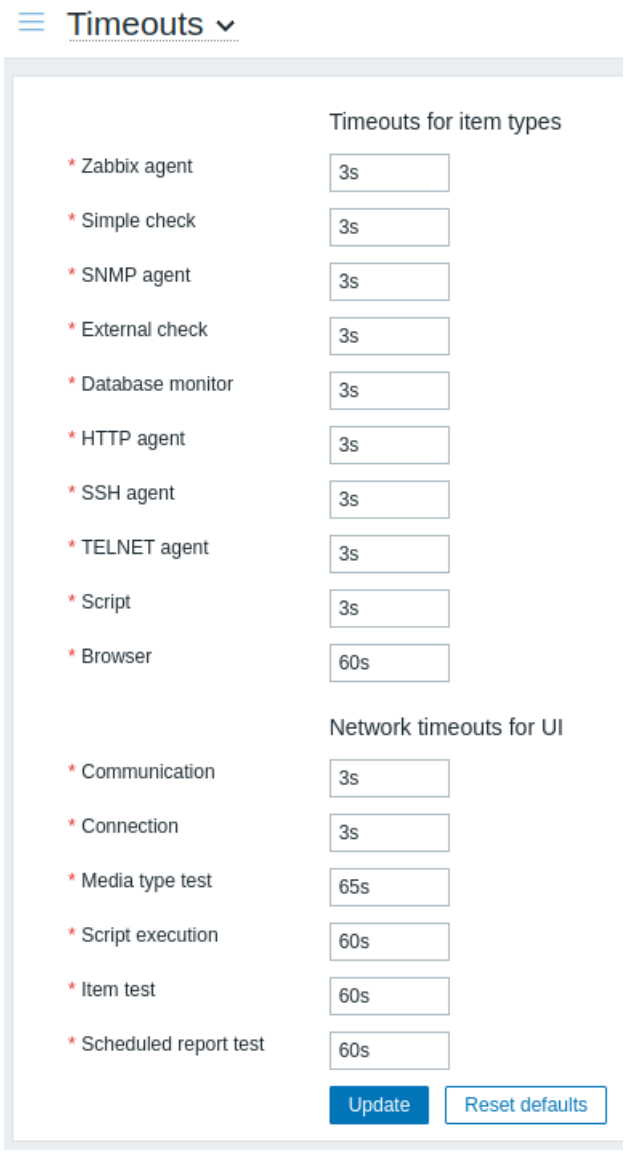
Parametri di configurazione:

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	Inserire 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 in chiaro sulla rete per informare il destinatario su quale PSK utilizzare.
PSK	Inserire 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*.

Vedere anche: Registrazione automatica sicura

Timeout

In questa sezione è possibile impostare i timeout globali per tipo di item e i timeout di rete. Tutti i campi di questo modulo sono obbligatori.

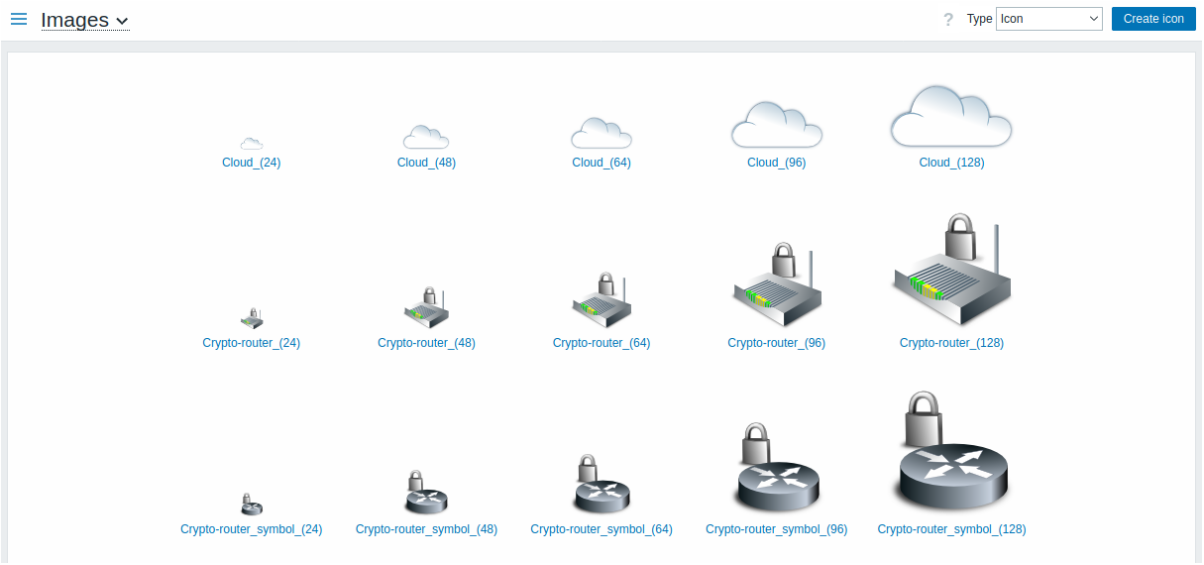


Parametro		Descrizione
Timeout per i tipi di item		Per quanti secondi Zabbix deve attendere prima di interrompere il controllo (in base al suo tipo). Per ulteriori dettagli e limitazioni, vedere timeout dell'item. Intervallo consentito: 1 - 600s (predefinito: 3s; predefinito per gli item browser: 60s). 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, sovrascrivono queste impostazioni globali. Inoltre, se sono configurati timeout per specifici item, questi sovrascrivono sia le impostazioni globali sia quelle del proxy.
Timeout di rete per la UI		
	Comunicazione	Per quanti secondi attendere prima di chiudere un socket inattivo (se una connessione al server Zabbix è stata stabilita in precedenza, ma il frontend non riesce a completare l'operazione di lettura/invio dei dati entro questo tempo, la connessione verrà interrotta). Intervallo consentito: 1 - 300s (predefinito: 3s).
	Connessione	Per quanti secondi attendere prima di interrompere un tentativo di connessione al server Zabbix. Intervallo consentito: 1 - 30s (predefinito: 3s).

Parametro		Descrizione
	Test del tipo di supporto	Per quanti secondi attendere una risposta durante il test di un tipo di supporto. Intervallo consentito: 1 - 300s (predefinito: 65s).
	Esecuzione script	Per 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 lunghezza variabile. Ad esempio, se lo script viene eseguito sull'agent, si tratta di un percorso di andata e ritorno attraverso il server (eventualmente anche il proxy) fino all'agent e ritorno. Per gli script eseguiti sull'agent, si applica il timeout del server o del 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 dei controlli attivi, altrimenti il timeout di server/proxy potrebbe essere superato prima ancora che l'agent attivo riceva lo script. Vedere anche: Timeout dello script
	Test item	Per quanti secondi attendere i dati restituiti durante il test di un item. Intervallo consentito: 1 - 600s (predefinito: 60s).
	Test del report pianificato	Per quanti secondi attendere i dati restituiti durante il test di un report pianificato. Intervallo consentito: 1 - 300s (predefinito: 60s).

Immagini

La sezione Immagini visualizza tutte le immagini disponibili in Zabbix. Le immagini sono memorizzate nel database.

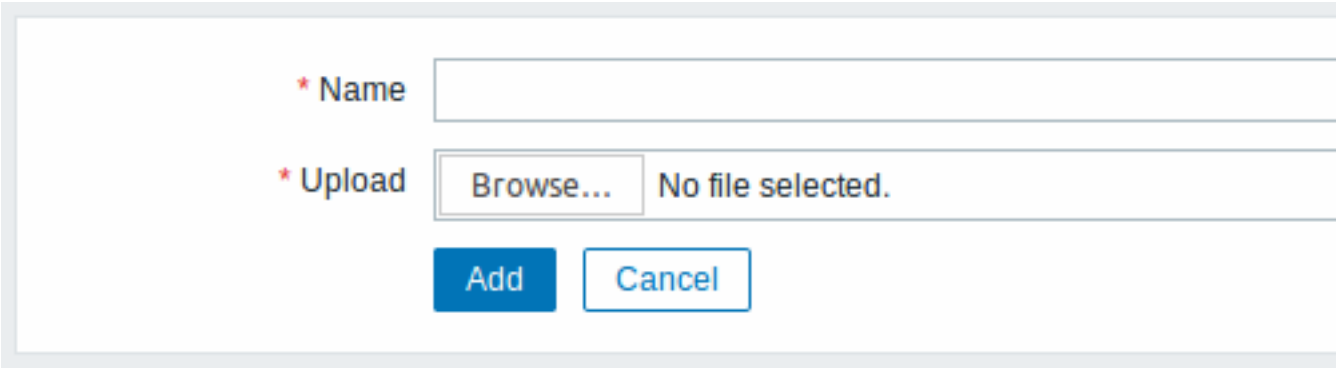


Il menu a discesa *Tipo* consente di passare tra immagini icona e di sfondo:

- Le icone sono utilizzate per visualizzare gli elementi delle mappe di rete
- Gli sfondi sono utilizzati come immagini di sfondo delle mappe di rete

Aggiunta di un'immagine

Puoi aggiungere una tua immagine facendo clic sul pulsante *Crea icona* o *Crea sfondo* nell'angolo superiore destro.



Attributi dell'immagine:

Parameter	Description
Name	Nome univoco di un'immagine.
Upload	Seleziona il file (PNG, JPEG, GIF o WebP) da un sistema locale da caricare in Zabbix. *Nota* che 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 in uso (Zabbix richiede la versione 2.0.28 o successiva).

Note:
La dimensione massima del file caricato è limitata dal valore di ZBX_MAX_IMAGE_SIZE, che è 1024x1024 byte o 1 MB.

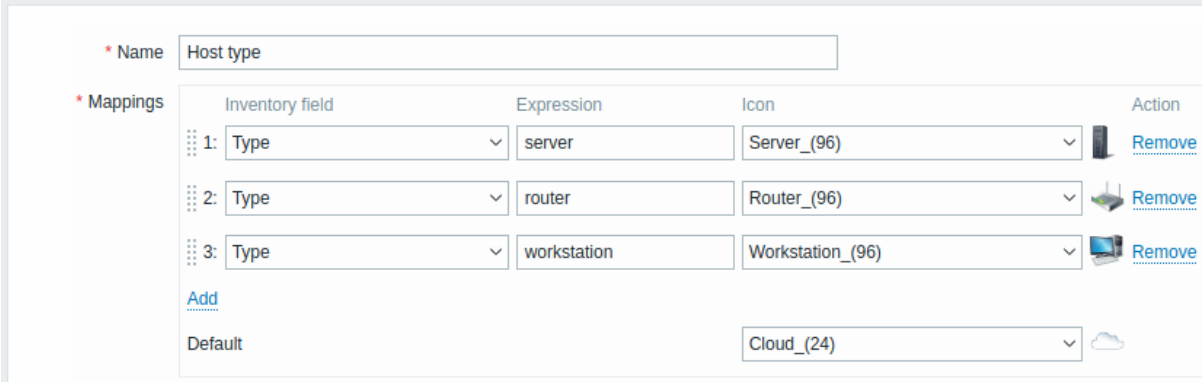
Il caricamento di un'immagine potrebbe non riuscire se la dimensione dell'immagine è vicina a 1 MB e il parametro di configurazione MySQL `max_allowed_packet` è impostato sul valore predefinito di 1 MB. In questo caso, aumenta il parametro max_allowed_packet.

Mappatura delle icone

Questa sezione consente di creare la mappatura di determinati host con determinate icone. Le informazioni del campo dell'inventario host vengono utilizzate per creare la mappatura.

Le mappature possono quindi essere utilizzate nella configurazione delle mappe di rete per assegnare automaticamente le icone appropriate agli host corrispondenti.

Per creare una nuova mappa di icone, fare clic su *Crea mappa di icone* nell'angolo in alto a destra.



Parametri di configurazione:

Parametro	Descrizione
Nome	Nome univoco della mappa di icone.
Mappature	Un elenco di mappature. L'ordine delle mappature determina quale avrà la priorità. È possibile spostare le mappature in alto e in basso nell'elenco tramite trascinamento.
Campo inventario	Campo dell'inventario host in cui verrà cercata una corrispondenza.
Espressione	Espressione regolare che descrive la corrispondenza.
Icona	Icona da utilizzare se viene trovata una corrispondenza per l'espressione.
Predefinita	Icona predefinita da utilizzare.

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

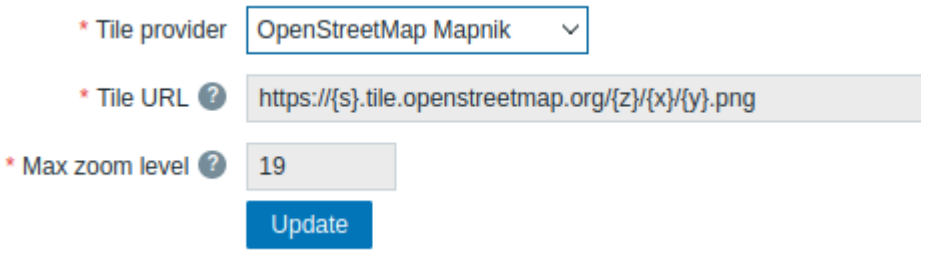
Questa sezione consente di personalizzare come lo stato dei trigger viene visualizzato nel frontend e i nomi e i colori della gravità dei trigger.



Parametro	Descrizione
Usa colori personalizzati per lo stato degli eventi *Eventi PROBLEM non confermati*, *Eventi PROBLEM confermati*, *Eventi RESOLVED non confermati*, *Eventi RESOLVED confermati*	Selezionando questo parametro si attiva la personalizzazione dei colori per i problemi confermati/non confermati. Inserire un nuovo codice colore oppure fare clic sul colore per selezionarne uno nuovo dalla tavolozza fornita. Se la casella *lampeggiante* è selezionata, i trigger lampeggeranno per un certo periodo di tempo al cambio di stato, per risultare più visibili.
Visualizza i trigger OK per	Periodo di tempo per la visualizzazione dei trigger OK. Intervallo consentito: 0 - 24 ore. Sono supportati i suffissi temporali, ad esempio 5m, 2h, 1d.
Al cambio di stato i trigger lampeggiano per	Durata del lampeggiamento dei trigger. Intervallo consentito: 0 - 24 ore. Sono supportati i suffissi temporali, ad esempio 5m, 2h, 1d.
Non classificato, *Informazione*, *Avviso*, *Medio*, *Alto*, *Disastro*	Nomi e/o colori di gravità personalizzati da visualizzare al posto di quelli predefiniti del sistema. Inserire un nuovo codice colore oppure fare clic sul colore per selezionarne uno nuovo dalla tavolozza fornita. Si noti che i nomi di gravità personalizzati inseriti qui verranno utilizzati in tutte le impostazioni locali. Se è necessario tradurli in altre lingue per determinati utenti, vedere la pagina Personalizzazione delle gravità dei trigger.

Mappe geografiche

Questa sezione consente di selezionare il provider del servizio di tile per mappe geografiche e di configurarne le impostazioni per il widget Geomap della dashboard. Per fornire la visualizzazione tramite mappe geografiche, Zabbix utilizza Leaflet, una libreria open-source JavaScript per mappe interattive. Si noti che Zabbix non ha alcun controllo sulla qualità delle immagini fornite da provider di tile di terze parti, inclusi i provider di tile predefiniti.



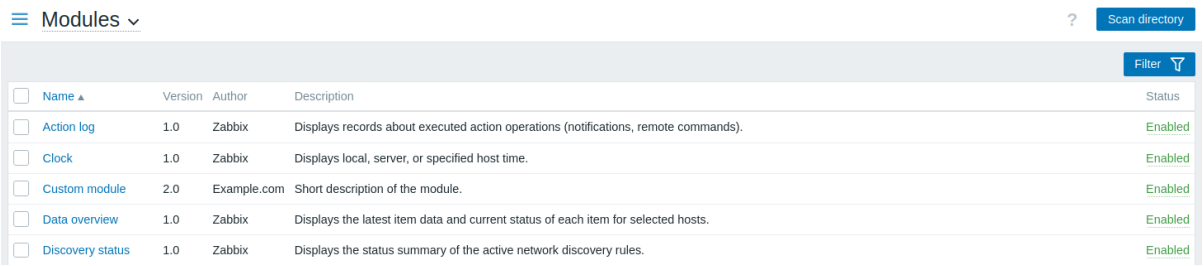
Parametro	Descrizione
Provider di tile	Selezionare uno dei provider di servizi di tile disponibili oppure selezionare *Altro* per aggiungere un altro provider di tile o tile self-hosted (vedere Utilizzo di un provider di servizi di tile personalizzato).
URL tile	Il modello di URL (fino a 2048 caratteri) per il caricamento e la visualizzazione del layer di tile sulle mappe geografiche. Questo campo è modificabile solo se *Provider di tile* è impostato su *Altro*. 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 utilizzato per aggiungere "@2x" all'URL per caricare tile retina. Esempio: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Testo di attribuzione	Testo di attribuzione del provider di tile da visualizzare in una piccola casella di testo sulla mappa. Questo campo è visibile solo se *Provider di tile* è impostato su *Altro*.
Livello di zoom massimo	Livello di zoom massimo della mappa. Questo campo è modificabile solo se *Provider di tile* è impostato su *Altro*.

Utilizzo di un fornitore di servizi di tile personalizzate

Il widget Geomap è in grado di caricare immagini tile raster da un file personalizzato self-hosted o un servizio di provider di tile di terze parti. Per utilizzare un file personalizzato servizio di provider di tile di terze parti o una cartella di tile self-hosted o server, selezionare *Altro* nel campo *Tile provider* e specificare il URL personalizzato nel campo *Tile URL* utilizzando i segnaposto appropriati.

Moduli

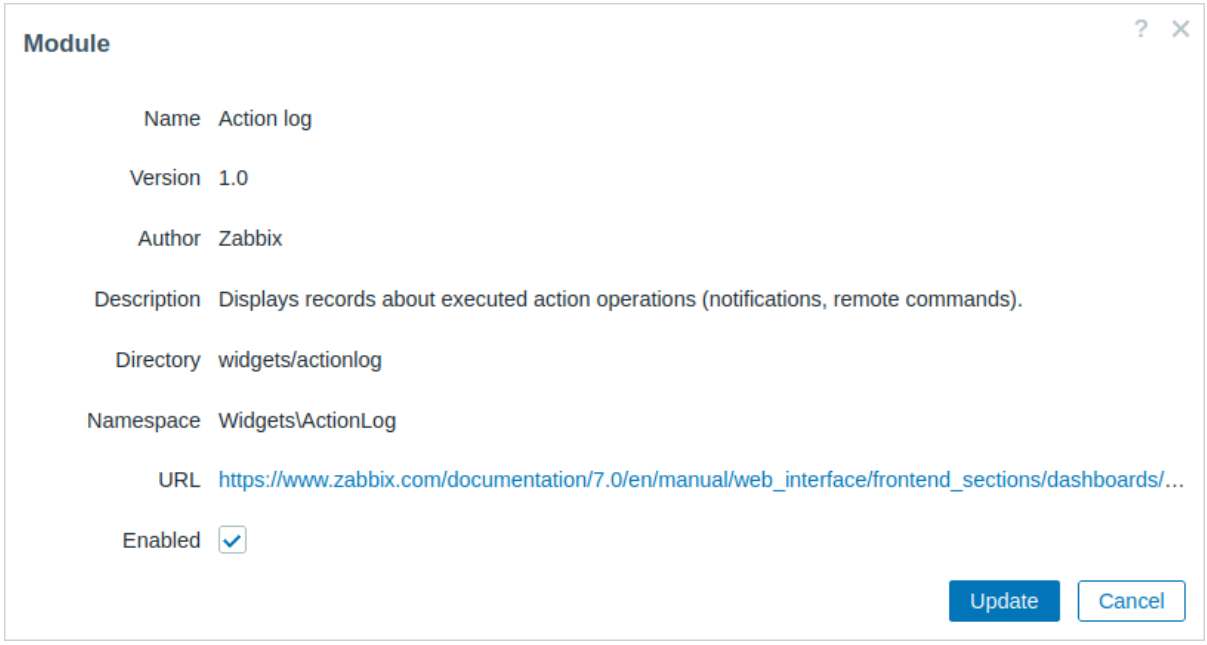
Questa sezione consente di amministrare i moduli del frontend personalizzati, così come quelli integrati.



Fare clic su *Scan directory* per registrare/annullare la registrazione di eventuali moduli personalizzati. I moduli registrati appariranno nell'elenco; i moduli non registrati verranno rimossi dall'elenco.

Fare clic sullo stato del modulo nell'elenco per abilitare/disabilitare un modulo. È inoltre possibile abilitare/disabilitare più moduli contemporaneamente selezionandoli nell'elenco e facendo poi clic sui pulsanti *Enable/Disable* sotto l'elenco.

Fare clic sul nome del modulo nell'elenco per visualizzarne i dettagli in una finestra pop-up.

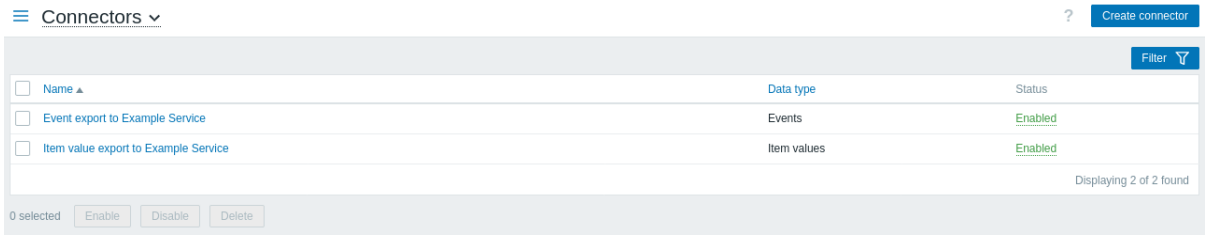


Lo stato del modulo può essere aggiornato anche nella finestra pop-up dei dettagli del modulo; per farlo, selezionare/deselezionare la casella di controllo *Enabled* e quindi fare clic su *Update*.

È possibile filtrare i moduli per nome o stato (abilitato/disabilitato).

Connettori

Questa sezione consente di configurare i connettori per lo streaming verso sistemi esterni dei dati di Zabbix 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 abilitare/disabilitare un connettore.
È inoltre possibile abilitare/disabilitare più connettori contemporaneamente selezionandoli nell'elenco e facendo quindi 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 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/autoregistrazione dell'host dall'operazione *Imposta modalità inventario host*.

Parametro	Descrizione
Gruppo utenti per il messaggio di database non disponibile	Gruppo utenti per l'invio del messaggio di allarme oppure 'Nessuno'. 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 dei supporti utente abilitate. Le notifiche vengono trasmesse utilizzando i seguenti tipi di supporto (se abilitati): email; SMS; script di avviso personalizzati. Anche se una voce di supporto 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 spesso di una volta ogni 15 minuti.
Registra trap SNMP senza corrispondenza	Registra SNMP trap se non sono state trovate interfacce SNMP corrispondenti.

Autorizzazione

Parametro	Descrizione
Tentativi di accesso	Numero di tentativi di accesso non riusciti prima che venga bloccata la possibilità di effettuare l'accesso.
Intervallo di blocco dell'accesso	Periodo di tempo durante il quale l'accesso sarà vietato quando viene superato il limite di *Tentativi di accesso*. Intervallo consentito: 0 - 3600 secondi. Sono supportati i suffissi temporali, ad esempio 90s, 5m, 1h.

Archiviazione dei segreti

Parameter	Description
Vault provider	Selezionare il software di gestione dei segreti per memorizzare i valori delle macro utente: *HashiCorp Vault* (predefinito) oppure *CyberArk Vault*.
Resolve secret vault macros by	Risolvere 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.

Vedere anche: Archiviazione dei segreti.

Sicurezza

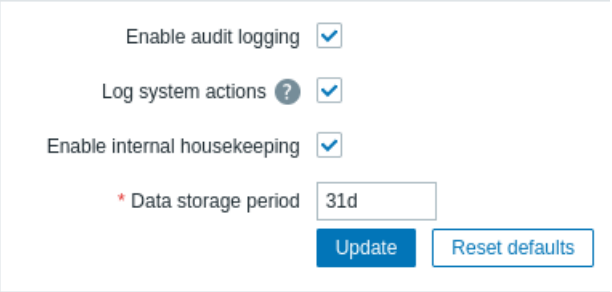
Parameter	Description
Convalida schemi URI	Deselezionare questa casella di controllo per disabilitare la convalida degli schemi URI (abilitata per impostazione predefinita). Se selezionata, è possibile specificare un elenco separato da virgole degli schemi URI consentiti (predefinito: http,https,ftp,file,mailto,tel,ssh). 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`.

2 Registro di controllo

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.

3 Pulizia

Panoramica

La sezione *Administration > Housekeeping* consente di configurare l'housekeeping interno di Zabbix che rimuove automaticamente dal database i dati obsoleti o eliminati dall'utente. Questo impedisce al database di crescere indefinitamente, consumando inutilmente spazio su disco e peggiorando le prestazioni del database.

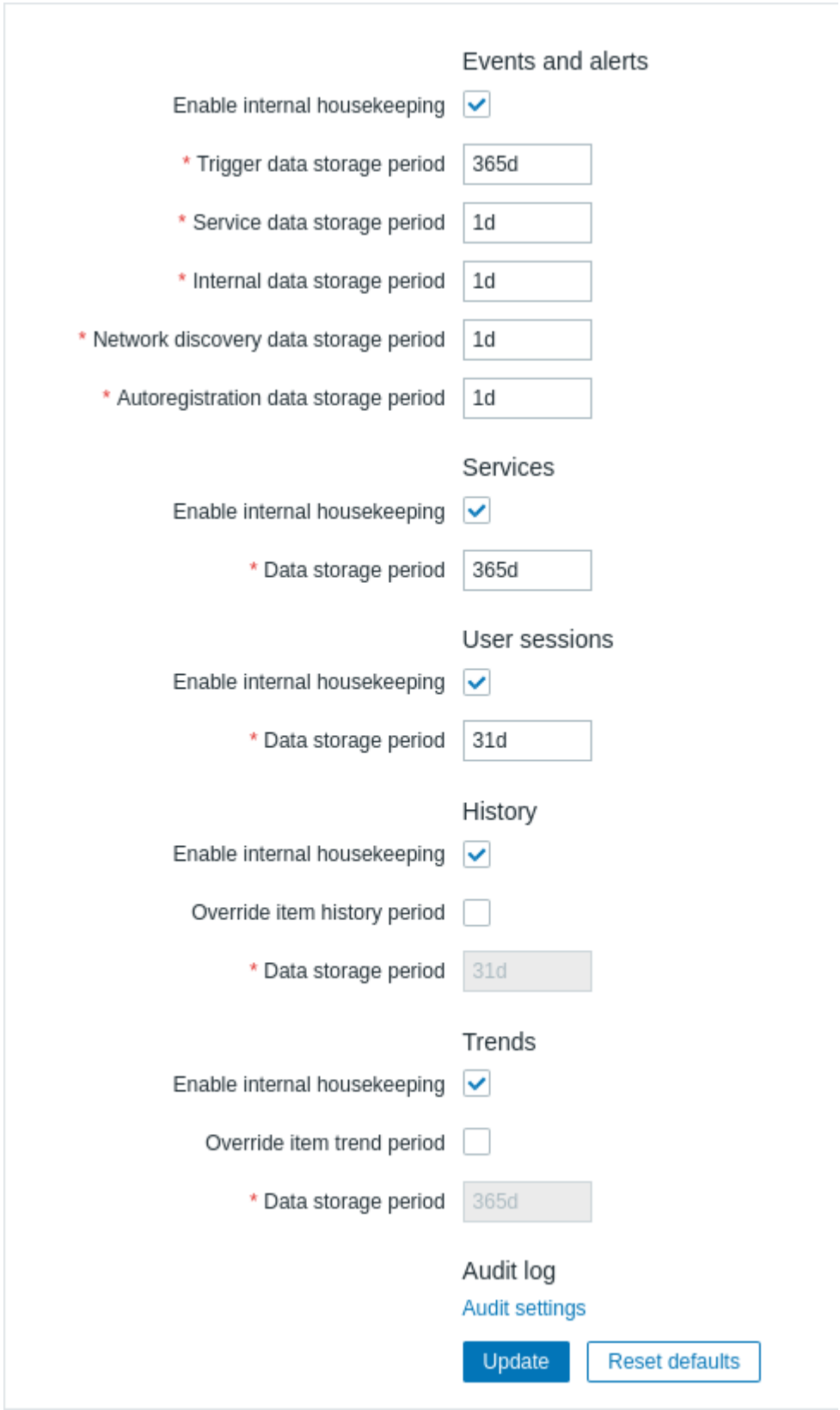
L’housekeeping può essere abilitato e configurato per i seguenti tipi di dati:

- Eventi e avvisi da trigger, servizi, dati interni, network discovery e autoregistrazione
- Servizi
- Sessioni utente
- Storico e trend per gli item

> **Note:**
> L’housekeeping del log di audit viene configurato separatamente.

Configurazione

Sono disponibili i seguenti parametri:



Parametro	Descrizione
Abilita housekeeping interno	Abilita o disabilita l'housekeeping interno (abilitato per impostazione predefinita). Se abilitato, la procedura di housekeeping rimuove automaticamente dal database i dati che superano il *Periodo di conservazione dei dati*.

Parametro	Descrizione
Periodo di conservazione dei dati	Specifica per quanto tempo conservare i dati prima che vengano rimossi dall'housekeeper. Obbligatorio se l'housekeeping interno è abilitato. Intervallo: 1 giorno (1 ora per la cronologia) - 25 anni; oppure "0". Sono supportati i suffissi temporali (ad esempio, 1d, 1w). Per *Eventi e avvisi*, il periodo di conservazione dei dati viene impostato separatamente per trigger, servizi, dati interni, network discovery e autoregistrazione. L'housekeeper rimuove solo gli eventi che non sono associati a problemi. Ad esempio, un evento di problema/ripristino più vecchio del *Periodo di conservazione dei dati* non verrà rimosso se è associato a un problema non risolto. Quando l'housekeeper rimuove entità obsolete, rimuove prima i problemi e poi gli eventi. Per *Cronologia* e *Trend*, i periodi di conservazione dei dati determinano anche per quanto tempo i dati rimangono visibili in *Monitoring > Latest data*, anche se l'housekeeping interno è disabilitato. Si noti che quando un item o un trigger viene eliminato, anche i relativi problemi e gli eventi associati vengono eliminati.
Sovrascrivi periodo cronologia item	Se abilitato, il periodo di conservazione di cronologia/trend specificato nella configurazione dell'item viene sovrascritto dall'impostazione *Periodo di conservazione dei dati* (tranne per gli item con l'opzione *Non memorizzare* abilitata). Questa opzione può essere utilizzata anche quando l'housekeeping interno è disabilitato e viene usato un housekeeper esterno.
Sovrascrivi periodo trend item	

Se si utilizza **TimescaleDB**, diventa disponibile la sezione *Compressione di cronologia, trend e log di audit*.

Note:
Per TimescaleDB, abilitare *Sovrascrivi periodo cronologia item*, *Sovrascrivi periodo trend item* e *Abilita housekeeping interno* per cronologia e trend, per beneficiare pienamente del partizionamento automatico. Se queste opzioni sono disabilitate, i dati conservati nelle tabelle di cronologia e trend saranno comunque 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 cronologia di tali item verrà cancellata quando una partizione obsoleta viene eliminata.

Il pulsante *Ripristina predefiniti* consente di annullare tutte le modifiche apportate.

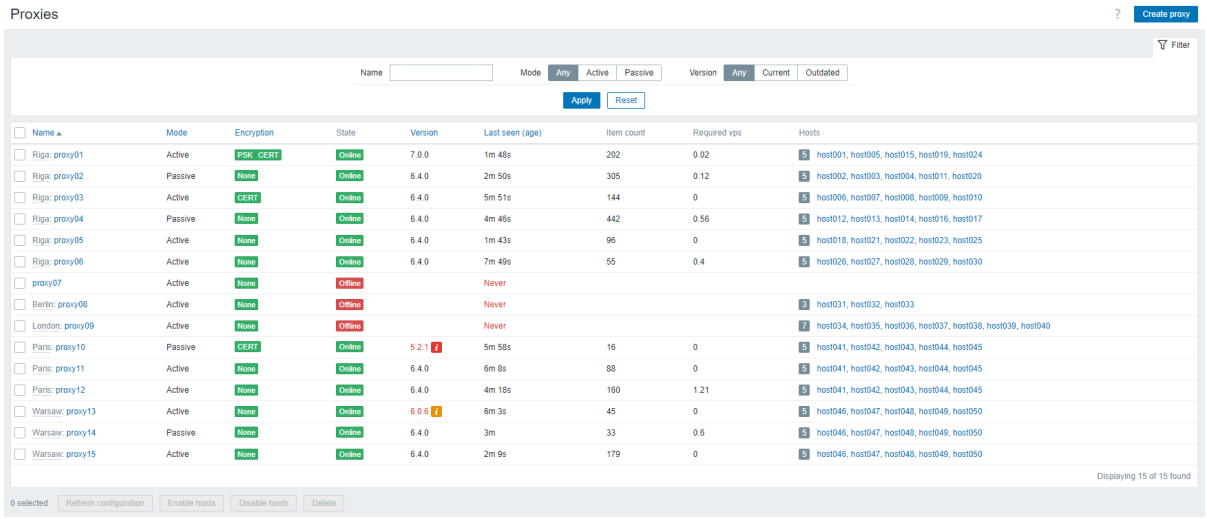
4 Proxy

Panoramica

Nella sezione *Amministrazione → Proxies* 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:

Column	Description
Name	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.
Mode	Modalità del proxy - *Active* o *Passive*.
Encryption	Stato della crittografia per le connessioni dal proxy: **None** - nessuna crittografia; **PSK** - utilizzo di una chiave precondivisa; **Cert** - utilizzo di un certificato.
State	Stato del proxy: **Unknown** - il proxy è stato creato mentre Zabbix server era inattivo, oppure il server non ha ancora aggiornato lo stato; **Online** - il proxy ha comunicato con Zabbix server (il proxy passivo ha risposto a una richiesta del server; il proxy attivo ha inviato una richiesta) entro il periodo di failover; **Offline** - il proxy non ha comunicato con Zabbix server entro il periodo di failover.
Version	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 l'icona di stato informativo (gialla o rossa). Passare il puntatore sull'icona per i dettagli.
Last seen (age)	L'ora in cui il proxy è stato visto l'ultima volta dal server.
Item count	Il numero di item abilitati sugli host abilitati assegnati al proxy.
Required vps	Prestazioni richieste del proxy (il numero di valori che devono essere raccolti al secondo).
Hosts	Numero di host abilitati assegnati al proxy e un elenco di 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 *Create proxy* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

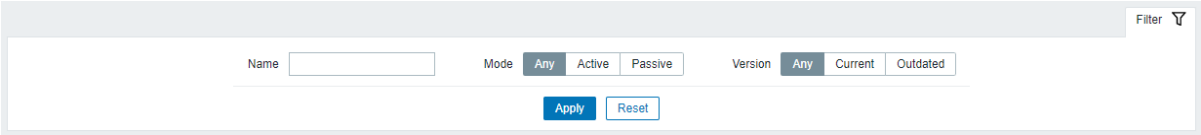
- *Refresh configuration* - aggiorna la configurazione dei proxy;
- *Enable hosts* - cambia lo stato degli host monitorati dal proxy in *Monitored*;
- *Disable hosts* - cambia lo stato degli host monitorati dal proxy in *Not monitored*;
- *Delete* - elimina i proxy.

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi proxy, quindi fare clic sul pulsante richiesto.

Uso del filtro

È possibile utilizzare il filtro per visualizzare solo i proxy di interesse. Per migliori prestazioni di 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, diventa disponibile un filtro con cui è possibile filtrare i proxy per nome, modalità e versione. Si noti che l'opzione del filtro *Obsoleti* visualizza sia i proxy obsoleti (parzialmente supportati) sia quelli non supportati.



5 Gruppi di proxy

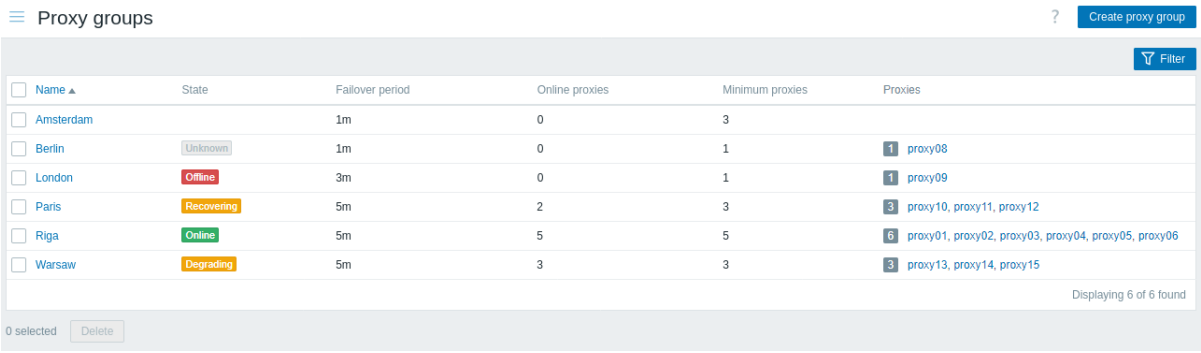
Panoramica

In *Amministrazione* → *Gruppi di proxy* è possibile configurare i gruppi di proxy.

I gruppi di proxy vengono utilizzati nel bilanciamento del carico dei proxy con distribuzione automatizzata 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:

Column	Description
Name	Nome del gruppo di proxy. Facendo clic sul nome del gruppo di proxy si apre il modulo di configurazione del gruppo di proxy.
State	Stato del gruppo di proxy (visualizzato se il gruppo contiene almeno un proxy): **Unknown** - il gruppo è stato creato mentre lo Zabbix server era inattivo, oppure il server non ha ancora aggiornato lo stato; **Online** - il numero minimo di proxy ha comunicato con lo Zabbix server entro il periodo di failover; **Degrading** - il gruppo sta per andare offline a breve perché i proxy online sono scesi sotto la soglia di *Minimum proxies*; **Offline** - meno del numero minimo di proxy ha comunicato con lo Zabbix server entro il periodo di failover; **Recovering** - il gruppo sta per tornare online a breve perché i proxy online hanno raggiunto la soglia di *Minimum proxies*. Lo stato del gruppo di proxy può essere monitorato con gli item zabbix[proxy group,<name>,state] e zabbix[proxy group,discovery].
Failover period	Periodo durante il quale un proxy nel gruppo di proxy deve comunicare con lo Zabbix server per essere considerato online.
Online proxies	Numero di proxy online (visualizzato in rosso se inferiore al minimo del gruppo).
Minimum proxies	Numero minimo di proxy online richiesto per mantenere online il gruppo di proxy.
Proxies	Numero di proxy nel gruppo e relativo elenco, con collegamenti al modulo di configurazione del proxy. Il numero massimo di proxy elencati è limitato dal valore *Max count of elements to show inside table cell*.

Per configurare un nuovo gruppo di proxy, fare clic sul pulsante *Create proxy groups* nell'angolo superiore destro.

Opzioni di modifica di massa

I pulsanti sotto l'elenco offrono alcune opzioni di modifica di massa:

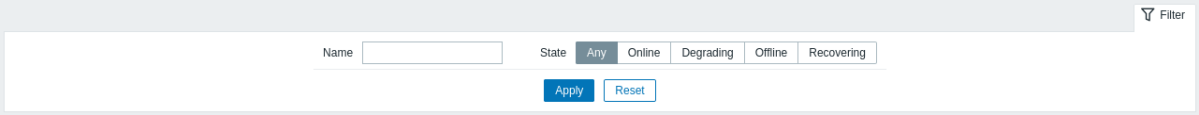
- *Delete* - elimina i gruppi di proxy.

Per utilizzare queste opzioni, selezionare le caselle di controllo davanti ai rispettivi gruppi di proxy, quindi fare clic sul pulsante richiesto.

Utilizzo del filtro

È possibile utilizzare il filtro per visualizzare solo i gruppi di proxy di interesse. Per ottenere prestazioni di ricerca migliori, i dati vengono cercati con le macro non risolte.

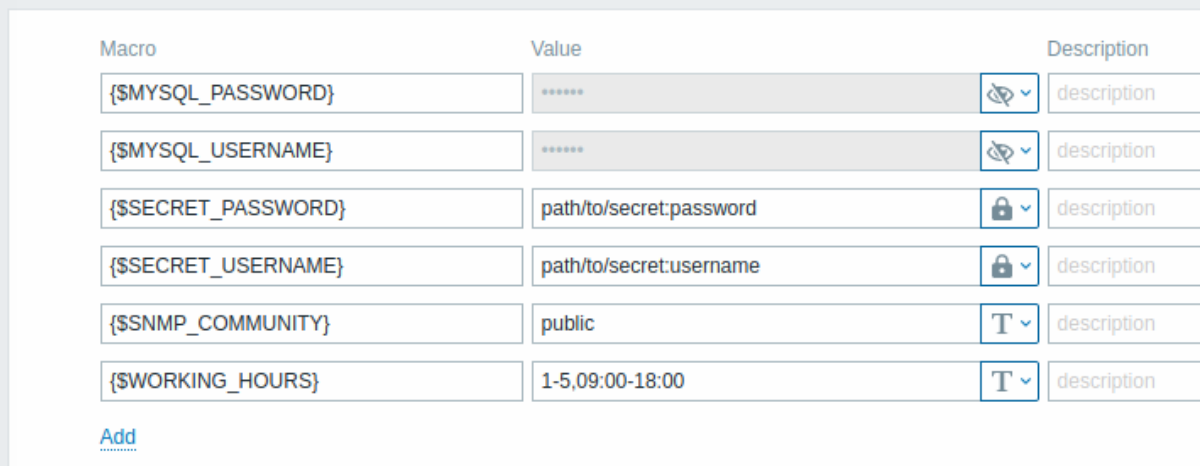
Il link *Filtro* è disponibile sopra l'elenco dei gruppi di proxy. Facendo clic su di esso, diventa disponibile un filtro in cui è possibile filtrare i gruppi di proxy per nome e stato.



6 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 di Vault. È inoltre supportata l'aggiunta di una descrizione.



7 Coda

Panoramica

Nella sezione *Amministrazione* → *Coda* vengono visualizzati gli item in attesa di essere aggiornati.

Idealmente, quando si apre questa sezione dovrebbe essere tutto "verde", cioè 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 possono subire ritardi e le informazioni vengono visualizzate in questa sezione. Per maggiori dettagli, vedere la sezione Queue.

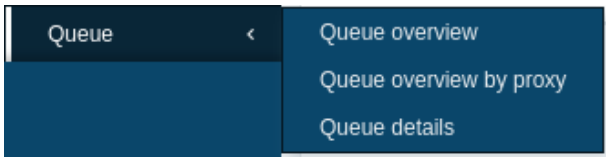
Note:
Lo Zabbix proxy non è a conoscenza dei periodi di manutenzione; per i dettagli vedere Calculation of queues during maintenance.

Note:
La coda è disponibile solo se Zabbix server 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.

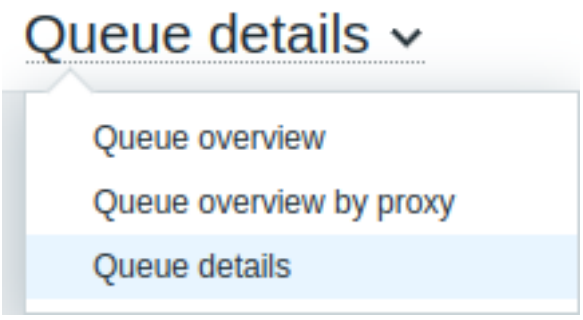
La sezione *Amministrazione* → *Coda* contiene le seguenti pagine:

- 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 di item in ritardo.

L'elenco delle pagine disponibili appare facendo clic su *Coda* nella sezione del menu *Amministrazione*. È anche possibile passare da una pagina all'altra utilizzando il menu a discesa del titolo nell'angolo superiore sinistro.



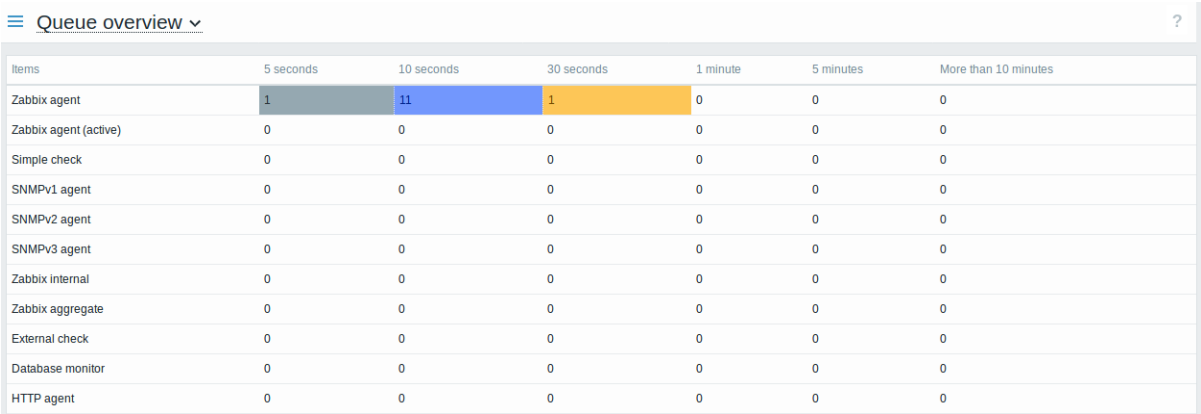
Menu di terzo livello.



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.

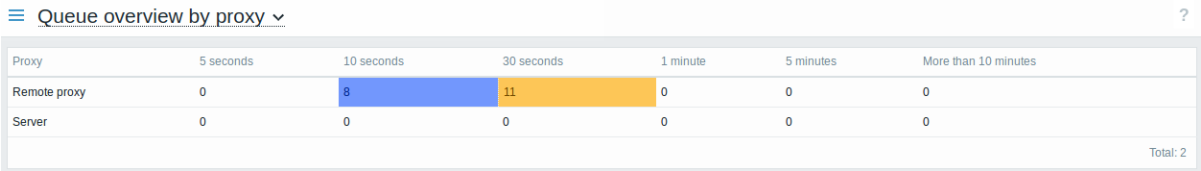


Items	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Zabbix agent	1	11	1	0	0	0
Zabbix agent (active)	0	0	0	0	0	0
Simple check	0	0	0	0	0	0
SNMPv1 agent	0	0	0	0	0	0
SNMPv2 agent	0	0	0	0	0	0
SNMPv3 agent	0	0	0	0	0	0
Zabbix internal	0	0	0	0	0	0
Zabbix aggregate	0	0	0	0	0	0
External check	0	0	0	0	0	0
Database monitor	0	0	0	0	0	0
HTTP agent	0	0	0	0	0	0

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.



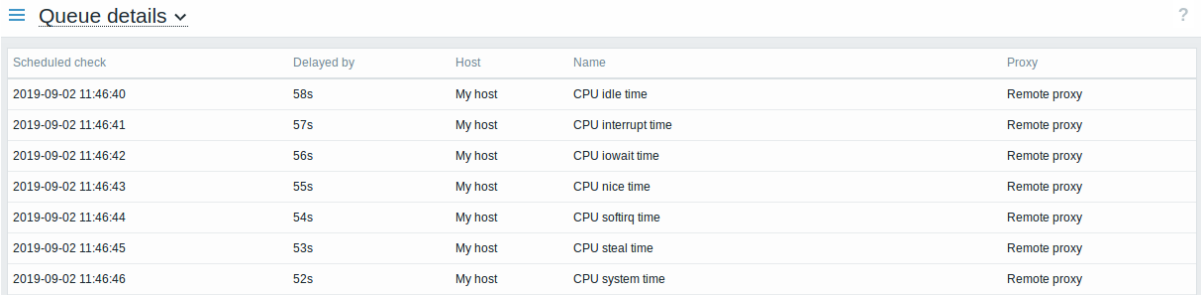
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

Total: 2

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.



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

Dati visualizzati:

Colonna	Descrizione
Assegno programmato	Viene visualizzata l'ora in cui l'assegno era scaduto.

Colonna	Descrizione
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 viene visualizzato alcun dato e compare il seguente messaggio di errore:



In questo caso, il messaggio di errore è il seguente:

```
Cannot display item queue. Permission denied
```

Ciò accade quando i parametri di configurazione PHP nel file *zabbix.conf.php* - `$ZBX_SERVER` oppure entrambi `$ZBX_SERVER` e `$ZBX_SERVER_PORT` - puntano a un server Zabbix esistente che utilizza un database diverso.

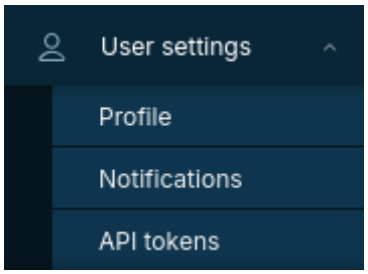
3 Impostazioni utente

Panoramica

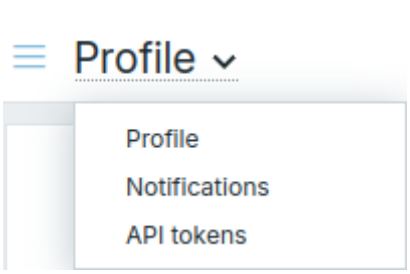
A seconda dei permessi del ruolo utente, la sezione *Impostazioni utente* può contenere le seguenti pagine:

- *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.

L'elenco delle pagine disponibili viene visualizzato facendo clic sull'icona utente vicino alla 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.



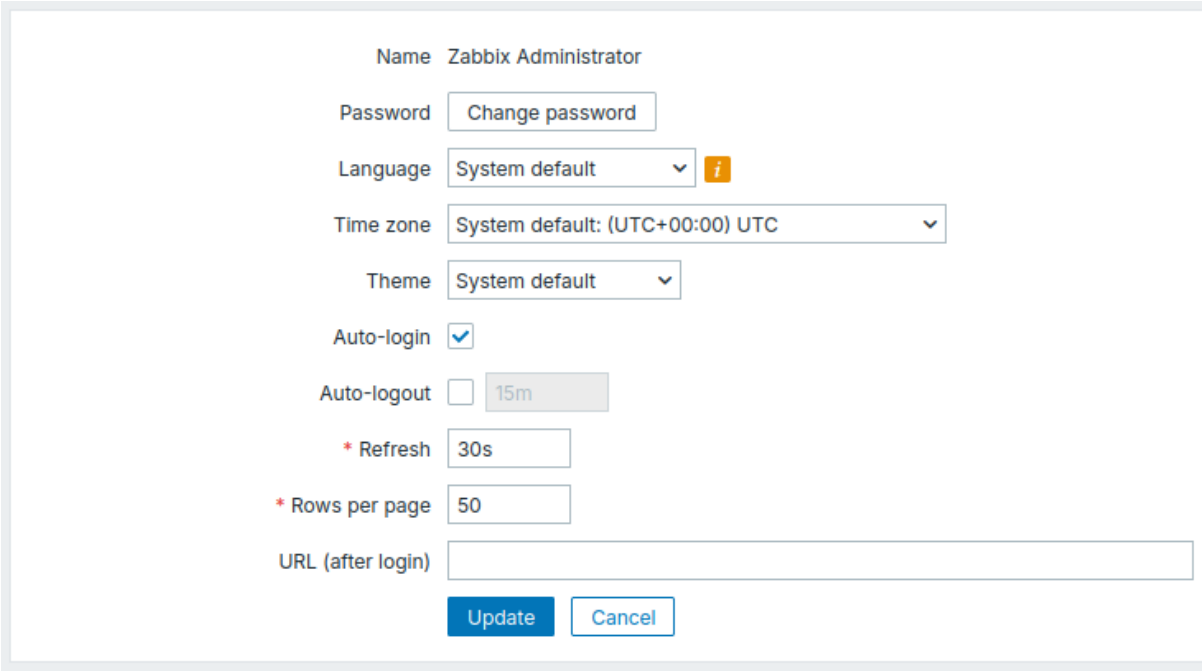
Menu di terzo livello.



Menu a discesa del titolo.

Profilo utente

La sezione *Profilo utente* offre opzioni per impostare una lingua dell'interfaccia personalizzata, il tema dei colori, il numero di righe visualizzate negli elenchi, ecc. Le modifiche apportate qui verranno applicate solo all'utente corrente.



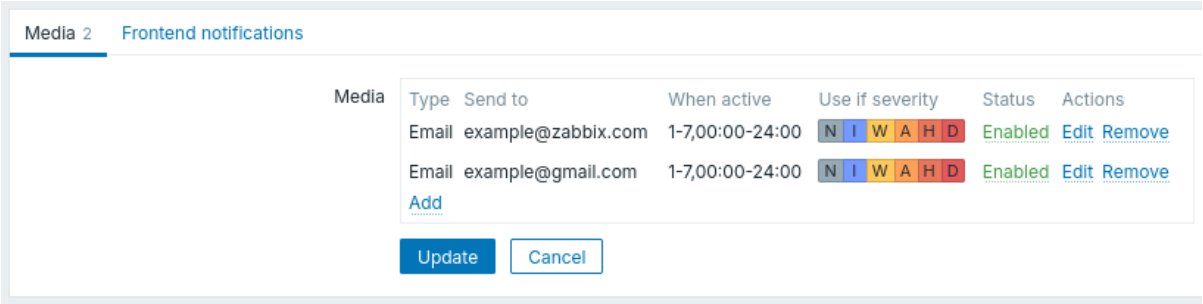
Parametro	Descrizione
Name	Vengono visualizzati il nome e il cognome dell'utente. In assenza di nome e cognome, viene visualizzato il nome utente.
Password	Fare clic sul pulsante *Change password* per aprire tre campi: *Old password*, *New password*, *New password (once again)*. Se la password viene modificata correttamente, l'utente verrà disconnesso da tutte le sessioni attive. Si noti che la password può essere modificata solo per gli utenti che utilizzano l'autenticazione interna di Zabbix.
Language	Selezionare la lingua dell'interfaccia desiderata oppure **System default** per utilizzare le impostazioni predefinite del sistema. Selezionando *English (en_US)* verrà abilitato anche il formato data/ora US nel frontend. Per ulteriori informazioni, vedere Installazione di lingue aggiuntive del frontend.
Time zone	Selezionare il fuso orario per sovrascrivere il fuso orario globale a livello di utente oppure selezionare **System default** per utilizzare le impostazioni globali del fuso orario.
Theme	Selezionare un tema di colori specifico per il proprio profilo: **System default** - utilizza le impostazioni predefinite del sistema; **Blue** - tema blu standard; **Blue (classic)** - tema blu standard che utilizza i caratteri legacy; **Dark** - tema scuro alternativo; **Dark (classic)** - tema scuro alternativo che utilizza i caratteri legacy; **High-contrast light** - tema chiaro ad alto contrasto; **High-contrast dark** - tema scuro ad alto contrasto.
Auto-login	Selezionare questa casella per fare in modo che Zabbix si ricordi di te ed esegua automaticamente l'accesso per 30 giorni. Quando si accede con *Remember for 30 days* 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 nelle visite successive entro 30 giorni. Quando si accede senza *Remember for 30 days*: - Il parametro di accesso automatico viene cancellato. - La disconnessione automatica segue normalmente il timeout configurato. Per questo vengono utilizzati i cookie del browser.

Parametro	Descrizione
Auto-logout	Con questa casella selezionata, si verrà disconnessi automaticamente dopo il numero di secondi impostato (minimo 90 secondi, massimo 1 giorno). Sono supportati i suffissi temporali, ad esempio: 90s, 5m, 2h, 1d. Si noti che questa opzione non funzionerà nei seguenti casi: * Quando le pagine del menu *Monitoring* eseguono aggiornamenti delle informazioni in background. Se vengono lasciate aperte pagine che aggiornano i dati a intervalli di tempo specifici (dashboard, grafici, dati più recenti, ecc.), la durata della sessione viene estesa, disabilitando di fatto la funzione di disconnessione automatica. * Se si accede con l'opzione *Remember me for 30 days* selezionata; in questo caso il timer di disconnessione automatica viene ignorato, mantenendo attiva la sessione per l'intero periodo. * Quando l'autenticazione viene eseguita tramite un provider di identità SSO (IdP), la disconnessione automatica termina solo la sessione Zabbix. La sessione SSO con l'IdP rimane attiva, quindi l'utente potrebbe essere in grado di accedere nuovamente senza inserire una password. *Auto-logout* può anche accettare "0", il che significa che la funzione di disconnessione automatica viene disabilitata dopo l'aggiornamento delle impostazioni del profilo.
Refresh	Impostare la frequenza con cui verranno aggiornate le informazioni nelle pagine del menu *Monitoring* (minimo 0 secondi, massimo 1 ora). Sono supportati i suffissi temporali, ad esempio: 30s, 90s, 1m, 1h.
Rows per page	Impostare quante righe verranno visualizzate per pagina negli elenchi. Un numero inferiore di righe (e quindi di record da visualizzare) comporta tempi di caricamento più rapidi.
URL (after login)	Impostare un URL specifico da visualizzare dopo l'accesso. Invece della pagina predefinita *Dashboards*, può essere, ad esempio, l'URL di *Monitoring* > *Triggers*.

Notifiche

La sezione *Notifiche* fornisce opzioni per personalizzare le notifiche dell'utente corrente.

La scheda *Media* consente di specificare i dettagli dei media per l'utente, come i tipi di media e gli indirizzi da utilizzare e quando utilizzarli per recapitare le notifiche.



Se il tipo di media 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 media 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 media di altri utenti possono essere concesse/revocate in base al relativo ruolo utente (vedere l'opzione *Create and edit user media*)

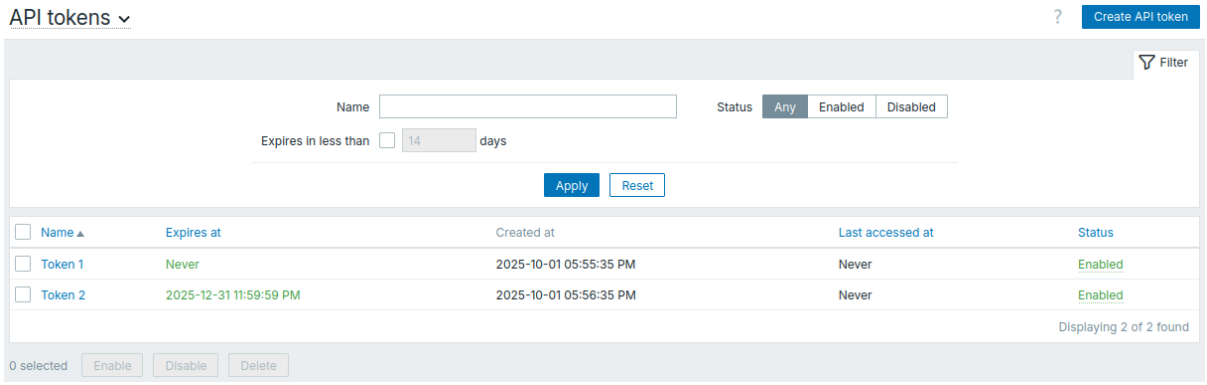
Si noti che per gli utenti provisioned:

- i media dell'utente provisioned non possono essere eliminati;
- i media dell'utente provisioned possono essere disabilitati/abilitati;
- i campi dei media dell'utente provisioned come *When active*, *Use if severity* e *Enabled* possono essere modificati manualmente;
- per gli utenti provisioned possono essere aggiunti manualmente media utente aggiuntivi (ad esempio, indirizzi email aggiuntivi);
- i media utente aggiunti manualmente possono essere eliminati.

La scheda *Notifiche frontend* 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*). 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 poi clic sui pulsanti *Enable/Disable* sotto l'elenco.

Attention:
Gli utenti non possono visualizzare il valore di *Auth token* dei token assegnati a loro in Zabbix. Il valore di *Auth token* viene visualizzato una sola volta, immediatamente dopo la creazione di un token. Se il token è stato smarrito, deve essere rigenerato.

1 Notifiche globali

Panoramica

Le notifiche globali consentono di visualizzare i problemi in tempo reale direttamente sulla schermata corrente nel frontend di Zabbix.

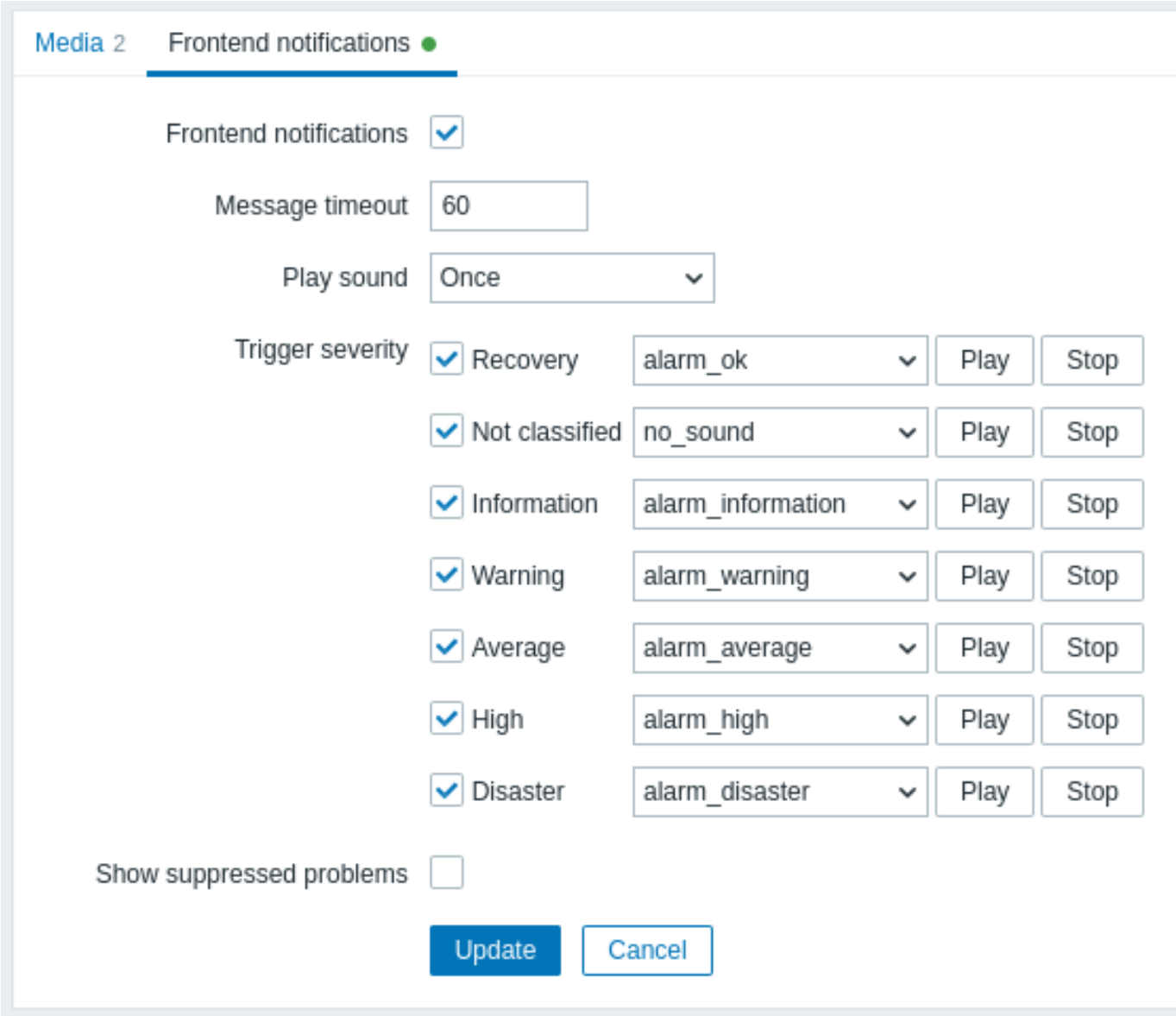
Senza le notifiche globali, quando si lavora al di fuori delle sezioni *Problemi* o *Dashboard*, non si riceverebbe alcuna informazione sui problemi correnti. Le notifiche globali garantiscono che queste informazioni vengano visualizzate, indipendentemente dalla posizione corrente all'interno del frontend 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

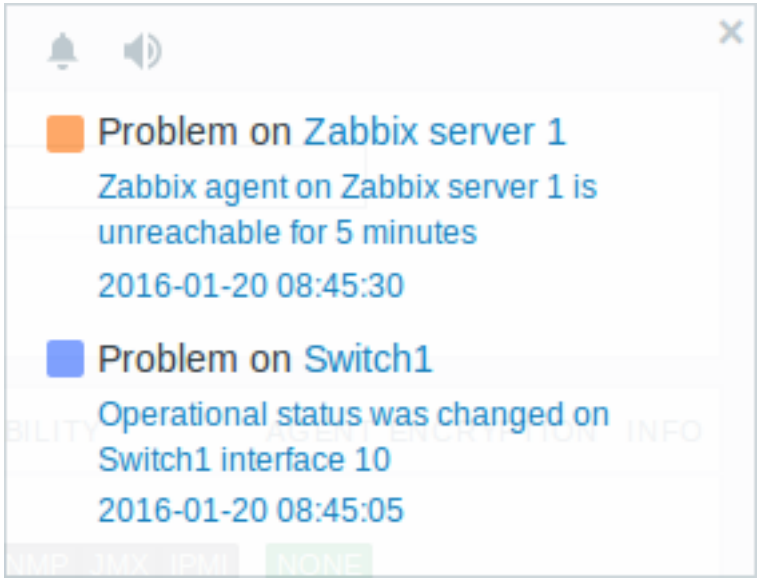
Le notifiche globali possono essere abilitate per ciascun utente nella scheda **Frontend notifications** della sezione Notifications.



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.

2 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.

4 Ricerca globale

È possibile effettuare ricerche nel frontend di Zabbix per 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 *Invio* oppure facendo clic sull'icona di ricerca .



Se esiste un host che contiene la stringa inserita in una 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:

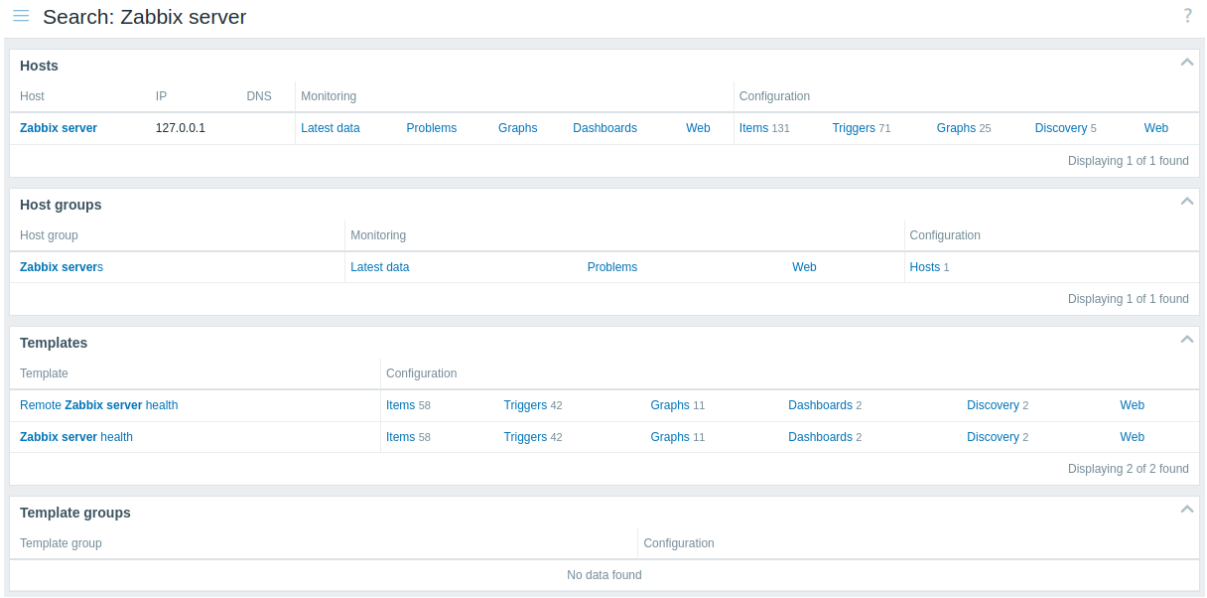
- Nome host
- Nome visibile
- Indirizzo IP
- Nome DNS

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 ogni blocco, ad esempio, *Visualizzazione di 13 su 13 trovati*. Se non ci sono voci, il numero di voci non viene visualizzato. Il totale delle 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), la quantità di entità trovate viene visualizzata da un numero accanto al nome dell'entità, in grigio. **Nota** che se ci sono zero entità, 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

5 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 server web) e decommentare le seguenti righe:

```
// Modalità di manutenzione.
define('ZBX_DENY_GUI_ACCESS', 1);

// Array di indirizzi IP autorizzati a connettersi al frontend (opzionale).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Messaggio mostrato nella schermata di avviso (opzionale).
$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 server web. Tuttavia, alcuni sistemi operativi e server web potrebbero utilizzare un percorso diverso.
Ad esempio, il percorso per:

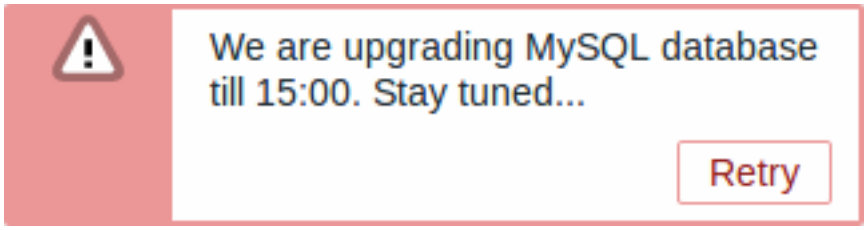
- SUSE e RedHat è `/etc/zabbix/web/maintenance.inc.php`.
- I sistemi basati su Debian è `/usr/share/zabbix/conf/`.

Vedi anche Copia dei file PHP.

Parametro	Dettagli
ZBX_DENY_GUI_ACCESS	Se definito con qualsiasi valore, la modalità di manutenzione verrà abilitata. Per disabilitare la modalità di manutenzione, commentare o eliminare.
ZBX_GUI_ACCESS_IP_RANGE	Array di indirizzi IP a cui è consentito connettersi al frontend (opzionale). Ad esempio: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Un messaggio per informare gli utenti della manutenzione (opzionale). 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.

6 Parametri di 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 (&).

Monitoraggio > Problemi

Sono supportati i seguenti parametri:

Parameter	Description	Example
show	Opzione di filtro *Mostra*. Valori possibili: 1 - problemi recenti; 2 - tutti; 3 - nello stato di problema.	show=1
name	Opzione di filtro *Problema*: stringa libera.	name=Zabbix agent

Parameter	Description	Example
severities	Opzione di filtro *Gravità*: array delle gravità selezionate nel formato `severities[*]=*` (sostituire * con il livello di gravità). Valori possibili: 0 - non classificato; 1 - informazione; 2 - avviso; 3 - media; 4 - alta; 5 - disastro.	severities[3]=3
inventory	Opzione di filtro *Inventario host*: array di campi inventario `[field]`, `[value]`	inventory[0][field]=type&invento
evaltype	Opzione di filtro *Tag*: metodo di valutazione dei tag. Valori possibili: 0 - and/or; 2 - or.	evaltype=0
tags	Opzione di filtro *Tag*: 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 di filtro *Mostra tag*. Valori possibili: 0 - nessuno; 1 - uno; 2 - due; 3 - tre.	show_tags=3
tag_name_format	Opzione di filtro *Nome tag*. Valori possibili: 0 - nome completo; 1 - nome abbreviato; 2 - nessuno.	tag_name_format=1
tag_priority	Opzione di filtro *Priorità di visualizzazione tag*: stringa separata da virgole della priorità di visualizzazione dei tag	tag_priority=customer, target
show_suppressed	Opzione di filtro *Mostra problemi soppressi*. Valori possibili: 1 - mostra; 2 - non mostrare.	show_suppressed=1
acknowledgement_status	Opzione di filtro *Stato di riconoscimento*. Valori possibili: 0 - tutti; 1 - non riconosciuti; 2 - riconosciuti.	acknowledgement_status=0
acknowledged_by_me	Opzione di filtro *Da me*; supportata solo con `acknowledgement_status=2`. Valori possibili: 0 - disabilitato; 1 - abilitato.	acknowledged_by_me=1
compact_view	Opzione di filtro *Vista compatta*. Valori possibili: 0 - disabilitato; 1 - abilitato.	compact_view=1

Parameter	Description	Example
highlight_row	Opzione di filtro *Evidenzia l'intera riga*. Valori possibili: 0 - disabilitato; 1 - abilitato.	highlight_row=1
filter_name	Opzione delle proprietà del filtro *Nome*: stringa libera	filter_name=Databases
filter_show_counter	Opzione delle proprietà del filtro *Mostra numero di record*. Valori possibili: 0 - disabilitato; 1 - abilitato.	filter_show_counter=1
filter_custom_time	Opzione delle proprietà del filtro *Imposta periodo di tempo personalizzato*. Valori possibili: 0 - disabilitato; 1 - abilitato.	filter_custom_time=1
sort	Colonna in base alla quale ordinare. Valori possibili: clock - ordina per la colonna *Ora*; host - ordina per la colonna *Host*; severity - ordina per la colonna *Gravità*; name - ordina per la colonna *Problema*.	sort=clock
sortorder	Ordine di ordinamento dei risultati. Valori possibili: DESC - decrescente; ASC - crescente.	sortorder=DESC
age_state	Opzione di filtro *Età inferiore a*; supportata solo con `show`=3. Valori possibili: 0 - disabilita il parametro age; 1 - abilita il parametro age.	age_state=1
age	Opzione di filtro *Età inferiore a*: integer, numero di giorni; supportata solo con `age_state`=1 e `show`=3.	age=7
groupids	Opzione di filtro *Gruppi host*: array di ID dei gruppi host.	groupids[]=4
hostids	Opzione di filtro *Host*: array di ID host	hostids[]=10084
triggerids	Opzione di filtro *Trigger*: array di ID trigger	triggerids[]=22382
show_timeline	Opzione di filtro *Mostra timeline*; non supportata con `compact_view`=1. Valori possibili: 0 - non mostrare; 1 - mostra.	show_timeline=1
details	Opzione di filtro *Mostra dettagli*. Valori possibili: 0 - non mostrare; 1 - mostra.	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 relativo (ad esempio, now-1m); supportato solo con `filter_custom_time`=1	to=now

Vedere anche: opzioni di filtro della pagina Problemi.

Modalità chiosco

La modalità chiosco nelle pagine del frontend supportate può essere attivata utilizzando i parametri URL. Ad esempio, nelle dashboard:

- `/zabbix.php?action=dashboard.view&kiosk=1` - attiva la modalità chiosco
- `/zabbix.php?action=dashboard.view&kiosk=0` - attiva la modalità normale

Presentazione

È possibile attivare uno slideshow nella dashboard:

- `/zabbix.php?action=dashboard.view&slideshow=1` - attiva la presentazione

7 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 è di 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 utilizzata come nome del cookie di sessione del frontend Zabbix.

Predefinito: zbx_sessionid

- ZBX_DATA_CACHE_TTL

Timeout TTL in secondi utilizzato per invalidare la cache dei dati della risposta di Vault. Impostare 0 per disabilitare la memorizzazione nella cache della risposta di Vault.

Predefinito: 60

- SUBFILTER_VALUES_PER_GROUP

Numero di valori del sottofiltro per gruppo (ad esempio, nel sottofiltro di ultimi dati).

Predefinito: 1000

- ZBX_MAX_WIDGET_LINES

Numero massimo di righe del widget da visualizzare.

Predefinito: 1000

8 Creare il tuo tema

Panoramica

Per impostazione predefinita, Zabbix fornisce una serie di temi predefiniti. Potresti seguire la procedura passo passo qui fornita per creare il tuo tema. Sentiti libero di condividere il risultato del tuo lavoro con la comunità Zabbix se hai creato qualcosa di carino

Passo 1

Per definire il tuo tema dovrai creare un file CSS e salvarlo la cartella `assets/styles/` (ad esempio, *custom-theme.css*). Puoi copia i file da un tema diverso e crea il tuo tema basato su di esso o ricominciare da zero.

Passo 2

Aggiungi il tuo tema all'elenco dei temi restituiti dall'APP::getThemes() metodo. Puoi farlo sovrascrivendo il metodo ZBase::getThemes() in la classe APP. Questo può essere fatto aggiungendo il codice seguente prima di parentesi di chiusura in *include/classes/core/APP.php*:

```
funzione statica pubblica getThemes() {
    return array_merge(parent::getThemes(), [
        'tema-personalizzato' => _('Tema personalizzato')
    ]);
}
```

::: nota importante Nota che il nome che specifichi all'interno del primo la coppia di virgolette deve corrispondere al nome del file del tema senza estensione. :::

Per aggiungere più temi, basta elencarli sotto il primo tema, per esempio:

```
funzione statica pubblica getThemes() {
    return array_merge(parent::getThemes(), [
        'tema-personalizzato' => _('Tema personalizzato'),
        'un altro tema' => _('Un altro tema'),
        'onemoretheme' => _('Un altro tema')
    ]);
}
```

Nota che ogni tema tranne l'ultimo deve avere una virgola finale.

Per modificare i colori del grafico, è necessario aggiungere la voce nel file *grafico_tema* tabella del database.

Passaggio 3

Attiva il nuovo tema.

Nel frontend Zabbix, puoi impostare questo tema come predefinito o cambiare il tuo tema nel profilo utente.

Goditi il nuovo look!

9 Modalità di debug

Panoramica

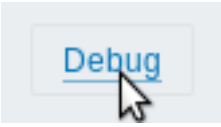
La modalità di debug può essere utilizzata per diagnosticare problemi di prestazioni con le pagine di frontend.

Configurazione

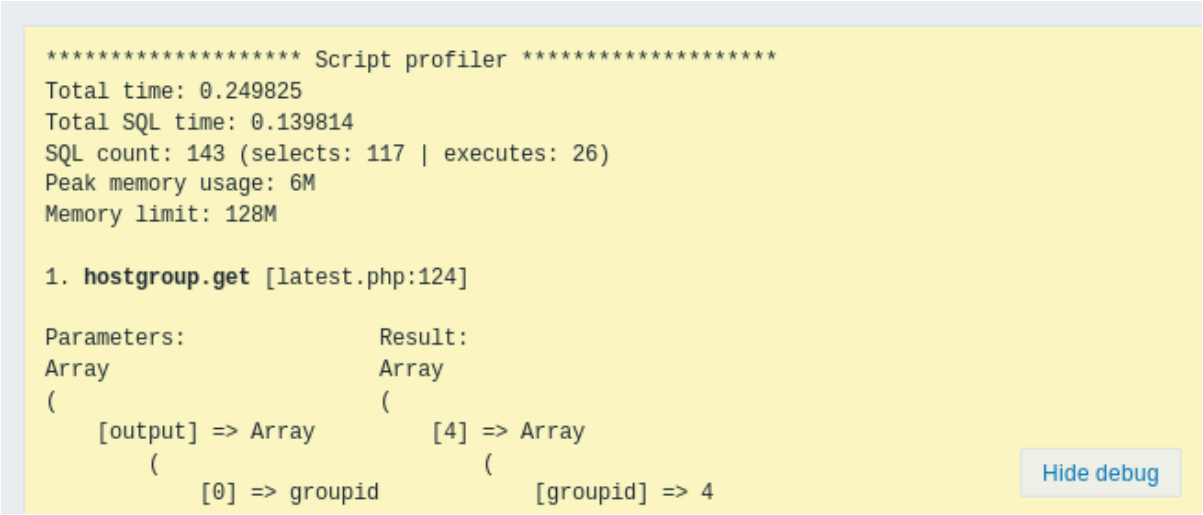
La modalità di debug può essere attivata per singoli utenti che appartengono a un gruppo di utenti:

- durante la configurazione di un gruppo di utenti;
- durante la visualizzazione dei gruppi di utenti configurati.

Quando la *modalità di debug* è abilitata per un gruppo di utenti, i relativi utenti vedranno un pulsante *Debug* nell'angolo inferiore destro 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 cercare la causa principale del problema.

Warning:
L'abilitazione della *modalità di debug* influisce negativamente sulle prestazioni del frontend.

10 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)
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

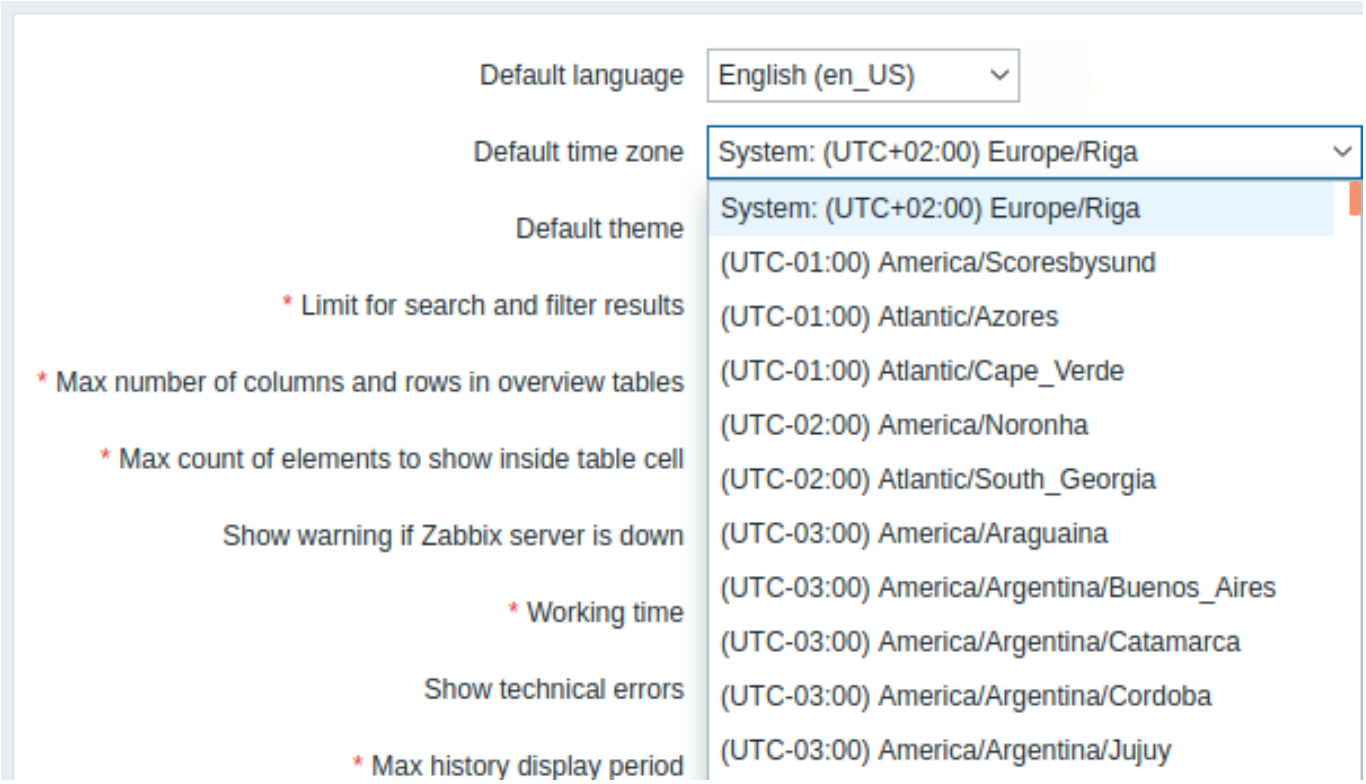
[1] Quando `HttpOnly` è impostato su 'true', il cookie sarà accessibile solo tramite il protocollo HTTP. Ciò significa che il cookie non sarà accessibile dai linguaggi di scripting, come JavaScript. Questa impostazione può contribuire efficacemente a ridurre il furto di identità tramite attacchi XSS (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.

11 Fusi orari

Panoramica

Il fuso orario del frontend può essere impostato globalmente nel frontend e regolato per i singoli utenti.



Se è selezionato *System*, per il frontend verrà utilizzato il fuso orario del web server (incluso il valore di 'date.timezone' di php.ini, se impostato), mentre Zabbix server utilizzerà il fuso orario della macchina su cui è in esecuzione.

Note:
Zabbix server 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 di tempo entro cui vengono inviate le notifiche (vedere l'impostazione "When active" nella configurazione dei media dell'utente).

La scelta del fuso orario non influisce sul formato di data/ora del frontend. Al contrario, è possibile regolare la lingua dell'interfaccia (durante l'installazione oppure nelle impostazioni utente) - selezionando *English (en_US)* verrà abilitato anche il formato data/ora statunitense nel frontend.

Configurazione

Il fuso orario globale:

- può essere impostato manualmente durante l'installazione del frontend
- può essere modificato in *Amministrazione* → *Generale* → *GUI*

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 gli intervalli di pianificazione.

12 Reimpostazione della password

Panoramica Questa sezione descrive i passaggi per reimpostare le password degli utenti in Zabbix.

Passaggi Rivolgiti al tuo amministratore Zabbix se hai dimenticato la password di Zabbix e non riesci ad accedere.

Un utente Super administrator può modificare le password di tutti gli utenti nel modulo di configurazione dell'utente.

Se un Super administrator 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 admin (sostituisci 'Admin' con il nome utente appropriato del Super admin):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Dopo aver eseguito questa query, la password dell'utente sarà impostata su *zabbix*. Assicurati di modificare la password predefinita al primo accesso.

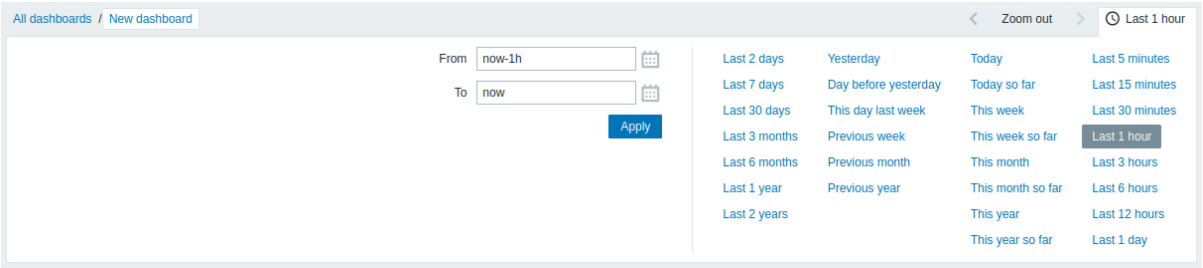
13 Selettori del periodo di tempo e dell'host

Panoramica

Questa pagina descrive i selettori di periodo temporale e di host disponibili nelle dashboard. Questi selettori consentono di regolare dinamicamente i dati visualizzati in tutti i widget della dashboard compatibili senza modificare le impostazioni dei singoli widget.

Selettore dell'intervallo di tempo

Il selettore dell'intervallo di tempo consente di selezionare i periodi richiesti più spesso con un solo clic del mouse. Può essere espanso o compresso facendo clic sul pulsante del selettore dell'intervallo di tempo nell'angolo superiore destro della dashboard.



Opzioni come *Oggi*, *Questa settimana*, ecc. mostrano l'intero periodo, incluse le ore/i giorni futuri. Opzioni come *Oggi finora*, *Questa settimana finora*, ecc. mostrano solo le ore trascorse.

Una volta selezionato un periodo, può essere spostato avanti e indietro nel tempo facendo clic sui pulsanti freccia . Il pulsante *Riduci zoom* consente di ridurre lo zoom del periodo del 50% in ciascuna direzione.

Note:
Per i grafici (tranne quelli con intervallo di tempo *Personalizzato*), è anche possibile selezionare l'intervallo di tempo visualizzato evidenziando un'area nel grafico con il pulsante sinistro del mouse. Una volta rilasciato il pulsante sinistro del mouse, il grafico eseguirà lo zoom sull'area evidenziata. È anche possibile ridurre lo zoom facendo doppio clic nel grafico.

I campi *Da/A* mostrano il periodo selezionato usando la sintassi del tempo assoluto (nel formato `Y-m-d H:i:s`) oppure la sintassi del tempo relativo. Un intervallo di tempo relativo può contenere una o più operazioni matematiche (- o +), ad esempio `now-1d` oppure `now-1d-2h+5m`.

Sono supportate le seguenti abbreviazioni di tempo relativo:

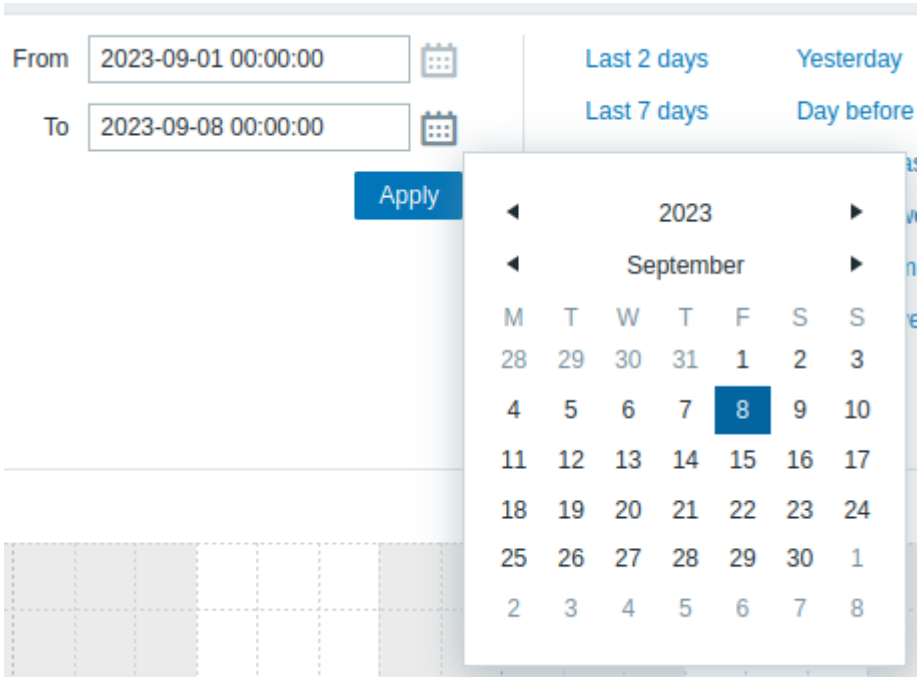
- `now`
- `s` (secondi)
- `m` (minuti)

- `h` (ore)
- `d` (giorni)
- `w` (settimane)
- `M` (mesi)
- `y` (anni)

La precisione è supportata nel selettore dell'intervallo di tempo (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 intervallo di tempo usando il selettore di data. Per aprirlo, fare clic sull'icona del calendario accanto ai campi *Da/A*.



Note:
All'interno del selettore di data, è possibile spostarsi tra anno/mese/data usando `Tab`, `Shift+Tab` e i tasti freccia della tastiera. Premendo `Enter` si conferma 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.



17 Best practice

Panoramica Questa sezione illustra le best practices per la configurazione di Zabbix.

Sebbene queste pratiche non siano obbligatorie per il funzionamento di Zabbix, la loro implementazione è altamente consigliata per un'esperienza di utilizzo ottimale e sicura.

1 Best practices per la 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.

Codifica UTF-8

UTF-8 è l'unica codifica supportata da Zabbix. È risaputo che funziona senza alcuna falla di sicurezza. Gli utenti devono essere consapevoli che esistono problemi di sicurezza noti se si utilizzano altre codifiche.

Percorsi dei programmi di installazione di Windows

Quando si utilizzano i programmi di installazione di Windows, si consiglia di utilizzare i percorsi predefiniti forniti dal programma. L'utilizzo 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 utilizzare le funzioni macro invece delle macro semplici negli script globali definiti dall'utente, poiché le macro non vengono sottoposte automaticamente all'escape.

Avvisi di sicurezza di Zabbix e database CVE

Vedi *Avvisi di sicurezza di Zabbix e database CVE*.

Modelli email HTML

Quando si creano o modificano i modelli di messaggio utilizzati per le email HTML, racchiudere sempre ogni macro con la 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'uso di `htmlencode()` garantisce che tutti i caratteri HTML presenti nei valori delle macro vengano sottoposti a escape e impedisce l'iniezione di HTML nelle notifiche (ad esempio, un utente malintenzionato che inserisce un link malevolo/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.

1 Controllo degli accessi

Panoramica

Questa sezione contiene le best practices per impostare il controllo degli accessi in modo sicuro.

Principio dei privilegi minimi

Gli account utente dovrebbero sempre essere eseguiti 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 server/proxy/agente Zabbix dovrebbero avere solo i privilegi essenziali per l'esecuzione delle 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 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 senza privilegi. Ad esempio, mentre *Avvisi* → *Script* è disponibile solo per gli utenti *Super Admin*, gli script possono essere recuperati anche tramite l'API Zabbix. In questo caso, limitare i permessi degli script ed escludere informazioni sensibili dagli script (ad esempio, le credenziali di accesso) può aiutare a evitare di esporre 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 deve 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 abbandonare i privilegi e passare 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 caricherà le impostazioni SSL da un percorso che può essere modificato da utenti senza privilegi, 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 di script globali sul server Zabbix può essere disabilitata impostando EnableGlobalScripts=0 nella configurazione del server;
- l'esecuzione di script globali sul proxy Zabbix è disabilitata per impostazione predefinita (può essere abilitata impostando EnableRemoteCommands=1 nella configurazione del proxy);
- l'esecuzione di script globali su Zabbix agent è disabilitata per impostazione predefinita (può essere abilitata aggiungendo un parametro AllowKey=system.run[<command>,*] per ogni comando consentito nella configurazione dell'agente);
- l'autenticazione HTTP dell'utente può essere disabilitata impostando `$ALLOW_HTTP_AUTH=false` nel file di configurazione del frontend (zabbix.conf.php). Si noti che la reinstallazione del frontend (eseguendo setup.php) rimuoverà questo parametro.

Accesso ai percorsi UNC su Windows da parte di Zabbix agent

Gli agent Zabbix su Windows seguono i percorsi UNC (condivisioni SMB come `\\server\share\file.txt`) negli item come `vfs.file.*`, `vfs.dir.*`, `modbus.get` e `perf_counter*`. In alcuni contesti questo 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 malevola a Zabbix agent può esporre l'hash NTLM al server del richiedente. Se necessario, gli utenti possono mitigare questo rischio con i parametri di configurazione AllowKey e DenyKey.

1 Protezione di MySQL/MariaDB

Panoramica

Questa sezione contiene le best practices per la protezione di un database MySQL/MariaDB.

> **Note:**
> Per una configurazione di base, consultare 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 modificarne la struttura durante l'aggiornamento di Zabbix.

Per migliorare la sicurezza, si consiglia di creare ruoli e utenti aggiuntivi per il database con privilegi minimi. Questi ruoli e utenti devono essere configurati in base al principio del privilegio minimo, ovvero, devono disporre solo dei privilegi essenziali per l'esecuzione delle funzioni previste.

Creazione dei ruoli utente

Creare i seguenti ruoli con i privilegi corrispondenti:

- **zbx_srv** - ruolo per eseguire Zabbix server e proxy:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - ruolo per eseguire il frontend e l'API di Zabbix:

```
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 che il database è stato creato, 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';
-- Per MariaDB: saltare la riga successiva (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>` secondo necessità.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- Per MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Ad esempio, per creare e assegnare il ruolo per l'esecuzione del server e del proxy Zabbix:

```
CREATE USER 'usr_srv'@'localhost' IDENTIFIED BY 'password';
GRANT 'zbx_srv' TO 'usr_srv'@'localhost';
SET DEFAULT ROLE ALL TO 'usr_srv'@'localhost';
FLUSH PRIVILEGES;
```

2 Protezione di PostgreSQL/TimescaleDB

Panoramica

Questa sezione contiene le best practices per la protezione di un database PostgreSQL.

> **Note:**
> Per una configurazione di base, consultare 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 modificarne la struttura durante l'aggiornamento di Zabbix.

Per migliorare la sicurezza, si consiglia di configurare un modello di utilizzo sicuro dello schema e di creare ruoli e utenti aggiuntivi del database con privilegi minimi. Questi ruoli e utenti devono essere configurati in base al principio del privilegio minimo, ovvero, devono disporre solo dei privilegi essenziali per l'esecuzione delle 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 stabilita 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:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Nota: search_path deve puntare allo schema "zabbix":
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Dopo aver configurato il database, procedere con la 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, 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>, <role> e <password> secondo necessità.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Ad esempio, per creare e assegnare il ruolo per l'esecuzione del server e del proxy Zabbix:

```
CREATE USER usr_srv WITH ENCRYPTED password '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 di proprietario del database.

Se lo schema zabbix di PostgreSQL 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
```

2 Crittografia

Panoramica

Questa sezione contiene le best practices per impostare la crittografia in modo sicuro.

Configurazione SSL per il frontend Zabbix

Sui sistemi basati su RHEL, installa il pacchetto mod_ssl:

```
dnf install mod_ssl
```

Crea una directory per le chiavi SSL:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Crea il certificato SSL:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Compila i prompt in modo appropriato. La riga più importante è quella che richiede il Common Name. Devi inserire il nome di dominio che desideri associare al tuo server. Se non hai un nome di dominio, puoi 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 []:
```

Modifica 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
```

Riavvia il servizio Apache per Applica le modifiche:

```
systemctl restart httpd.service
```

3 Server web

Panoramica

Questa sezione contiene le best practices per configurare il server web 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 dalla 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 Zabbix dagli attacchi di downgrade del protocollo, consigliamo di abilitare la policy HSTS sul server web.

Per abilitare la policy HSTS per il frontend Zabbix nella configurazione di Apache, segui questi passaggi:

1. Individua il file di configurazione del tuo host virtuale:

   - `/etc/httpd/conf/httpd.conf` sui sistemi basati su RHEL
   - `/etc/apache2/sites-available/000-default.conf` su Debian/Ubuntu

2. Aggiungi la seguente direttiva al file di configurazione del tuo host virtuale:

```
<VirtualHost *:*>
Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Riavvia il servizio Apache per applicare le modifiche:

```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Applicazione dei cookie di sessione Secure e SameSite in Zabbix

Durante la configurazione di Zabbix, è essenziale applicare gli attributi Secure e SameSite ai cookie di sessione per migliorare la sicurezza e prevenire gli attacchi cross-site request forgery (CSRF). Tuttavia, l'applicazione di `SameSite=Strict` può causare problemi in alcuni scenari, ad esempio:

- I widget URL della dashboard mostrano "utente non autenticato" quando incorporano iframe dello stesso dominio.
- Gli utenti che accedono alla dashboard tramite HTTP invece che HTTPS possono riscontrare problemi di accesso.
- Impossibilità di condividere URL verso sezioni specifiche del menu di Zabbix o verso host specifici.

Per mitigare questi problemi, gli utenti dovrebbero avere la possibilità di regolare il criterio SameSite.

1. Cookie Secure

L'impostazione del flag `secure` garantisce che i cookie vengano trasmessi solo tramite HTTPS, evitando l'esposizione su connessioni non crittografate.

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; # Sostituire ~ con 'zbx_session' per maggiore specificità
```

Abilitazione di Content Security Policy (CSP) sul server web

Per proteggere il frontend di Zabbix da Cross Site Scripting (XSS), iniezione di dati e attacchi simili, consigliamo di abilitare Content Security Policy sul server web. Per farlo, configura il server web in modo che restituisca l'header HTTP.

> **Attention:**
> La seguente configurazione dell'header 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'header CSP se, ad esempio, stai 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 utilizzi il metodo di autenticazione a più fattori Duo Universal Prompt, assicurati di aggiungere "duo.com" alla direttiva CSP nel file di configurazione del tuo virtual host.

Per abilitare CSP per il frontend di Zabbix nella configurazione di Apache, segui questi passaggi:

1. Individua il file di configurazione del tuo virtual host:

- `/etc/httpd/conf/httpd.conf` sui sistemi basati su RHEL
- `/etc/apache2/sites-available/000-default.conf` su Debian/Ubuntu

2. Aggiungi la seguente direttiva al file di configurazione del tuo virtual host:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

3. Riavvia il servizio Apache per applicare le modifiche:

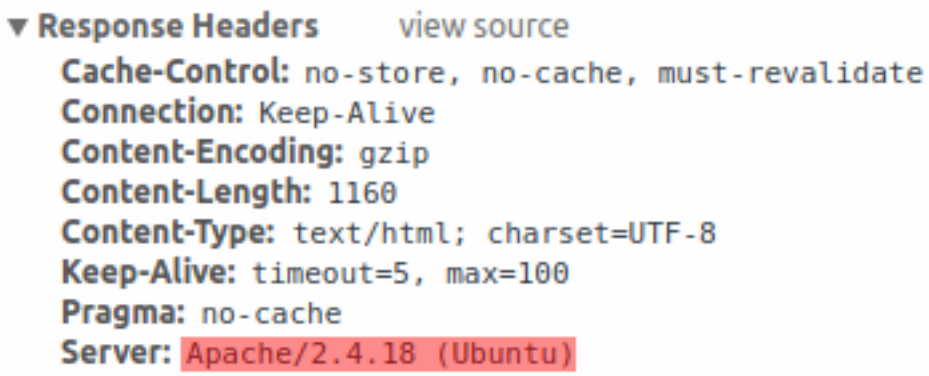
```
#### Sui sistemi basati su RHEL:
systemctl restart httpd.service

#### Su Debian/Ubuntu
systemctl restart apache2.service
```

Disabilitazione dell'esposizione delle informazioni del server web

Per migliorare la sicurezza, si consiglia di disabilitare tutte le firme del server web.

Per impostazione predefinita, il server web espone la firma 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 server web affinché le modifiche al file di configurazione vengano applicate.

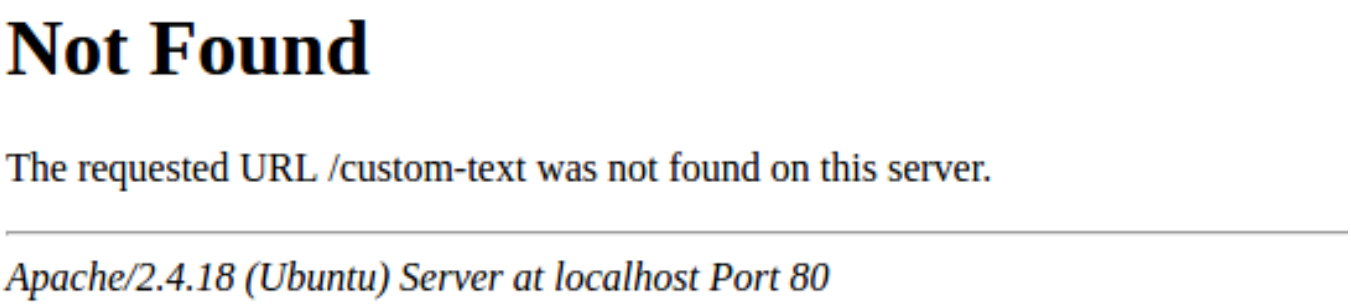
Per una maggiore sicurezza, è possibile utilizzare lo strumento *mod_security* con Apache (pacchetto *libapache2-mod-security2*). Questo strumento consente di rimuovere la firma del server anziché rimuoverne solo la versione. La firma del server può essere modificata impostando "SecServerSignature" su qualsiasi valore desiderato dopo l'installazione di *mod_security*.

Consultare la documentazione del server web per informazioni su come rimuovere/modificare le firme software.

Disabilitazione delle pagine di errore predefinite del server web

Per evitare l'esposizione di informazioni, si consiglia di disabilitare le pagine di errore predefinite.

Per impostazione predefinita, un server web utilizza pagine di errore integrate:



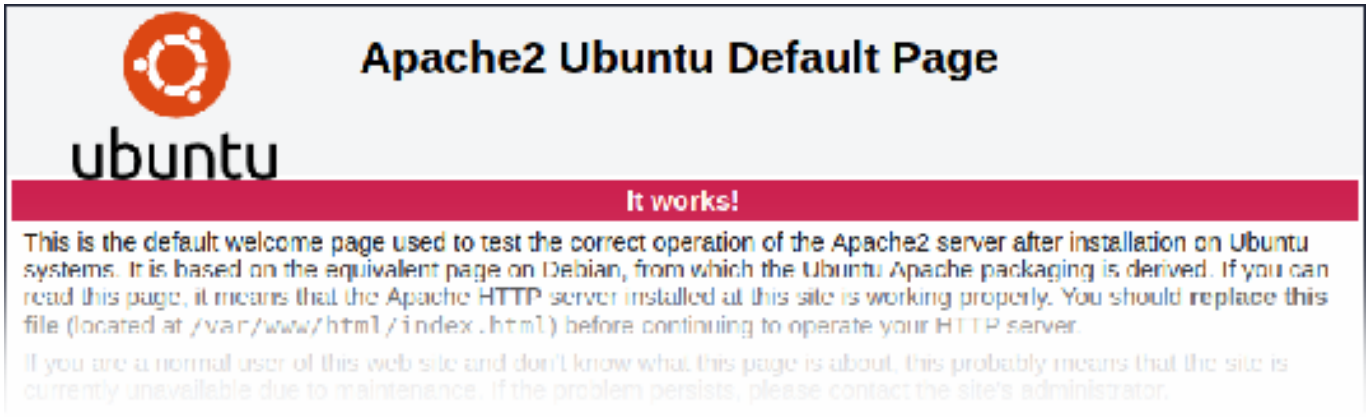
Queste pagine di errore predefinite devono essere sostituite/rimosse. Ad esempio, la direttiva "ErrorDocument" può essere utilizzata per definire una pagina/testo di errore personalizzato per il server web Apache.

Consultare la documentazione del server web per informazioni su come sostituire/rimuovere le pagine di errore predefinite.

Rimozione della pagina di test del server web

Per evitare la divulgazione di informazioni, si consiglia di rimuovere la pagina di test del server web.

Per impostazione predefinita, la radice web del server web Apache contiene la pagina di test `index.html`:



Consultare la documentazione del server web per informazioni su come rimuovere le pagine di test predefinite.

Imposta l'intestazione di risposta HTTP X-Frame-Options

Di default, Zabbix è configurato con il parametro *Usa l'intestazione HTTP X-Frame-Options* impostato su `SAMEORIGIN`. Ciò significa che il contenuto può essere caricato solo in un frame che abbia la stessa origine della pagina stessa.

Gli elementi del frontend di Zabbix che estraggono il contenuto da URL esterni (ovvero, il widget URL dashboard) visualizzano il contenuto recuperato in una sandbox con tutte le restrizioni del sandboxing abilitate.

Queste impostazioni migliorano la sicurezza del frontend di Zabbix e forniscono protezione contro gli attacchi XSS e clickjacking. Gli utenti *Super admin* possono modificare i parametri *Usa sandboxing iframe* e *Usa l'intestazione HTTP X-Frame-Options* in base alle proprie esigenze. Si prega di valutare attentamente i rischi e i benefici prima di modificare le impostazioni predefinite. Si sconsiglia di disattivare completamente il sandboxing degli iframe o l'intestazione HTTP X-Frame-Options.

Nascondere il file con l'elenco delle password comuni

Per aumentare la complessità degli attacchi brute force alle password, si consiglia di limitare l'accesso al file `ui/data/top_passwords.txt`. Questo file contiene un elenco delle password più comuni e specifiche per il contesto e impedisce agli utenti di impostarle (se il parametro *Evita password facili da indovinare* è abilitato nella policy password).

Per limitare l'accesso al file `top_passwords.txt`, modifica la configurazione del tuo server web.

Su Apache, l'accesso ai file può essere limitato utilizzando il file `.htaccess`:

```
<Files "top_passwords.txt">
Order Allow,Deny
Deny from all
</Files>
```

Su NGINX, l'accesso ai file può essere limitato utilizzando la direttiva `location`:

```
location = /data/top_passwords.txt {
deny all;
return 404;
}
```

2 Best practices per la configurazione

Panoramica

Questa sezione illustra le migliori pratiche per configurare Zabbix in modo da ottenere prestazioni ottimali e facilità d'uso. Le raccomandazioni si basano sui consigli degli sviluppatori di Zabbix e sull'esperienza pratica dei formatori e degli ingegneri del supporto Zabbix.

Ogni installazione di Zabbix è unica e alcune di queste linee guida potrebbero non essere adatte alla tua configurazione specifica. Tuttavia, si consiglia di cercare di attenersi il più possibile a queste linee guida per evitare problemi comuni e potenziali.

Note:
Se ritieni che questa pagina possa essere migliorata, ci farebbe piacere ricevere il tuo parere! Evidenzia il testo in questione e premi **ctrl+Enter** per segnalare un errore o condividere il tuo feedback.

Host e elementi Definizione di un host

In Zabbix, un host non è una macchina fisica o un dispositivo, ma un'entità logica. Ai fini del monitoraggio, è possibile creare host separati per un database o, ad esempio, una macchina virtuale. In alternativa, è possibile creare un host generico *Laptop di John* e monitorare tutte le metriche sotto quell'host.

La procedura consigliata consiste nel creare un host separato per ogni istanza indipendente, come una macchina virtuale, un database, un container o uno switch di rete. Utilizzando questo approccio, sarà possibile:

1. Evitare confusione nei dati di monitoraggio, avendo item, trigger e notifiche di avviso separati per ciascun host.
2. Ottimizzare i livelli di accesso degli utenti. È possibile configurare i ruoli utente per concedere l'accesso alla visualizzazione e/o alla configurazione solo di host specifici. Vedere anche il principio del privilegio minimo.

Host con item duplicati

Se hai diversi host simili, come *Network switch 1* e *Network switch 2*, Zabbix offre più modi per ricreare rapidamente l'host. Puoi semplicemente clonare un host con tutte le sue metriche premendo il pulsante Clone, ma in questo caso, per aggiornare un item in seguito, dovrai farlo manualmente su ciascun host.

La procedura consigliata consiste nel creare un template con tutte le metriche richieste, ad esempio *Network switch template*. Quindi raggruppa gli host simili in un gruppo di host; nell'esempio sopra potrebbe essere *Network switches*. A questo punto, nella sezione *Data Collection -> Hosts* puoi filtrare tutti gli host per gruppo di host e usare il pulsante *Mass update* per collegare il template a tutti i tuoi switch di rete.

Item dipendenti

Per ridurre al minimo il numero di richieste all'entità di destinazione, Zabbix consente di creare item master e item dipendenti. In questo caso, l'item master raccoglie un ampio insieme di informazioni con una singola richiesta. Successivamente, gli item dipendenti possono essere configurati per estrarre specifiche porzioni di dati da tale raccolta tramite il preprocessing e memorizzarle come metriche individuali.

Ad esempio, l'item master può raccogliere una risposta JSON o XML contenente più metriche oppure eseguire una query al database che restituisce più colonne di dati (ad esempio, numero di connessioni aperte, connessioni interrotte, numero massimo consentito di connessioni simultanee e totale cumulativo delle connessioni dall'avvio), mentre gli item dipendenti analizzeranno e memorizzeranno separatamente ciascun valore richiesto.

La procedura consigliata per questa configurazione consiste nello scartare la cronologia dell'item master subito dopo la raccolta e conservare solo i dati degli item dipendenti.

Server e proxy

Se tutti gli host si trovano nella stessa rete locale del server Zabbix e non vi sono problemi di scalabilità o prestazioni, potrebbe non essere necessario un proxy. In ambienti più grandi o complessi, il monitoraggio diretto degli host con il server Zabbix potrebbe non essere sufficiente. L'aggiunta di un proxy e l'assegnazione di una parte degli host a tale proxy consentono una distribuzione del carico più uniforme.

La procedura consigliata è aggiungere un proxy Zabbix quando:

1. Si stanno monitorando più host utilizzando vari metodi di raccolta delle metriche dietro un firewall. Il proxy raccoglierà i dati dagli host e li inoltrerà al server Zabbix, riducendo la necessità di aprire più porte del firewall.
2. Si stanno monitorando sedi remote, filiali e/o reti. In caso di interruzione della rete tra il server Zabbix e le sedi remote, i proxy Zabbix distribuiti nelle sedi remote continueranno a raccogliere i dati e invieranno i dati raccolti al server Zabbix non appena la connessione di rete verrà ripristinata.
3. Si dispone di una distribuzione su larga scala e si desidera ridurre il carico sul server Zabbix e migliorare le prestazioni. La definizione di distribuzione su larga scala è molto ampia e dipende non solo dal numero di host, ma anche dal numero di valori raccolti al secondo.

Macro segrete

Potresti voler utilizzare le macro utente segrete come testo segreto oppure come macro secret vault.

Per una maggiore sicurezza quando si utilizzano macro secret vault, si consiglia di configurare i valori delle macro in modo che vengano recuperati in modo indipendente da Zabbix server e dai proxy Zabbix. Per impostazione predefinita, i valori delle macro segrete vengono recuperati da Zabbix server e propagati ai proxy Zabbix.

18 API

Panoramica L'API di Zabbix consente di recuperare e modificare programmaticamente la configurazione di Zabbix e fornisce accesso ai dati storici. È ampiamente utilizzata per:

- creare nuove applicazioni che funzionino con Zabbix;
- integrare Zabbix in software di terze parti;
- automatizzare attività di routine.

L'API di Zabbix è un'API basata su HTTP ed è distribuita come parte del frontend web. Utilizza il protocollo JSON-RPC 2.0, il che significa due cose:

- l'API consiste in un insieme di metodi separati;
- le richieste e le risposte tra i client e l'API sono codificate utilizzando 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 applicazioni Python, vedere libreria Python per Zabbix.

> **Note:**
> L'accesso degli utenti in Zabbix, inclusi sia i dati di configurazione sia quelli storici, dipende dal tipo di utente, dal ruolo utente assegnato e dai gruppi di utenti.

Struttura L'API è composta da una serie di metodi che sono nominalmente raggruppati in API separate. Ciascuno di questi metodi esegue un'attività specifica. Ad esempio, il metodo `host.create` appartiene all'API *host* e viene utilizzato per creare nuovi host. Storicamente, le API sono talvolta chiamate "classi".

> **Note:**
> La maggior parte delle API contiene almeno quattro metodi: `get`, `create`, `update` e `delete`, rispettivamente per il recupero, la creazione, l'aggiornamento e l'eliminazione dei dati, ma alcune API possono fornire un insieme di metodi completamente diverso.

Esecuzione delle richieste Una volta configurato il frontend, è possibile utilizzare richieste HTTP remote per chiamare l'API. Per farlo, è necessario inviare richieste HTTP POST al file `api_jsonrpc.php` situato nella directory del frontend. Ad esempio, se il frontend di Zabbix è installato in `https://example.com/zabbix`, una richiesta HTTP per chiamare il metodo `apiinfo.version` può apparire così:

```
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'header `Content-Type` impostato su uno di questi valori: `application/json-rpc`, `application/json` oppure `application/jsonrequest`.

L'oggetto della richiesta deve contenere le seguenti proprietà:

- `jsonrpc` - la versione del protocollo JSON-RPC utilizzata dall'API (l'API di Zabbix implementa JSON-RPC versione 2.0);
- `method` - il metodo API che viene chiamato;
- `params` - i parametri che verranno passati al metodo API;
- `id` - un identificatore arbitrario della richiesta (se omesso, l'API tratta la richiesta come una notifica).

Se la richiesta è corretta, la risposta restituita dall'API dovrebbe apparire così:

```
{
    "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.

Flusso di lavoro di esempio La sezione seguente ti guiderà attraverso alcuni esempi di utilizzo in maggiore dettaglio.

Autenticazione Per accedere a qualsiasi dato in Zabbix, è necessario:

- utilizzare un token API esistente (creato nel frontend di Zabbix o usando la Token API);
- utilizzare un token di autenticazione ottenuto con il metodo user.login.

Ad esempio, se si desidera ottenere un nuovo token di autenticazione effettuando l'accesso come utente *Admin* standard, una richiesta JSON sarà 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":"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).

Recupero degli host Ora disponi di un token di autenticazione utente valido (rappresentato come variabile negli esempi seguenti) che può essere utilizzato per accedere ai dati in Zabbix. Ad esempio, puoi usare il metodo host.get per recuperare gli ID, i nomi 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. Invece di 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 dell'item appena creato, che può essere utilizzato 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 create*, può anche accettare array di oggetti e creare più item con una sola chiamata API.

Creazione di più trigger Pertanto, se i *metodi create* accettano array, è possibile aggiungere più trigger, ad esempio questi:

```
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":"Il carico del processore è
```

La risposta restituita in caso di esito positivo conterrà gli ID dei trigger appena creati:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Aggiornamento di un item Abilitare 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 Abilitare 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 non corretta 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, invece 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 in presenza di valori di input non corretti, di una sessione scaduta o nel tentativo di accedere a oggetti inesistenti. La tua applicazione dovrebbe essere in grado di gestire correttamente questi tipi di errori.

Versioni dell'API Per semplificare il versionamento dell'API, a partire da Zabbix 2.0.4, la versione dell'API corrisponde alla versione di Zabbix stessa. È possibile utilizzare il metodo apiinfo.version per conoscere 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 retrocompatibilità delle funzionalità all'interno di una versione principale. Quando apporta modifiche non retrocompatibili tra release principali, Zabbix di solito mantiene le vecchie funzionalità come deprecate nella release successiva, e le rimuove solo nella release seguente. Occasionalmente, Zabbix può rimuovere funzionalità tra release principali senza fornire alcuna retrocompatibilità. È importante non fare mai affidamento su funzionalità deprecate e migrare alle alternative più recenti il prima possibile.

> **Note:**
> È possibile seguire tutte le modifiche apportate all'API nel registro delle modifiche dell'API.

Approfondimenti Ora hai conoscenze sufficienti per iniziare a lavorare con la API di Zabbix, tuttavia non fermarti qui. Per ulteriori approfondimenti, ti consigliamo di consultare l'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

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 del livello di servizio e recupera informazioni dettagliate sugli SLA di qualsiasi servizio.

API del servizio

Raccolta dati L'API Zabbix ti consente di gestire la configurazione del tuo sistema di monitoraggio.

Host e gruppi di host

Gestisci gruppi di host, host e tutto ciò che li riguarda, incluse le interfacce dell'host, le macro dell'host e periodi di manutenzione.

API dell'host | API del gruppo di host | API dell'interfaccia di host | API della macro utente | API della mappa valori | API della manutenzione

Elementi

Definisci gli elementi da monitorare.

API dell'elemento

Trigger

Configura i trigger per ricevere notifiche sui problemi del tuo sistema. Gestisci le dipendenze tra trigger.

API del trigger

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

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

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 sull'API Recupera la versione dell'API Zabbix in modo che la tua applicazione possa utilizzare funzionalità specifiche della versione.

API delle Informazioni sull'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.

Proprietà	Tipo	Descrizione
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"
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 permette di recuperare gli alert in base ai parametri forniti.

> **Note:**
> Questo metodo è disponibile a qualunque tipo di utente. I permessi per la chiamate del metodo possono essere revocati nei settaggi del ruolo utene. Vedi User roles per maggiori 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.

Parametro	Tipo	Descrizione
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.
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 authentication ha le seguenti proprietà.

Property	Type	Description
authentication_type	integer	Autenticazione predefinita. Valori possibili: 0 - *(predefinito)* Interna; 1 - LDAP.
http_auth_enabled	integer	Autenticazione HTTP. Valori possibili: 0 - *(predefinito)* Disabilitata; 1 - Abilitata. Comportamento della proprietà: - *supportata* se `$ALLOW_HTTP_AUTH` è abilitato nel file di configurazione del frontend (*zabbix.conf.php*).
http_login_form	integer	Modulo di accesso predefinito. Valori possibili: 0 - *(predefinito)* Modulo di accesso Zabbix; 1 - Modulo di accesso HTTP. Comportamento della proprietà: - *supportata* se `$ALLOW_HTTP_AUTH` è abilitato nel file di configurazione del frontend (*zabbix.conf.php*).
http_strip_domains	string	Nome di dominio da rimuovere. Comportamento della proprietà: - *supportata* se `$ALLOW_HTTP_AUTH` è abilitato nel file di configurazione del frontend (*zabbix.conf.php*).
http_case_sensitive	integer	Accesso HTTP con distinzione tra maiuscole e minuscole. Valori possibili: 0 - Disattivato; 1 - *(predefinito)* Attivato. Comportamento della proprietà: - *supportata* se `$ALLOW_HTTP_AUTH` è abilitato nel file di configurazione del frontend (*zabbix.conf.php*).
ldap_auth_enabled	integer	Autenticazione LDAP. Valori possibili: 0 - *(predefinito)* 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 - *(predefinito)* Disabilitata; 1 - Abilitata.

Property	Type	Description
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.
passwd_check_rules	integer	Regole di controllo della password. Valori bitmap possibili: 0 - Controlla la lunghezza della password; 1 - Controlla se la password utilizza lettere latine maiuscole e minuscole; 2 - Controlla se la password utilizza cifre; 4 - Controlla se la password utilizza caratteri speciali; 8 - *(predefinito)* Controlla che la password non sia nell'elenco delle password comunemente usate e non contenga derivazioni della parola "Zabbix" o del nome, cognome o nome utente dell'utente. Questo è un campo bitmask; è accettata qualsiasi somma dei valori bitmap possibili (ad esempio, 15 per controllare 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 che ha effettuato l'accesso. Accetta secondi e unità di tempo con suffisso, con supporto per mesi e anni (3600s,60m,1h,1d,1M,1y). Valore minimo: 1h. Predefinito: 1h. Disponibile solo per il provisioning LDAP.
disabled_usrgrpid	ID	ID del gruppo utenti a cui assegnare l'utente deprovisionato. 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 gli utenti di tipo *Super admin*. Le autorizzazioni per richiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 action ha le seguenti proprietà.

Proprietà	Type	Descrizione
actionid	ID	ID dell'azione. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
esc_period	string	Durata predefinita del passo dell'operazione. Deve essere di almeno 60 secondi. Accetta secondi, un'unità di tempo con suffisso oppure una macro utente. Comportamento della proprietà: - *supportata* se `eventsource` è impostato su "evento creato da un trigger", "evento interno" oppure "evento creato all'aggiornamento dello stato del servizio"
eventsource	integer	Tipo di eventi che l'azione gestirà. Fare riferimento alla proprietà `source` dell'evento per un elenco dei tipi di evento supportati. 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 se 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 oppure no. 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"

Proprietà	Type	Descrizione
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 un'azione viene eseguita. Ha le seguenti proprietà.

Property	Type	Description
operationtype	integer	Tipo di operazione. Valori possibili: 0 - invia messaggio; 1 - script globale; 2 - aggiungi host; 3 - rimuovi host; 4 - aggiungi al gruppo host; 5 - rimuovi dal gruppo host; 6 - collega template; 7 - scollega template; 8 - abilita host; 9 - disabilita host; 10 - imposta la modalità di inventario dell'host; 13 - aggiungi tag host; 14 - rimuovi tag host. Valori possibili se `eventsource` di Action object è impostato su "event created by a trigger" o "event created on service status update": 0 - "send message"; 1 - "global script". Valori possibili se `eventsource` di Action object è impostato su "internal event": 0 - "send message". Property behavior: - *required*
esc_period	string	Durata di un passaggio di escalation in secondi. Deve essere maggiore di 60 secondi. Accetta secondi, un'unità di tempo con suffisso o una macro utente. Se impostato a 0 o 0s, verrà utilizzato il periodo di escalation predefinito dell'azione. Predefinito: 0s. Property behavior: - *supported* se `eventsource` di Action object è impostato su "event created by a trigger", "internal event" o "event created on service status update"

Property	Type	Description
esc_step_from	integer	Passaggio da cui iniziare l'escalation. Predefinito: 1. Property behavior: - *supported* se `eventsource` di Action object è impostato su "event created by a trigger", "internal event" o "event created on service status update"
esc_step_to	integer	Passaggio in cui terminare l'escalation. Predefinito: 1. Property behavior: - *supported* se `eventsource` di Action object è impostato su "event created by a trigger", "internal event" o "event created on service status update"
evaltype	integer	Metodo di valutazione della condizione dell'operazione. Valori possibili: 0 - *(predefinito)* And/Or; 1 - And; 2 - Or.
opcommand	object	Script globale da eseguire. Lo script globale deve avere definita la proprietà `scriptid`. Property behavior: - *required* se `operationtype` è impostato su "global script"
opcommand_grp	array	Gruppi host su cui eseguire gli script globali. I gruppi host devono avere definita la proprietà `groupid`. Property behavior: - *required* se `operationtype` è impostato su "global script" e `opcommand_hst` non è impostato
opcommand_hst	array	Host su cui eseguire gli script globali. Gli host devono avere definita la proprietà `hostid`. Property behavior: - *required* se `operationtype` è impostato su "global script" e `opcommand_grp` non è impostato
opconditions	array	Condizioni dell'operazione utilizzate per le azioni dei trigger. L'oggetto condizione dell'operazione è descritto in dettaglio di seguito.
opgroup	array	Gruppi host a cui aggiungere gli host. I gruppi host devono avere definita la proprietà `groupid`. Property behavior: - *required* se `operationtype` è impostato su "add to host group" o "remove from host group"
opmessage	object	Oggetto contenente i dati del messaggio inviato dall'operazione. L'oggetto messaggio dell'operazione è descritto in dettaglio di seguito. Property behavior: - *required* se `operationtype` è impostato su "send message"

Property	Type	Description
opmessage_grp	array	Gruppi utenti a cui inviare messaggi. I gruppi utenti devono avere definita la proprietà `usrgrpid`. Property behavior: - *required* se `operationtype` è impostato su ”send message” 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 ”send message” e `opmessage_grp` non è impostato
optemplate	array	Template da collegare agli host. I template devono avere definita la proprietà `templateid`. Property behavior: - *required* se `operationtype` è impostato su ”link template” o ”unlink template”
opinventory	object	Modalità di inventario da impostare per l’host. L’inventario deve avere definita la proprietà `inventory_mode`. Property behavior: - *required* se `operationtype` è impostato su ”set host inventory mode”
optag	array	Tag host da aggiungere o rimuovere. I tag devono avere definita la proprietà `tag`. La proprietà `value` è facoltativa. Property behavior: - *supported* se `operationtype` è impostato su ”add host tags” o ”remove host tags”.

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 condizione del filtro dell'azione definisce una condizione specifica che deve essere verificata prima di eseguire le operazioni dell'azione.

Property	Type	Description
conditiontype	integer	Tipo di condizione. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger": 0 - gruppo host; 1 - host; 2 - trigger; 3 - nome evento; 4 - gravità del trigger; 6 - periodo di tempo; 13 - template host; 16 - il problema è soppresso; 25 - tag evento; 26 - valore del tag evento. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da una regola di discovery": 7 - IP 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 di discovery. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato da autoregistrazione active agent": 20 - proxy; 22 - nome host; 24 - metadati host. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento interno": 0 - gruppo host; 1 - host; 13 - template host; 23 - tipo di evento; 25 - tag evento; 26 - valore del tag evento. Valori possibili se `eventsource` dell'oggetto Action è impostato su "evento creato durante l'aggiornamento dello stato del servizio": 25 - tag evento; 26 - valore del tag evento; 27 - servizio; 28 - nome del servizio. Comportamento della proprietà: - *required*
value	string	Valore con cui confrontare. Comportamento della proprietà: - *required*

Property	Type	Description
value2	string	Valore secondario con cui confrontare. Comportamento della proprietà: - *required* se `eventsource` dell'oggetto Action è impostato su "evento creato da un trigger", `conditiontype` è impostato su uno qualsiasi dei valori possibili per le azioni trigger e il tipo di condizione (vedi sotto) è "26" - *required* se `eventsource` dell'oggetto Action è impostato su "evento interno", `conditiontype` è impostato su uno qualsiasi dei valori possibili per le azioni interne e il tipo di condizione (vedi sotto) è "26" - *required* se `eventsource` dell'oggetto Action è impostato su "evento creato durante l'aggiornamento dello stato del servizio", `conditiontype` è impostato su uno qualsiasi dei valori possibili per le azioni di servizio e il tipo di condizione (vedi sotto) 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 verranno richieste in seguito.
operator	integer	Operatore della condizione. Valori possibili: 0 - *(predefinito)* uguale a; 1 - diverso da; 2 - contiene; 3 - non contiene; 4 - in; 5 - maggiore o uguale a; 6 - minore o uguale a; 7 - non 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.

I seguenti operatori e valori sono supportati per ciascun tipo di condizione.

Condition	Condition name	Supported operators	Expected value
0	Gruppo host	uguale a, diverso da	ID del gruppo host.
1	Host	uguale a, diverso da	ID host.
2	Trigger	uguale a, diverso da	ID trigger.
3	Nome evento	contiene, non contiene	Nome evento.
4	Gravità del trigger	uguale a, diverso da, maggiore o uguale a, minore o uguale a	Gravità del trigger. Fare riferimento alla proprietà `severity` del trigger per un elenco delle gravità trigger supportate.
5	Valore del trigger	uguale a	Valore del trigger. Fare riferimento alla proprietà `value` del trigger per un elenco dei valori trigger supportati.
6	Periodo di tempo	in, non in	Ora in cui l'evento è stato attivato come periodo di tempo.

Condition	Condition name	Supported operators	Expected value
7	IP host	uguale a, diverso da	Uno o più intervalli IP da controllare, separati da virgole. Fare riferimento alla sezione configurazione della network discovery per maggiori informazioni sui formati supportati degli intervalli IP.
8	Tipo di servizio rilevato	uguale a, diverso da	Tipo di servizio rilevato. Il tipo di servizio corrisponde al tipo del controllo di discovery utilizzato per rilevare il servizio. Fare riferimento alla proprietà `type` del controllo di discovery per un elenco dei tipi supportati.
9	Porta del servizio rilevato	uguale a, diverso da	Uno o più intervalli di porte, separati da virgole.
10	Stato della discovery	uguale a	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 perso.
11	Durata di uptime o downtime	maggiore o uguale a, minore o uguale a	Tempo che indica da quanto il oggetto rilevato si trova nello stato corrente, in secondi.
12	Valori ricevuti	uguale a, diverso da, maggiore o uguale a, minore o uguale a, contiene, non contiene	Valore restituito durante l'esecuzione di un controllo di discovery Zabbix agent, SNMPv1, SNMPv2 o SNMPv3.
13	Template host	uguale a, diverso da	ID del template collegato.
16	Il problema è soppresso	Sì, No	Nessun valore richiesto: l'uso dell'operatore "Sì" significa che il problema deve essere soppresso, "No" - non soppresso.
18	Regola di discovery	uguale a, diverso da	ID della regola di discovery.
19	Controllo di discovery	uguale a, diverso da	ID del controllo di discovery.
20	Proxy	uguale a, diverso da	ID del proxy.
21	Oggetto di discovery	uguale a	Tipo di oggetto che ha attivato l'evento di discovery. Valori possibili: 1 - host rilevato; 2 - servizio rilevato.
22	Nome host	contiene, non contiene, corrisponde, non corrisponde	Nome 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 a	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 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*.

Condition	Condition name	Supported operators	Expected value
25	Tag	uguale a, diverso da, contiene, non contiene	Tag evento.
26	Valore del tag	uguale a, diverso da, contiene, non contiene	Valore del tag evento.
27	Servizio	uguale a, diverso da	ID del servizio.
28	Nome del servizio	uguale a, diverso da	Nome del servizio.

action.create

Descrizione

`object action.create(object/array actions)`

Questo metodo consente di creare nuove azioni.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 azioni 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. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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.
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"

Parametro	Tipo	Descrizione
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*.

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 dei ruoli utente. Consultare Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono gli oggetti da esportare e il formato da utilizzare.

Parametro	Tipo	Descrizione
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: - *obbligatorio*
prettyprint	boolean	Rende l'output più leggibile aggiungendo l'indentazione. Valori possibili: `true` - aggiunge l'indentazione; `false` - *(predefinito)* non aggiunge l'indentazione.
options	object	Oggetti da esportare. L'oggetto `options` ha i seguenti parametri: `host_groups` - `(array)` ID dei gruppi di 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 supporto da esportare; `template_groups` - `(array)` ID dei gruppi di template da esportare; `templates` - `(array)` ID dei template da esportare. Gli utenti di tipo *Admin* e *User* possono esportare solo quegli oggetti per i quali dispongono del permesso di sola lettura o lettura-scrittura, nonché le immagini, ma non i tipi di supporto. Comportamento del parametro: - *obbligatorio*

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 i 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. Consultare Ruoli utente per ulteriori informazioni.

Parametri

`(object)` Parametri contenenti i dati da importare e le regole su come tali dati devono essere gestiti.

Parametro	Type	Descrizione
format	string	Formato della stringa serializzata. Valori possibili: `yaml` - YAML; `xml` - XML; `json` - JSON. Comportamento del parametro: - *required*
source	string	Stringa serializzata contenente i dati di configurazione. Comportamento del parametro: - *required*
rules	object	Regole su come devono essere importati gli oggetti nuovi ed esistenti. Gli utenti di tipo *Admin* possono importare solo quegli oggetti per i quali dispongono del permesso di *read-write*, nonché le mappe. Ad esempio, un host e le sue entità (items, triggers, grafici, ecc.) possono essere importati solo se il gruppo utenti dell'utente dispone del permesso per il gruppo host a cui apparterrà l'host importato. Le immagini e i tipi di supporto non possono essere importati dagli utenti di tipo *Admin*. Il parametro `rules` è descritto in dettaglio nella tabella seguente. Comportamento del parametro: - *required*

Attention:
Se non vengono fornite regole, la configurazione non verrà aggiornata.

L'oggetto `rules` supporta i seguenti parametri.

Parametro	Type	Descrizione
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`.
graphs	object	Regole su come importare i grafici. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi grafici; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i grafici esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i grafici 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`.
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`.
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`.

Parametro	Type	Descrizione
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`.
mediaTypes	object	Regole su come importare i tipi di supporto. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi tipi di supporto; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i tipi di supporto 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 che sono collegati all'host o al template in fase di importazione, ma non sono 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`.
templateDashboards	object	Regole su come importare le dashboard dei template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove dashboard dei template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le dashboard dei template esistenti verranno aggiornate; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le dashboard dei template non presenti nei dati importati verranno eliminate dal database; predefinito: `false`.

Parametro	Type	Descrizione
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 le mappe dei valori di host o template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove mappe dei valori; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le mappe dei valori esistenti verranno aggiornate; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le mappe dei valori non presenti nei dati importati verranno eliminate 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 di sistema correnti e mostra cosa verrebbe modificato se questo file di importazione venisse importato.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni dei ruoli utente. Consultare Ruoli utente per ulteriori informazioni.

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. Comportamento del parametro: - *obbligatorio*
source	string	Stringa serializzata contenente i dati di configurazione. Comportamento del parametro: - *obbligatorio*
rules	object	Regole su come devono essere confrontati 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 del permesso di *sola lettura* o *lettura-scrittura*. Il parametro `rules` è descritto in dettaglio nella tabella seguente. Comportamento del parametro: - *obbligatorio*

Attention:
Se non viene fornita alcuna regola, 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
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`.
graphs	object	Regole su come importare i grafici. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi grafici; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i grafici esistenti verranno aggiornati; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, i grafici 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`.
template_groups	object	Regole su come importare i gruppi di template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi gruppi di template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i gruppi di template 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`. Questo parametro non farà alcuna differenza nell'output. È consentito solo per coerenza con `configuration.import`.
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`.

Parameter	Type	Description
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`. 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`, 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`. 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 supporto. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno creati nuovi tipi di supporto; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, i tipi di supporto esistenti verranno aggiornati; predefinito: `false`. Questo parametro non farà alcuna differenza nell'output. È consentito solo per coerenza con `configuration.import`.
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 (item, trigger, 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`.

Parameter	Type	Description
templateDashboards	object	Regole su come importare le dashboard dei template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove dashboard dei template; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le dashboard dei template esistenti verranno aggiornate; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le dashboard dei template non presenti nei dati importati verranno eliminate dal database; 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 le mappe dei valori di host o template. Parametri supportati: `createMissing` - `(boolean)` se impostato su `true`, verranno create nuove mappe dei valori; predefinito: `false`; `updateExisting` - `(boolean)` se impostato su `true`, le mappe dei valori esistenti verranno aggiornate; predefinito: `false`; `deleteMissing` - `(boolean)` se impostato su `true`, le mappe dei valori non presenti nei dati importati verranno eliminate dal database; predefinito: `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.

Proprietà	Tipo	Descrizione
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, cioè l'URL del destinatario. Le macro utente sono supportate. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
protocol	integer	Protocollo di comunicazione. Valori possibili: 0 - *(predefinito)* Zabbix Streaming Protocol v1.0.

Proprietà	Tipo	Descrizione
data_type	integer	Tipo di dati. Valori possibili: 0 - *(predefinito)* Valori degli item; 1 - Eventi.
item_value_type	integer	Una somma dei tipi di valore degli item da inviare. Valori possibili: 1 - Numerico (float); 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 di ripetizione. Accetta secondi. Valori possibili: 0s-10s. Predefinito: 5s. Comportamento della proprietà: - *supportata* se `max_attempts` è maggiore di 1.
timeout	string	Timeout. I suffissi temporali sono supportati (ad esempio, 30s, 1m). Le macro utente sono supportate. Valori possibili: 1s-60s. Predefinito: 5s.
http_proxy	string	Stringa di connessione del proxy HTTP(S) fornita come *[protocol]://[username[:password]@]proxy.example.com[:port]*. Le macro utente sono supportate.

Proprietà	Tipo	Descrizione
authtype	integer	Metodo di autenticazione HTTP. Valori possibili: 0 - *(predefinito)* Nessuno; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.
username	string	Nome utente. Le macro utente sono supportate. Comportamento della proprietà: - *supportata* se `authtype` è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”
password	string	Password. Le macro utente sono supportate. Comportamento della proprietà: - *supportata* se `authtype` è impostato su ”Basic”, ”NTLM”, ”Kerberos” o ”Digest”
token	string	Token Bearer. Le macro utente sono supportate. Comportamento della proprietà: - *obbligatoria* se `authtype` è impostato su ”Bearer”
verify_peer	integer	Indica se verificare che il certificato del host sia autentico. Valori possibili: 0 - Non verificare; 1 - *(predefinito)* Verificare.
verify_host	integer	Indica se verificare che il nome host della connessione corrisponda a quello nel certificato del host. Valori possibili: 0 - Non verificare; 1 - *(predefinito)* Verificare.
ssl_cert_file	string	Percorso del file della chiave SSL pubblica. Le macro utente sono supportate.
ssl_key_file	string	Percorso del file della chiave SSL privata. Le macro utente sono supportate.
ssl_key_password	string	Password per il file della chiave SSL. Le macro utente sono supportate.
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 è impostato, verrà esportato tutto. L’oggetto filtro tag ha le seguenti proprietà.

Proprietà	Type	Descrizione
tag	string	Nome del tag. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione. Valori possibili: 0 - *(predefinito)* Uguale; 1 - Diverso da; 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 connettore.

Note:
Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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 connettore.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 connettore in base ai parametri specificati.

Note:
Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 connettori esistenti.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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"

Proprietà	Type	Descrizione
snmpv3_privpassphrase	string	Passphrase di privacy. Comportamento della proprietà: - *supportata* se `type` è impostato su ”SNMPv3 agent” e `snmpv3_securitylevel` è impostato su ”authPriv”
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*

Proprietà	Type	Descrizione
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”
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

`intero/array dcheck.get(parametri oggetto)`

Il metodo consente di recuperare i controlli di rilevamento in base a quanto indicato parametri.

Questo metodo è disponibile per utenti di qualsiasi tipo. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere Utente ruoli per maggiori 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.

Parametro	Tipo	Descrizione
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	
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:

- Correlation
- Correlation operation
- Correlation filter
  - Correlation filter condition

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.

Proprietà	Tipo	Descrizione
conditions	array	Insieme di condizioni del filtro da utilizzare per filtrare i risultati. Le condizioni verranno ordinate nell'ordine in cui sono posizionate nella formula. 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*
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. 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 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. Comportamento della proprietà: - *obbligatoria* 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”
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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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

Description

`object correlation.delete(array correlationids)`

Questo metodo permette di cancellare le correlazioni.

> **Note:**
> Questo metodo è disponibile solamente agli utenti di tipo *Super admin*. I permessi per invocare il metodo possono essere revocati nei settaggi di dei ruoli utente. Vedi User roles per maggiori 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. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Consultare Ruoli utente per ulteriori informazioni.

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`.
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 le correlazioni esistenti.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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.

Proprietà	Tipo	Descrizione
dashboardid	ID	ID della dashboard. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome della dashboard. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
userid	ID	ID dell'utente proprietario della dashboard.
private	integer	Tipo di condivisione della dashboard. Valori possibili: 0 - dashboard pubblica; 1 - *(predefinita)* dashboard privata.
display_period	integer	Periodo di visualizzazione predefinito della pagina (in secondi). Valori possibili: 10, 30, 60, 120, 600, 1800, 3600. Predefinito: 30.
auto_start	integer	Avvio automatico della presentazione. Valori possibili: 0 - non avviare automaticamente la presentazione; 1 - *(predefinito)* avviare automaticamente la presentazione.

Pagina della dashboard

L'oggetto pagina della dashboard ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
dashboard_pageid	ID	ID della pagina della dashboard. Comportamento della proprietà: - *sola lettura*
name	string	Nome della pagina della dashboard. Predefinito: stringa vuota.

Proprietà	Tipo	Descrizione
display_period	integer	Periodo di visualizzazione della pagina della dashboard (in secondi). Valori possibili: 0, 10, 30, 60, 120, 600, 1800, 3600. Predefinito: 0 (verrà utilizzato il periodo di visualizzazione predefinito della pagina).
widgets	array	Array degli oggetti widget della dashboard.

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.

Property	Type	Description
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 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*
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 nuove dashboard.

> **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)` 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 le dashboard.

> **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

`(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 le dashboard 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. Consultare 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 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 le dashboard 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. Consultare 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 campo del widget dashboard consentono di configurare il widget *Action log* nei metodi `dashboard.create` e `dashboard.update`.

> **Attention:**
> Le proprietà `fields` del widget non vengono validate 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 *Action log*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Registro azioni*.

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 utente. Nota: per configurare più utenti, creare un oggetto campo widget dashboard per ciascun utente con un numero incrementato nel nome della proprietà.
Azioni	12	actionids.0	ID azione. Nota: per configurare più azioni, creare un oggetto campo widget dashboard per ciascuna azione con un numero incrementato nel nome della proprietà.
Tipi di supporto	13	mediatypeids.0	ID tipo di supporto. Nota: per configurare più tipi di supporto, creare un oggetto campo widget dashboard per ciascun tipo di supporto con 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 widget dashboard per ciascun valore con un numero incrementato nel nome della proprietà.
Stringa di ricerca	1	message	Qualsiasi valore stringa.
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 *Riferimento* 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
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).

Parametro	type	name	value
Mostra righe	0	show_lines	I valori possibili vanno da 1 a 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 campo 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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.

Parametro	type	name	value
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.

Parameter	type	name	value
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 campo del widget dashboard consentono di configurare il widget *Stato della discovery* 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 campo del widget 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 campo del widget 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
Intervallo di aggiornamento	0	rf_rate

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 campo del widget della dashboard consentono di configurare il widget *Misuratore* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Misuratore*, 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	type	name	value
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	Valori possibili nell'intervallo 1-100. Predefinito: 15.
Posizione verticale	0	desc_v_pos	0 - In alto; 1 - *(predefinito)* In basso.
Grassetto	0	desc_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	desc_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto).
Valore			
Posizioni decimali	0	decimal_places	Valori possibili nell'intervallo 1-10. Predefinito: 2.
Dimensione	0	value_size	Valori possibili nell'intervallo 1-100. Predefinito: 25.
Grassetto	0	value_bold	0 - *(predefinito)* Disabilitato; 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 - *(predefinito)* Abilitato.
Unità (valore)	1	units	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su "Abilitato"

Parametro	type	name	value
Dimensione	0	units_size	Valori possibili nell'intervallo 1-100. Predefinito: 25. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su "Abilitato"
Grassetto	0	units_bold	0 - *(predefinito)* Disabilitato; 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 - *(predefinito)* Dopo il valore; 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 arco	0	value_arc_size	Valori possibili nell'intervallo 1-100. Predefinito: 20.
Ago			
Colore	1	needle_color	Codice colore esadecimale (ad esempio FF0000). Predefinito: "" (vuoto). Comportamento del parametro: - *supportato* se è impostato un oggetto campo widget dashboard per *Show* con il valore "Value arc", oppure *Show arc* è impostato su "Abilitato"
Scala			
Mostra unità	0	scale_show_units	0 - Disabilitato; 1 - *(predefinito)* Abilitato. Comportamento del parametro: - *supportato* se *Unità* (casella di controllo) è impostato su "Abilitato" e se è impostato un oggetto campo widget dashboard per *Show* con il valore "Value arc", oppure *Show arc* è impostato su "Abilitato"
Dimensione	0	scale_size	Valori possibili nell'intervallo 1-100. Predefinito: 15. Comportamento del parametro: - *supportato* se è impostato un oggetto campo widget dashboard per *Show* con il valore "Value arc", oppure *Show arc* è impostato su "Abilitato"
Posizioni decimali	0	scale_decimal_places	Valori possibili nell'intervallo 1-10. Predefinito: 0. Comportamento del parametro: - *supportato* se è impostato un oggetto campo widget dashboard per *Show* con il valore "Value arc", oppure *Show arc* è impostato su "Abilitato"

Parametro	type	name	value
Soglie			
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio FF0000).
Soglia	1	thresholds.0.threshold	Qualsiasi valore numerico. I suffissi (ad esempio ”1d”, ”2w”, ”4K”, ”8G”) sono supportati.
Mostra etichette	0	th_show_labels	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se le *Soglie* sono impostate e se è impostato un oggetto campo widget dashboard per *Show* con il valore ”Value arc” oppure *Show arc* è impostato su ”Abilitato”
Mostra arco	0	th_show_arc	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se le *Soglie* sono impostate
Dimensione arco	0	th_arc_size	Valori possibili nell’intervallo 1-100. Predefinito: 5. Comportamento del parametro: - *supportato* se *Show arc* 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 campo del widget dashboard consentono di configurare il widget *Geomappa* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Geomappa*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Geomap*.

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 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 incrementale 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.
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 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			
Tipo di valutazione	0	evaltype	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard template.
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* Questo parametro non è supportato se si configura il widget in una dashboard template.

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 del tag nell'elenco di valutazione dei tag. Comportamento del parametro: - *obbligatorio* se si configura *Tag*
Valore tag	1	tags.0.value	Questo parametro non è supportato se si configura il widget in una dashboard template. 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* Questo parametro non è supportato se si configura il widget in una dashboard template.
Vista iniziale	1	default_view	*latitudine*, *longitudine*, *livello di zoom* separati da virgole (*facoltativo*, valori possibili da 0 a 30). Esempio: `40.6892494,-74.0466891,10`.
Raggruppamento			
Modalità di raggruppamento	0	clustering_mode	0 - *(predefinito)* Auto; 1 - Livello di zoom.
Livello di zoom del raggruppamento	0	clustering_zoom_level	I valori possibili vanno da 0 a 30. Predefinito: 0. Comportamento del parametro: - *supportato* se *Modalità di raggruppamento* è impostato su "Livello di zoom"
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 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 campo del widget dashboard consentono di configurare il widget *Graph* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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 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 il host o item configurato.

Parametro	type	name	value
Tipo di set di dati	0	ds.0.dataset_type	0 - Elenco item; 1 - *(predefinito)* Modelli di item.
Item	4	ds.0.itemids.0	ID 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 widget della 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 *Item (Widget)* non è impostato
Item (Widget)	1	ds.0.itemids.0._reference	Riferimento dell'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 della 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 *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 item"

Parametro	type	name	value
Modelli di 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”
Modelli di item	1	ds.0.items.0	Questo parametro non è supportato se si configura il widget in una dashboard di template. Nome o modello di Item (ad esempio, ”*: Number of processed *values per second”). Quando si configura il widget in una dashboard di 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”
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 impostato
Palette colori	0	ds.0.color_palette	Indice della palette colori. I valori possibili vanno da 0 a 11. Predefinito: 0. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su ”Modelli di item” e *Colore* non è impostato
Disegna	0	ds.0.type	0 - *(predefinito)* Linea; 1 - Punti; 2 - Scala; 3 - Barre.
Impilato	0	ds.0.stacked	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Disegna* è impostato su ”Linea”, ”Scala” o ”Barre”
Larghezza	0	ds.0.width	I valori possibili vanno da 1 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Disegna* è impostato su ”Linea” o ”Scala”
Dimensione punto	0	ds.0.pointsize	I valori possibili vanno da 1 a 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Disegna* 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 *Disegna* è impostato su ”Linea” o ”Scala”

Parametro	type	name	value
Dati mancanti	0	ds.0.missingdatafunc	0 - *(predefinito)* Nessuno; 1 - Collegati; 2 - Tratta come 0; 3 - Ultimo noto. Comportamento del parametro: - *supportato* se *Disegna* è impostato su ”Linea” o ”Scala”
Sostituisci host	1	ds.0.override_hostid	ABCDE - 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 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 usare 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 usare 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 - all.
Etichetta set di dati	1	ds.0.data_set_label	Qualsiasi valore stringa. Predefinito: "" (vuoto).
Tag item			
Tipo di valutazione	0	ds.0.item_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome 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 item*

Parametro	type	name	value
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 item*
Valore 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 item*

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.
Trigger semplici	0	simple_triggers	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Orario di lavoro	0	working_time	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Nomi host nelle etichette	0	show_hostnames	0 - *(predefinito)* Auto; 1 - Mostra; 2 - Nascondi. Questo parametro non è supportato durante la configurazione del widget in una template dashboard.
Linea del percentile (sinistra)			
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 vanno da 1 a 100. Comportamento del parametro: - *supportato* se *Asse Y* (nella configurazione di *Set di dati*) è impostato su "Sinistra"

Parametro	type	name	value
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 vanno da 1 a 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"

Parametro	type	name	value
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 seguenti parametri sono supportati per la configurazione della *Legenda*.

Parametro	type	name	value
Mostra legenda	0	legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Mostra min/media/max	0	legend_statistic	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitato"
Mostra funzione di aggregazione	0	legend_aggregation	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitato"
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	I valori possibili vanno da 1 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "Abilitato"

Parametro	type	name	value
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 ”Abilitato” e *Mostra min/media/max* è impostato su ”Disabilitato”

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.
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*

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 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*

Sostituzioni

I seguenti parametri sono supportati per la configurazione delle *Sostituzioni*.

Note:
Il primo numero nel nome della proprietà (ad esempio or.0.hosts.0, or.0.items.0) rappresenta il particolare set di dati, mentre il secondo numero, se presente, rappresenta il host o item configurato.

Parametro	type	name	value
Pattern host	1	or.0.hosts.0	Nome o pattern del Host (ad esempio `Zabbix*`). Questo parametro non è supportato se si configura il widget in una dashboard template. Comportamento del parametro: - *obbligatorio* se si configurano le *Sostituzioni*
Pattern 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 template, devono essere impostati solo i pattern per gli item configurati sul template. Comportamento del parametro: - *obbligatorio* se si configurano le *Sostituzioni*
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 - Scala; 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 punti	0	or.0.pointsize	I valori possibili vanno da 1 a 10.
Dati mancanti	0	or.0.missingdatafunc	0 - Nessuno; 1 - Collegati; 2 - Tratta 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.

Parametro	type	name	value
Spostamento temporale	1	or.0.timeshift	Stringa temporale valida (ad esempio 3600, 1h, ecc.). È possibile usare 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 campo del widget dashboard consentono di configurare il widget *Graph (classic)* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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"

Parametro	type	name	value
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 leg-enda	0	show_legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
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 campo 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 validate 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.
Sorgente	0	source_type	2 - *(predefinito)* Graph prototype; 3 - Simple graph prototype.
Graph prototype	7	graphid.0	ID del Graph prototype. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Graph prototype"
Item prototype	5	itemid.0	ID dell'Item prototype. Comportamento del parametro: - *obbligatorio* se *Sorgente* è impostato su "Simple graph prototype"
Periodo di tempo	1	time_period._referenc	`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 *From* e *To*.

Parametro	type	name	value
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 leg-enda	0	show_legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
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 template dashboard.
Colonne	0	columns	I valori possibili vanno da 1 a 24. Predefinito: 2.
Righe	0	rows	I valori possibili vanno da 1 a 16. Predefinito: 1.
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 *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 campo del widget della dashboard consentono di configurare il widget *Nido d'ape* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Nido d'ape*, 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.
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			

Parametro	type	name	value
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.
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*

Parametro	type	name	value
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.

Parametro	type	name	value
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_decimals	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)* Auto; 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 impostato su "Personalizzata"
Grassetto	0	primary_label_bold	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colore	1	primary_label_color	Codice colore esadecimale (ad esempio `FF0000`). Predefinito: in base a `theme` di Settings object e User object: `1F2C33` per "blue-theme" o "hc-light"; `EEEEEE` per "dark-theme" o "hc-dark".
Unità (casella di controllo)	0	primary_label_units_show	0 - Disabilitato; 1 - *(predefinito)* Abilitato. 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 impostato su "Abilitato"
Posizione	0	primary_label_units_pos	0 - Prima del valore; 1 - *(predefinito)* Dopo il valore. Comportamento del parametro: - *supportato* se *Tipo* è impostato su "Valore" e *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.
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"

Parametro	type	name	value
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 - *(predefinito)* Auto; 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 impostato su ”Personalizzata”
Grassetto	0	secondary_label_bold	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Colore	1	secondary_label_color	Codice colore esadecimale (ad esempio `FF0000`). Predefinito: in base a `theme` di Settings object e User object: `1F2C33` per ”blue-theme” o ”hc-light”; `EEEEEE` per ”dark-theme” o ”hc-dark”.
Unità (casella di controllo)	0	secondary_label_units_show	0 - Disabilitato; 1 - *(predefinito)* Abilitato. 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 impostato su ”Abilitato”
Posizione	0	secondary_label_position	0 - Prima del valore; 1 - *(predefinito)* Dopo il valore. Comportamento del parametro: - *supportato* se *Tipo* è impostato su ”Valore” e *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 di sfondo			
Colore di sfondo	1	bg_color	Codice colore esadecimale (ad esempio `FF0000`). Predefinito: in base a `theme` di Settings object e User object: `D9E7ED` per ”blue-theme”; `3D5059` per ”dark-theme”; `AAD7E9` per ”hc-light”; `335463` per ”hc-dark”.
Soglie			
Interpolazione del colore	0	interpolation	0 - Disabilitata; 1 - *(predefinito)* Abilitata.
Colore	1	thresholds.0.color	Codice colore esadecimale (ad esempio `FF0000`).
Soglia	1	thresholds.0.threshold	Qualsiasi valore numerico. I suffissi (ad esempio ”1d”, ”2w”, ”4K”, ”8G”) sono supportati.

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 campo del widget dashboard consentono di configurare il widget *Disponibilità dell'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.
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 campo del widget della dashboard consentono di configurare il widget *Scheda dell'host* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Scheda dell'host*, 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.

Parametro	type	name	value
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 campo del widget della dashboard consentono di configurare il widget *Navigatore degli host* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Navigatore degli host*, 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 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
Pattern host	1	hosts.0	Nome o pattern dell'host. Nota: Per configurare più pattern host, creare un oggetto campo widget dashboard per ciascun pattern host con un numero incrementale nel nome della proprietà. Per più pattern host, il parametro *Gruppi host* non deve essere configurato affatto oppure deve essere configurato con almeno un gruppo host a cui appartengono gli host che corrispondono ai pattern host configurati. Questo parametro non è supportato se si configura il widget in una dashboard template.
Stato host	0	status	-1 - *(predefinito)* Qualsiasi; 0 - Abilitato; 1 - Disabilitato. 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; 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
Gravità	0	severities.0	0 - Non classificato; 1 - Informazione; 2 - Avviso; 3 - Media; 4 - Alta; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: Per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementale nel nome della proprietà.
Mostra host in manuten-zione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Mostra prob-lemi	0	show_problems	0 - Tutti; 1 - *(predefinito)* Non soppressi; 2 - Nessuno.
Raggruppa per			
Attributo	0	group_by.0.attribute	0 - Gruppo host; 1 - Valore tag; 2 - Gravità. Nota: Il numero nel nome della proprietà fa riferimento all'ordine degli attributi 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 tag"
Limite 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 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 campo 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 validate 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

Parametro	type	name	value
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.
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 "Ultimi dati".

Parametro	type	name	value
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 esempio `FF0000`). Predefinito: `42A5F5`.
Periodo di tempo	1	sparkline.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 relativo parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `""` (vuoto) In alternativa, è possibile impostare il periodo di tempo solo nei parametri *Da* e *A*.
Da	1	sparkline.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.). Predefinito: now-1h. Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `sparkline.time_period.to` è impostato

Parametro	type	name	value
A	1	sparkline.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.). Predefinito: now. Comportamento del parametro: - *supportato* se *Periodo di tempo* non è impostato - *obbligatorio* se `time_period.to` è impostato
Dati storici	0	sparkline.history	0 - *(predefinito)* Auto; 1 - Storico; 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 campo del widget dashboard consentono di configurare il widget *Item history* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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 history*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Cronologia 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.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.
Colonne (vedere sotto)			
Mostra righe	0	show_lines	I valori possibili vanno da 1 a 100. Predefinito: 25.
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 (vedere sotto)			
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*

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.value	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 di colonna sono supportati se l'*Item* configurato è un item di tipo carattere, testo o log.

Parametro	type	name	value
Evidenziazioni			
Evidenziazione	1	columns.0.highlights.color	Codice colore esadecimale (ad es. FF0000).
Soglia	1	columns.0.highlights.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 font monospazio	0	columns.0.monospace_font	0 - *(predefinito)* Usa il font predefinito; 1 - Usa font monospazio.
Visualizza ora locale	0	columns.0.local_time	0 - *(predefinito)* Visualizza timestamp; 1 - Visualizza 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._referenceDASHBOARD	`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 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 campo del widget della dashboard consentono di configurare il widget *Navigatore di elementi* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Navigatore di elementi*, 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 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.

Parametro	type	name	value
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.
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.hostid` - imposta il selettore host della dashboard come origine dati per gli host; `ABCDE._hostid` - 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.
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 incrementato nel nome della proprietà.
Tag item			
Tipo di valutazione	0	item_tags_evaltype	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*
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			

Parametro	type	name	value
Attributo	0	group_by.0.attribute	0 - Gruppo di host; 1 - Nome host; 2 - Valore del tag host; 3 - Valore del tag 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 host" o "Valore del tag item"
Limite item	0	show_lines	I valori possibili vanno da 1 a 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 campo del widget dashboard consentono di configurare il widget *Item value* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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 value*, 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.
Mostra	0	show.0	Comportamento del parametro: - *obbligatorio* se *Item* non è impostato 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”
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 avanzata sono supportati se *Mostra* è impostato su ”Descrizione”.

Parametro	type	name	value
Descrizione	1	description	Qualsiasi valore stringa, incluse le macro. Macro supportate: {HOST.*}, {ITEM.*}, {INVENTORY.*}, macro utente. Predefinito: {ITEM.NAME}.

Parametro	type	name	value
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 es. FF0000). Predefinito: "" (vuoto).

Valore

I seguenti parametri di configurazione avanzata 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 - Alto; 1 - *(predefinito)* Centro; 2 - 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.

Parametro	type	name	value
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 avanzata sono supportati se *Mostra* è impostato su ”Time”.

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*.
Posizione verticale	0	time_v_pos	0 - *(predefinito)* Alto; 1 - Centro; 2 - 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 vanno da 1 a 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 *Mostra* è impostato su ”Sparkline”.

Parametro	type	name	value
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 esempio `FF0000`). Predefinito: `42A5F5`.
Periodo di tempo	1	sparkline.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 relativo 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*.
Da	1	sparkline.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: - *obbligatorio* se `sparkline.time_period.to` è impostato
A	1	sparkline.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: - *obbligatorio* se `sparkline.time_period.from` è impostato
Dati storici	0	sparkline.history	0 - *(predefinito)* Auto; 1 - Storico; 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 campo 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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

Parametro	type	name	value
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

Descrizione

Questi parametri e i possibili valori delle proprietà per i rispettivi oggetti campo del widget della dashboard consentono di configurare il widget *Albero di navigazione della 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Albero di navigazione della mappa*, 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 campo 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 validate 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 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	0	ds.0.dataset_type	0 - Elenco item; 1 - *(predefinito)* Modelli di item.
Item	4	ds.0.itemids.0	ID Item. Quando si configura il widget in una dashboard template, devono essere impostati solo gli item configurati sul template. Nota: Per configurare più item, creare un oggetto campo widget della 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 *Item (Widget)* non è impostato
Item (Widget)	1	ds.0.itemids.0._reference	Riferimento dell'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 della 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 *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 item"
Tipo di item	0	ds.0.type.0	0 - *(predefinito)* Normale; 1 - Totale. Il valore "Totale" può essere impostato solo per un item nell'intero grafico. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su "Elenco item"

Parametro	type	name	value
Modelli di 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”
Modelli di item	1	ds.0.items.0	Questo parametro non è supportato se si configura il widget in una dashboard template. 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 sul template. Comportamento del parametro: - *obbligatorio* se *Tipo di set di dati* è impostato su ”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 *Tavolozza colori* non è impostata
Tavolozza colori	0	ds.0.color_palette	Indice della tavolozza colori. I valori possibili vanno da 0 a 11. Predefinito: 0. Comportamento del parametro: - *supportato* se *Tipo di set di dati* è impostato su ”Modelli 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 item			
Tipo di valutazione	0	ds.0.item_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome 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 item*

Parametro	type	name	value
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 item*
Valore 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 item*

Opzioni di visualizzazione

I seguenti parametri sono supportati per configurare le *Opzioni di visualizzazione*.

Parameter	type	name	value
Selezione dei dati storici	0	source	0 - *(predefinito)* Auto; 1 - Storico; 2 - Trend.
Disegno	0	draw_type	0 - *(predefinito)* Torta; 1 - Ciambella.
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 *Disegno* è impostato su "Ciambella"
Larghezza del bordo	0	stroke	I valori possibili vanno da 0 a 10. Predefinito: 0. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Ciambella"
Mostra valore totale	0	total_show	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Disegno* è impostato su "Ciambella"
Dimensione	0	value_size_type	0 - *(predefinito)* Auto; 1 - Personalizzata. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"
Dimensione (valore per dimensione personalizzata)	0	value_size	I valori possibili vanno da 1 a 100. Predefinito: 20. Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su "Abilitato"

Parameter	type	name	value
Posizioni decimali	0	decimal_places	I valori possibili vanno da 0 a 6. Predefinito: 2.
Unità (casella di controllo)	0	units_show	Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su ”Abilitato” 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Unità (valore)	1	units	Comportamento del parametro: - *supportato* se *Mostra valore totale* è impostato su ”Abilitato” 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 vanno da 0 a 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 vanno da 1 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Unisci i settori più piccoli di N%* (casella di controllo) è impostato su ”Abilitato”

Parameter	type	name	value
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._referenceDA	`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	type	name	value
Mostra legenda	0	legend	0 - Disabilitato; 1 - *(predefinito)* Abilitato.
Mostra valore	0	legend_value	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su "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.
Numero di righe/ Numero massimo di righe	0	legend_lines	Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato” I valori possibili vanno da 1 a 10. Predefinito: 1.
Numero di colonne	0	legend_columns	Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato” I valori possibili vanno da 1 a 4. Predefinito: 4. Comportamento del parametro: - *supportato* se *Mostra legenda* è impostato su ”Abilitato” 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 campo del widget della dashboard consentono di configurare il widget *Host problematici* 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Host problematici*, 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.

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 *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.
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*

Parametro	type	name	value
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.
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 campo del widget dashboard consentono di configurare il widget *Problems* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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 *Problems*.

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 - 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 widget dashboard per ciascun gruppo di host con un numero incrementale 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.

Parametro	type	name	value
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 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 dashboard per ciascun host con un numero incrementale 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.
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 - Media; 4 - Alta; 5 - Disastro. Predefinito: vuoto (tutti abilitati). Nota: Per configurare più valori, creare un oggetto campo widget dashboard per ciascun valore con un numero incrementale nel nome della proprietà.
Tag del 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: - *required* se si configurano i *Tag del problema*

Parametro	type	name	value
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: - *required* se si configurano i *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: - *required* se si configurano i *Tag del problema*
Mostra tag	0	show_tags	0 - *(predefinito)* Nessuno; 1 - 1; 2 - 2; 3 - 3.
Nome tag (formato)	0	tag_name_format	0 - *(predefinito)* Completo; 1 - Abbreviato; 2 - Nessuno. Comportamento del parametro: - *supported* 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: - *supported* 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 del riconoscimento	0	acknowledgement_status	0 - *(predefinito)* tutti; 1 - Non riconosciuti; 2 - Riconosciuti.
Da me	0	acknowledged_by_me	0 - *(predefinito)* Disabilitato; 1 - Abilitato.

Parametro	type	name	value
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, eccetto ”Ora (decrescente)” e ”Ora (crescente)”, il parametro *Mostra timeline* deve essere impostato su ”Disabilitato”. I valori ”Host (crescente)” e ”Host (decrescente)” non sono supportati se si configura il widget in una dashboard template.
Mostra time-line	0	show_timeline	0 - Disabilitato; 1 - *(predefinito)* Abilitato. Comportamento del parametro: - *supported* se *Ordina voci per* è impostato su ”Ora (decrescente)” o ”Ora (crescente)”
Evidenzia l’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 vanno da 1 a 100. Predefinito: 25.
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: - *required*

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 campo del widget 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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.
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.

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 *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.
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 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 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	0	show_type	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".

Parametro	type	name	value
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.
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 campo del widget dashboard consentono di configurare il widget *Scatter plot* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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 *Scatter plot*, 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

Parametro	type	name	value
Asse X (Widget)	1	ds.0.x_axis_itemids.0	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 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
Asse Y (Widget)	1	ds.0.y_axis_itemids.0	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

Parametro	type	name	value
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”
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.

Parametro	type	name	value
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.
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._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 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	Qualsiasi valore numerico. Predefinito: "" (vuoto).
		y_axis_min	
Max	1	x_axis_max	Qualsiasi valore numerico. Predefinito: "" (vuoto).
		y_axis_max	
Unità (tipo)	0	x_axis_units	0 - *(predefinito)* Automatico; 1 - Statico.
		y_axis_units	

Parametro	type	name	value
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"
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 campo del widget dashboard consentono di configurare il widget *Rapporto degli 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'aggiornamento del widget *Rapporto degli SLA*, fare riferimento al comportamento dei parametri descritto nelle tabelle seguenti.

Parametri

I seguenti parametri sono supportati per il widget *Report SLA*.

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.
SLA	10	slaid.0	ID SLA. Comportamento del parametro: - *obbligatorio*
Servizio	9	serviceid.0	ID 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 campo del widget della 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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 *Informazioni di sistema*.

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	0	info_type	0 - *(predefinito)* Statistiche di sistema; 1 - Nodi ad alta disponibilità.
Mostra i dettagli del controllo degli aggiornamenti software	0	show_software_update_check_details	0 - *(predefinito)* Disabilitato; 1 - Abilitato. Comportamento del parametro: - *supported* se `AllowSoftwareUpdateCheck` nella configurazione del server Zabbix è abilitato e *Mostra* è impostato su "Statistiche di sistema"

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 campo del widget dashboard consentono di configurare il widget *Top hosts* nei metodi `dashboard.create` e `dashboard.update`.

Attention:
Le proprietà `fields` del widget non vengono validate 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	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.
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.

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 *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	0 - *(predefinito)* And/Or; 2 - Or. Questo parametro non è supportato se si configura il widget in una dashboard template.
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 host* Questo parametro non è supportato se si configura il widget in una dashboard template.
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 configurano i *Tag host* Questo parametro non è supportato se si configura il widget in una dashboard template.
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 host* Questo parametro non è supportato se si configura il widget in una dashboard template.
Mostra host in manutenzione	0	maintenance	0 - *(predefinito)* Disabilitato; 1 - Abilitato.
Colonne (vedi sotto)			
Ordina per	0	column	Valore numerico della colonna tra le colonne configurate.

Parametro	type	name	value
Ordine	0	order	2 - *(predefinito)* Top N; 3 - Bottom N.
Limite host	0	show_lines	I valori possibili vanno da 1 a 1000. Predefinito: 10. Questo parametro non è supportato se si configura il widget in una dashboard 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*
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 *Data* è impostato su "Item value".

Note:
Il primo numero nel nome della proprietà *Thresholds* (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 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 `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parametro	type	name	value
Item	1	columns.0.item	Nome item valido. Quando si configura il widget in una dashboard template, devono essere impostati solo gli item configurati sul template.
Visualizza il valore dell'item come	0	columns.0.display_value	0 - *(predefinito)* Numerico; 1 - Testo; 2 - Binario.

Parametro	type	name	value
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 "Numeric"
Min	1	columns.0.min	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Bar" o "Indicators"
Max	1	columns.0.max	Qualsiasi valore numerico. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Bar" o "Indicators"
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 "Numeric"
Sparkline			
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 "Numeric" 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 "Numeric" 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 "Numeric" 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 suo parametro `reference` uguale a `ABCDE`) come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *From* e *To*. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Sparkline"

Parametro	type	name	value
Da	1	columns.0.sparkline.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 *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Sparkline" - *richiesto* se `columns.0.sparkline.time_period.to` è impostato
A	1	columns.0.sparkline.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 *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Sparkline" - *richiesto* se `columns.0.sparkline.time_period.from` è impostato
Dati cronologici	0	columns.0.sparkline.history	0 - *(predefinito)* Auto; 1 - History; 2 - Trends. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numeric" e *Visualizzazione* è impostato su "Sparkline"
Thresholds			
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 "Numeric"
Soglia	1	columnsthresholds.0.threshold	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Numeric"
Highlights			
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 "Text"
Pattern	1	columns.0.highlights.pattern	Qualsiasi valore stringa. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su "Text"
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 "Binary"
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.

Parametro	type	name	value
Periodo di tempo	1	columns.0.time_period.reference	DASHBOARD - imposta il selettore del periodo di tempo della dashboard come origine dati; ABCDE._timeperiod - imposta un widget compatibile (con il suo parametro reference uguale a ABCDE) come origine dati. Predefinito: DASHBOARD._timeperiod In alternativa, è possibile impostare il periodo di tempo solo nei parametri *From* e *To*. 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 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 e *Funzione di aggregazione* è impostato su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *richiesto* se columns.0.time_period.to è impostato
A	1	columns.0.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 e *Funzione di aggregazione* è impostato su ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *richiesto* se columns.0.time_period.from è impostato
Dati crono-logici	0	columns.0.history	0 - *(predefinito)* Auto; 1 - History; 2 - Trends. Comportamento del parametro: - *supportato* se *Visualizza il valore dell'item come* è impostato su ”Numeric”
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: - *richiesto*

Testo

I seguenti parametri sono supportati se *Data* è impostato su ”Text”.

Parametro	type	name	value
Testo	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 campo 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 validate 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.
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.

Parametro	type	name	value
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.
Layout	0	layout	0 - *(predefinito)* Orizzontale; 1 - Verticale.
Mostra problemi	0	show_problems	0 - Tutti; 1 - *(predefinito)* Non soppressi; 2 - Nessuno.
Item (vedere sotto)			

Configurazione avanzata

I seguenti parametri di configurazione avanzata sono supportati per il widget *Top items*.

Parametro	type	name	value
Ordinamento host			
Ordina per	0	host_ordering_order_by	0 - *(predefinito)* Nome host; 3 - Valore item.
Pattern item	1	host_ordering_item.0	Nome o pattern di Item (ad esempio, "*: Number of processed *values per second"). Nota: per configurare più pattern di item, creare un oggetto campo del widget dashboard per ciascun pattern di item con un numero incrementato nel nome della proprietà. Quando si configura il widget in una template dashboard, devono essere impostati solo i pattern per gli item configurati sul template. Comportamento del parametro: - *obbligatorio* se *Ordina per* è impostato su "Valore item"
Ordine	0	host_ordering_order	2 - *(predefinito)* Primi N; 3 - Ultimi N.
Limite	0	host_ordering_limit	I valori possibili vanno da 1 a 100. Predefinito: 10.

Parametro	type	name	value
Ordinamento item			
Ordina per	0	item_ordering_order_by	1 - Host; 2 - Nome item; 3 - *(predefinito)* Valore item.
Pattern host	1	item_ordering_host.0	Nome o pattern di Host. Nota: per configurare più pattern di host, creare un oggetto campo del widget dashboard per ciascun pattern di host con un numero incrementato nel nome della proprietà. Comportamento del parametro: - *obbligatorio* se *Ordina per* è impostato su ”Host”
Ordine	0	item_ordering_order	2 - *(predefinito)* Primi N; 3 - Ultimi N.
Limite	0	item_ordering_limit	I valori possibili vanno da 1 a 100. Predefinito: 10.

Colonne

Note:
Il primo numero nel nome della proprietà (ad esempio, columns.0.items.0, columns.0.item_tags_evaltype) rappresenta la colonna specifica, mentre il secondo numero, se presente, rappresenta l’entità configurata (ad esempio, pattern di item, tag).

Parametro	type	name	value
Pattern di item	1	columns.0.items.0	Nome o pattern di Item (ad esempio, ”*: Number of processed *values per second”). Nota: per configurare più pattern di item, creare un oggetto campo widget dashboard per ciascun pattern di item con il secondo numero nel nome della proprietà incrementato. Quando si configura il widget su una template dashboard, devono essere impostati solo i pattern per gli item configurati sulla template. Comportamento del parametro: - *obbligatorio*
Tag di item			
Tipo di valutazione	0	columns.0.item_tags_evaltype	0 - *(predefinito)* And/Or; 2 - Or.
Nome 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 di item*

Parametro	type	name	value
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 di item*
Valore 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 di item*
Colore di base	1	columns.0.base_color	Codice colore esadecimale (ad esempio FF0000).
Visualizza valore come	0	columns.0.display_value_as	*(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. I suffissi (ad esempio "1d", "2w", "4K", "8G") sono supportati. 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. I suffissi (ad esempio "1d", "2w", "4K", "8G") sono supportati. 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 vanno da 0 a 10. Predefinito: 1. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Riempimento	0	columns.0.sparkline.fill	I valori possibili vanno da 0 a 10. Predefinito: 3. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Colore	1	columns.0.sparkline.color	Codice colore esadecimale (ad esempio FF0000). Predefinito: 42A5F5. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"

Parametro	type	name	value
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 *Visualizzazione* è impostata su "Sparkline"
Da	1	columns.0.sparkline.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 *Visualizzazione* è impostata su "Sparkline" - *obbligatorio* se `columns.0.sparkline.time_period.to` è impostato
A	1	columns.0.sparkline.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 *Visualizzazione* è impostata su "Sparkline" - *obbligatorio* se `columns.0.sparkline.time_period.from` è impostato
Dati storici	0	columns.0.sparkline.history	0 - *(predefinito)* Auto; 1 - Storico; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizzazione* è impostata su "Sparkline"
Soglie			
Colore	1	columns.0.thresholds.color	Codice colore esadecimale (ad esempio `FF0000`). Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico"
Soglia	1	columns.0.thresholds.threshold	Un intero valore numerico. I suffissi (ad esempio "1d", "2w", "4K", "8G") sono supportati. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico"
Evidenziazioni			
Evidenziazione colore	1	columns.0.highlights.color	Codice colore esadecimale (ad esempio `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 vanno da 0 a 10. Predefinito: 2.
Configurazione avanzata			

Parametro	type	name	value
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.
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 impostata su "min", "max", "avg", "count", "sum", "first", "last"
Da	1	columns.0.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 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 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 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)* Auto; 1 - Storico; 2 - Trend. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico"
Aggrega	0	columns.0.aggregate_grouping	0 - *(predefinito)* Ogni item 1 - Combinato. Comportamento del parametro: - *supportato* se *Visualizza valore come* è impostato su "Numerico" e *Visualizzazione* è impostata su "Così com'è", "Barra" o "Indicatori".

Parametro	type	name	value
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 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 campo 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 validate 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 per il widget *Top triggers*.

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.
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.
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.
Problema	1	problem	Nome 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 - 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 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. Comportamento del parametro: - *obbligatorio* se si configurano i *Tag 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 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*
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 relativo parametro *Reference* impostato su "`ABCDE`") come origine dati. Predefinito: `DASHBOARD._timeperiod` In alternativa, è possibile impostare il periodo di tempo solo nei parametri *From* e *To*.
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
Limite trig-ger	0	show_lines	I valori possibili vanno da 1 a 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 campo del widget della 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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.

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.
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 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 campo 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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 campo 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. Questo consente agli utenti di modificare i widget built-in e creare widget personalizzati, ma comporta anche il rischio di creare o aggiornare i widget in modo errato. Per garantire la corretta creazione o l'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.

Parametro	type	name	value
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*
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 manutenzione	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. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del dashboard del template. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
templateid	ID	ID del template a cui appartiene il dashboard. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
display_period	integer	Periodo di visualizzazione predefinito della pagina (in secondi). Valori possibili: 10, 30, 60, 120, 600, 1800, 3600. Predefinito: 30.
auto_start	integer	Avvio automatico della presentazione. Valori possibili: 0 - non avviare automaticamente la presentazione; 1 - *(predefinito)* avviare automaticamente la presentazione.
uuid	string	Identificatore univoco universale, utilizzato per collegare i dashboard del template importati a quelli già esistenti. Generato automaticamente, se non specificato.

Pagina della dashboard del template

L'oggetto pagina della dashboard del template ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
dashboard_pageid	ID	ID della pagina della dashboard. Comportamento della proprietà: - *sola lettura*
name	string	Nome della pagina della dashboard. Predefinito: stringa vuota.

Proprietà	Tipo	Descrizione
display_period	integer	Periodo di visualizzazione della pagina della dashboard (in secondi). Valori possibili: 0, 10, 30, 60, 120, 600, 1800, 3600. Predefinito: 0 (verrà utilizzato il periodo di visualizzazione predefinito della pagina).
widgets	array	Array degli oggetti widget della dashboard del template.

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*
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.

Proprietà	Tipo	Descrizione
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*

Proprietà	Type	Descrizione
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 di 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)` 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 le dashboard del 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

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)`

Questo metodo consente di recuperare le dashboard dei template 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 del ruolo 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	

Parametro	Type	Descrizione
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 dashboard di template 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 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 utenti

L'oggetto directory utenti ha le seguenti proprietà.

Property	Type	Description
userdirectoryid	ID	ID della directory utenti. Se una directory utenti viene eliminata, il valore della proprietà `userdirectoryid` dell'oggetto User viene impostato su "0" per tutti gli utenti collegati alla directory utenti eliminata. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
idp_type	integer	Tipo di protocollo di autenticazione utilizzato dal provider di identità per la directory utenti. Si noti che può esistere una sola directory utenti di tipo SAML. Valori possibili: 1 - Directory utenti di tipo LDAP; 2 - Directory utenti di tipo SAML. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione

Property	Type	Description
group_name	string	Attributo della directory utenti LDAP/SAML che contiene il nome del gruppo utilizzato per mappare i gruppi tra la directory utenti 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 utenti 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 provisionato. Esempi: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	Attributo della directory utenti 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 provisionato. Esempi: *sn*, *surname*, *lastName*
provision_status	integer	Stato del provisioning della directory utenti. Valori possibili: 0 - *(predefinito)* Disabilitato (il provisioning degli utenti creati da questa directory utenti è disabilitato); 1 - Abilitato (il provisioning degli utenti creati da questa directory utenti è abilitato; inoltre, deve essere abilitato lo stato del provisioning LDAP o SAML (`ldap_jit_status` o `saml_jit_status` dell’oggetto Authentication)).
provision_groups	array	Array di oggetti provisioning groups mappings per mappare il pattern del gruppo utenti LDAP/SAML al gruppo utenti e al ruolo utente di Zabbix. Comportamento della proprietà: - *obbligatoria* se `provision_status` è impostato su ”Enabled”
provision_media	array	Array di oggetti media type mappings per mappare gli attributi media LDAP/SAML dell’utente (ad esempio, email) ai supporti utente di Zabbix per l’invio di notifiche.
Proprietà specifiche di LDAP:		
name	string	Nome univoco della directory utenti. 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 schema (`ldap://` o `ldaps://`), host e porta (opzionale). 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”

Property	Type	Description
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 utenti LDAP verso gli 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 utenti LDAP tramite il quale identificare l'account utente dalle 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"
bind_dn	string	Account del server LDAP 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 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 utenti. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"
group_basedn	string	Percorso di base della directory utenti LDAP verso i gruppi; utilizzato per configurare un controllo di appartenenza dell'utente nella directory utenti 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"

Property	Type	Description
group_filter	string	Stringa di filtro per recuperare i gruppi della directory utenti LDAP di cui l'utente è membro; utilizzata per configurare un controllo di appartenenza dell'utente nella directory utenti LDAP. Ignorata durante il provisioning di un utente se `group_membership` è impostato. Segnaposto supportati per `group_filter`: *%{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}* - username dell'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 utenti LDAP che contiene informazioni sui membri del gruppo; utilizzato per configurare un controllo di appartenenza dell'utente nella directory utenti 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 utenti 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 utenti LDAP in base alle informazioni fornite nella richiesta di accesso. Segnaposto supportati per `search_filter`: *%{attr}* - nome dell'attributo di ricerca (ad esempio, *uid*, *sAMAccountName*); *%{user}* - username dell'utente Zabbix. Predefinito: *(%{attr}=%{user})* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type LDAP"

Property	Type	Description
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 utenti LDAP utilizzato per fare riferimento a un oggetto utente. Il valore di `user_ref_attr` viene utilizzato per ottenere valori dall’attributo specificato nella directory utenti 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 il provider di identità e viene utilizzato per comunicare con il provider di identità 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 del provider di identità. 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 utenti SAML (anche attributo SCIM se `scim_status` è impostato su ”SCIM provisioning is enabled”) che contiene lo username 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 del provider di identità, 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”

Property	Type	Description
slo_url	string	URL del servizio SAML single log-out del provider di identità, a cui Zabbix invierà le richieste di logout 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 - Crittografa 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 - Crittografa 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 del provider di identità 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 - *(predefinito)* Non firmare le asserzioni; 1 - Firma 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 - *(predefinito)* Non firmare le richieste AuthN; 1 - Firma le richieste AuthN. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"

Property	Type	Description
sign_messages	integer	Indica se i messaggi SAML devono essere firmati con una firma SAML. Valori possibili: 0 - *(predefinito)* Non firmare i messaggi; 1 - Firma 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 logout SAML devono essere firmate con una firma SAML. Valori possibili: 0 - *(predefinito)* Non firmare le richieste di logout; 1 - Firma le richieste di logout. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML"
sign_logout_responses	integer	Indica se le risposte di logout SAML devono essere firmate con una firma SAML. Valori possibili: 0 - *(predefinito)* Non firmare le risposte di logout; 1 - Firma le risposte di logout. 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 `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola scrittura*
sp_private_key	string	Contenuto della chiave privata del Service Provider (SP) per la configurazione del servizio SAML single sign-on (SSO). Fornisce autenticazione sicura e scambio di dati con l'Identity Provider (IdP). Esempi: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola scrittura*

Property	Type	Description
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 `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola scrittura*
idp_certificate_hash	string	Hash md5 del valore di `idp_certificate`. Restituirà una stringa vuota se `idp_certificate` è vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola lettura*
sp_private_key_hash	string	Hash md5 del valore di `sp_private_key`. Restituirà una stringa vuota se `sp_private_key` è vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola lettura*
sp_certificate_hash	string	Hash md5 del valore di `sp_certificate`. Restituirà una stringa vuota se `sp_certificate` è vuoto. Comportamento della proprietà: - *supportata* se `idp_type` è impostato su "User directory of type SAML" e `$SSO['CERT_STORAGE']` è impostato su `database` in zabbix.conf.php - *sola lettura*

Mappature dei tipi di supporto

L'oggetto delle mappature dei tipi di supporto ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
userdirectory_media	mediaid	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 nuove directory utenti.

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 utenti. Una directory utenti non può essere eliminata quando è utilizzata direttamente da almeno un gruppo utenti.
La directory utenti LDAP predefinita non può essere eliminata quando `authentication.ldap_configured` è impostato su 1 o quando rimangono altre directory utenti.

> **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)`

Questo 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 utenti.

Note:
Questo metodo consente anche di verificare quali dati configurati corrispondono alle impostazioni della directory utenti per il provisioning degli utenti (ad esempio, quale ruolo utente, gruppi utenti e media utente verranno assegnati all'utente). Per questo tipo di test, la richiesta API deve essere effettuata per una directory utenti che abbia `provision_status` impostato su abilitato.

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

”oggetto userdirectory.update(object/array userDirectory)”.

Questo metodo consente di aggiornare le directory utente esistenti.

Questo metodo è disponibile solo per il tipo di utente *Super amministratore*.

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*.

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
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.

Property	Type	Description
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)`

Questo metodo consente di recuperare i graph 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 del ruolo utente. Vedere User roles 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.
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.

Proprietà	Tipo	Descrizione
regexpid	ID	ID dell'espressione regolare. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome dell'espressione regolare. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
test_string	string	Stringa di test.

Espressioni

L'oggetto expressions ha le seguenti proprietà.

Proprietà	Type	Descrizione
expression	string	Espressione regolare. Comportamento della proprietà: - *obbligatoria*
expression_type	integer	Tipo di espressione regolare. Valori possibili: 0 - Stringa di caratteri inclusa; 1 - Qualsiasi stringa di caratteri inclusa; 2 - Stringa di caratteri non inclusa; 3 - Il risultato è TRUE; 4 - Il risultato è FALSE. Comportamento della proprietà: - *obbligatoria*

Proprietà	Type	Descrizione
exp_delimiter	string	Delimitatore dell'espressione. Valore predefinito: ",". Valori possibili: "," o ".", oppure "/". Comportamento della proprietà: - *supportata* se `expression_type` è impostato su "Any character string included"
case_sensitive	integer	Sensibilità alle maiuscole/minuscole. Valore predefinito: 0. Valori possibili: 0 - Senza distinzione tra maiuscole e minuscole; 1 - Con 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

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.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.create",
    "params": {
      "name": "Storage devices for SNMP discovery",
      "test_string": "/boot",
      "expressions": [
        {
          "expression": "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$",
          "expression_type": "4",
          "case_sensitive": "1"
        }
```

```
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16"
        ]
    },
    "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*. I permessi per chiamare il metodo possono essere revocati 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 User roles 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	
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"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Risposta:


```
{
  "jsonrpc": "2.0",
  "result": {
    "1": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|
          "expression_type": "3"
        }
      ]
    },
    "2": {
      "regexpid": "2",
      "name": "Network interfaces for discovery",
      "expressions": [
        {
          "expression": "^Software Loopback Interface",
          "expression_type": "4"
        },
        {
          "expression": "^(In)?[Ll]oop[Bb]ack[0-9._]*$",
          "expression_type": "4"
        },
        {
          "expression": "^NULL[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ll]o[0-9.]*$",
          "expression_type": "4"
        },
        {
          "expression": "^[Ss]ystem$",
          "expression_type": "4"
        },
        {
          "expression": "^Nu[0-9.]*$",
          "expression_type": "4"
        }
      ]
    }
  },
  "id": 1
}
```


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 il rilevamento dei file system.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.update",
    "params": {
      "regexpid": "1",
      "name": "File systems for discovery",
      "test_string": "",
      "expressions": [
        {
          "expression": "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|zfs)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        },
        {
          "expression": "^(ntfs|fat32|fat16)$",
          "expression_type": "3",
          "exp_delimiter": ",",
          "case_sensitive": "0"
        }
      ]
    },
    "id": 1
}
```

Risposta:

```
{
    "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.
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.

Property	Type	Description
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.
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.
- Confermare l'evento. Se l'evento è già confermato, questa azione verrà ignorata.
- Annullare la conferma dell'evento. Se l'evento non è confermato, questa azione verrà ignorata.
- Aggiungere un messaggio.
- Modificare la gravità dell'evento. Se l'evento ha già la stessa gravità, questa azione verrà ignorata.
- Sopprimere l'evento. Se l'evento è già soppresso, questa azione verrà ignorata.
- Annullare la soppressione dell'evento. Se l'evento non è soppresso, questa azione verrà ignorata.
- Modificare il rango dell'evento in causa. Se l'evento è già classificato come causa, questa azione verrà ignorata.
- Modificare il rango 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 problema.
Per chiudere l'evento o modificarne la gravità sono richiesti i diritti di lettura/scrittura per il trigger.
Per chiudere un evento, nel trigger deve essere consentita la chiusura manuale.

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 che devono essere eseguite.

Parametro	Type	Descrizione
eventids	ID/array	ID degli eventi da riconoscere. Comportamento del parametro: - *obbligatorio*

Parametro	Type	Descrizione
action	integer	Azione/i di aggiornamento dell'evento. Valori bitmap possibili: 1 - chiudi problema; 2 - riconosci evento; 4 - aggiungi messaggio; 8 - cambia gravità; 16 - annulla riconoscimento evento; 32 - sopprimi evento; 64 - annulla soppressione evento; 128 - cambia il rango dell'evento in causa; 256 - cambia il rango dell'evento in sintomo. Questo è un campo bitmask; è accettata qualsiasi somma dei possibili valori bitmap (ad esempio, 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 "change event rank to symptom"
message	string	Testo del messaggio. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "add message"
severity	integer	Nuova gravità per gli eventi. Valori possibili: 0 - non classificato; 1 - informazione; 2 - avviso; 3 - medio; 4 - alto; 5 - disastro. Comportamento del parametro: - *obbligatorio* se `action` contiene il bit "change severity"
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 "suppress event"

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID degli eventi aggiornati sotto la 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
}
```

Fonte

CEvent::acknowledge() in *ui/include/classes/api/services/CEvent.php*.

event.get

Descrizione

`integer/array event.get(object parameters)`

Questo 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 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 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.
source	integer	Restituisce solo gli eventi del tipo specificato. Fare riferimento alla pagina dell'oggetto event per un elenco dei tipi di evento supportati. Predefinito: 0 - eventi trigger.
object	integer	Restituisce solo gli eventi creati da oggetti del tipo specificato. Fare riferimento alla pagina dell'oggetto event per un elenco dei tipi di oggetto supportati. Predefinito: 0 - trigger.
acknowledged	boolean	Se impostato su `true`, restituisce solo gli eventi confermati.
action	integer	Restituisce solo gli eventi per i quali sono state eseguite le azioni di aggiornamento evento specificate. Per più azioni, utilizzare la somma di qualsiasi valore bitmap accettabile come maschera di bit (ad esempio, 34 per confermare 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 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 evento specificate. Si applica solo se object è trigger.
trigger_severities	integer/array	Restituisce solo gli eventi con le severità 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 operator possibili: 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 al momento specificato o successivamente.

Parameter	Type	Description
time_till	timestamp	Restituisce solo gli eventi creati al momento specificato o precedentemente.
problem_time_from	timestamp	Restituisce solo gli eventi che erano nello stato di problema a partire da `problem_time_from`, indipendentemente dal loro stato corrente. Si applica solo quando source è un evento trigger e object è un trigger. Questo parametro è obbligatorio se viene specificato `problem_time_till`.
problem_time_till	timestamp	Restituisce solo gli eventi che erano nello stato di problema fino a `problem_time_till`, indipendentemente dal loro stato corrente. Si applica solo quando source è un evento trigger e object è un trigger. Questo parametro è obbligatorio se viene specificato `problem_time_from`.
value	integer/array	Restituisce solo gli eventi con i valori specificati.
selectAcknowledges	query	Restituisce una proprietà `acknowledges` con gli aggiornamenti dell'evento. Gli aggiornamenti dell'evento sono ordinati in ordine cronologico inverso. L'oggetto aggiornamento evento ha le seguenti proprietà: `acknowledgeid` - `(ID)` ID della conferma; `userid` - `(ID)` ID dell'utente che ha aggiornato l'evento; `clock` - `(timestamp)` ora 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)` ora fino alla quale l'evento sarà soppresso; `taskid` - `(ID)` ID del task se l'evento corrente è soggetto a un cambio di rango; `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 avvisi generati dall'evento. Gli avvisi sono ordinati in ordine cronologico inverso.
selectHosts	query	Restituisce una proprietà `hosts` con gli host che contengono l'oggetto che ha creato l'evento. Supportato solo per gli 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 maintenance attive e delle soppressioni manuali: `maintenanceid` - `(ID)` ID della maintenance; `userid` - `(ID)` ID dell'utente che ha soppresso l'evento; `suppress_until` - `(integer)` ora fino alla 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 proprietà del tipo di dati `text`.

Parameter	Type	Description
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `eventid`, `objectid`, `clock`. Valori possibili se utilizzato insieme a `groupBy`: `objectid`. Valori possibili se utilizzato 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	
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`, ma non è stato utilizzato il parametro `groupBy`
- un array di oggetti con i risultati dell'aggregazione, se è stato utilizzato 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": "La media di carico è troppo alta (carico per CPU superiore a 1.5 per 5m)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Medie di carico(1m 5m 15m): (1.6328 3.0522 2.2515), n. di CPU: 2",
            "suppression_data": [],
            "suppressed": "0",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        },
        {
            "eventid": "4",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728657737",
            "value": "1",
            "acknowledged": "1",
            "ns": "460759366",
            "name": "La media di carico è troppo alta (carico per CPU superiore a 1.5 per 5m)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Ambiente di test. Ignorare questo avviso.",
```

```
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Medie di carico(1m 5m 15m): (1.6328 3.0522 2.2515), n. di CPU: 2",
            "suppression_data": [
                {
                    "maintenanceid": "0",
                    "suppress_until": "1728744338",
                    "userid": "1"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "scope",
                    "value": "capacity"
                },
                {
                    "tag": "scope",
                    "value": "performance"
                },
                {
                    "tag": "component",
                    "value": "cpu"
                },
                {
                    "tag": "class",
                    "value": "os"
                },
                {
                    "tag": "target",
                    "value": "linux"
                }
            ],
            "urls": []
        }
    ],
    "id": 1
}
```


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 dall'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",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "opdata": "",
            "suppressed": "0",
            "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.

Proprietà	Tipo	Descrizione
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.
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 grafici.

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 User roles.

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 grafici.

> **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 grafici 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 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.
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	

Parametro	Type	Descrizione
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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 gruppo di proxy ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
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. I suffissi temporali sono supportati, ad esempio 30s, 1m. Le macro utente sono supportate. Valori possibili: 10s-15m. Predefinito: 1m.
min_online	string	Numero minimo di proxy online richiesti per mantenere il gruppo di proxy online. Le macro utente sono supportate. Intervallo di valori possibili: 1-1000. Predefinito: 1.

Proprietà	Tipo	Descrizione
state	integer	Stato del gruppo di proxy. Valori possibili: 0 - Sconosciuto; 1 - Offline; 2 - In ripristino; 3 - Online; 4 - In degrado. 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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 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)` 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`.
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 di proxy esistenti.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Vedere User roles per ulteriori informazioni.

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 chiamare 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)`

Questo metodo consente di recuperare i gruppi di template 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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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.
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 chiamare 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 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 chiamare 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 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 gruppi di template 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 User roles.

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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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 gruppi di host.

Un gruppo di host non può essere eliminato se:

- contiene host che appartengono solo a questo gruppo;
- è contrassegnato come interno;
- è utilizzato da un prototipo di host;
- è utilizzato in uno script globale;
- è utilizzato in una condizione di correlazione.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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)`

Questo 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 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
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 di host specificati.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 di host.

Note:
Questo metodo è disponibile solo per gli utenti di tipo *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 tag a tutti i sottogruppi di un gruppo di host.

Note:
Questo metodo è disponibile solo per gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Vedi Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 utenti

L'oggetto gruppo utenti ha le seguenti proprietà.

Property	Type	Description
usrgrpid	ID	ID del gruppo utenti. Property behavior: - *sola lettura* - *obbligatorio* per le operazioni di aggiornamento
name	string	Nome del gruppo utenti. Property behavior: - *obbligatorio* per le operazioni di creazione
debug_mode	integer	Indica se la modalità di debug è abilitata o disabilitata. Valori possibili: 0 - *(predefinito)* disabilitata; 1 - abilitata.
gui_access	integer	Metodo di autenticazione del frontend per gli utenti del gruppo. Valori possibili: 0 - *(predefinito)* usa il metodo di autenticazione predefinito del sistema; 1 - usa l'autenticazione interna; 2 - usa l'autenticazione LDAP; 3 - disabilita l'accesso al frontend.
mfa_status	integer	Indica se l'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. Property behavior: - *supportato* per utenti di tipo *Super admin* - *supportato* se `mfa_status` dell'oggetto Authentication è impostato su "Enabled"
users_status	integer	Indica se il gruppo utenti è abilitato o disabilitato. Per gli utenti deprovisioned, il gruppo utenti non può essere abilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.

Property	Type	Description
userdirectoryid	ID	ID della directory utenti utilizzata per l'autenticazione. Property behavior: - *supportato* per utenti di tipo *Super admin* - *supportato* se `gui_access` è impostato su "use the system default authentication method" o "use LDAP authentication"

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.

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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

`(object/array)` Gruppi utenti da creare.

Oltre alle proprietà standard del gruppo utenti, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
hostgroup_rights	object/array	Permessi dei gruppi host da assegnare al gruppo utenti.
templategroup_rights	object/array	Permessi dei gruppi template da assegnare al gruppo utenti.

Parametro	Type	Descrizione
tag_filters	array	Permessi basati sui tag da assegnare al gruppo utenti.
users	object/array	Utenti da aggiungere al gruppo 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)`

Questo metodo consente di recuperare i gruppi di utenti 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Tipo	Descrizione
mfaids	ID/array	Restituisce solo i gruppi utenti con i metodi MFA specificati. Comportamento del parametro: - *supportato* per utenti di tipo *Super admin*
mfa_status	integer	Restituisce solo i gruppi utenti con lo stato MFA specificato. Fare riferimento alla pagina del gruppo utenti per un elenco degli stati supportati.
status	integer	Restituisce solo i gruppi utenti con lo stato specificato. Fare riferimento alla pagina del gruppo utenti per un elenco degli stati supportati.
userids	ID/array	Restituisce solo i gruppi utenti che contengono gli utenti specificati.
usrgrpids	ID/array	Restituisce solo i gruppi utenti con gli ID specificati.
selectTagFilters	query	Restituisce i permessi del gruppo utenti basati sui tag nella proprietà `tag_filters`.
selectUsers	query	Restituisce gli utenti del gruppo utenti nella proprietà `users`. Vedere `user.get` per le restrizioni basate sul tipo di utente.
selectHostGroupRights	query	Restituisce i permessi del gruppo utenti sui gruppi host nella proprietà `hostgroup_rights`. Fare riferimento alla pagina del gruppo utenti per un elenco dei livelli di accesso ai gruppi host.
selectTemplateGroupRights	query	Restituisce i permessi del gruppo utenti sui gruppi template nella proprietà `templategroup_rights`. Fare riferimento alla pagina del gruppo utenti per un elenco dei livelli di accesso ai gruppi template.
limitSelects	integer	Limita il numero di record restituiti dalle sottoquery.
output	query	Proprietà dell'oggetto gruppo 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 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. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà del gruppo utenti da aggiornare.

La proprietà `usrgrpid` deve essere definita per ciascun gruppo utenti, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, tutte le altre rimarranno invariate.

Oltre alle proprietà standard del gruppo utenti, il metodo accetta i seguenti parametri.

Parametro	Tipo	Descrizione
hostgroup_rights	object/array	Permessi del gruppo host per sostituire i permessi correnti assegnati al gruppo utenti.
templategroup_rights	object/array	Permessi del gruppo template per sostituire i permessi correnti assegnati al gruppo utenti.
tag_filters	array	Permessi basati sui tag per sostituire i permessi correnti assegnati al gruppo utenti.
users	object/array	Utenti per sostituire gli utenti correnti assegnati al gruppo 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 normale; 4 - un host convertito da prototipo. Comportamento della proprietà: - *sola lettura*
inventory_mode	integer	Modalità di popolamento dell'inventario host. Valori possibili: -1 - *(predefinito)* disabilitato; 0 - manuale; 1 - automatico.
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 - admin; 5 - OEM.
ipmi_username	string	Nome utente IPMI.
maintenance_from	timestamp	Ora di inizio della manutenzione effettiva. Comportamento della proprietà: - *sola lettura*
maintenance_status	integer	Stato della manutenzione effettiva. Valori possibili: 0 - *(predefinito)* nessuna manutenzione; 1 - manutenzione in corso. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
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 effettiva 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.
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 "Proxy group"
status	integer	Stato e funzione dell'host. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.
tls_connect	integer	Connessioni verso l'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 bitmask; è accettata qualsiasi somma dei valori bitmap possibili (ad esempio, 6 per PSK e certificato).
tls_issuer	string	Emittente del certificato.
tls_subject	string	Soggetto del certificato.
tls_psk_identity	string	Identità PSK; deve essere associata a una sola PSK (tra autoregistration, hosts e proxies). Non includere informazioni sensibili nell'identità PSK, poiché viene inviata in rete senza crittografia per informare il destinatario su quale PSK utilizzare. Comportamento della proprietà: - *sola scrittura* - *obbligatoria* se `tls_connect` è impostato su "PSK", oppure se `tls_accept` contiene il bit "PSK"

Property	Type	Description
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 attiva 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
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

ID	Proprietà	Type	Descrizione	Lunghezza massima
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*.

Proprietà	Tipo	Descrizione
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 gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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

Creare un host denominato "PSK host" con la crittografia PSK configurata. Si noti che l'host deve essere preconfigurato per utilizzare 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 gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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. I permessi per chiamare il metodo possono essere revocati nelle impostazioni dei ruoli utente. Consultare 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.
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.

Parametro	Tipo	Descrizione
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 oggetti correlati da più 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)` 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 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 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 del host

Aggiorna il host "10590" in modo che utilizzi la crittografia PSK solo per le connessioni dal host al server Zabbix e modifica l'identità PSK e la chiave PSK. Si noti che il host deve essere preconfigurato per utilizzare 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)`

Questo 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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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*. I permessi per chiamare il metodo possono essere revocati 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo 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 del ruolo utente. Vedere User roles 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 immagini 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. Per ulteriori informazioni, vedere Ruoli utente.

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 settings ha 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 di sistema. Per l'elenco completo dei fusi orari supportati, fare riferimento alla documentazione PHP.
default_theme	string	Tema predefinito. Valori possibili: `blue-theme` - *(predefinito)* Blu; `dark-theme` - Scuro; `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 mostrare all'interno di una cella della tabella. Predefinito: 50.
server_check_interval	integer	Mostra un avviso se Zabbix server non è disponibile. Valori possibili: 0 - Non mostrare l'avviso; 10 - *(predefinito)* Mostrare l'avviso.
work_period	string	Orario di lavoro. Predefinito: 1-5,09:00-18:00.
show_technical_errors	integer	Mostra errori tecnici (PHP/SQL) agli utenti non Super admin e agli utenti che non fanno parte di gruppi utenti con la modalità debug abilitata. Valori possibili: 0 - *(predefinito)* Non mostrare errori tecnici; 1 - Mostrare errori tecnici.
history_period	string	Periodo massimo per visualizzare i dati storici nei widget dashboard Dati recenti, Web e Panoramica dati. Accetta secondi e unità di tempo con suffisso. Predefinito: 24h.
period_default	string	Periodo predefinito del filtro temporale. Accetta secondi e unità di tempo con suffisso, con supporto per mesi e anni (30s, 1m, 2h, 1d, 1M, 1y). Predefinito: 1h.

Property	Type	Description
max_period	string	Periodo massimo per il filtro temporale. Accetta secondi e unità di tempo con suffisso, con supporto per mesi e anni (30s, 1m, 2h, 1d, 1M, 1y). Predefinito: 2y.
severity_color_0	string	Colore per la severità ”Non classificato” come codice colore esadecimale. Predefinito: 97AAB3.
severity_color_1	string	Colore per la severità ”Informazione” come codice colore esadecimale. Predefinito: 7499FF.
severity_color_2	string	Colore per la severità ”Avviso” come codice colore esadecimale. Predefinito: FFC859.
severity_color_3	string	Colore per la severità ”Media” come codice colore esadecimale. Predefinito: FFA059.
severity_color_4	string	Colore per la severità ”Alta” come codice colore esadecimale. Predefinito: E97659.
severity_color_5	string	Colore per la severità ”Disastro” come codice colore esadecimale. Predefinito: E45959.
severity_name_0	string	Nome per la severità ”Non classificato”. Predefinito: Non classificato.
severity_name_1	string	Nome per la severità ”Informazione”. Predefinito: Informazione.
severity_name_2	string	Nome per la severità ”Avviso”. Predefinito: Avviso.
severity_name_3	string	Nome per la severità ”Media”. Predefinito: Media.
severity_name_4	string	Nome per la severità ”Alta”. Predefinito: Alta.
severity_name_5	string	Nome per la severità ”Disastro”. Predefinito: Disastro.
custom_color	integer	Usa colori personalizzati per lo stato degli eventi. Valori possibili: 0 - *(predefinito)* Non usare colori personalizzati per lo stato degli eventi; 1 - Usare colori personalizzati per lo stato degli eventi.
ok_period	string	Periodo di visualizzazione dei trigger OK. Accetta secondi e unità di tempo con suffisso. Predefinito: 5m.
blink_period	string	Periodo di lampeggiamento al cambio di stato del trigger. Accetta secondi e unità di tempo con suffisso. Predefinito: 2m.
problem_unack_color	string	Colore per gli eventi PROBLEM non riconosciuti come codice colore esadecimale. Predefinito: CC0000.

Property	Type	Description
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 mostrare il lampeggiamento; 1 - *(predefinito)* Mostrare il lampeggiamento.
problem_ack_style	integer	Lampeggiamento per gli eventi PROBLEM riconosciuti. Valori possibili: 0 - Non mostrare il lampeggiamento; 1 - *(predefinito)* Mostrare il lampeggiamento.
ok_unack_style	integer	Lampeggiamento per gli eventi RESOLVED non riconosciuti. Valori possibili: 0 - Non mostrare il lampeggiamento; 1 - *(predefinito)* Mostrare il lampeggiamento.
ok_ack_style	integer	Lampeggiamento per gli eventi RESOLVED riconosciuti. Valori possibili: 0 - Non mostrare il lampeggiamento; 1 - *(predefinito)* Mostrare il lampeggiamento.
url	string	URL del frontend.
discovery_groupid	ID	ID del gruppo 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 utenti a cui verrà inviato il messaggio di allarme per database non disponibile. Se impostato su "0", il messaggio di allarme non verrà inviato.
snmptrap_logging	integer	Registra gli SNMP trap non corrispondenti. Valori possibili: 0 - Non registrare gli SNMP trap non corrispondenti; 1 - *(predefinito)* Registrare gli SNMP trap non corrispondenti.
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 secondi e unità di tempo con suffisso. Predefinito: 30s.

Property	Type	Description
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	Header HTTP X-Frame-Options. Predefinito: SAMEORIGIN.
iframe_sandboxing_enabled	integer	Usa il sandboxing iframe. Valori possibili: 0 - Non usare; 1 - *(predefinito)* Usare.
iframe_sandboxing_exceptions	string	Eccezioni del sandboxing iframe.
connect_timeout	string	Timeout di connessione con Zabbix server. Intervallo di valori possibili: 1-30s. Predefinito: 3s. Comportamento della proprietà: - *required*
socket_timeout	string	Timeout predefinito di rete. Intervallo di valori possibili: 1-300s. Predefinito: 3s. Comportamento della proprietà: - *required*
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: - *required*
item_test_timeout	string	Timeout di rete per i test degli item. Intervallo di valori possibili: 1-600s. Predefinito: 60s. Comportamento della proprietà: - *required*
script_timeout	string	Timeout di rete per l’esecuzione degli script. Intervallo di valori possibili: 1-300s. Predefinito: 60s. Comportamento della proprietà: - *required*

Property	Type	Description
report_test_timeout	string	Timeout di rete per il test dei report pianificati. Intervallo di valori possibili: 1-300s. Predefinito: 60s. Comportamento della proprietà: - *required*
auditlog_enabled	integer	Indica se abilitare il logging di audit. Valori possibili: 0 - Disabilitare; 1 - *(predefinito)* Abilitare.
auditlog_mode	integer	Indica se abilitare il logging di audit delle attività di low-level discovery, network discovery e autoregistrazione eseguite dal server (utente di sistema). Valori possibili: 0 - Disabilitare; 1 - *(predefinito)* Abilitare.
ha_failover_delay	string	Ritardo di failover in secondi. Predefinito: 1m. Comportamento della proprietà: - *read-only*
geomaps_tile_provider	string	Provider dei 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 dei tile della geomappa. Comportamento della proprietà: - *supported* se `geomaps_tile_provider` è impostato su stringa vuota
geomaps_max_zoom	integer	Livello massimo di zoom della geomappa. Valori possibili: 0-30. Comportamento della proprietà: - *supported* se `geomaps_tile_provider` è impostato su stringa vuota
geomaps_attribution	string	Testo di attribuzione della geomappa. Comportamento della proprietà: - *supported* se `geomaps_tile_provider` è impostato su stringa vuota
vault_provider	integer	Provider del vault. Valori possibili: 0 - *(predefinito)* HashiCorp Vault; 1 - CyberArk Vault.

Property	Type	Description
proxy_secrets_provider	integer	Risolvi i valori delle macro segrete tramite: Valori possibili: 0 - *(predefinito)* solo server; 1 - server e proxy in modo indipendente.
timeout_zabbix_agent	string	Non impiegare più di `timeout_*` secondi per l'elaborazione. Accetta secondi o unità di tempo con suffisso (ad esempio, 30s, 1m). Accetta anche macro utente. Intervallo di valori possibili: 1-600s. Predefinito: 3s. Predefinito per `timeout_browser`: 60s. Comportamento della proprietà: *- required*
timeout_simple_check		
timeout_snmp_agent		
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

Descrizione

`object settings.get(object parameters)`

Questo metodo consente di recuperare l'oggetto delle impostazioni in base ai parametri forniti.

> **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 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": {
        "default_theme": "blue-theme",
        "search_limit": "1000",
        "max_in_table": "50",
        "server_check_interval": "10",
        "work_period": "1-5,09:00-18:00",
        "show_technical_errors": "0",
        "history_period": "24h",
        "period_default": "1h",
        "max_period": "2y",
        "severity_color_0": "97AAB3",
        "severity_color_1": "7499FF",
        "severity_color_2": "FFC859",
        "severity_color_3": "FFA059",
        "severity_color_4": "E97659",
        "severity_color_5": "E45959",
        "severity_name_0": "Not classified",
        "severity_name_1": "Information",
        "severity_name_2": "Warning",
        "severity_name_3": "Average",
        "severity_name_4": "High",
        "severity_name_5": "Disaster",
        "custom_color": "0",
        "ok_period": "5m",
        "blink_period": "2m",
        "problem_unack_color": "CC0000",
        "problem_ack_color": "CC0000",
        "ok_unack_color": "009900",
        "ok_ack_color": "009900",
        "problem_unack_style": "1",
        "problem_ack_style": "1",
        "ok_unack_style": "1",
        "ok_ack_style": "1",
        "discovery_groupid": "5",
        "default_inventory_mode": "-1",
        "alert_usrgrpid": "7",
        "snmptrap_logging": "1",
        "default_lang": "en_US",
        "default_timezone": "system",
        "login_attempts": "5",
        "login_block": "30s",
        "validate_uri_schemes": "1",
        "uri_valid_schemes": "http,https,ftp,file,mailto,tel,ssh",
        "x_frame_options": "SAMEORIGIN",
        "iframe_sandboxing_enabled": "1",
        "iframe_sandboxing_exceptions": "",
        "max_overview_table_size": "50",
        "connect_timeout": "3s",
        "socket_timeout": "3s",
        "media_type_test_timeout": "65s",
        "script_timeout": "60s",
        "item_test_timeout": "60s",
        "url": "",
        "report_test_timeout": "60s",
        "auditlog_enabled": "1",
        "auditlog_mode": "1",
        "ha_failover_delay": "1m",
        "geomaps_tile_provider": "OpenStreetMap.Mapnik",
        "geomaps_tile_url": "",
```

```
        "geomaps_max_zoom": "0",
        "geomaps_attribution": "",
        "vault_provider": "0",
        "timeout_zabbix_agent": "3s",
        "timeout_simple_check": "3s",
        "timeout_snmp_agent": "3s",
        "timeout_external_check": "3s",
        "timeout_db_monitor": "3s",
        "timeout_http_agent": "3s",
        "timeout_ssh_agent": "3s",
        "timeout_telnet_agent": "3s",
        "timeout_script": "3s"
    },
    "id": 1
}
```


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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

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 dell'API 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*.

Interfaccia host

Questa classe è progettata per lavorare con le interfacce host.

Riferimenti agli oggetti:

- Interfaccia host
  - Dettagli

Metodi disponibili:

- hostinterface.create - crea nuove interfacce host
- hostinterface.delete - elimina interfacce host
- hostinterface.get - recupera interfacce host
- hostinterface.massadd - aggiunge interfacce host agli host

- hostinterface.massremove - rimuove interfacce host dagli host
- hostinterface.update - aggiorna interfacce host

Oggetto interfaccia dell'host

I seguenti oggetti sono direttamente correlati all'API `hostinterface`.

Interfaccia host

L'oggetto interfaccia host ha le seguenti proprietà.

Attention:
Si noti che entrambe le proprietà `ip` e `dns` sono *obbligatorie* per le operazioni di creazione. Se non si desidera utilizzare DNS, impostarlo come stringa vuota.

Proprietà	Type	Descrizione
interfaceid	ID	ID dell'interfaccia. Property behavior: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
available	integer	Disponibilità dell'interfaccia host. Valori possibili: 0 - *(predefinito)* sconosciuta; 1 - disponibile; 2 - non disponibile. Property behavior: - *sola lettura*
hostid	ID	ID dell'host a cui appartiene l'interfaccia. Property behavior: - *costante* - *obbligatoria* per le operazioni di creazione
type	integer	Tipo di interfaccia. Valori possibili: 1 - Agent; 2 - SNMP; 3 - IPMI; 4 - JMX. Property behavior: - *obbligatoria* per le operazioni di creazione
ip	string	Indirizzo IP utilizzato dall'interfaccia. Può essere vuoto se la connessione viene effettuata tramite DNS. Property behavior: - *obbligatoria* per le operazioni di creazione
dns	string	Nome DNS utilizzato dall'interfaccia. Può essere vuoto se la connessione viene effettuata tramite IP. Property behavior: - *obbligatoria* per le operazioni di creazione
port	string	Numero di porta utilizzato dall'interfaccia. Può contenere macro utente. Property behavior: - *obbligatoria* per le operazioni di creazione

Proprietà	Type	Descrizione
useip	integer	Indica se la connessione deve essere effettuata tramite IP. Valori possibili: 0 - connetti utilizzando il nome DNS dell'host; 1 - connetti utilizzando l'indirizzo IP dell'host. Property behavior: - *obbligatoria* per le operazioni di creazione
main	integer	Indica se l'interfaccia viene utilizzata 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. Property behavior: - *obbligatoria* per le operazioni di creazione
details	object	Oggetto details aggiuntivo per l'interfaccia. Property behavior: - *obbligatorio* se `type` è impostato su "SNMP"
disable_until	timestamp	Ora del prossimo polling di un'interfaccia host non disponibile. Property behavior: - *sola lettura*
error	string	Testo dell'errore se l'interfaccia host non è disponibile. Property behavior: - *sola lettura*
errors_from	timestamp	Ora in cui l'interfaccia host è diventata non disponibile. Property behavior: - *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"

Proprietà	Type	Descrizione
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.create

Descrizione

`object hostinterface.create(object/array hostInterfaces)`

Questo metodo consente di creare nuove interfacce 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)` Interfacce host da creare.

Il metodo accetta interfacce host con le proprietà standard dell'interfaccia host.

Restituisce i valori

`(object)` Restituisce un oggetto contenente gli ID delle interfacce host create nella proprietà `interfaceids`. L'ordine degli ID restituiti corrispondono all'ordine delle interfacce host passate.

Esempi

Creare una nuova interfaccia

Creare un'interfaccia agent IP secondaria sull'host "30052".

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "30052",
        "main": "0",
        "type": "1",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "10050"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Creare un'interfaccia con dettagli SNMP

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.create",
    "params": {
        "hostid": "10456",
        "main": "0",
        "type": "2",
        "useip": "1",
        "ip": "127.0.0.1",
        "dns": "",
        "port": "1601",
        "details": {
            "version": "2",
            "bulk": "1",
            "community": "{$SNMP_COMMUNITY}"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Guarda anche

- hostinterface.massadd
- host.massadd

Fonte

CHostInterface::create() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Descrizione

`object hostinterface.delete(array hostInterfaceIds)`

Questo metodo consente di eliminare le interfacce 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 delle interfacce host da eliminare.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle interfacce host eliminate sotto la proprietà `interfaceids`.

Esempi

Eliminare un'interfaccia host

Eliminare l'interfaccia host con ID 30062.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Guarda anche

- hostinterface.massremove
- host.massremove

Fonte

CHostInterface::delete() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Descrizione

`integer/array hostinterface.get(object parameters)`

Questo metodo consente di recuperare le interfacce host 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 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
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*.

hostinterface.massadd

Descrizione

`object hostinterface.massadd(object parameters)`

Questo metodo consente di aggiungere simultaneamente interfacce host a più host.

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 del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object)` Parametri contenenti le interfacce host da creare sugli host specificati.

Il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
interfaces	object/array	Interfacce host da creare sugli host specificati. Comportamento del parametro: - *obbligatorio*
hosts	object/array	Host da aggiornare. Gli host devono avere definita solo la proprietà `hostid`. Comportamento del parametro: - *obbligatorio*

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle interfacce dell'host create sotto la proprietà `interfaceids`.

Esempi

Creazione delle interfacce

Crea un'interfaccia su due host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30052"
            }
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "main": 0,
            "port": "10050",
            "type": 1,
            "useip": 1
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
```

```
        ]
    },
    "id": 1
}
```

Guarda anche

- hostinterface.create
- host.massadd
- Host

Fonte

CHostInterface::massAdd() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Descrizione

`object hostinterface.massremove(object parameters)`

Questo metodo consente di rimuovere le interfacce host dagli host specificati.

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 User roles.

Parametri

`(object)` Parametri contenenti gli ID degli host da aggiornare e le interfacce da rimuovere.

Parametro	Type	Descrizione
interfaces	object/array	Interfacce host da rimuovere dagli host specificati. L'oggetto interfaccia host deve avere definite solo le proprietà `ip`, `dns` e `port`. Comportamento del parametro: - *obbligatorio*
hostids	ID/array	ID degli host da aggiornare. Comportamento del parametro: - *obbligatorio*

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle interfacce host eliminate sotto la proprietà `interfaceids`.

Esempi

Rimozione delle interfacce

Rimuovere l'interfaccia SNMP "127.0.0.1" da due host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
```

```
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Guarda anche

- hostinterface.delete
- host.massremove

Fonte

CHostInterface::massRemove() in *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

Descrizione

`object hostinterface.update(object/array hostInterfaces)`

Questo metodo consente di aggiornare le interfacce host esistenti.

> **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 del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Proprietà dell'interfaccia host da aggiornare.

La proprietà `interfaceid` deve essere definita per ciascuna interfaccia host, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà specificate, tutte le altre rimarranno invariate.

Valori di ritorno

`(object)` Restituisce un oggetto contenente gli ID delle interfacce dell'host aggiornate sotto la proprietà `interfaceids`.

Esempi

Modifica della porta di un'interfaccia host

Modificare la porta di un'interfaccia host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "10055"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

Modifica di più porte di interfaccia host

Modificare la porta di più interfacce host.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": [
        {
            "interfaceid": "30049",
            "port": "10055"
        },
        {
            "interfaceid": "30050",
            "port": "10055"
        },
        {
            "interfaceid": "30051",
            "port": "10055"
        }
    ],
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30049",
            "30050",
            "30051"
        ]
    },
    "id": 1
}
```

Fonte

CHostInterface::update() 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:**
> I web item non possono essere creati, aggiornati o eliminati direttamente tramite la Zabbix API.

L'oggetto item ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
itemid	ID	ID dell'item. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
delay	string	Intervallo di aggiornamento dell'item. Accetta secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente. Se utilizzato, il valore deve essere una singola macro. Macro multiple o macro combinate con testo non sono supportate. 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), oppure se `type` è impostato su "Zabbix agent (active)" (7) e `key_` non contiene "mqtt.get"
hostid	ID	ID del host o del template a cui appartiene l'item. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
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 individuati automaticamente

Proprietà	Tipo	Descrizione
key_	string	Chiave dell'item. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o individuati automaticamente
name	string	Nome dell'item. Supporta macro utente. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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 individuati automaticamente
value_type	integer	Tipo di informazione dell'item. Valori possibili: 0 - numerico in 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 o individuati automaticamente

Proprietà	Tipo	Descrizione
allow_traps	integer	Consente di popolare il valore in modo simile a un item trapper. 0 - *(predefinito)* Non consentire l'accettazione di dati in ingresso; 1 - Consentire l'accettazione di dati in ingresso. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti individuati automaticamente
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 individuati automaticamente
description	string	Descrizione dell'item. Comportamento della proprietà: - *sola lettura* per gli oggetti individuati automaticamente
error	string	Testo dell'errore se ci sono problemi nell'aggiornamento del valore dell'item. Comportamento della proprietà: - *sola lettura*
flags	integer	Origine dell'item. Valori possibili: 0 - un item normale; 4 - un item convertito da prototipo. Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Seguire i reindirizzamenti della risposta durante l'interrogazione 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 individuati automaticamente
headers	array	Array di header 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 o individuati automaticamente

Proprietà	Tipo	Descrizione
history	string	Unità di tempo che definisce per quanto tempo devono essere conservati i dati storici. Accetta anche macro utente. Predefinito: 31d. Comportamento della proprietà: - *sola lettura* per gli oggetti individuati automaticamente
http_proxy	string	Stringa di connessione del proxy HTTP(S). Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
inventory_link	integer	ID del campo di inventario host popolato dall’item. Fare riferimento alla pagina dell’inventario host per un elenco dei campi di 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 individuati automaticamente
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 individuati automaticamente
jmx_endpoint	string	Stringa di connessione personalizzata di JMX agent. 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 individuati automaticamente
lastclock	timestamp	Ora dell’ultimo aggiornamento del valore dell’item. Per impostazione predefinita, vengono visualizzati solo i valori che rientrano nelle ultime 24 ore. È possibile estendere questo intervallo di tempo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration → General*. Comportamento della proprietà: - *sola lettura*
lastns	integer	Nanosecondi dell’ultimo aggiornamento del valore dell’item. Per impostazione predefinita, vengono visualizzati solo i valori che rientrano nelle ultime 24 ore. È possibile estendere questo intervallo di tempo modificando il valore del parametro *Max history display period* nella sezione di menu *Administration → General*. Comportamento della proprietà: - *sola lettura*

Proprietà	Tipo	Descrizione
lastvalue	string	Ultimo valore dell'item. Per impostazione predefinita, vengono visualizzati solo i valori che rientrano nelle ultime 24 ore. È possibile estendere questo intervallo di tempo 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 individuati automaticamente
master_itemid	ID	ID del master item. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Dependent item" - *sola lettura* per gli oggetti ereditati o individuati automaticamente
output_format	integer	Indica se la risposta deve essere convertita in JSON. 0 - *(predefinito)* Memorizza il dato grezzo; 1 - Converti in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati o individuati automaticamente
params	string	Parametri aggiuntivi a seconda del 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 individuati automaticamente
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 o individuati automaticamente
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 individuati automaticamente

Proprietà	Tipo	Descrizione
post_type	integer	Tipo del corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati grezzi; 2 - Dati JSON; 3 - Dati XML. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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 individuati automaticamente
prevvalue	string	Valore precedente dell’item. Per impostazione predefinita, vengono visualizzati solo i valori che rientrano nelle ultime 24 ore. È possibile estendere questo intervallo di tempo 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 individuati automaticamente
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 individuati automaticamente
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 o individuati automaticamente
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 individuati automaticamente

Proprietà	Tipo	Descrizione
retrieve_mode	integer	Quale parte della risposta deve essere memorizzata. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Header; 2 - Verranno memorizzati sia il corpo sia gli header. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Header. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
snmp_oid	string	OID SNMP. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SNMP agent” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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 individuati automaticamente
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 individuati automaticamente
ssl_key_password	string	Password per il file della chiave SSL. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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.
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 o individuati automaticamente

Proprietà	Tipo	Descrizione
templateid	ID	ID dell'item 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 secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m). Accetta anche macro utente. Intervallo di valori possibile: 1-600s. Predefinito: "" - usa 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 individuati automaticamente
trapper_hosts	string	Host consentiti. Comportamento della proprietà: - *sola lettura* per gli oggetti individuati automaticamente - *supportata* se `type` è impostato su "Zabbix trapper", oppure se `type` è impostato su "HTTP agent" e `allow_traps` è impostato su "Allow to accept incoming data"
trends	string	Unità di tempo che definisce per quanto tempo devono essere conservati i dati di trend. Accetta anche macro utente. Predefinito: 365d. Comportamento della proprietà: - *supportata* se `value_type` è impostato su "numeric float" o "numeric unsigned" - *sola lettura* per gli oggetti individuati automaticamente
units	string	Unità di misura del valore. Comportamento della proprietà: - *supportata* se `value_type` è impostato su "numeric float" o "numeric unsigned" - *sola lettura* per gli oggetti ereditati o individuati automaticamente
username	string	Nome utente per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SSH agent", "TELNET agent", oppure se `type` è impostato su "JMX agent" e `password` è impostato - *supportata* 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") o individuati automaticamente
uuid	string	Identificatore univoco universale, utilizzato per collegare un item importato a quelli già esistenti. Viene generato automaticamente se non fornito. Comportamento della proprietà: - *supportata* se l'item appartiene a un template

Proprietà	Tipo	Descrizione
valuemapid	ID	ID della mappa valori associata. Comportamento della proprietà: - *supportata* se `value_type` è impostato su ”numeric float”, ”character” o ”numeric unsigned” - *sola lettura* per gli oggetti ereditati o individuati automaticamente
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 o individuati automaticamente
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 o individuati automaticamente

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*

Proprietà	Type	Descrizione
value	string	Valore del parametro. Comportamento della proprietà: - *obbligatorio*

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 degli item

L'oggetto di preelaborazione degli item ha le seguenti proprietà.

Proprietà	Type	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 - Variazione semplice; 10 - Variazione al secondo; 11 - XML XPath; 12 - JSONPath; 13 - Nell'intervallo; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Controlla errore in JSON; 17 - Controlla errore in XML; 18 - Controlla errore usando un'espressione regolare; 19 - Scarta se invariato; 20 - Scarta se invariato con heartbeat; 21 - JavaScript; 22 - Pattern Prometheus; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituisci; 26 - Controlla 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 usati dall'opzione di preelaborazione. I parametri multipli sono separati dal carattere newline (\n). Se `type` è impostato su "Check unsupported", 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 - controlla se il messaggio di errore corrisponde a *pattern*; 1 - controlla se il messaggio di errore non corrisponde a *pattern*. Comportamento della proprietà: - *required* se `type` è impostato su "Custom multiplier" (1), "Right trim" (2), "Left trim" (3), "Trim" (4), "Regular expression" (5), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Discard unchanged with heartbeat" (20), "JavaScript" (21), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Replace" (25), Check unsupported (26), "SNMP walk value" (28), "SNMP walk to JSON" (29), oppure "SNMP get value" (30)

Proprietà	Type	Descrizione
error_handler	integer	Tipo di azione usato in caso di errore del passaggio di preelaborazione. Valori possibili: 0 - Il messaggio di errore è impostato da Zabbix server; 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Valori possibili se `type` è impostato su ”Check unsupported”: 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Comportamento della proprietà: - *required* se `type` è impostato su ”Custom multiplier” (1), ”Regular expression” (5), ”Boolean to decimal” (6), ”Octal to decimal” (7), ”Hexadecimal to decimal” (8), ”Simple change” (9), ”Change per second” (10), ”XML XPath” (11), ”JSONPath” (12), ”In range” (13), ”Matches regular expression” (14), ”Does not match regular expression” (15), ”Check for error in JSON” (16), ”Check for error in XML” (17), ”Check for error using regular expression” (18), ”Prometheus pattern” (22), ”Prometheus to JSON” (23), ”CSV to JSON” (24), ”Check unsupported” (26), ”XML to JSON” (27), ”SNMP walk value” (28), ”SNMP walk to JSON” (29), oppure ”SNMP get value” (30)
error_handler_params	string	Parametri del gestore errori. Comportamento della proprietà: - *required* se `error_handler` è impostato su ”Set custom value” oppure ”Set custom error message”

I seguenti parametri e gestori errori sono supportati per ciascun tipo di preelaborazione.

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori errori supportati
1	Moltiplicatore personalizzato	number[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 errori supportati
8	Esadecimale in decimale				0, 1, 2, 3
9	Variazione semplice				0, 1, 2, 3
10	Variazione al secondo				0, 1, 2, 3
11	XML XPath	path[4]			0, 1, 2, 3
12	JSONPath	path[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	Controlla errore in JSON	path[4]			0, 1, 2, 3
17	Controlla errore in XML	path[4]			0, 1, 2, 3
18	Controlla errore usando un’espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
19	Scarta se invariato				
20	Scarta se invariato con heartbeat	seconds[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

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori errori supportati
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	Sostituisci	stringa di ricerca[2]	sostituzione[2]		
26	Controlla 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 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 in virgola mobile (gli interi vengono convertiti implicitamente in valori float)
[2] stringa
[3] espressione regolare
[4] JSONPath o XML XPath
[5] intero positivo (con supporto dei suffissi temporali, ad esempio 30s, 1m, 2h, 1d)
[6] macro utente
[7] pattern Prometheus che segue la sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome etichetta, valore etichetta e valore della metrica) può essere una macro utente.
[8] output Prometheus che segue la 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 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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

Elimina 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 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
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 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 User roles.

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 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.
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 log 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 User roles.

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 nuove macro 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”.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Risposta:

```
{
    "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 nuove macro globali.

> **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)` 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 del ruolo 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)`

Questo metodo consente di recuperare le macro del host e globali 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 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
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 le macro del 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 le 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. Vedere Ruoli utente per ulteriori informazioni.

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`.

Manutenzione

L'oggetto manutenzione ha le seguenti proprietà.

Proprietà	Type	Descrizione
maintenanceid	ID	ID della manutenzione. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome della manutenzione. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
active_since	timestamp	Ora in cui la manutenzione diventa attiva (inclusa). Il valore specificato verrà arrotondato per difetto ai minuti. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
active_till	timestamp	Ora in cui la manutenzione cessa di essere attiva (esclusa). Il valore specificato verrà arrotondato per difetto ai minuti. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
description	string	Descrizione della manutenzione.
maintenance_type	integer	Tipo di manutenzione. Valori possibili: 0 - *(predefinito)* con raccolta dati; 1 - senza raccolta dati.
tags_evaltype	integer	Metodo di valutazione dei tag del problema. Valori possibili: 0 - *(predefinito)* 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”

Tag del problema

L’oggetto tag del problema viene utilizzato per definire quali problemi devono essere soppressi quando la manutenzione entra in vigore. I tag possono essere specificati solo se `maintenance_type` dell’oggetto Maintenance è impostato su ”with data collection”. Ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
tag	string	Nome del tag del problema. Comportamento della proprietà: - *obbligatorio*

Proprietà	Tipo	Descrizione
operator	integer	Operatore della condizione. Valori possibili: 0 - Uguale; 2 - *(predefinito)* 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

Parametri

`(object/array)` Maintenance da creare.

Oltre alle proprietà standard della maintenance, il metodo accetta i seguenti parametri.

Parametro	Type	Descrizione
groups	object/array	Gruppi di host che saranno sottoposti a maintenance. I gruppi di host devono avere definita solo la proprietà `groupid`. Comportamento del parametro: - *obbligatorio* se `hosts` non è impostato
hosts	object/array	Host che saranno sottoposti a maintenance. Gli host devono avere definita solo la proprietà `hostid`. Comportamento del parametro: - *obbligatorio* se `groups` non è impostato
timeperiods	object/array	Periodi di tempo della maintenance. Comportamento del parametro: - *obbligatorio*
tags	object/array	Tag dei problemi. Definiscono quali problemi devono essere soppressi. Se non vengono forniti tag, saranno soppressi tutti i problemi degli host in maintenance attiva. Comportamento del parametro: - *supportato* se `maintenance_type` dell'oggetto Maintenance è impostato su "with data collection"

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 host con ID "2" e con i tag problema **service:mysqld** ed **error**. Deve essere attiva dal 17.03.2026 fino 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": "2"}
        ],
        "timeperiods": [
            {
                "period": 3600,
                "timeperiod_type": 3,
                "start_time": 64800,
                "every": 1,
                "dayofweek": 64
            }
        ],
        "tags": [
            {
                "tag": "service",
                "operator": "0",
                "value": "mysqld"
            },
            {
                "tag": "error",
                "operator": "2",
                "value": ""
            }
        ]
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Vedi anche

- 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 chiamare 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)`

Questo metodo consente di recuperare le maintenance in base ai parametri forniti.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
groupids	ID/array	Restituisce solo le maintenance assegnate ai gruppi di host specificati.

Parametro	Type	Descrizione
hostids	ID/array	Restituisce solo le maintenance assegnate agli host specificati.
maintenanceids	ID/array	Restituisce solo le maintenance con gli ID specificati.
selectHostGroups	query	Restituisce una proprietà `hostgroups` con i gruppi di host assegnati alla maintenance.
selectHosts	query	Restituisce una proprietà `hosts` con gli host assegnati alla maintenance.
selectTags	query	Restituisce una proprietà `tags` con i tag problema della maintenance.
selectTimeperiods	query	Restituisce una proprietà `timeperiods` con i periodi di tempo della maintenance.
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 e i dati relativi ai gruppi di host assegnati, ai periodi di tempo definiti e ai tag dei problemi.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Risposta:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "maintenanceid": "3",
            "name": "Sunday maintenance",
            "maintenance_type": "0",
            "description": "",
            "active_since": "1358844540",
```

```
            "active_till": "1390466940",
            "tags_evaltype": "0",
            "hostgroups": [
                {
                    "groupid": "4",
                    "name": "Zabbix servers",
                    "flags": "0",
                    "uuid": "6f6799aa69e844b4b3918f779f2abf08"
                }
            ],
            "timeperiods": [
                {
                    "timeperiod_type": "3",
                    "every": "1",
                    "month": "0",
                    "dayofweek": "1",
                    "day": "0",
                    "start_time": "64800",
                    "period": "3600",
                    "start_date": "2147483647"
                }
            ],
            "tags": [
                {
                    "tag": "service",
                    "operator": "0",
                    "value": "mysqld",
                },
                {
                    "tag": "error",
                    "operator": "2",
                    "value": ""
                }
            ]
        }
    ],
    "id": 1
}
```


Vedi anche

- Host
- Gruppo host
- 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 maintenance esistenti.

> **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 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, tutte le altre rimarranno invariate.

Oltre alle proprietà standard della manutenzione, il metodo accetta i seguenti parametri.

Parameter	Type	Description
groups	object/array	Gruppi di host per 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 per sostituire gli host correnti. Gli host devono avere definita solo la proprietà `hostid`. Comportamento del parametro: - *obbligatorio* se `groups` non è impostato
timeperiods	object/array	Periodi temporali della manutenzione per sostituire i periodi correnti.
tags	object/array	Tag dei problemi per sostituire i tag correnti. Comportamento del parametro: - *supportato* se `maintenance_type` dell'oggetto manutenzione è impostato su "with data collection"

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
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.

Proprietà	Tipo	Descrizione
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.
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"

Proprietà	Tipo	Descrizione
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.
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.

Proprietà	Tipo	Descrizione
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*
elements	array	Oggetto dati Element. Comportamento della proprietà: - *obbligatoria* se `elementtype` è impostato su "host", "map", "trigger" o "host group"

Proprietà	Type	Descrizione
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.
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.

Proprietà	Type	Descrizione
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:

Proprietà	Type	Descrizione
sysmap_shapeid	ID	ID dell'elemento forma della mappa. Property behavior: - *read-only*
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: - *required*
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.

Proprietà	Type	Descrizione
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 nuove mappe.

> **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 User roles 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 del ruolo utente. Vedere User roles 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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 User roles 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 value map. Ha le seguenti proprietà.

Property	Type	Description
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 nell'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 nuove mappe dei valori.

> **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)` 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 mappe dei valori 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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 mappe dei valori 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 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 nuove mappe di icone.

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)` 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 di 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 User roles 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)`

Questo metodo consente di recuperare le mappe di icone 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 del ruolo utente. Vedere User roles 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 mappe di icone 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*.

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	Tipo	Descrizione
mfaid	ID	ID del metodo MFA. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
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 - lungo 6 cifre; 8 - lungo 8 cifre. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”TOTP”
api_hostname	string	Hostname 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	Secret del 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 gli utenti di tipo *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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`.

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, verrà eseguita una ricerca `LIKE "%…%"`. Proprietà supportate: `name`.
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

`(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 gli utenti di tipo *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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`.

Modulo

L'oggetto modulo ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
moduleid	ID	ID del modulo come memorizzato nel database. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
id	string	ID univoco del modulo come definito da uno sviluppatore nel file manifest.json del modulo. Valori possibili per i moduli integrati: vedere la descrizione della proprietà "type" in Dashboard widget. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
relative_path	string	Percorso della directory del modulo relativo alla directory del frontend di Zabbix. Valori possibili: `widgets/*` - per i moduli widget integrati; `modules/*` - per i moduli di terze parti. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
status	integer	Indica se il modulo è abilitato o disabilitato. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato.
config	object	Configurazione del modulo.

module.create

Descrizione

`object module.create(object/array modules)`

Questo metodo consente di installare nuovi moduli del frontend.

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. Per ulteriori informazioni, vedere User roles.

Attention:
I file del modulo devono essere estratti manualmente nelle sottodirectory corrette, in modo che corrispondano 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

Attention:
I file del modulo 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)`

Questo metodo consente di recuperare i moduli in base ai parametri specificati.

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. Per ulteriori informazioni, vedere Ruoli utente.

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	
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 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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 di alta disponibilità

> **Note:**
> I nodi vengono creati dal server Zabbix e non possono essere modificati tramite l'API.

L'oggetto nodo di alta disponibilità ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
ha_nodeid	ID	ID del nodo.
name	string	Nome assegnato al nodo, utilizzando il parametro di configurazione HANodeName di zabbix_server.conf. Vuoto per un server in esecuzione in modalità standalone.
address	string	Nome IP o DNS da cui il nodo si connette.
port	integer	Porta su cui il nodo è in esecuzione.
lastaccess	integer	Ora dell'heartbeat, cioè l'ora 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)`

Questo metodo consente di recuperare un elenco di nodi del cluster ad alta disponibilità in base ai parametri forniti.

Note:
Questo metodo è disponibile solo per i tipi di utente *Super admin*. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo metodo consente di recuperare i problemi in base ai parametri specificati.

Questo metodo serve per recuperare i problemi non risolti. È anche possibile, se specificato, recuperare inoltre i problemi risolti di recente. Il periodo che determina quanto indietro nel tempo si intende per "di recente" è definito in *Administration* → *General*. I problemi che sono stati risolti prima di tale periodo non vengono mantenuti nella tabella dei problemi. Per recuperare i problemi che sono stati risolti più indietro nel passato, utilizzare 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. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere User roles per ulteriori informazioni.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
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 del tipo specificato. Fare riferimento alla pagina dell'oggetto evento problema per un elenco dei tipi di evento supportati. Predefinito: 0 - problema creato da un trigger.
object	integer	Restituisce solo i problemi creati da oggetti del tipo specificato. Fare riferimento alla pagina dell'oggetto evento problema per un elenco dei tipi di oggetto supportati. Predefinito: 0 - trigger.
acknowledged	boolean	Se impostato su `true`, restituisce solo i problemi confermati.
action	integer	Restituisce solo i problemi per i quali sono state eseguite le azioni di aggiornamento evento specificate. Per più azioni, utilizzare la somma di qualsiasi valore bitmap accettabile come bitmask (ad esempio, 34 per confermare e sopprimere l'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.

Parametro	Type	Descrizione
symptom	boolean	Se impostato su `true`, restituisce solo gli eventi problema di tipo 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 solo i problemi risolti di recente (dipende da `ok_period`).
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 aggiornamento 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)` ora fino alla quale l'evento sarà soppresso; `taskid` - `(ID)` ID del task se l'evento corrente è soggetto a una modifica di rango. 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)` ora fino alla quale il problema è soppresso.

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 dato `text`.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `eventid`.
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 o:

- una matrice di oggetti;
- il conteggio degli oggetti recuperati, se il parametro `countOutput` è stato usato.

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",
            "opdata": "",
            "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"
                }
            ],
            "suppression_data": [
                {
                    "maintenanceid": "15",
                    "suppress_until": "1472511600",
                    "userid": "0"
                }
            ],
            "suppressed": "1",
            "tags": [
                {
                    "tag": "test-tag",
                    "value": "test-value"
                }
            ]
        }
    ],
    "id": 1
}
```


Recupero dei problemi riconosciuti dall'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 su localhost non è raggiungibile da 5 minuti",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "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.
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.

Proprietà	Type	Descrizione
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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo 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 chiamare 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)

Questo 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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.

Parametro	Type	Descrizione
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	
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 prototipi di grafico esistenti.

> **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 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.
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.

Property	Type	Description
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*
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*
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"

Proprietà	Type	Descrizione
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.
securityname	string	Nome di sicurezza SNMPv3. Comportamento della proprietà: - *supportata* se `version` è impostato su "SNMPv3"

Property	Type	Description
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 prototipi di host.

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 del ruolo 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)`

Questo 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 i prototipi di host esistenti.

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 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 ha 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 secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente e macro LLD. Se utilizzato, il valore deve essere una singola macro. Macro multiple o macro combinate con testo non sono supportate. 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), oppure se `type` è impostato su "Zabbix agent (active)" (7) e `key_` non contiene "mqtt.get"
hostid	ID	ID del host a cui appartiene il prototipo di item. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
interfaceid	ID	ID dell'interfaccia del 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 - item dipendente; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser. 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 in 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 simile a un item trapper. 0 - *(predefinito)* Non consentire l’accettazione di dati in ingresso; 1 - Consentire l’accettazione di 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 individuato automaticamente 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 header HTTP 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
history	string	Unità di tempo che definisce per quanto tempo devono essere conservati i dati storici. Accetta anche macro utente e macro LLD. Predefinito: 31d.
http_proxy	string	Stringa di connessione del 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 di JMX agent. 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 del master item. 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)* Memorizza il dato grezzo; 1 - Converti 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; - lo 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 grezzi. 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 "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
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"
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"
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	Quale parte della risposta deve essere memorizzata. Valori possibili se `request_method` è impostato su "GET", "POST" o "PUT": 0 - *(predefinito)* Corpo; 1 - Header; 2 - Verranno memorizzati sia il corpo sia gli header. Valori possibili se `request_method` è impostato su "HEAD": 1 - Header. 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 per il 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 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 di item del template padre. Comportamento della proprietà: - *sola lettura*
timeout	string	Timeout della richiesta di polling dei dati dell’item. Accetta secondi o un’unità di tempo con suffisso (ad esempio, 30s, 1m). Accetta anche macro utente e macro LLD. Intervallo di valori possibile: 1-600s. Predefinito: ”” - usa le impostazioni globali/del proxy. 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
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 to accept incoming data”

Proprietà	Tipo	Descrizione
trends	string	Unità di tempo che definisce per quanto tempo devono essere conservati i dati di 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 del valore. 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: - *obbligatoria* se `type` è impostato su ”SSH agent” o ”TELNET agent”, oppure se `type` è impostato su ”JMX agent” e `password` è impostato - *supportata* 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 fornito. Comportamento della proprietà: - *supportata* 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 del host per la connessione corrisponda a quello nel certificato del 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
verify_peer	integer	Indica se verificare che il certificato del 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 individuazione del prototipo di item. Valori possibili: 0 - *(predefinito)* i nuovi item verranno individuati; 1 - i nuovi item non verranno individuati e gli item esistenti verranno contrassegnati come persi.

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*

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*.

Proprietà	Tipo	Descrizione
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 ha le seguenti proprietà.

Property	Type	Description
type	integer	Il tipo di opzione di pre-elaborazione. Valori possibili: 1 - Moltiplicatore personalizzato; 2 - Taglio a destra; 3 - Taglio a sinistra; 4 - Taglio; 5 - Espressione regolare; 6 - Booleano in decimale; 7 - Ottale in decimale; 8 - Esadecimale in decimale; 9 - Variazione semplice; 10 - Variazione al secondo; 11 - XML XPath; 12 - JSONPath; 13 - Nell'intervallo; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Controlla errore in JSON; 17 - Controlla errore in XML; 18 - Controlla errore usando un'espressione regolare; 19 - Scarta se invariato; 20 - Scarta se invariato con heartbeat; 21 - JavaScript; 22 - Pattern Prometheus; 23 - Prometheus in JSON; 24 - CSV in JSON; 25 - Sostituisci; 26 - Controlla non supportato; 27 - XML in JSON; 28 - Valore SNMP walk; 29 - SNMP walk in JSON; 30 - Valore SNMP get. Property behavior: - *required*

Property	Type	Description
params	string	Parametri aggiuntivi usati dall'opzione di pre-elaborazione. I parametri multipli sono separati dal carattere newline (\n). Se `type` è impostato su "Check unsupported", 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 - controlla se il messaggio di errore corrisponde a *pattern*; 1 - controlla se il messaggio di errore non corrisponde a *pattern*. Property behavior: - *required* se `type` è impostato su "Custom multiplier" (1), "Right trim" (2), "Left trim" (3), "Trim" (4), "Regular expression" (5), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Discard unchanged with heartbeat" (20), "JavaScript" (21), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Replace" (25), "Check unsupported" (26), "SNMP walk value" (28), "SNMP walk to JSON" (29) o "SNMP get value" (30)
error_handler	integer	Tipo di azione usato in caso di errore del passaggio di pre-elaborazione. Valori possibili: 0 - Il messaggio di errore è impostato da Zabbix server; 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Valori possibili se `type` è impostato su "Check unsupported": 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Property behavior: - *required* se `type` è impostato su "Custom multiplier" (1), "Regular expression" (5), "Boolean to decimal" (6), "Octal to decimal" (7), "Hexadecimal to decimal" (8), "Simple change" (9), "Change per second" (10), "XML XPath" (11), "JSONPath" (12), "In range" (13), "Matches regular expression" (14), "Does not match regular expression" (15), "Check for error in JSON" (16), "Check for error in XML" (17), "Check for error using regular expression" (18), "Prometheus pattern" (22), "Prometheus to JSON" (23), "CSV to JSON" (24), "Check unsupported" (26), "XML to JSON" (27), "SNMP walk value" (28), "SNMP walk to JSON" (29) o "SNMP get value" (30)
error_handler_params	string	Parametri del gestore errori. Property behavior: - *required* se `error_handler` è impostato su "Set custom value" o "Set custom error message"

I seguenti parametri e gestori errori sono supportati per ciascun tipo di pre-elaborazione.

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
1	Moltiplicatore per-son-aliz-zato	numero[1, 6]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
2	Taglio a destra	elenco di caratteri[2]			
3	Taglio a sinistra	elenco di caratteri[2]			
4	Taglio	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
8	Esadecimale in decimale				0, 1, 2, 3
9	Variazione semplice				0, 1, 2, 3
10	Variazione al secondo				0, 1, 2, 3
11	XML XPath	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	Controlla errore in JSON	percorso[4]			0, 1, 2, 3
17	Controlla errore in XML	percorso[4]			0, 1, 2, 3

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
18	Controlla errore usando un'espressione regolare	pattern[3]	output[2]		0, 1, 2, 3
19	Scarta se invariato				
20	Scarta 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	Sostituisci	stringa di ricerca[2]	sostituzione[2]		
26	Controlla 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 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 in virgola mobile
[2] stringa
[3] espressione regolare
[4] JSONPath o XML XPath
[5] intero positivo (con supporto dei suffissi temporali, ad esempio 30s, 1m, 2h, 1d)
[6] macro utente, macro LLD
[7] Pattern Prometheus che segue la sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome etichetta, valore etichetta e valore della metrica) può essere una macro utente o una macro LLD.
[8] Output Prometheus che segue la 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 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 prototipi di item.

> **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)` 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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

Elimina 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 del ruolo utente. Vedere User roles 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.
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.

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 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.
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 prototipi di item 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 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 di regola di low-level discovery ha le seguenti proprietà.

Property	Type	Description
itemid	ID	ID del prototipo di regola LLD. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
ruleid	ID	ID della regola LLD/prototipo di regola LLD padre. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di creazione
delay	string	Intervallo di aggiornamento del prototipo di regola LLD. Accetta secondi o un'unità di tempo con suffisso (ad esempio 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente. Se utilizzate, il valore deve essere una singola macro. Macro multiple o macro mescolate con testo non sono supportate. 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 del host a cui appartiene il prototipo di regola LLD. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione
flags	integer	Origine del prototipo di regola LLD. Valori possibili: 3 - un prototipo di regola di low-level discovery; 7 - un prototipo di regola di low-level discovery rilevato. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
interfaceid	ID	ID dell'interfaccia del host del prototipo di regola LLD. Comportamento della proprietà: - *obbligatoria* se il prototipo di regola LLD appartiene a un host e `type` è impostato su "Zabbix agent", "IPMI agent", "JMX agent" o "SNMP agent" - *supportata* se il prototipo di 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 di regola LLD. È richiesta almeno una macro LLD. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
name	string	Nome del prototipo di regola LLD. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione - *sola lettura* per gli oggetti ereditati
type	integer	Tipo del prototipo di 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 - item dipendente; 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 simile a un item trapper. Valori possibili: 0 - *(predefinito)* Non consentire l'accettazione di dati in ingresso; 1 - Consentire l'accettazione di 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. 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 regola LLD.
error	string	Testo dell’errore se ci sono problemi nell’aggiornamento del valore del prototipo di regola LLD. Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Seguire 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 header 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
http_proxy	string	Stringa di connessione del 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 di JMX agent. 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 secondi, un’unità di tempo con suffisso o una macro utente. Predefinito: `7d`.

Property	Type	Description
lifetime_type	integer	Scenario per eliminare le risorse LLD perse. Valori possibili: 0 - *(predefinito)* Elimina dopo il raggiungimento della soglia lifetime; 1 - Non eliminare; 2 - Elimina immediatamente.
enabled_lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno disabilitati. Accetta secondi, un'unità di tempo con suffisso o una macro utente. Predefinito: 0.
enabled_lifetime_type	integer	Scenario per disabilitare le risorse LLD perse. Valori possibili: 0 - Disabilita dopo il raggiungimento della soglia lifetime; 1 - Non disabilitare; 2 - *(predefinito)* Disabilita immediatamente.
master_itemid	ID	ID del master item. Una regola di discovery non può essere un master item per 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)* Memorizza il dato grezzo; 1 - Converti in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati
params	string	Parametri aggiuntivi a seconda del tipo del prototipo di regola LLD: - script eseguito per regole LLD SSH e Telnet; - query SQL per regole LLD di monitoraggio database; - formula per regole LLD calcolate; - lo script per 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 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")

Property	Type	Description
post_type	integer	Tipo del corpo dei dati POST memorizzato nella proprietà `posts`. Valori possibili: 0 - *(predefinito)* Dati grezzi; 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 "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
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"
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"
query_fields	array	Array di campi 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	Quale parte della risposta deve essere memorizzata. Valori possibili se `request_method` è impostato su "GET", "POST" o "PUT": 0 - *(predefinito)* Corpo; 1 - Header; 2 - Verranno memorizzati sia il corpo che gli header. Valori possibili se `request_method` è impostato su "HEAD": 1 - Header. Comportamento della proprietà: - *supportata* se `type` è impostato su "HTTP agent" - *sola lettura* per gli oggetti ereditati

Property	Type	Description
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 per il 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 di regola LLD. Valori possibili: 0 - *(predefinito)* normale; 1 - non supportato. Comportamento della proprietà: - *sola lettura*
status	integer	Stato del prototipo di regola LLD. Valori possibili: 0 - *(predefinito)* prototipo di regola LLD abilitato; 1 - prototipo di regola LLD disabilitato.
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 del prototipo di regola LLD del template padre. Comportamento della proprietà: - *sola lettura*
timeout	string	Timeout della richiesta di polling dei dati dell’item. Accetta secondi o un’unità di tempo con suffisso (ad esempio 30s, 1m). Accetta anche macro utente. Intervallo di valori possibile: 1-600s. Predefinito: ”” - usa le impostazioni globali/del proxy. 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

Property	Type	Description
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 to accept incoming data”
username	string	Nome utente per l’autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH agent”, ”TELNET agent”, oppure se `type` è impostato su ”JMX agent” e `password` è impostato - *supportata* 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 regola LLD importati a quelli già esistenti. Generato automaticamente, se non fornito. Comportamento della proprietà: - *supportata* se il prototipo di regola LLD appartiene a un template
verify_host	integer	Indica se verificare che il nome host per la connessione corrisponda a quello nel certificato del 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
verify_peer	integer	Indica se verificare che il certificato del 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

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*

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*
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*
path	string	Selettore del valore che verrà assegnato alla macro corrispondente. Comportamento della proprietà: - *obbligatorio*

Preelaborazione del prototipo di regola LLD

L’oggetto di preelaborazione del prototipo di regola LLD ha le seguenti proprietà.

Proprietà	Type	Descrizione
type	integer	Il tipo di opzione di preelaborazione. Valori possibili: 5 - Espressione regolare; 11 - XML XPath; 12 - JSONPath; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Controlla errore in JSON; 17 - Controlla errore in XML; 20 - Scarta 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. Property behavior: - *obbligatoria*
params	string	Parametri aggiuntivi utilizzati dall'opzione di preelaborazione. Più parametri sono separati dal carattere newline (\n). Property behavior: - *obbligatoria* se `type` è impostato su "Espressione regolare" (5), "XML XPath" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Controlla errore in JSON" (16), "Controlla errore in XML" (17), "Scarta 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 è impostato da Zabbix server; 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Property behavior: - *obbligatoria* se `type` è impostato su "Espressione regolare" (5), "XML XPath" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Controlla errore in JSON" (16), "Controlla errore 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 errori. Property behavior: - *obbligatoria* se `error_handler` è impostato su "Imposta valore personalizzato" o "Imposta messaggio di errore personalizzato"

I seguenti parametri e gestori errori sono supportati per ciascun tipo di preelaborazione.

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori errori supportati
5	Espressione regolare	pattern[1]	output[2]		0, 1, 2, 3
11	XML XPath	path[3]			0, 1, 2, 3
12	JSONPath	path[3]			0, 1, 2, 3
14	Corrisponde all'espressione regolare	pattern[1]			0, 1, 2, 3
15	Non corrisponde all'espressione regolare	pattern[1]			0, 1, 2, 3
16	Controlla errore in JSON	path[3]			0, 1, 2, 3
17	Controlla errore in XML	path[3]			0, 1, 2, 3
20	Scarta invariato con heartbeat	seconds[4, 5]			
21	JavaScript	script[2]			
23	Prometheus in JSON	pattern[5, 6]			0, 1, 2, 3
24	CSV in JSON	character[2]	character[2]	0,1	0, 1, 2, 3
25	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 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

Tipo di preelaborazione	Nome	Parametro 1	Parametro 2	Parametro 3	Gestori errori supportati
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 XML XPath
[4] intero positivo (con supporto dei suffissi temporali, ad es. 30s, 1m, 2h, 1d)
[5] macro utente
[6] pattern Prometheus che segue la sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ogni componente del pattern Prometheus (metrica, nome etichetta, valore etichetta e valore della metrica) può essere una macro utente.
[7] Supporta più record "Nome campo,Prefisso OID,Formato" 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*
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*

Proprietà	Type	Descrizione
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.
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 operativo di override del prototipo di regola LLD

Il periodo operativo di override del prototipo di regola LLD è un valore di intervallo di aggiornamento impostato per l'item individuato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
delay	string	Esegue l'override dell'intervallo di aggiornamento del prototipo di item. Accetta secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente o macro LLD. Se utilizzato, il valore deve essere una singola macro. Macro multiple o macro combinate con testo non sono supportate. 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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 di regola 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 del ruolo 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)`

Questo metodo consente di recuperare i prototipi di regola 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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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`.

Parametro	Type	Descrizione
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.
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 prototipi di regole LLD esistenti.

Si noti che l'aggiornamento dei prototipi già individuati è limitato.

> **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 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.
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.

Property	Type	Description
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.
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 prototipi di trigger.

> **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 del ruolo 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 prototipi di 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 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)`

Questo metodo consente di recuperare i prototipi di 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
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.
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.

Parametro	Tipo	Descrizione
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`.
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 prototipi di 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 del ruolo 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 ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
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
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: - *obbligatoria* se `proxy_groupid` non è 0
local_port	string	Numero di porta del proxy locale a cui connettersi. Le macro utente sono supportate. Predefinito: `10051`. Comportamento della proprietà: - *supportata* se `proxy_groupid` non è 0
operating_mode	integer	Tipo di proxy. Valori possibili: 0 - proxy attivo; 1 - proxy passivo. Comportamento della proprietà: - *obbligatoria* 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: - *obbligatoria* se la modalità operativa del proxy Zabbix è passiva

Proprietà	Tipo	Descrizione
port	string	Numero di porta a cui connettersi. Le macro utente sono supportate. Predefinito: 10051. Comportamento della proprietà: - *supportata* 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 verso l'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 bitmask; è accettata qualsiasi somma dei possibili valori bitmap (ad esempio, 6 per PSK e certificato).
tls_issuer	string	Autorità emittente del certificato.
tls_subject	string	Soggetto del certificato.
tls_psk_identity	string	Identità PSK; deve essere associata a una sola 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* - *obbligatoria* se `tls_connect` è impostato su "PSK", oppure `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 `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 - sovrascrivi i timeout.
timeout_zabbix_agent	string	Non impiegare più di `timeout_*` secondi per l'elaborazione. Accetta secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m). Accetta anche macro utente. Intervallo di valori possibili: 1-600s. Predefinito: "". Comportamento della proprietà: - *obbligatoria* se `custom_timeouts` è impostato su 1.
timeout_simple_check		
timeout_snmp_agent		

Proprietà	Tipo	Descrizione
timeout_external_check timeout_db_monitor timeout_http_agent timeout_ssh_agent timeout_telnet_agent timeout_script timeout_browser		
version	integer	Versione del proxy. Numero di versione Zabbix in 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, ecc. 0 - Versione del proxy sconosciuta. Comportamento della proprietà: - *sola lettura*
compatibility	integer	Versione del proxy confrontata con la 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 precedente release LTS del server oppure la versione principale del server è precedente a quella principale 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*

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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere User roles per ulteriori informazioni.

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. Per gli host deve essere definita solo la proprietà `hostid`.

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
}
```

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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 permette di recuperare i proxy in base ai parametri passati.

> **Note:**
> Questo metodo è disponibile per ogni tipo di utente. I permessi per chiamare questo metodo possono essere revocati nelle impostazioni dell'user role. Guarda User roles per maggiori informazioni.

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 gruppi di proxy specificati.
selectAssignedHosts	query	Restituisce una proprietà `assignedHosts` con gli host assegnati al proxy. Supporta `count`.

Parametro	Tipo	Descrizione
selectHosts	query	Restituisce una proprietà `hosts` con gli host monitorati dal proxy. Supporta `count`.
selectProxyGroup	query	Restituisce una proprietà `proxyGroup` con l'oggetto gruppo di proxy.
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `proxyid`, `name`, `operating_mode`.
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

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 i 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 ciascun proxy, tutte le altre proprietà sono facoltative. Verranno aggiornate solo le proprietà passate, 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. Per gli host deve essere definita solo la proprietà `hostid`.

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 l'oggetto di registrazione automatica in base ai parametri specificati.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin* . I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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 la registrazione automatica esistente.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Super admin* . I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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 regola di discovery definisce una regola di network discovery. Ha le seguenti proprietà.

Property	Type	Description
druleid	ID	ID della regola di discovery. Property behavior: - *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, fare riferimento alla sezione configurazione della network discovery. Property behavior: - *obbligatoria* per le operazioni di creazione
name	string	Nome della regola di discovery. Property behavior: - *obbligatoria* per le operazioni di creazione
delay	string	Intervallo di esecuzione della regola di discovery. Accetta secondi o un'unità di tempo con suffisso (ad esempio 30s, 1m, 2h, 1d), oppure 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 regola di discovery. Valori possibili: 0 - *(predefinito)* numero illimitato di controlli; 1 - un controllo; 2-999 - numero personalizzato di controlli.

Property	Type	Description
error	string	Testo dell'errore se si sono verificati problemi durante l'esecuzione della regola di discovery. Property behavior: - *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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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 User roles.

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)`

Questo metodo consente di recuperare le regole di discovery 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 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
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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 low-level discovery ha 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 secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente. Se utilizzato, il valore deve essere una singola macro. Macro multiple o macro combinate con testo non sono supportate. 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 del host a cui appartiene la regola LLD. Comportamento della proprietà: - *costante* - *obbligatoria* per le operazioni di creazione

Property	Type	Description
flags	integer	Origine della regola di discovery. Valori possibili: 1 - una regola di low-level discovery; 5 - una regola di low-level discovery convertita da prototipo. Comportamento della proprietà: - *sola lettura*
interfaceid	ID	ID dell'interfaccia 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"
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: - *obbligatoria* 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 - item dipendente; 19 - HTTP agent; 20 - SNMP agent; 21 - Script; 22 - Browser; 23 - Nidificato. Questo tipo è consentito solo se il genitore della regola è un template (che si presume sia collegato a un prototipo host durante la discovery) o un host rilevato (che ha una regola LLD alla sua radice). 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

Property	Type	Description
allow_traps	integer	Consente di popolare il valore in modo simile al trapper item. Valori possibili: 0 - *(predefinito)* Non consentire l'accettazione di dati in ingresso; 1 - Consentire l'accettazione di 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; 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 se si verificano problemi durante l'aggiornamento del valore della regola LLD. Comportamento della proprietà: - *sola lettura*
follow_redirects	integer	Segui 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 header 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
http_proxy	string	Stringa di connessione del 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

Property	Type	Description
jmx_endpoint	string	Stringa di connessione personalizzata di JMX agent. 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 secondi, un’unità di tempo con suffisso o una macro utente. Predefinito: `7d`.
lifetime_type	integer	Scenario per eliminare le risorse LLD perse. Valori possibili: 0 - *(predefinito)* Elimina dopo il raggiungimento della soglia di lifetime; 1 - Non eliminare; 2 - Elimina immediatamente.
enabled_lifetime	string	Periodo di tempo dopo il quale gli item non più rilevati verranno disabilitati. Accetta secondi, un’unità di tempo con suffisso o una macro utente. Predefinito: `0`.
enabled_lifetime_type	integer	Scenario per disabilitare le risorse LLD perse. Valori possibili: 0 - Disabilita dopo il raggiungimento della soglia di lifetime; 1 - Non disabilitare; 2 - *(predefinito)* Disabilita immediatamente.
master_itemid	ID	ID del master item. La regola di discovery non può essere un master item per 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)* Memorizza grezzo; 1 - Converti in JSON. Comportamento della proprietà: - *supportata* se `type` è impostato su ”HTTP agent” - *sola lettura* per gli oggetti ereditati
params	string	Parametri aggiuntivi a seconda del tipo di regola LLD: - script eseguito per le regole LLD SSH e Telnet; - query SQL per le regole LLD database monitor; - formula per le regole LLD calcolate; - lo 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”)

Property	Type	Description
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”, ”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 grezzi; 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 ”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
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”
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”
query_fields	array	Array di campi 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

Property	Type	Description
retrieve_mode	integer	Quale parte della risposta deve essere memorizzata. Valori possibili se `request_method` è impostato su ”GET”, ”POST” o ”PUT”: 0 - *(predefinito)* Corpo; 1 - Header; 2 - Verranno memorizzati sia il corpo che gli header. Valori possibili se `request_method` è impostato su ”HEAD”: 1 - Header. 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
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 per il 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 supportato. 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 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 della regola LLD del template padre. Comportamento della proprietà: - *sola lettura*

Property	Type	Description
timeout	string	Timeout della richiesta di polling dei dati dell'item. Accetta secondi o un'unità di tempo con suffisso (ad esempio, 30s, 1m). Accetta anche macro utente. Intervallo di valori possibili: 1-600s. Predefinito: "" - usa le impostazioni globali/del proxy. 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
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 to accept incoming data"
username	string	Nome utente per l'autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "SSH agent", "TELNET agent", oppure se `type` è impostato su "JMX agent" e `password` è impostato - *supportata* 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 fornito. Comportamento della proprietà: - *supportata* se la regola LLD appartiene a un template
verify_host	integer	Indica se verificare che il nome host per la connessione corrisponda a quello 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
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

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*

Proprietà	Type	Descrizione
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"
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 ha le seguenti proprietà.

Property	Type	Description
type	integer	Il tipo di opzione di preelaborazione. Valori possibili: 5 - Espressione regolare; 11 - XML XPath; 12 - JSONPath; 14 - Corrisponde all'espressione regolare; 15 - Non corrisponde all'espressione regolare; 16 - Verifica errore in JSON; 17 - Verifica errore in XML; 20 - Scarta invariato con heartbeat; 21 - JavaScript; 23 - Da Prometheus a JSON; 24 - Da CSV a JSON; 25 - Sostituisci; 27 - Da XML a JSON; 28 - Valore SNMP walk; 29 - Da SNMP walk a JSON; 30 - Valore SNMP get. Property behavior: - *obbligatorio*
params	string	Parametri aggiuntivi utilizzati dall'opzione di preelaborazione. Più parametri sono separati dal carattere newline (\n). Property behavior: - *obbligatorio* se `type` è impostato su "Espressione regolare" (5), "XML XPath" (11), "JSONPath" (12), "Corrisponde all'espressione regolare" (14), "Non corrisponde all'espressione regolare" (15), "Verifica errore in JSON" (16), "Verifica errore in XML" (17), "Scarta invariato con heartbeat" (20), "JavaScript" (21), "Da Prometheus a JSON" (23), "Da CSV a JSON" (24), "Sostituisci" (25), "Valore SNMP walk" (28), "Da SNMP walk a JSON" (29) o "Valore SNMP get" (30)

Property	Type	Description
error_handler	integer	Tipo di azione utilizzato in caso di errore del passaggio di preelaborazione. Valori possibili: 0 - Il messaggio di errore è impostato da Zabbix server; 1 - Scarta valore; 2 - Imposta valore personalizzato; 3 - Imposta messaggio di errore personalizzato. Property behavior: - *obbligatorio* se `type` è impostato su ”Espressione regolare” (5), ”XML XPath” (11), ”JSONPath” (12), ”Corrisponde all’espressione regolare” (14), ”Non corrisponde all’espressione regolare” (15), ”Verifica errore in JSON” (16), ”Verifica errore in XML” (17), ”Da Prometheus a JSON” (23), ”Da CSV a JSON” (24), ”Da XML a JSON” (27), ”Valore SNMP walk” (28), ”Da SNMP walk a JSON” (29) o ”Valore SNMP get” (30)
error_handler_params	string	Parametri del gestore degli errori. Property behavior: - *obbligatorio* se `error_handler` è impostato su ”Imposta valore personalizzato” o ”Imposta 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	XML XPath	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 errore in JSON	path[3]			0, 1, 2, 3
17	Verifica errore in XML	path[3]			0, 1, 2, 3
20	Scarta invariato con heartbeat	secondi[4, 5]			
21	JavaScript	script[2]			

Preprocessing type	Name	Parameter 1	Parameter 2	Parameter 3	Supported error handlers
23	Da Prometheus a JSON	pattern[5, 6]			0, 1, 2, 3
24	Da CSV a JSON	carattere[2]	carattere[2]	0,1	0, 1, 2, 3
25	Sostituisci	stringa di ricerca[2]	sostituzione[2]		
27	Da XML a 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	Da SNMP walk a JSON[7]	Nome 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 XML XPath
[4] intero positivo (con supporto dei suffissi temporali, ad esempio 30s, 1m, 2h, 1d)
[5] macro utente
[6] pattern Prometheus che segue la sintassi: `<metric name>{<label name>="<label value>", ...} == <value>`. Ciascun componente del pattern Prometheus (metrica, nome etichetta, valore etichetta e valore della metrica) può essere una macro utente.
[7] Supporta più record "Nome campo,Prefisso OID,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*

Property	Type	Description
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*
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.

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 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 operativo di override della regola LLD

Il periodo operativo di override della regola LLD è un valore di intervallo di aggiornamento impostato per l'item rilevato. Ha le seguenti proprietà:

Property	Type	Description
delay	string	Esegue l'override dell'intervallo di aggiornamento del prototipo di item. Accetta secondi oppure un'unità di tempo con suffisso (ad esempio, 30s, 1m, 2h, 1d) e, facoltativamente, uno o più intervalli personalizzati, tutti separati da punto e virgola. Gli intervalli personalizzati possono essere una combinazione di intervalli flessibili e pianificati. Accetta macro utente o macro LLD. Se utilizzato, il valore deve essere una singola macro. Macro multiple o macro combinate con testo non sono supportate. 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 e il valore del tag impostati sull'oggetto individuato. Ha le seguenti proprietà:

Proprietà	Type	Descrizione
tag	string	Nuovo nome del tag. Comportamento della proprietà: - *obbligatorio*
value	string	Nuovo valore del tag.

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 chiamare 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 del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo 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 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
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`.

Parametro	Tipo	Descrizione
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`.
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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

Parametro	Type	Descrizione
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

Aggiungere un filtro in modo che il contenuto della macro *{#FSTYPE}* corrisponda all'espressione regolare *@File systems for discovery*.

Richiesta:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Risposta:

```
{
    "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 ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
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
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 di ripetizione del periodo. Valori possibili: 0 - *(predefinito)* giornaliera; 1 - settimanale; 2 - mensile; 3 - annuale.

Proprietà	Tipo	Descrizione
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 bitmask; è accettata 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 del giorno (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 del giorno (23:59:59)).
subject	string	Oggetto del messaggio del report.
message	string	Testo del messaggio del report.
status	integer	Indica se il report è abilitato o disabilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
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*

Proprietà	Tipo	Descrizione
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*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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

Parametro	Type	Descrizione
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo 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 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 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 report pianificati esistenti.

Note:
Questo metodo è disponibile solo per il tipo di utente *Admin* e *Super admin*. Le autorizzazioni per chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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
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

Proprietà	Tipo	Descrizione
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 del ruolo

L'oggetto delle regole del ruolo ha le seguenti proprietà:

Proprietà	Type	Descrizione
ui	array	Array di oggetti elemento UI.
ui.default_access	integer	Indica se l'accesso ai nuovi elementi UI è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* Abilitato.
services.read.mode	integer	Accesso in sola lettura ai servizi. Valori possibili: 0 - Accesso in sola lettura ai servizi specificati dalla proprietà `services.read.list` o corrispondenti alla proprietà `services.read.tag`; 1 - *(predefinito)* Accesso in sola lettura a tutti i servizi.
services.read.list	array	Array di oggetti Service. Ai servizi specificati, inclusi i servizi figli, verrà concesso l'accesso in sola lettura per il ruolo utente. L'accesso in sola lettura non sovrascriverà l'accesso in lettura-scrittura ai servizi. Comportamento della proprietà: - *supportato* se `services.read.mode` è impostato su "0"
services.read.tag	object	Array di oggetti tag del servizio. Ai servizi corrispondenti al tag, inclusi i servizi figli, verrà concesso l'accesso in sola lettura per il ruolo utente. L'accesso in sola lettura non sovrascriverà l'accesso in lettura-scrittura ai servizi. Comportamento della proprietà: - *supportato* 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 dalla proprietà `services.write.list` o corrispondenti alla proprietà `services.write.tag`; 1 - Accesso in lettura-scrittura a tutti i servizi.

Proprietà	Type	Descrizione
services.write.list	array	Array di oggetti Service. Ai servizi specificati, inclusi i servizi figli, verrà concesso l'accesso in lettura-scrittura per il ruolo utente. L'accesso in lettura-scrittura sovrascriverà l'accesso in sola lettura ai servizi. Comportamento della proprietà: - *supportato* se `services.write.mode` è impostato su "0"
services.write.tag	object	Array di oggetti tag del servizio. Ai servizi corrispondenti al tag, inclusi i servizi figli, verrà concesso l'accesso in lettura-scrittura per il ruolo utente. L'accesso in lettura-scrittura sovrascriverà l'accesso in sola lettura ai servizi. Comportamento della proprietà: - *supportato* 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 - Disabilitato; 1 - *(predefinito)* Abilitato.
api.mode	integer	Modalità di gestione dei metodi API elencati nella proprietà `api`. Valori possibili: 0 - *(predefinito)* Elenco di negazione; 1 - Elenco di autorizzazione.
api	array	Array di metodi API.
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.

Elemento dell'interfaccia utente

L'oggetto elemento dell'interfaccia utente ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
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* → *Dati più recenti*; `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 di 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* → *Discovery*; `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* → *Discovery*; `configuration.trigger_actions` - *Avvisi* → *Azioni* → *Azioni trigger*; `configuration.service_actions` - *Avvisi* → *Azioni* → *Azioni servizio*; `configuration.discovery_actions` - *Avvisi* → *Azioni* → *Azioni discovery*; `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* → *Log di audit*; `reports.action_log` - *Report* → *Log delle azioni*; `configuration.event_correlation` - *Raccolta dati* → *Correlazione 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.general` - *Amministrazione* → *Generale*; `administration.audit_log` - *Amministrazione* → *Log di audit*; `administration.housekeeping` - *Amministrazione* → *Housekeeping*; `administration.proxy_groups` - *Amministrazione* → *Gruppi proxy*; `administration.proxies` - *Amministrazione* → *Proxy*; `administration.macros` - *Amministrazione* → *Macro*; `administration.queue` - *Amministrazione* → *Coda*. Comportamento della proprietà: - *obbligatoria*

Proprietà	Tipo	Descrizione
status	integer	Indica se l'accesso all'elemento dell'interfaccia utente è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* 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 ha le seguenti proprietà:

Proprietà	Tipo	Descrizione
name	string	Nome dell'azione. Valori possibili se `type` dell'oggetto Role è impostato su "User", "Admin" o "Super admin": `edit_dashboards` - Crea e modifica dashboard; `edit_maps` - Crea e modifica mappe; `add_problem_comments` - Aggiunge commenti ai problemi; `change_severity` - Modifica la gravità del problema; `acknowledge_problems` - Riconosce i problemi; `suppress_problems` - Sopprime i problemi; `close_problems` - Chiude i problemi; `execute_scripts` - Esegue script; `manage_api_tokens` - Gestisce i token API; `change_problem_ranking` - Modifica la classificazione del problema da causa a sintomo e viceversa; `edit_own_media` - Consente di creare/modificare i propri media. Valori possibili se `type` dell'oggetto Role è impostato su "Admin" o "Super admin": `edit_maintenance` - Crea e modifica periodi di manutenzione; `manage_scheduled_reports` - Gestisce i report pianificati, `manage_sla` - Gestisce gli SLA. Valori possibili se `type` dell'oggetto Role è impostato su "User" o "Admin": `invoke_execute_now` - consente di eseguire i controlli degli item agli utenti che hanno solo permessi di lettura sul host. Valori possibili se `type` dell'oggetto Role è impostato su "Super admin": `edit_user_media` - Consente di creare/modificare i media per gli utenti. Comportamento della proprietà: - *obbligatoria*
status	integer	Indica se l'accesso per eseguire l'azione è abilitato. Valori possibili: 0 - Disabilitato; 1 - *(predefinito)* 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. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo metodo consente di recuperare i ruoli 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 i seguenti parametri.

Parametro	Tipo	Descrizione
roleids	ID/array	Restituisce solo i ruoli con gli ID specificati.
selectRules	query	Restituisce una proprietà `rules` con le regole del ruolo.

Parametro	Tipo	Descrizione
selectUsers	query	Restituisce una proprietà `users` con gli utenti a cui è assegnato il ruolo. Vedere `user.get` per le restrizioni 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",
        "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*. 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 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 ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
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 che verrà utilizzata dallo scenario web. Predefinito: Zabbix
authentication	integer	Metodo di autenticazione che verrà utilizzato dallo scenario web. Valori possibili: 0 - *(predefinito)* nessuna; 1 - autenticazione HTTP di base; 2 - autenticazione NTLM; 3 - autenticazione Kerberos; 4 - autenticazione Digest.
delay	string	Intervallo di esecuzione dello scenario web. Accetta secondi o un'unità di tempo con suffisso (ad es. 30s, 1m, 2h, 1d), oppure una macro utente. Predefinito: 1m.
headers	array	Header HTTP che verranno inviati durante l'esecuzione di una richiesta.
http_password	string	Password utilizzata per l'autenticazione HTTP di base, NTLM, Kerberos o Digest.

Proprietà	Tipo	Descrizione
http_proxy	string	proxy che verrà utilizzato dallo scenario web, specificato come *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 fallire. 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.
templateid	ID	ID dello scenario web del template padre. 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)* salta la verifica dell'host; 1 - verifica host.
verify_peer	integer	Indica se verificare che il certificato dell'host sia autentico. Valori possibili: 0 - *(predefinito)* salta la verifica del peer; 1 - verifica 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: - *supportata* se lo scenario web appartiene a un template

Passaggio dello scenario

L'oggetto passaggio dello scenario definisce uno specifico controllo di uno scenario web. Ha le seguenti proprietà.

Proprietà	Type	Descrizione
name	string	Nome del passaggio dello scenario. Comportamento della proprietà: - *obbligatoria*
no	integer	Numero di sequenza del passaggio in uno scenario web. Comportamento della proprietà: - *obbligatoria*
url	string	URL da controllare. Comportamento della proprietà: - *obbligatoria*

Proprietà	Type	Descrizione
follow_redirects	integer	Indica se seguire i reindirizzamenti HTTP. Valori possibili: 0 - non seguire i reindirizzamenti; 1 - *(predefinito)* seguire i reindirizzamenti.
headers	array	Header HTTP che verranno inviati durante l'esecuzione di una richiesta. Gli header del passaggio dello scenario sovrascriveranno gli header specificati per lo scenario web.
posts	string/array	Variabili HTTP POST come stringa (dati POST grezzi) 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 header; 2 - header e corpo.
status_codes	string	Intervalli dei codici di stato HTTP richiesti, separati da virgole.
timeout	string	Timeout della richiesta in secondi. Accetta secondi, un'unità di tempo con suffisso o una macro utente. Predefinito: 15s. Massimo: 1h. Minimo: 1s.
variables	array	Variabili del passaggio dello scenario.
query_fields	array	Campi di 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*.

Proprietà	Tipo	Descrizione
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 scenari web.

Note:
La creazione di uno scenario web comporterà automaticamente la creazione di un insieme di item di monitoraggio 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 del ruolo 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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)`

Questo metodo consente di recuperare gli scenari web in base ai parametri forniti.

Note:
Questo metodo è disponibile per utenti di qualsiasi tipo. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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.

Parametro	Type	Descrizione
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`.
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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 ha le seguenti proprietà.

Proprietà	Type	Descrizione
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

Proprietà	Type	Descrizione
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
execute_on	integer	Dove eseguire lo script. Valori possibili: 0 - esegui su Zabbix agent; 1 - esegui su Zabbix server. È *supportato* solo se l’esecuzione degli script globali è abilitata su Zabbix server; 2 - *(predefinito)* esegui su Zabbix server o proxy. Comportamento della proprietà: - *supportato* se `type` è impostato su ”Script”
menu_path	string	Cartelle separate da slash che formano una navigazione simile a un menu nel frontend quando si fa clic su host o evento. Comportamento della proprietà: - *supportato* se `scope` è impostato su ”manual host action” o ”manual event action”
authtype	integer	Metodo di autenticazione usato per il tipo di script SSH. Valori possibili: 0 - password; 1 - chiave pubblica. Comportamento della proprietà: - *supportato* se `type` è impostato su ”SSH”
username	string	Nome utente usato per l’autenticazione. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”SSH” o ”TELNET”

Proprietà	Type	Descrizione
password	string	Password usata per gli script SSH con autenticazione tramite password e per gli script TELNET. Comportamento della proprietà: - *supportato* se `type` è impostato su ”SSH” e `authtype` è impostato su ”password”, oppure `type` è impostato su ”TELNET”
publickey	string	Nome del file della chiave pubblica usato 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 usato 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 usato per gli script SSH e TELNET. Comportamento della proprietà: - *supportato* se `type` è impostato su ”SSH” o ”TELNET”
groupid	ID	ID del gruppo host su cui lo script può essere eseguito. Se impostato su ”0”, lo script sarà disponibile per tutti i gruppi host. Predefinito: 0.
usrgrpid	ID	ID del gruppo utenti a cui sarà consentito eseguire lo script. Se impostato su ”0”, lo script sarà disponibile per tutti i gruppi utenti. Predefinito: 0. Comportamento della proprietà: - *supportato* se `scope` è impostato su ”manual host action” o ”manual event action”
host_access	integer	Permessi host necessari per eseguire lo script. Valori possibili: 2 - *(predefinito)* lettura; 3 - scrittura. Comportamento della proprietà: - *supportato* se `scope` è impostato su ”manual host action” o ”manual event action”
confirmation	string	Testo del popup di conferma. Il popup apparirà quando si tenta di eseguire lo script dal frontend di Zabbix. Comportamento della proprietà: - *supportato* se `scope` è impostato su ”manual host action” o ”manual event action”
timeout	string	Timeout di esecuzione dello script webhook in secondi. I suffissi temporali sono supportati (ad esempio, 30s, 1m). Valori possibili: 1-60s. Predefinito: 30s. Comportamento della proprietà: - *obbligatoria* se `type` è impostato su ”Webhook”

Proprietà	Type	Descrizione
parameters	array	Array dei parametri di input del webhook. Comportamento della proprietà: - *supportato* 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 URL in una nuova finestra. Valori possibili: 0 - No; 1 - *(predefinito)* Sì. Comportamento della proprietà: - *supportato* se `type` è impostato su ”URL”
manualinput	integer	Indica se lo script accetta input fornito dall’utente. Valori possibili: 0 - *(predefinito)* Disabilitato; 1 - Abilitato; Comportamento della proprietà: - *supportato* se `scope` è impostato su ”manual host action” o ”manual event action”
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 usato per convalidare l’input fornito dall’utente. La stringa consiste in un’espressione regolare oppure in 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 dei possibili valori di input. Comportamento della proprietà: - *supportato* se `manualinput` è impostato su ”Enabled”
manualinput_default_value	string	Valore predefinito per il riempimento automatico dell’input utente. Comportamento della proprietà: - *supportato* 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 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo metodo consente di recuperare gli script 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 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.

Parametro	Tipo	Descrizione
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 del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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 del ruolo 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 script 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 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`.

Servizio

L'oggetto servizio ha le seguenti proprietà.

Proprietà	Tipo	Descrizione
serviceid	ID	ID del servizio. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
algorithm	integer	Regola di calcolo dello stato. Applicabile solo se esistono servizi figli. Valori possibili: 0 - imposta lo stato su OK; 1 - il più critico se tutti i figli hanno problemi; 2 - il più critico tra i servizi figli. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
name	string	Nome del servizio. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
sortorder	integer	Posizione del servizio utilizzata per l'ordinamento. Valori possibili: 0-999. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
weight	integer	Peso del servizio. Valori possibili: 0-1000000. Predefinito: 0.

Proprietà	Tipo	Descrizione
propagation_rule	integer	Regola di propagazione dello stato. Valori possibili: 0 - *(predefinito)* propaga lo stato del servizio così com'è, senza alcuna modifica; 1 - aumenta lo stato propagato di un determinato `propagation_value` (da 1 a 5 livelli di gravità); 2 - diminuisce lo stato propagato di un determinato `propagation_value` (da 1 a 5 livelli di gravità); 3 - ignora questo servizio - lo stato non viene propagato al servizio padre in alcun modo; 4 - imposta uno stato fisso del servizio usando un determinato `propagation_value`. Comportamento della proprietà: - *obbligatoria* se `propagation_value` è impostato
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. Comportamento della proprietà: - *obbligatoria* se `propagation_rule` è impostato
status	integer	Indica se il servizio è nello stato OK o in stato di problema. Se il servizio è in stato di problema, `status` è uguale a uno dei seguenti valori: - la gravità del problema più critico; - lo stato più alto di un servizio figlio in stato di problema. Se il servizio è nello stato OK, `status` è uguale a: -1. Comportamento della proprietà: - *sola lettura*
description	string	Descrizione del servizio.
uuid	string	Identificatore univoco universale, utilizzato per collegare i servizi importati a quelli già esistenti. Generato automaticamente, se non fornito.
created_at	integer	Timestamp Unix della creazione del servizio.

Proprietà	Tipo	Descrizione
readonly	integer	Accesso al servizio. Valori possibili: 0 - Lettura-scrittura; 1 - Sola lettura. Comportamento della proprietà: - *sola lettura*

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*

Proprietà	Type	Descrizione
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 di servizio

> **Note:**
> Gli allarmi di servizio non possono essere creati, aggiornati o eliminati direttamente tramite la API di Zabbix.

Gli oggetti allarme di servizio rappresentano un cambiamento di stato di un servizio. Hanno le seguenti proprietà.

Proprietà	Tipo	Descrizione
clock	timestamp	Ora in cui si è verificato il cambiamento di 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.

Proprietà	Type	Descrizione
tag	string	Nome del tag del problema. Comportamento della proprietà: - *obbligatorio*
operator	integer	Operatore della condizione di mappatura. Valori possibili: 0 - *(predefinito)* 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 del ruolo utente. Vedere User roles 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`.
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 del ruolo utente. Vedere User roles 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)`

Questo metodo consente di recuperare i servizi 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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.
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`.

Parametro	Type	Descrizione
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.
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 servizi esistenti.

> **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

`(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`.

Housekeeping

L'oggetto impostazioni ha le seguenti proprietà.

Property	Type	Description
hk_events_mode	integer	Abilita l'housekeeping interno per eventi e avvisi. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_events_trigger	string	Periodo di conservazione dei dati dei trigger. Accetta secondi e unità di tempo con suffisso. Predefinito: 365d.
hk_events_service	string	Periodo di conservazione dei dati dei servizi. Accetta secondi e unità di tempo con suffisso. Predefinito: 1d.
hk_events_internal	string	Periodo di conservazione dei dati interni. Accetta secondi e unità di tempo con suffisso. Predefinito: 1d.
hk_events_discovery	string	Periodo di conservazione dei dati di network discovery. Accetta secondi e unità di tempo con suffisso. Predefinito: 1d.
hk_events_autoreg	string	Periodo di conservazione dei dati di autoregistrazione. Accetta secondi e unità di tempo con suffisso. Predefinito: 1d.
hk_services_mode	integer	Abilita l'housekeeping interno per i servizi. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_services	string	Periodo di conservazione dei dati dei servizi. Accetta secondi e unità di tempo con suffisso. Predefinito: 365d.

Property	Type	Description
hk_audit_mode	integer	Abilita l'housekeeping interno per l'audit. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_audit	string	Periodo di conservazione dei dati di audit. Accetta secondi e unità di tempo con suffisso. Predefinito: 31d.
hk_sessions_mode	integer	Abilita l'housekeeping interno per le sessioni. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_sessions	string	Periodo di conservazione dei dati delle sessioni. Accetta secondi e unità di tempo con suffisso. Predefinito: 31d.
hk_history_mode	integer	Abilita l'housekeeping interno per lo storico. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_history_global	integer	Sovrascrive il periodo dello storico dell'item. Valori possibili: 0 - Non sovrascrivere; 1 - *(predefinito)* Sovrascrivi.
hk_history	string	Periodo di conservazione dei dati dello storico. Accetta secondi e unità di tempo con suffisso. Predefinito: 31d.
hk_trends_mode	integer	Abilita l'housekeeping interno per i trend. Valori possibili: 0 - Disabilita; 1 - *(predefinito)* Abilita.
hk_trends_global	integer	Sovrascrive il periodo dei trend dell'item. Valori possibili: 0 - Non sovrascrivere; 1 - *(predefinito)* Sovrascrivi.
hk_trends	string	Periodo di conservazione dei dati dei trend. Accetta secondi e unità di tempo con suffisso. 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 l'housekeeping e l'eliminazione degli item. Property behavior: - *read-only*
compression_availability	integer	Indica se la compressione dei dati è supportata dal database (o dalla sua estensione). Valori possibili: 0 - Non disponibile; 1 - Disponibile. Property behavior: - *read-only*

Property	Type	Description
compression_status	integer	Abilita la compressione TimescaleDB per storico e trend. Valori possibili: 0 - *(predefinito)* Disattivata; 1 - Attivata.
compress_older	string	Comprime i record di storico e trend più vecchi del periodo specificato. Accetta secondi e unità di tempo con suffisso. Predefinito: 7d.

housekeeping.get

Descrizione

`object housekeeping.get(object parameters)`

Questo metodo consente di recuperare un oggetto housekeeping 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

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)`

Questo metodo consente di recuperare i servizi rilevati 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 del ruolo 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.
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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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)`

Questo 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 del ruolo utente. Vedere User roles 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 del Service Level Indicator (SLI) per un Service Level Agreement (SLA).

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 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": 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 voci SLA 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 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*.

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.

Proprietà	Tipo	Descrizione
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.
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 ripulire lo storico degli elementi.

> **Note:**
> Questo metodo è disponibile solo per gli utenti di tipo *Admin* e *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare Ruoli utente per ulteriori informazioni.

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 dello storico in base ai parametri specificati.

Attention:
Questo metodo può restituire i dati dello storico di un'entità eliminata se questi dati non sono ancora stati rimossi dal processo di housekeeping.

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)` 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 - numerico in virgola mobile; 1 - carattere; 2 - log; 3 - *(predefinito)* numerico senza segno; 4 - testo; 5 - binario; 6 - JSON.
hostids	ID/array	Restituisce solo la history dei host specificati.
itemids	ID/array	Restituisce solo la history degli item specificati.
maxValueSize	integer	Numero massimo di byte da restituire nella proprietà `value`. Intervallo di 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 al momento specificato.
time_till	timestamp	Restituisce solo i valori ricevuti prima o al momento specificato.

Parametro	Tipo	Descrizione
sortfield	string/array	Ordina il risultato in base alle proprietà specificate. Valori possibili: `itemid`, `clock`, `ns`.
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 "%…%"`. Non supporta la ricerca per `value` se `history` è impostato su 6 (JSON).
countOutput	boolean	Questi parametri sono descritti nel commento di riferimento.
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. I permessi per chiamare il metodo possono essere revocati 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 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
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 qualsiasi host o template a cui è collegato. Per mantenere queste entità, scollegare prima il template dagli host o template pertinenti utilizzando 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 del ruolo utente. Per ulteriori informazioni, vedere 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)`

Questo 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 User roles.

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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo 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 oggetti correlati da più template.

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 del ruolo 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 template 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 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 supporto

L'oggetto tipo di supporto ha le seguenti proprietà.

Proprietà	Type	Descrizione
mediatypeid	ID	ID del tipo di supporto. Comportamento della proprietà: - *sola lettura* - *obbligatoria* per le operazioni di aggiornamento
name	string	Nome del tipo di supporto. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
type	integer	Trasporto utilizzato dal tipo di supporto. Valori possibili: 0 - Email; 1 - Script; 2 - SMS; 4 - Webhook. Comportamento della proprietà: - *obbligatoria* per le operazioni di creazione
exec_path	string	Nome del file di 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"

Proprietà	Type	Descrizione
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)* SMTP generico; 1 - Gmail; 2 - Relay Gmail; 3 - Office365; 4 - Relay Office365.
smtp_email	string	Indirizzo email da cui 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. Predefinita: 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)* Nessuno; 1 - STARTTLS; 2 - SSL/TLS. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Email”
smtp_verify_host	integer	Verifica host SSL per SMTP. Valori possibili: 0 - *(predefinito)* No; 1 - Sì. Comportamento della proprietà: - *supportata* se `smtp_security` è impostato su ”STARTTLS” o ”SSL/TLS”
smtp_verify_peer	integer	Verifica peer SSL per SMTP. Valori possibili: 0 - *(predefinito)* No; 1 - Sì. Comportamento della proprietà: - *supportata* se `smtp_security` è impostato su ”STARTTLS” o ”SSL/TLS”

Proprietà	Type	Descrizione
smtp_authentication	integer	Metodo di autenticazione SMTP da utilizzare. Valori possibili: 0 - *(predefinito)* Nessuno; 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 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 a un utente 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 di 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 di aggiornamento. Questo 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"
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"

Proprietà	Type	Descrizione
access_token_updated	timestamp	Timestamp dell'ultima modifica di `access_token` eseguita dal server durante l'aggiornamento con `refresh_token` o tramite API in caso di modifiche ai 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 tramite API in caso di modifiche ai 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 supporto 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 per inviare un avviso. Valori possibili: 1-100. Predefinito: 3.
attempt_interval	string	Intervallo tra i tentativi di ritrasmissione. Accetta secondi e unità di tempo con suffisso. Valori possibili: 0-1h. Predefinito: 10s.
message_format	integer	Formato del messaggio. Valori possibili: 0 - Testo semplice; 1 - *(predefinito)* HTML.
script	text	Corpo dello script webhook (JavaScript). Comportamento della proprietà: - *supportata* se `type` è impostato su "Email" Comportamento della proprietà: - *obbligatoria* se `type` è impostato su "Webhook"

Proprietà	Type	Descrizione
timeout	string	Timeout dello script webhook. Accetta secondi e unità di tempo con suffisso. Valori possibili: 1-60s. Predefinito: 30s. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook”
process_tags	integer	Elabora i valori delle proprietà JSON nella risposta dello script Webhook come tag. Questi tag vengono aggiunti a eventuali tag problema esistenti. Valori possibili: 0 - *(predefinito)* Ignora la risposta dello script webhook; 1 - Elabora la risposta dello script webhook come tag. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook”
show_event_menu	integer	Include una voce nel menu evento 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 del menu evento né la proprietà `urls`; 1 - Includere la voce del menu evento e la proprietà `urls`. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook”
event_menu_url	string	URL utilizzato nella voce del menu evento 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”
event_menu_name	string	Nome utilizzato per la voce del menu evento 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”
parameters	array	Parametri di Webhook o dello script. Comportamento della proprietà: - *supportata* se `type` è impostato su ”Webhook” o ”Script”
description	text	Descrizione del tipo di supporto.

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.
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 del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

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 del ruolo utente. Per ulteriori informazioni, vedere Ruoli utente.

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)`

Questo metodo consente di recuperare i tipi di supporto 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.

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.
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`.

Parametro	Type	Descrizione
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 tipi di supporto 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 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.

Proprietà	Tipo	Descrizione
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 richiesta 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 *Manage API tokens* 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:
Per il ruolo utente è necessaria l'autorizzazione *Manage API tokens* per 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)`

Questo 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	

Parametro	Type	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 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 token esistenti.

Note:
Per il ruolo utente è necessaria l'autorizzazione *Manage API tokens* 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*.

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.
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)`

Questo 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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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
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*

Property	Type	Description
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.
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*.
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 del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 User roles.

Parametri

`(object)` Parametri che definiscono l'output desiderato.

Il metodo supporta i seguenti parametri.

Parametro	Type	Descrizione
triggerids	ID/array	Restituisce solo i trigger con gli ID specificati.
groupids	ID/array	Restituisce solo i trigger che appartengono a host o template dei gruppi host o gruppi template specificati.

Parametro	Type	Descrizione
templateids	ID/array	Restituisce solo i trigger che appartengono ai template specificati.
hostids	ID/array	Restituisce solo i trigger che appartengono agli host specificati.
itemids	ID/array	Restituisce solo i trigger che contengono gli item specificati.
functions	string/array	Restituisce solo i trigger che utilizzano le funzioni specificate. Fare riferimento alla pagina delle funzioni supportate per un elenco delle funzioni supportate.
group	string	Restituisce solo i trigger che appartengono a host o template del gruppo host o gruppo template con il nome specificato.
host	string	Restituisce solo i trigger che appartengono 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 che appartengono ai template.
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 che appartengono a host monitorati e contengono solo item abilitati.
active	flag	Restituisce solo i trigger abilitati che appartengono a host monitorati.
maintenance	boolean	Se impostato su `true`, restituisce solo i trigger abilitati che appartengono a host in manutenzione.
withUnacknowledgedEvents	flag	Restituisce solo i trigger che hanno 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	Salta i trigger in stato di problema che dipendono da altri trigger. Si noti che gli altri trigger vengono ignorati se disabilitati, se hanno item disabilitati o host degli item disabilitati.
lastChangeSince	timestamp	Restituisce solo i trigger che hanno cambiato stato dopo il momento specificato.
lastChangeTill	timestamp	Restituisce solo i trigger che hanno cambiato stato prima del momento specificato.
only_true	flag	Restituisce solo i trigger che recentemente sono stati in stato di problema.
min_severity	integer	Restituisce solo i trigger con severità 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 di operator: 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 funzioni e macro nell'espressione del trigger.
selectHostGroups	query	Restituisce i gruppi 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 nel trigger nella proprietà `items`.

Parametro	Type	Descrizione
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`.
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 da cui è stato rilevato. Ha le seguenti proprietà: `parent_triggerid` - `(ID)` ID del prototipo di trigger da cui è stato creato il trigger; `status` - `(int)` stato della 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 in template/host/template collegati. Valori possibili: `true` - template/host/template collegati devono avere anch'essi i tag specificati; `false` - *(predefinito)* i tag di template/host/template collegati vengono ignorati.
selectInheritedTags	query	Restituisce una proprietà `inheritedTags` con i tag ereditati da template, template collegati e host, e dagli item referenziati nelle espressioni del trigger o di recovery.
selectTemplateGroups	query	Restituisce i gruppi 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 oppure un array di valori da confrontare. Non supporta proprietà del tipo di dato `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 sotto-selezioni. Si applica alle seguenti sotto-selezioni: `selectHosts` - i risultati saranno ordinati per `host`.

Parametro	Type	Descrizione
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 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	
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 da cui 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 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 del ruolo 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.

Proprietà	Type	Descrizione
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 directory utenti è impostato su "1") e lo stato di autenticazione di tutto il provisioning LDAP o SAML è abilitato (`ldap_jit_status` dell'oggetto autenticazione è impostato su "Enabled for configured LDAP IdPs" oppure `saml_jit_status` dell'oggetto autenticazione 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. Si noti che 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 utenti configurate in Zabbix.
attempt_clock	timestamp	Ora dell'ultimo tentativo di accesso non riuscito. Comportamento della proprietà: - *sola lettura*

Proprietà	Type	Descrizione
attempt_failed	integer	Numero recente di tentativi di accesso non riusciti. Comportamento della proprietà: - *sola lettura*
attempt_ip	string	Indirizzo IP da cui è provenuto l'ultimo tentativo di accesso non riuscito. Comportamento della proprietà: - *sola lettura*
autologin	integer	Indica se abilitare l'accesso automatico. Valori possibili: 0 - accesso automatico disabilitato *(predefinito)*; 1 - accesso automatico abilitato.
autologout	string	Durata della sessione utente. Accetta secondi e unità di tempo con suffisso. Se impostato su 0s, la sessione non scadrà mai. Predefinito: 15m.
lang	string	Codice della lingua dell'utente, ad esempio `en_US`. Predefinito: `default` - valore predefinito 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	Intervallo di aggiornamento automatico. Accetta secondi o unità di tempo con suffisso (ad esempio 30s, 90s, 1m, 1h). Predefinito: 30s.
rows_per_page	integer	Numero di righe di oggetti da mostrare per pagina. Predefinito: 50.
surname	string	Cognome dell'utente.
theme	string	Tema dell'utente. Valori possibili: `default` - *(predefinito)* valore predefinito del sistema; `blue-theme` - Blu; `dark-theme` - Scuro.
ts_provisioned	timestamp	Ora in cui è stata eseguita l'ultima operazione di provisioning. Comportamento della proprietà: - *sola lettura* - *supportata* per utenti di tipo *Super admin*
url	string	URL della pagina a cui reindirizzare l'utente dopo l'accesso.

Proprietà	Type	Descrizione
userdirectoryid	ID	ID della directory utenti a cui l'utente è collegato. Utilizzato per il provisioning (creazione o aggiornamento), nonché per consentire l'accesso a un utente collegato a una directory utenti. Per le operazioni di accesso, il valore di questa proprietà avrà priorità rispetto alla proprietà `userdirectoryid` dei gruppi utenti a cui l'utente appartiene. Predefinito: 0. Comportamento della proprietà: - *sola lettura* - *supportata* per utenti di tipo *Super admin*
timezone	string	Fuso orario dell'utente, ad esempio `Europe/London`, `UTC`. Predefinito: `default` - valore predefinito del sistema. Per l'elenco completo dei fusi orari supportati, fare riferimento alla documentazione PHP.

Supporto di notifica

L'oggetto media dispone delle seguenti proprietà.

Property	Type	Description
mediaid	ID	ID del supporto di notifica dell'utente. Comportamento della proprietà: - *sola lettura*
mediatypeid	ID	ID del tipo di supporto di notifica utilizzato dal supporto di notifica dell'utente. Comportamento della proprietà: - *obbligatoria*
sendto	string/array	Indirizzo, nome utente o altro identificatore del destinatario. Se `type` del tipo di supporto di notifica è impostato su "Email", i valori sono rappresentati come array. Per altri tipi di tipi di supporto di notifica, il valore è rappresentato come stringa. Comportamento della proprietà: - *obbligatoria*
active	integer	Indica se il supporto di notifica è abilitato. Valori possibili: 0 - *(predefinito)* abilitato; 1 - disabilitato.

Property	Type	Description
severity	integer	Gravità dei trigger per cui inviare notifiche. Valori bitmap possibili: 1 - Non classificato; 2 - Informazione; 4 - Avviso; 8 - Medio; 16 - Alto; 32 - Disastro. Questo è un campo bitmask; è accettata qualsiasi somma dei possibili valori bitmap (ad esempio, 48 per Medio, Alto e Disastro). Predefinito: 63.
period	string	Periodo di tempo in cui è possibile inviare le notifiche, come periodo di tempo o 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 supporto di notifica della directory utente per i supporti di notifica provisioned. Comportamento della proprietà: - *sola lettura* - *supportata* per utenti di tipo *Super admin*

m4rv1n

Descrizione

`string/object user.login(object parameters)`

Questo metodo consente di effettuare l'accesso all'API e generare un token di autenticazione.

Warning:
Quando si utilizza questo metodo, è inoltre necessario eseguire user.logout per evitare la generazione di un numero elevato di record di sessione aperti.

Attention:
Questo metodo è disponibile solo per gli utenti non autenticati che non appartengono ad alcun gruppo di utenti 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.

Parametro	Type	Descrizione
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: - *supportato* se `sessionid` è impostato Token di autenticazione dell'utente.
token	string	Comportamento del parametro: - *obbligatorio* se `token` non è impostato Token API dell'utente.
		Comportamento del parametro: - *obbligatorio* 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*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Vedere Ruoli utente per ulteriori informazioni.

Note:
La robustezza della password dell'utente viene convalidata in base alle regole della policy delle password definite dall'API di autenticazione. Vedere API di autenticazione per ulteriori informazioni.

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": "Doe123",
        "roleid": "5",
        "usrgrps": [
            {
                "usrgrpid": "7"
            }
        ],
        "medias": [
            {
                "mediatypeid": "1",
                "sendto": [
                    "support@company.com"
                ],
                "active": 0,
                "severity": 63,
                "period": "1-7,00:00-24:00"
            }
        ]
    },
    "id": 1
}
```

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 gli utenti di tipo *Super admin*. I permessi per chiamare il metodo possono essere revocati nelle impostazioni del ruolo utente. Consultare 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)`

Questo metodo consente di recuperare gli utenti 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 del ruolo utente. Vedere User roles per ulteriori informazioni.

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`.
sortfield	string/array	Possibili proprietà del User object per utenti di tipo *Admin* e *User* quando richiedono dati sugli utenti del proprio gruppo utenti: `name`, `surname`, `username`. 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",
            "refresh": "0s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "3",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "2",
            "username": "guest",
            "name": "",
            "surname": "",
            "url": "",
            "autologin": "0",
            "autologout": "15m",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "4",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        },
        {
            "userid": "3",
            "username": "user",
            "name": "Zabbix",
            "surname": "User",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "ru_RU",
            "refresh": "15s",
            "theme": "dark-theme",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "100",
            "timezone": "default",
            "roleid": "1",
            "userdirectoryid": "0",
            "ts_provisioned": "0"
        }
    ],
```

```
    "id": 1
}
```

Recupero degli utenti come *Admin*

Come utente di tipo *Admin*, recupera dati dettagliati sul tuo utente e dati limitati per gli utenti del tuo gruppo 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",
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ]
        },
        {
            "userid": "3",
            "username": "database-admin",
            "name": "John",
            "surname": "Doe",
            "url": "",
            "autologin": "0",
            "autologout": "0",
            "lang": "default",
            "refresh": "30s",
            "theme": "default",
            "attempt_failed": "0",
            "attempt_ip": "",
            "attempt_clock": "0",
            "rows_per_page": "50",
            "timezone": "default",
            "roleid": "2",
            "provisioned": "0",
            "gui_access": "0",
            "debug_mode": "0",
            "users_status": "0",
```

```
            "usrgrps": [
                {
                    "usrgrpid": "7",
                    "name": "Zabbix administrators",
                    "gui_access": "0",
                    "users_status": "0",
                    "debug_mode": "0",
                    "mfa_status": "0"
                }
            ],
            "medias": [
                {
                    "mediaid": "2",
                    "mediatypeid": "1",
                    "sendto": [
                        "john.doe@example.com"
                    ],
                    "active": "0",
                    "severity": "63",
                    "period": "1-7,00:00-24:00",
                    "provisioned": 0
                }
            ],
            "mediatypes": [
                {
                    "mediatypeid": "1",
                    "type": "0",
                    "name": "Email",
                    "status": "0",
                    "description": "",
                    "maxattempts": "3"
                }
            ],
            "role": {
                "roleid": "2",
                "name": "Admin role",
                "type": "2",
                "readonly": "0"
            }
        }
    ],
    "id": 1
}
```


Recupero dei dati utente

Recupera i dati di un utente con ID ”12”.

Richiesta:


```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```


Risposta:


```
{
  "jsonrpc": "2.0",
```

```
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "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 dei ruoli utente. Consultare 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*. I permessi per chiamare il metodo possono essere revocati 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 chiamare il metodo possono essere revocate nelle impostazioni del ruolo utente. Per ulteriori informazioni, vedere User roles.

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 utenti esistenti.

> **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.

> **Note:**
> La robustezza della password dell'utente viene convalidata in base alle regole della policy delle password definite dall'API di autenticazione. Vedere API di autenticazione 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 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.

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 (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.
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 cronologia

ZBXNEXT-10069 Aggiunto il nuovo parametro `maxValueSize` al metodo history.get. Se si stanno richiedendo dati per item con un valore binario (`history=5`), impostare `maxValueSize` su `null`; altrimenti `history.get` restituirà il valore dell'item troncato a 64KiB.

host

ZBXNEXT-10387 `host.massupdate`: Metodo rimosso. Utilizzare invece `host.update`.

hostgroup

ZBXNEXT-10387 `hostgroup.massupdate`: Metodo rimosso. Utilizzare invece `hostgroup.update`, `hostgroup.massadd` o `hostgroup.massremove`.

hostinterface

ZBXNEXT-10387 `hostinterface.replacehostinterfaces`: Metodo rimosso. Utilizzare invece `hostinterface.update`, `hostinterface.massadd` o `hostinterface.massremove`.

template

ZBXNEXT-10387 `template.massupdate`: Metodo rimosso. Utilizzare invece `template.update`, `template.massadd` o `template_massremove`.

templategroup

ZBXNEXT-10387 `templategroup.massupdate`: Metodo rimosso. Utilizzare invece `templategroup.update`, `templategroup.massadd` o `templategroup.massremove`.

Altre modifiche e correzioni di bug connettore

ZBXNEXT-10069 Aggiunto un nuovo valore `item_value_type` (64 - JSON) all'oggetto connector.

dashboard

ZBXNEXT-4769 Aggiunto il nuovo campo del widget dashboard `show_hostnames` al widget graph.
ZBXNEXT-9546 Aggiunti i nuovi campi del widget dashboard `ds.0.item_tags_evaltype`, `ds.0.item_tags.0.tag`, `ds.0.item_tags.0.operator` e `ds.0.item_tags.0.value` ai widget svggraph e piechart.
ZBXNEXT-10158 Aggiunti i nuovi campi del widget dashboard `columns.0.aggregate_columns`, `columns.0.combined_column_name` e `columns.0.column_aggregate_function` al widget topitems.
ZBXNEXT-9424, ZBXNEXT-4769 Aggiunto il nuovo tipo di widget dashboard scatterplot.
ZBXNEXT-8160 Aggiunti i nuovi campi del widget dashboard `clustering_mode` e `clustering_zoom_level` al widget geomap.
ZBXNEXT-943 Aggiunto il nuovo campo del widget dashboard `ds.0.invert_values` al widget graph.

templatedashboard

ZBXNEXT-9424 Aggiunto un nuovo tipo di widget dashboard del template scatterplot.

cronologia

ZBXNEXT-10069 Aggiunto un nuovo valore `history` (6 - JSON) al metodo history.get.
ZBXNEXT-10069 Aggiunto un nuovo oggetto `JSON history`.

hostprototype

ZBXNEXT-9546 Aggiunto il nuovo parametro `selectInheritedTags` al metodo hostprototype.get.

httptest

ZBXNEXT-9546 Aggiunti i nuovi parametri `inheritedTags` e `selectInheritedTags` al metodo httptest.get.

item

ZBXNEXT-10069 Aggiunto un nuovo valore `value_type` (6 - JSON) all'oggetto item.
ZBXNEXT-9546 Aggiunti i nuovi parametri `inheritedTags` e `selectInheritedTags` al metodo item.get.

itemprototype

ZBXNEXT-10069 Aggiunto un nuovo valore `value_type` (6 - JSON) all'oggetto itemprototype.
ZBXNEXT-9546 Aggiunto il nuovo parametro `selectInheritedTags` al metodo itemprototype.get.

template

ZBXNEXT-9546 Aggiunti i nuovi parametri `inheritedTags` e `selectInheritedTags` al metodo template.get.

trigger

ZBXNEXT-9546 Aggiunti i nuovi parametri `inheritedTags` e `selectInheritedTags` al metodo trigger.get.

triggerprototype

ZBXNEXT-9546 Aggiunto il nuovo parametro `selectInheritedTags` al metodo triggerprototype.get.

triggertag

ZBXNEXT-5467 Aggiunta la proprietà `automatic` all'oggetto tag del trigger.

userdirectory

ZBXNEXT-9834 Aggiunte le proprietà `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` e `sp_certificate_hash` all'oggetto directory utente.

Appendice 3. Modifiche nella versione 8.0

8.0.1 Questa versione non è ancora stata rilasciata.

19 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 al server o al proxy Zabbix, ad esempio dati periodici di disponibilità e prestazioni nel caso di script utente a lunga esecuzione.

L'invio di dati al server o al proxy Zabbix è possibile utilizzando l'utilità Zabbix sender o il protocol di Zabbix sender.

L'invio di dati al server Zabbix è possibile anche utilizzando 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 Zabbix agent `system.run[]`

L'agent item Zabbix `system.run[]` è un item per un comando definito dall'utente (associato a una chiave predefinita `system.run[]`, ad esempio `system.run[myscript.sh]`) che può essere eseguito sull'host in cui è in esecuzione Zabbix agent.

Nota: gli item `system.run[]` sono disabilitati per impostazione predefinita e, se utilizzati, devono essere abilitati (consentiti) e definiti nel file di configurazione di Zabbix agent o agent 2 (parametro di configurazione `AllowKey`).

> **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 agent HTTP

L'item HTTP agent è un item per eseguire richieste di dati tramite HTTP/HTTPS. Gli item HTTP agent sono utili per inviare richieste a endpoint HTTP per recuperare dati da servizi come *Elasticsearch* e *OpenWeatherMap*, per controllare lo stato della API di Zabbix o lo stato del server web Apache o Nginx, ecc. Gli item HTTP agent (con 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).

Un plugin è un pacchetto Go che definisce la struttura e implementa una o più interfacce di plugin (*Exporter*, *Collector*, *Configurator*, *Runner*, *Watcher*). Sono supportati due tipi di plugin di Zabbix agent 2:

- Plugin integrati (supportati da Zabbix 4.4.0)
- Plugin caricabili (supportati da Zabbix 6.0.0)

Vedere l'elenco dei plugin forniti da Zabbix.

Per istruzioni ed esercitazioni sulla creazione dei propri plugin, vedere il Centro sviluppatori.

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.

1 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. Pertanto, questa funzione deve restituire questa costante. La vecchia costante ZBX_MODULE_API_VERSION_ONE, usata a 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_write_cbs()** e **zbx_module_uninit()**:

```
int zbx_module_init(void);
```

Questa funzione deve eseguire l'inizializzazione necessaria per il modulo (se presente). In caso di successo, 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; per i dettagli vedere la sezione seguente. L'elenco termina con una struttura ZBX_METRIC il cui campo "key" è NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Se il modulo esporta **zbx_module_item_list()**, allora questa funzione viene utilizzata 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 utilizzerà per esportare la cronologia di 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 deinizializzazione necessaria (se presente), ad esempio liberando le risorse allocate, chiudendo 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 parametri o meno), **function** è una funzione C che implementa l'item (ad esempio, "zbx_module_dummy_random"), e **test_param** è l'elenco dei parametri da utilizzare quando Zabbix agent viene avviato con il flag "-p" (ad esempio, "1,1000", può essere NULL). Una definizione di esempio può apparire così:

```
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. Per i dettagli su come ottenere informazioni da AGENT_REQUEST e come impostare informazioni in AGENT_RESULT, vedere l'esempio del modulo "dummy" riportato di seguito.

Fornire callback di esportazione della cronologia

Attention:
L'esportazione della cronologia tramite modulo non è più supportata da Zabbix proxy.

Il modulo può specificare funzioni per esportare i dati della cronologia per tipo: Numerico (float), Numerico (unsigned), Carattere, Testo 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;
```

Ciascuna di esse deve accettare come argomenti l'array "history" di "history_num" elementi. A seconda del tipo di dati della cronologia da esportare, "history" è rispettivamente un array delle seguenti strutture:

```
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;
```

Le callback saranno utilizzate dai processi history syncer di Zabbix server al termine della procedura di sincronizzazione della cronologia, dopo che i dati sono stati scritti nel database di Zabbix e salvati nella cache dei valori.

> **Attention:**
> In caso di errore interno nel modulo di esportazione della cronologia, si raccomanda di scrivere il modulo in modo tale che non blocchi l'intero monitoraggio fino al ripristino, ma che invece scarti i dati e consenta a Zabbix server di continuare a funzionare.

Compilazione dei moduli

Attualmente i moduli devono 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. Altri header di sistema necessari affinché **include/module.h** funzioni correttamente sono **stdlib.h** e **stdint.h**.

Tenendo presenti queste informazioni, tutto è pronto per compilare il modulo. Il modulo deve includere **stdlib.h**, **stdint.h** e **module.h**, e lo script di build deve assicurarsi che questi file siano nel percorso di inclusione. Per i dettagli, vedere l'esempio del modulo "dummy" riportato di seguito.

Un altro header utile è **include/zbxcommon.h**, che definisce la funzione **zabbix_log()**, utilizzabile per finalità 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/i 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 che precede il nome è assoluto (inizia con '/'), allora LoadModulePath viene ignorato. È consentito includere più parametri LoadModule.

Ad esempio, per estendere Zabbix agent potremmo 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 verranno caricati i moduli mariadb.so, apache.so e kernel.so dalla directory /usr/local/lib/zabbix/agent, mentre dummy.so verrà caricato da /usr/local/lib/zabbix. L'agent non riuscirà ad avviarsi se manca un modulo, in caso di permessi non corretti oppure 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, il tipo di item deve essere "Zabbix agent" oppure "Zabbix agent (active)". Se il modulo viene caricato nel server o nel proxy, il tipo di item deve essere "Simple check".

L'esportazione della history tramite i moduli di 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 di callback non NULL, l'esportazione della history verrà abilitata automaticamente.

Modulo fittizio

Zabbix include un modulo di esempio scritto nel 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 utilizzato come template per i propri moduli.

Dopo aver eseguito ./configure nella directory radice dell'albero dei sorgenti di Zabbix come descritto sopra, è sufficiente eseguire **make** per compilare **dummy.so**.

```c
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
```

```
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}

static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param;

    if (1 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param = get_rparam(request, 0);

    SET_STR_RESULT(result, strdup(param));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: dummy_random                                                     *
 *                                                                            *
 * Purpose: a main entry point for processing of an item                      *
 *                                                                            *
 * Parameters: request - structure that contains item key and parameters      *
 *              request->key - item key without parameters                    *
 *              request->nparam - number of parameters                        *
 *              request->params[N-1] - pointers to item key parameters        *
 *              request->types[N-1] - item key parameters types:              *
```

```
 *                  REQUEST_PARAMETER_TYPE_UNDEFINED (key parameter is empty) *
 *                  REQUEST_PARAMETER_TYPE_ARRAY (array)                      *
 *                  REQUEST_PARAMETER_TYPE_STRING (quoted or unquoted string) *
 *                                                                            *
 *             result - structure that will contain result                    *
 *                                                                            *
 * Return value: SYSINFO_RET_FAIL - function failed, item will be marked      *
 *                                  as not supported by zabbix                *
 *               SYSINFO_RET_OK - success                                     *
 *                                                                            *
 * Comment: get_rparam(request, N-1) can be used to get a pointer to the Nth  *
 *          parameter starting from 0 (first parameter). Make sure it exists  *
 *          by checking value of request->nparam.                             *
 *          In the same manner get_rparam_type(request, N-1) can be used to   *
 *          get a parameter type.                                             *
 *                                                                            *
 ******************************************************************************/
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    char    *param1, *param2;
    int from, to;

    if (2 != request->nparam)
    {
        /* set optional error message */
        SET_MSG_RESULT(result, strdup("Invalid number of parameters."));
        return SYSINFO_RET_FAIL;
    }

    param1 = get_rparam(request, 0);
    param2 = get_rparam(request, 1);

    /* there is no strict validation of parameters and types for simplicity sake */
    from = atoi(param1);
    to = atoi(param2);

    if (from > to)
    {
        SET_MSG_RESULT(result, strdup("Invalid range specified."));
        return SYSINFO_RET_FAIL;
    }

    SET_UI64_RESULT(result, from + rand() % (to - from + 1));

    return SYSINFO_RET_OK;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_init                                                  *
 *                                                                            *
 * Purpose: the function is called on agent startup                           *
 *          It should be used to call any initialization routines             *
 *                                                                            *
 * Return value: ZBX_MODULE_OK - success                                      *
 *               ZBX_MODULE_FAIL - module initialization failed               *
 *                                                                            *
 * Comment: the module won't be loaded in case of ZBX_MODULE_FAIL             *
 *                                                                            *
 ******************************************************************************/
int zbx_module_init(void)
{
```

```
    /* initialization for dummy.random */
    srand(time(NULL));

    return ZBX_MODULE_OK;
}

/******************************************************************************
 *                                                                            *
 * 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; per esempio, `dummy.echo[ABC]` restituirà ABC
- `dummy.random[param1, param2]` - restituisce un numero casuale nell'intervallo param1-param2; per 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 di Windows.

In alcuni casi un modulo potrebbe dover leggere parametri di configurazione relativi al modulo da *zabbix_agentd.conf*. Attualmente questo non è supportato. Se hai bisogno che il tuo modulo utilizzi alcuni parametri di configurazione, probabilmente dovresti implementare il parsing di un file di configurazione specifico del modulo.

3 moduli frontend

Panoramica

È possibile estendere le funzionalità del frontend di Zabbix aggiungendo moduli di terze parti oppure sviluppando moduli personalizzati, senza la necessità di modificare il codice sorgente di Zabbix.

Si noti che il codice del modulo verrà eseguito con gli stessi privilegi del codice sorgente di Zabbix. Ciò significa che:

- i moduli di terze parti possono essere dannosi. È necessario fidarsi dei moduli che si stanno installando;
- gli errori nel codice di un modulo di terze parti possono causare il crash del frontend. Se ciò accade, è sufficiente rimuovere il codice del modulo dal frontend. Non appena si ricarica il frontend di Zabbix, verrà visualizzato un avviso che indica che alcuni moduli sono assenti. Andare a Amministrazione dei moduli (in *Amministrazione* → *Generale* → *Moduli*) e fare nuovamente clic su *Scansiona directory* per rimuovere dal database i moduli non esistenti.

Installazione

Leggere sempre il manuale di installazione del modulo specifico. Si consiglia di installare i nuovi moduli uno alla volta, per individuare facilmente eventuali errori.

Subito prima di installare un modulo:

- Assicurarsi di aver scaricato il modulo da una fonte affidabile. 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:

- Estrarre il modulo nella propria cartella all'interno della cartella `modules` del frontend di Zabbix
- Assicurarsi che la cartella del modulo contenga almeno il file manifest.json
- Andare a Amministrazione dei moduli e fare clic sul pulsante *Scansiona 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:

Problem	Solution
Il modulo non è apparso nell'elenco	Se il modulo non appare nell'elenco, assicurarsi che manifest.json esista in modules/your-module/. In caso contrario, il modulo potrebbe essere stato estratto nella directory sbagliata. Se invece esiste, il modulo potrebbe non essere compatibile con la propria versione di Zabbix. Inoltre, verificare che l'utente del web server abbia almeno accesso in lettura e ricerca (`r-x`) alla cartella del modulo e a tutte le sottodirectory, e accesso in lettura (`r--`) a tutti i file al loro interno.
Il frontend si è bloccato	Il codice del modulo non è compatibile con l'attuale versione 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 dei moduli e fare clic di nuovo su *Scansiona directory* per rimuovere dal database i moduli non esistenti.
Viene visualizzato un messaggio di errore relativo a namespace, ID o azioni identici	Il nuovo modulo ha tentato di registrare un namespace, un ID o delle azioni già registrati da altri moduli abilitati. Disabilitare il modulo in conflitto (indicato nel messaggio di errore) prima di abilitare quello nuovo.
Vengono visualizzati 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.

20 Appendici

Utilizzare la barra laterale per accedere ai contenuti della sezione Appendici.

1 Installazione e configurazione

Utilizzare la barra laterale per accedere ai contenuti di questa sezione.

1 Creazione del database

Panoramica

Durante l'installazione di Zabbix server o proxy deve essere creato un database Zabbix.

Questa sezione fornisce istruzioni per la creazione di un database Zabbix. È disponibile un insieme separato di istruzioni per ciascun database supportato.

> **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 Connessione sicura al database.

UTF-8 è l'unica codifica supportata da Zabbix. È noto che funziona senza vulnerabilità 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 Ripristino del set di caratteri e della collation del database Zabbix. Vedere anche Limiti del filtraggio con collation utf8mb4.

> **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

I set di caratteri utf8 (ovvero utf8mb3) e utf8mb4 sono supportati (rispettivamente con le regole di confronto utf8_bin e utf8mb4_bin) affinché Zabbix server/proxy funzioni correttamente con il database MySQL. Per le nuove installazioni si consiglia di utilizzare utf8mb4.

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 il binary logging è abilitato, non sono presenti privilegi di superuser e `log_bin_trust_function_creators = 1` non è impostato nel file di configurazione di MySQL.

Se si sta installando dai **pacchetti** Zabbix, procedere alle istruzioni per la propria piattaforma.

Se si sta installando Zabbix dai sorgenti:

- Creare e configurare un database e un utente.

```
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 utf8mb4 come set di caratteri predefinito (supponendo di trovarsi nella directory root dei sorgenti di Zabbix). Per un database Zabbix proxy, 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 il database per Zabbix proxy
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

`log_bin_trust_function_creators` 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 si esegue l'installazione dai **packages** di Zabbix, procedere con le istruzioni per la propria piattaforma.

Se si installa Zabbix dai sorgenti:

- Creare un utente del database.

Il seguente comando shell creerà l'utente `zabbix`. Specificare una password quando richiesto e ripetere la password (nota: potrebbe essere prima richiesta la password di `sudo`):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Creare 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
```

- Importare lo schema iniziale e i dati (supponendo di trovarsi nella directory root dei sorgenti di Zabbix). Per un database 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
#### fermarsi qui se si sta 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 riportati sono forniti come esempio e funzioneranno nella maggior parte delle installazioni GNU/Linux. È possibile utilizzare comandi diversi a seconda di come è configurato il sistema/database, ad esempio:

 `psql -U <username>`

 In caso di problemi durante la configurazione del database, consultare l'amministratore del database.

TimescaleDB

Le istruzioni per creare e configurare TimescaleDB sono fornite in una sezione separata.

SQLite

L'uso di SQLite è supportato solo per **Zabbix proxy**!

Il database verrà creato automaticamente se non esiste.

Torna alla sezione di installazione.

2 Riparazione del set di caratteri e delle regole di confronto del database Zabbix

MySQL/MariaDB

Storicamente, MySQL e i suoi derivati hanno usato 'utf8' come alias di utf8mb3, l'implementazione a 3 byte di MySQL dello standard UTF8, che invece 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'. 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 ai caratteri Unicode supplementari.

Warning:
Poiché le versioni precedenti a Zabbix 6.0 non riconoscono utf8mb4, assicurarsi prima di aggiornare Zabbix server e lo schema del DB alla versione 6.0.x o successiva prima di eseguire la conversione a utf8mb4.

1. Verificare il set di caratteri e la collation del database.

Per 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 può vedere, qui il set di caratteri non è 'utf8mb4' e la collation non è 'utf8mb4_bin', quindi è necessario correggerli.

2. Arrestare Zabbix.

3. Creare una copia di backup del database!

4. Correggere 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. Caricare 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. Eseguire 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;
```

Si noti che è previsto che 'utf8mb4' consumi leggermente più spazio su disco.

7. Se non ci sono errori, si consiglia di creare una copia di backup del database corretto.

8. Avviare Zabbix.

3 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 consultati, il che può velocizzare le query e risparmiare spazio. Migliora inoltre la gestione e la sincronizzazione dei dati nelle configurazioni cluster, facilitando 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 avanzati e potrebbero dover essere adattate alla configurazione specifica in uso. 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 potrebbe richiedere fino a 2,5 volte lo spazio attualmente utilizzato dalle tabelle dello storico.

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 Zabbix server alla versione 6.0 o successiva.

> **Attention:**
> A partire da Zabbix 7.0, l'aggiornamento delle tabelle alle chiavi primarie aggiorna anche le tabelle affinché utilizzino tipi di dati double precision.

 Se si utilizza Zabbix 7.0 (o successivo), le tabelle utilizzano già double precision. Tuttavia, è comunque possibile seguire le istruzioni in questa pagina per aggiornare le tabelle alle chiavi primarie senza influire sulle tabelle che utilizzano già double precision.

 Se si utilizza Zabbix 6.4 (o precedente), valutare prima l'aggiornamento delle tabelle a double precision. Per ulteriori informazioni, vedere Upgrading to numeric values of extended range nella documentazione di Zabbix 7.0.

Sono disponibili istruzioni per:

- 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.

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 tabelle 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 si riceve 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 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 aggiornamento richiede più tempo e deve essere utilizzato solo se non è possibile eseguire un aggiornamento con *mysqlsh*.

Aggiornamento delle tabelle

- Accedere alla console MySQL come root (consigliato) o come qualsiasi utente con privilegi FILE.
- Se si esegue la migrazione con un server Zabbix in esecuzione, avviare MySQL con la variabile local_infile abilitata.
- Rinominare le vecchie tabelle e creare le nuove tabelle 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) deve essere disabilitato 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;
```

Seguire le istruzioni post-migrazione per eliminare le vecchie tabelle.

Migrazione con server in esecuzione

Verificare per quali percorsi è abilitata l'importazione/esportazione:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Se il valore di *secure_file_priv* è un percorso a una directory, l'esportazione/importazione verrà eseguita per i file in quella directory. In questo caso, modificare di conseguenza i percorsi dei file nelle query oppure impostare il valore di *secure_file_priv* su una stringa vuota per la durata dell'aggiornamento.

Se il valore di *secure_file_priv* è vuoto, l'esportazione/importazione può essere eseguita da qualsiasi posizione.

Se il valore di *secure_file_priv* è NULL, impostarlo sul percorso che contiene i dati della tabella esportata (`/var/lib/mysql-files/` nell'esempio sopra).

Per ulteriori informazioni, vedere la documentazione MySQL o la documentazione MariaDB.

`max_execution_time` (in MySQL) o `max_statement_time` (in MariaDB) deve essere disabilitato 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
```

Seguire 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.

Vedere 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;
```

Vedere 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 spento durante l'aggiornamento.

Vedi anche: Note importanti

- Rinominare le tabelle utilizzando `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/opti`

```
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

4 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 la dimensione corrente 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 in Modifica della chiave primaria senza tempi di inattività. Sono disponibili anche altre opzioni:

- L'aumento del parametro MySQL `sort_buffer_size` può aiutare a ridurre l'utilizzo dello spazio su disco durante la ricostruzione manuale degli 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* di housekeeper prima di eseguire housekeeper.

Modifica della chiave primaria con downtime

1. Eliminare 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 tempi di inattività

Il metodo manuale per modificare la chiave primaria è descritto qui. In alternativa, è possibile utilizzare il toolkit pt-online-schema-change di Percona. Questo toolkit esegue automaticamente le seguenti azioni, riducendo al minimo anche lo spazio utilizzato per modificare la tabella `auditlog`.

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 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. Scambiare le tabelle.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Copiare i dati dalla vecchia tabella a quella nuova.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Questa operazione può essere eseguita in blocchi (più istruzioni `INSERT INTO` con clausole `WHERE clock` secondo necessità) per evitare un utilizzo eccessivo delle risorse.

4. Eliminare 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 attuale 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 downtime. Sono disponibili anche altre opzioni:

- Aumentare il parametro PostgreSQL `maintenance_work_mem` può aiutare a ridurre l'utilizzo di spazio su disco durante la ricostruzione manuale degli indici. Tuttavia, la modifica di questa variabile può influire sull'utilizzo complessivo della memoria del database.
- Se si dispone di un altro disco o tablespace con più spazio disponibile, si può valutare di modificare la posizione di archiviazione temporanea per la ricostruzione degli indici. 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* di housekeeper prima di eseguire housekeeper.

Modifica della chiave primaria con downtime

1. Eliminare l'attuale chiave primaria 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 (facoltativo ma fortemente consigliato, vedere Note importanti sulla ricostruzione degli indici).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Modifica della chiave primaria senza tempi di inattività

Il metodo manuale per modificare la chiave primaria è descritto qui. In alternativa, si può prendere in considerazione l'estensione `pg_repack` 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 a quella nuova.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

Questa operazione può essere eseguita in blocchi (più istruzioni `INSERT INTO` con clausole `WHERE clock` secondo necessità) per evitare un utilizzo eccessivo delle risorse.

4. Eliminare la vecchia tabella.

```
DROP TABLE auditlog_old;
```

Vedi anche

- Aggiornamento del database alle chiavi primarie

5 Connessione sicura al database

Panoramica

Questa sezione fornisce i passaggi di configurazione di Zabbix ed esempi di configurazione per connessioni TLS sicure tra:

Database	Componenti Zabbix
MySQL	Zabbix frontend, Zabbix server, Zabbix proxy
PostgreSQL	Zabbix frontend, Zabbix server, Zabbix proxy

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 che utilizzano CentOS 8.

Requisiti

Per configurare la crittografia è necessario quanto segue:

- Sistema operativo supportato dallo sviluppatore con OpenSSL >=1.1.X o alternativa.

> **Note:**
> Si consiglia di evitare sistemi operativi in stato di fine ciclo di vita, specialmente nel caso di nuove installazioni

- Motore di database (RDBMS) installato e mantenuto dal repository ufficiale fornito dallo sviluppatore. 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 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 suo certificato

Configurazione Zabbix

Frontend al database

Una connessione sicura al database può essere configurata durante l'installazione del frontend:

- Selezionare la casella *Database TLS encryption* nel passaggio Configure DB connection per abilitare la cifratura del trasporto.

- Selezionare la casella *Verify database certificate* che appare quando il campo *TLS encryption* è selezionato per abilitare la cifratura con certificati.

Note:
Per MySQL, la casella *Database TLS encryption* è disabilitata se *Database host* è impostato su localhost, perché una connessione che utilizza un file socket (su Unix) o memoria condivisa (su Windows) non può essere cifrata.
Per PostgreSQL, la casella *TLS encryption* è disabilitata se il valore del campo *Database host* inizia con una barra oppure se il campo è vuoto.

I seguenti parametri diventano disponibili nella modalità di cifratura TLS con certificati (se entrambe le caselle sono selezionate):

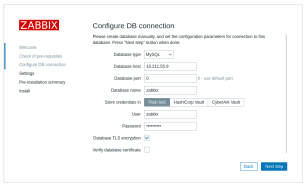
Parameter	Description
Database TLS CA file	Specificare il percorso completo di un file valido del certificato dell'autorità di certificazione TLS (CA).
Database TLS key file	Specificare il percorso completo di un file valido della chiave TLS.
Database TLS certificate file	Specificare il percorso completo di un file valido del certificato TLS.
Database host verification	Selezionare questa casella per attivare la verifica dell'host. Disabilitata per MYSQL, perché la libreria PHP MySQL non consente di saltare il passaggio di convalida del certificato del peer.
Database TLS cipher list	Specificare un elenco personalizzato di cifrari validi. Il formato dell'elenco dei cifrari deve essere conforme allo standard OpenSSL. Disponibile solo per MySQL.

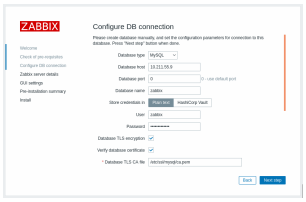
Attention:
I parametri TLS devono puntare a file validi. Se puntano a file inesistenti o non validi, ciò comporterà un errore di autorizzazione.
Se i file dei certificati sono scrivibili, il frontend genera un avviso nel report System information che indica "TLS certificate files must be read-only." (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 l'interfaccia GUI per definire le possibili opzioni: `required`, `verify_ca`, `verify_full`. Specificare le opzioni richieste nel passaggio della procedura guidata di installazione *Configure DB connections*. Queste opzioni sono mappate nel file di configurazione (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'] = ''; ...	Selezionare *Database TLS encryption* Lasciare deselezionato *Verify database certificate*	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 *Database TLS encryption* e *Verify database certificate* 2. Specificare il percorso di *Database TLS CA file*	Abilita la modalità `verify_ca`.

	...
	// Utilizzato per la connessione TLS con un elenco Cipher definito in modo rigoroso.
	$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 *Database TLS encryption* e *Verify database certificate*
		2. Specificare il percorso di *Database TLS key file*
		3. Specificare il percorso di *Database TLS CA file*
		4. Specificare il percorso di *Database TLS certificate file*
		5. Specificare *Database TLS cipher list* (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 *Database TLS encryption* e *Verify database certificate* 2. Specificare il percorso di *Database TLS key file* 3. Specificare il percorso di *Database TLS CA file* 4. Specificare il percorso di *Database TLS certificate file* 5. Selezionare *Database host verification*	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.
1. Impostare `DBTLSConnect=required`	Server/proxy stabiliscono una connessione TLS al database. Una connessione non crittografata non è consentita.
1. Impostare `DBTLSConnect=verify_ca` 2. Impostare `DBTLSCAFile` - specificare il file dell'autorità di certificazione TLS	Server/proxy stabiliscono 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	Server/proxy stabiliscono una connessione TLS al database dopo aver verificato il certificato del database e l'identità del 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	Server/proxy forniscono un certificato client durante la connessione al database.
1. 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 sarà considerata un errore.

1 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 Zabbix frontend 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 riportate in questa pagina. Sono disponibili molte altre combinazioni.

Prerequisiti

Installare il database MySQL dal repository ufficiale.

Per i dettagli su come utilizzare il repository MySQL, consultare la documentazione di MySQL.

Il server MySQL è pronto ad accettare connessioni sicure utilizzando un certificato autofirmato.

Per vedere quali utenti stanno utilizzando una connessione crittografata, eseguire la seguente query (Performance Schema deve essere attivato):

```
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`. Un certificato lato server verrà creato dopo la configurazione iniziale e l'avvio.

Creare utenti e ruoli per i componenti principali:

```
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 usare solo connessioni crittografate. Per maggiori dettagli sulla configurazione degli utenti, vedere la documentazione MySQL.

Eseguire il comando seguente per verificare la connessione (la connessione socket non può essere utilizzata per testare connessioni sicure):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

Controllare 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;
```

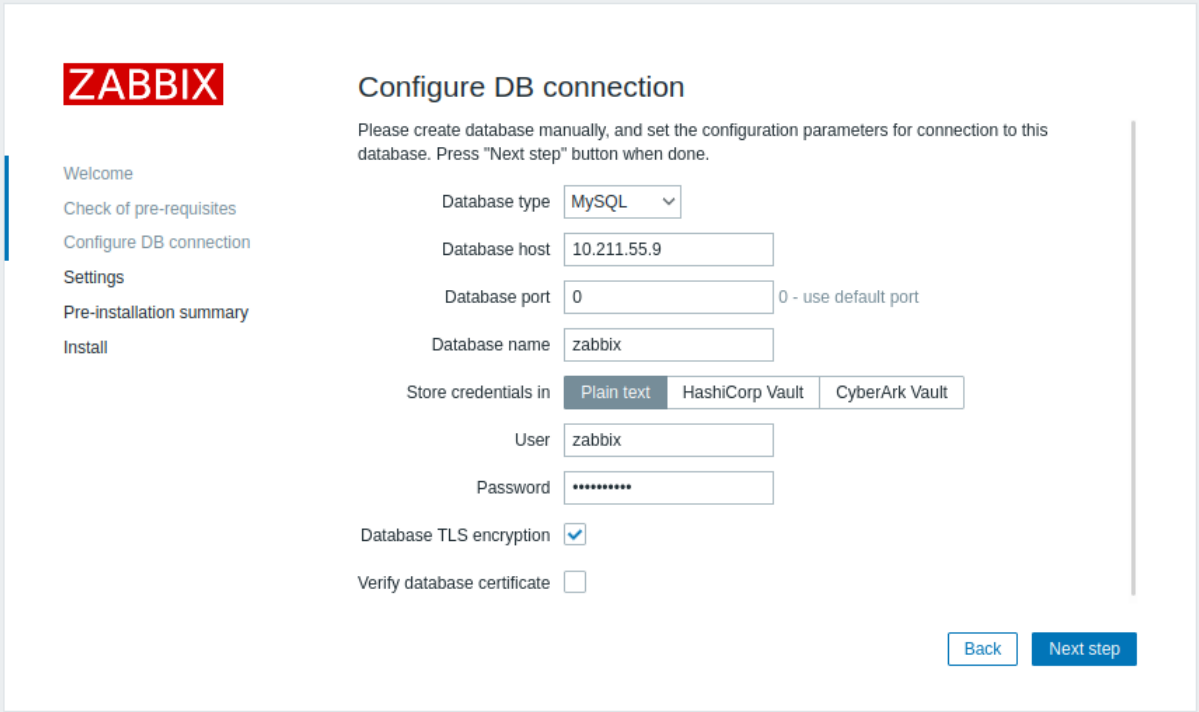
```
*************************** 1. row ***************************
Variable_name: Ssl_cipher_list
Value: TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_128_CCM_SHA256:E
1 row in set (0.00 sec)

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 *Verify database certificate* non selezionato



Server

Per abilitare la crittografia solo del trasporto per le connessioni tra server e il database, configurare */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Crittografia con verifica dell'autorità di certificazione

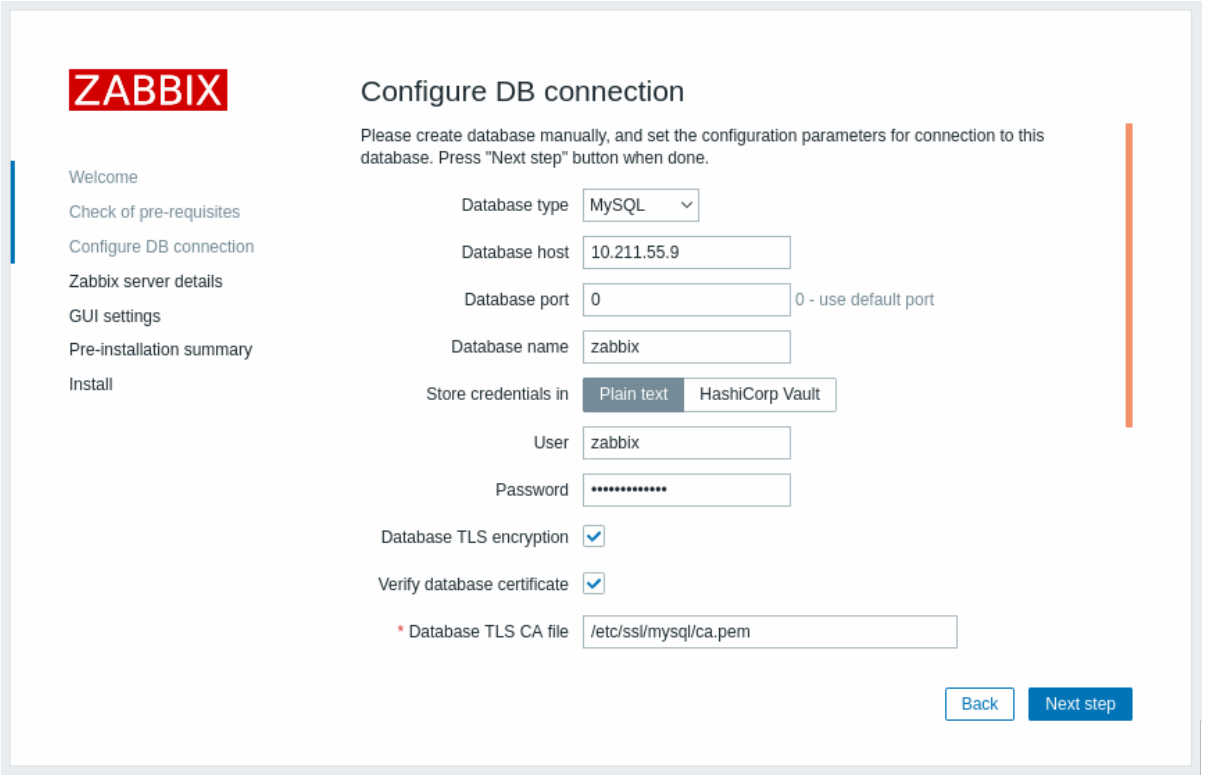
Copia la CA MySQL richiesta sul server frontend di Zabbix e assegna i permessi appropriati per consentire al webserver di leggere questo file.

> **Note:**
> Questa modalità non funziona su RHEL 7 a causa delle librerie MySQL meno recenti.

Frontend

Per abilitare la crittografia con verifica del certificato per le connessioni tra Zabbix frontend e il database:

- Selezionare *Database TLS encryption* e *Verify database certificate*
- Specificare il percorso del file CA TLS del database



In alternativa, è possibile impostarlo in */etc/zabbix/web/zabbix.conf.php*:

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Per la risoluzione dei problemi, utilizzare uno 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, configurare */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/mysql/ca.pem
...
```

Crittografia con verifica completa

Configurazione di MySQL

Impostare l'opzione di configurazione del server MySQL CE (*/etc/my.cnf.d/server-tls.cnf*) su:

```
[mysqld]
...
##### in questo esempio 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 il client (Zabbix frontend) devono essere create manualmente secondo la 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 sul nome FQDN, poiché Zabbix frontend utilizzerà il nome DNS per comunicare con il database oppure l'indirizzo IP del 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 effettuare l'accesso con tale 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
```

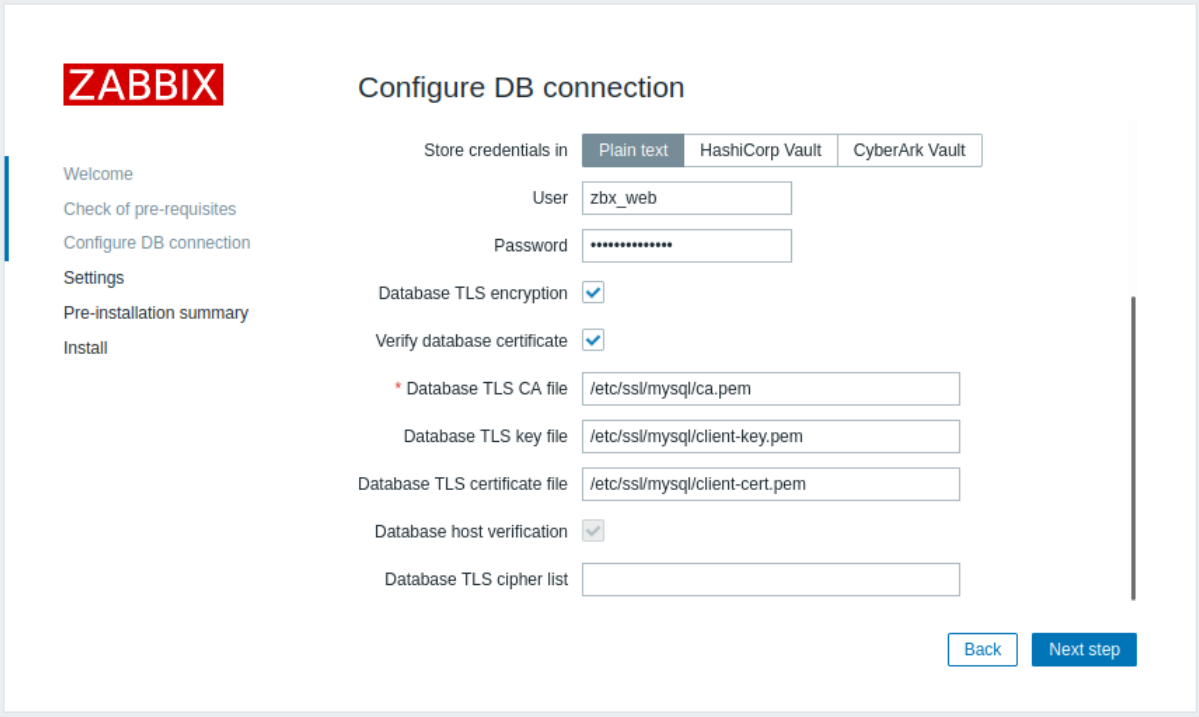
Frontend

Per abilitare la crittografia con verifica completa per le connessioni tra il frontend di Zabbix e il database:

- Seleziona *Database TLS encryption* e *Verify database certificate*
- Specifica il percorso di *Database TLS key file*
- Specifica il percorso di *Database TLS CA file*
- Specifica il percorso di *Database TLS certificate file*

Nota che *Database host verification* è selezionato e disattivato - questo passaggio non può essere saltato per MySQL.

Warning:
Se il campo *Database TLS cipher list* 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, questo può essere impostato in */etc/zabbix/web/zabbix.conf.php*:

```
...
// Utilizzato per la connessione TLS con un elenco di cifrari definito in modo rigoroso.
$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 Zabbix server e il database, configurare */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 Configurazione della crittografia PostgreSQL2

Panoramica

Questa sezione fornisce diversi esempi di configurazione della crittografia per CentOS 8.2 e PostgreSQL 13.

> **Note:**
> La connessione tra Zabbix frontend e PostgreSQL non può essere crittografata (i parametri nella GUI sono disabilitati) se il valore del campo *Database host* inizia con una barra oppure se il campo è vuoto.

Prerequisiti

Installare il database PostgreSQL utilizzando il repository ufficiale.

PostgreSQL non è configurato per accettare connessioni TLS immediatamente dopo l'installazione. Seguire le istruzioni della documentazione di PostgreSQL per la preparazione dei certificati con postgresql.conf e anche per il controllo dell'accesso utente tramite ph_hba.conf.

Per impostazione predefinita, il socket di PostgreSQL è associato a localhost; per consentire connessioni remote di rete, abilitare l'ascolto sull'interfaccia di rete reale.

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, modificare */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

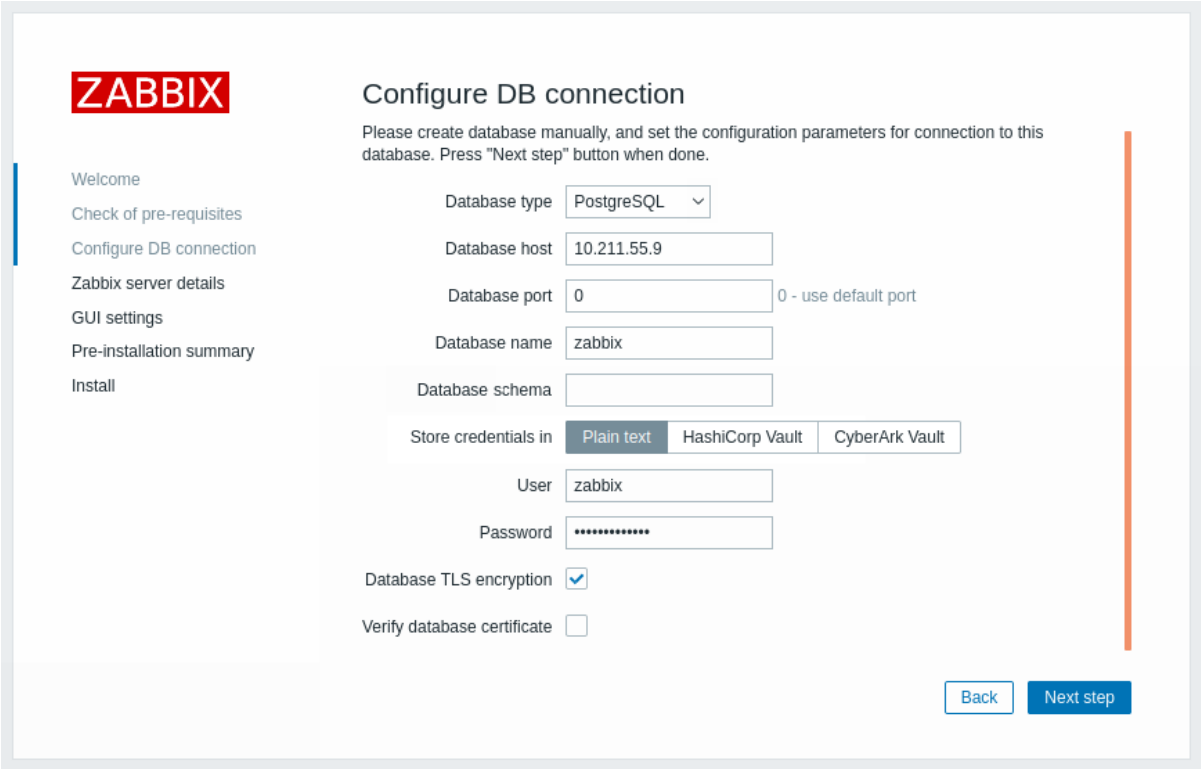
### verify full
hostssl all all 0.0.0.0/0 md5 clientcert=verify-full
...
```

Crittografia solo del trasporto

Frontend

Per abilitare la crittografia del solo 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 server e il database, configurare */etc/zabbix/zabbix_server.conf*:

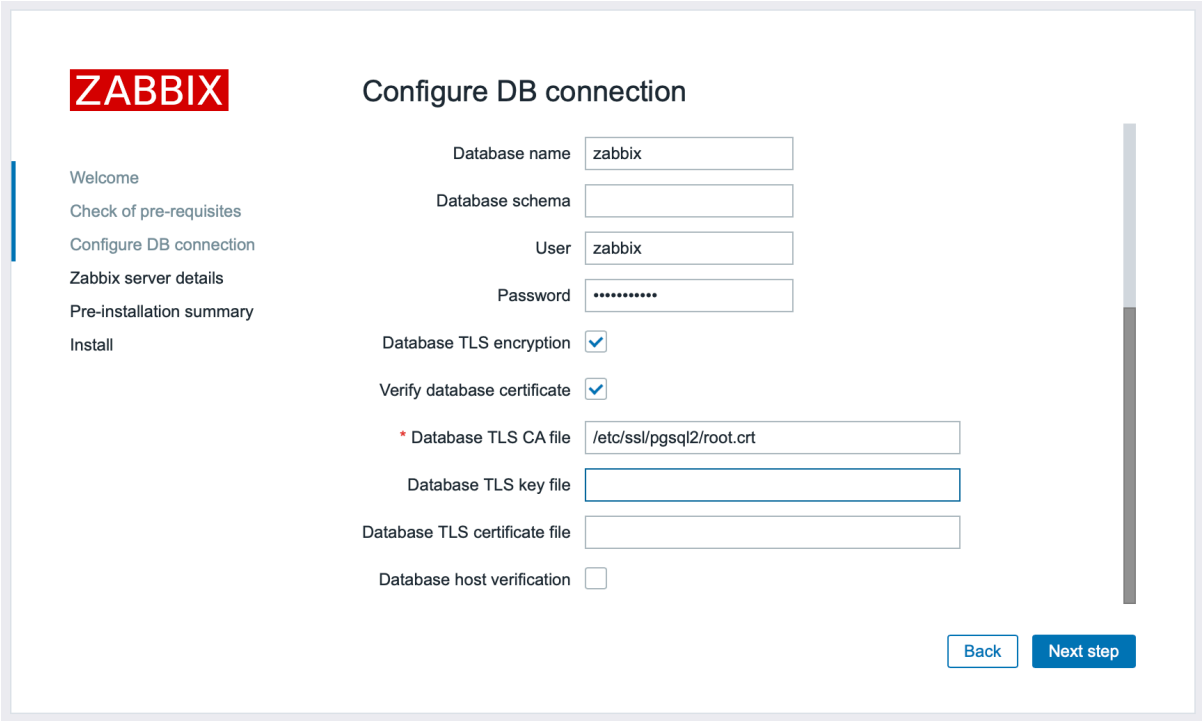
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

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 *Database TLS encryption* e *Verify database certificate*
- Specificare il percorso di *Database TLS CA file*



In alternativa, è possibile impostarlo in */etc/zabbix/web/zabbix.conf.php:*

```
...
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = false;
$DB['CIPHER_LIST'] = '';
...
```

Server

Per abilitare la crittografia con verifica del certificato per le connessioni tra Zabbix server e il database, configurare */etc/zabbix/zabbix_server.conf:*

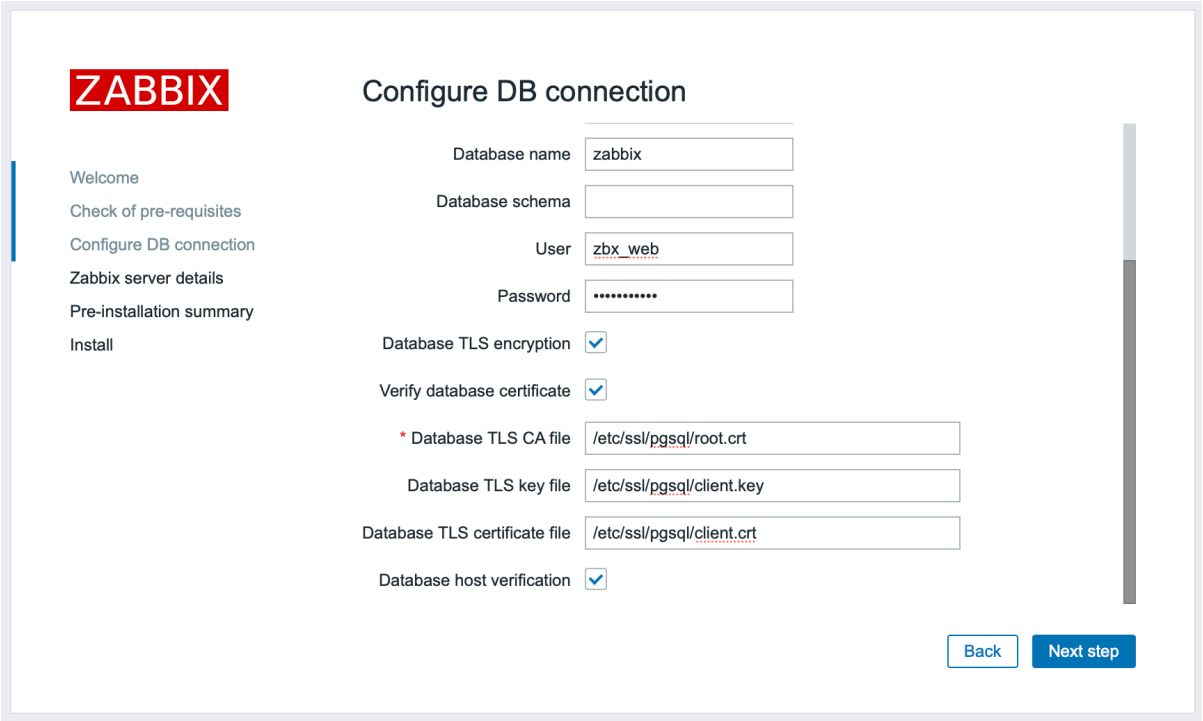
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Crittografia con verifica completa

Frontend

Per abilitare la crittografia con certificato e la verifica dell'identità del host del database per le connessioni tra il frontend di Zabbix e il database:

- Selezionare *Database TLS encryption* e *Verify database certificate*
- Specificare il percorso di *Database TLS key file*
- Specificare il percorso di *Database TLS CA file*
- Specificare il percorso di *Database TLS certificate file*
- Selezionare *Database host verification*



In alternativa, è possibile impostarlo in */etc/zabbix/web/zabbix.conf.php:*

```
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '';
$DB['CERT_FILE'] = '';
$DB['CA_FILE'] = '/etc/ssl/pgsql/root.crt';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Server

Per abilitare la crittografia con certificato e la verifica dell'identità del host del database per le connessioni tra Zabbix server e il database, configurare */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/pgsql/root.crt
DBTLSCertFile=/etc/ssl/pgsql/client.crt
DBTLSKeyFile=/etc/ssl/pgsql/client.key
...
```

6 Connessione sicura al frontend

Panoramica

Questa sezione fornisce i passaggi di configurazione di Zabbix ed esempi di configurazione per connessioni TLS sicure tra Zabbix frontend e Zabbix server.

Configurazione

Per impostazione predefinita, la comunicazione tra Zabbix frontend e Zabbix server non è crittografata. Per una maggiore sicurezza, abilitare TLS su entrambi i lati. Di seguito è riportato un esempio del modo più semplice per farlo.

1. Generare certificati e chiavi.

Creare una directory di lavoro:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Creare un certificato CA (adattare il valore `MyZabbixCA` in modo che corrisponda al common name effettivo):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Generare una chiave privata e un certificato per Zabbix server (adattare il valore `zabbix-server.example.com` in modo che corrisponda al common name 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
```

Generare una chiave privata e un certificato per Zabbix frontend (adattare il valore `zabbix-frontend-node` in modo che corrisponda al common name 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. Impostare i permessi corretti.

Per Zabbix server (adattare proprietario/gruppo in base all'utente del demone Zabbix server della propria 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 (adattare proprietario/gruppo in base all'utente del webserver della propria 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. Configurare Zabbix server.

In `zabbix_server.conf` aggiungere:

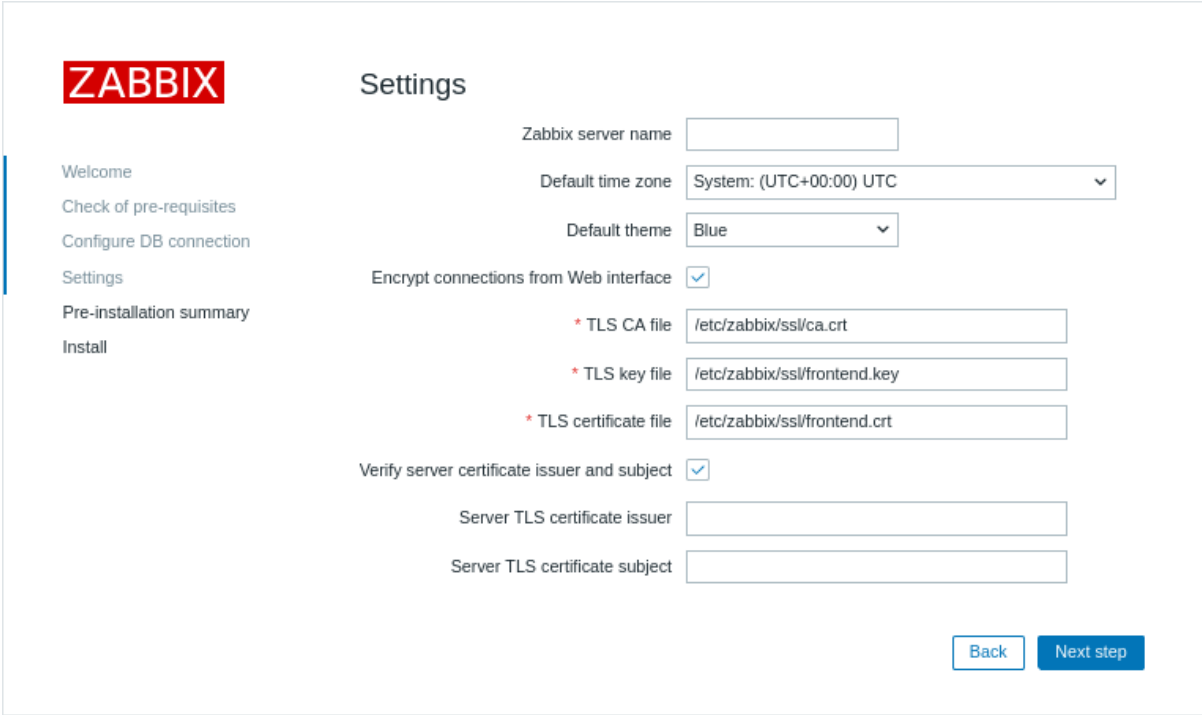
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Facoltativamente:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Quindi riavviare il server:

```
sudo systemctl restart zabbix-server
```

4. Configurare Zabbix frontend.

Durante l'installazione dell'interfaccia web, abilitare l'opzione *Encrypt connections from Web interface* (e l'opzione *Verify server certificate issuer and subject*, se necessario) e compilare i campi *TLS CA file*, *TLS key file*, *TLS certificate file* (e i campi *Server TLS certificate issuer* e *Server TLS certificate subject*, se necessario):



Parameter	Description
TLS CA file	Specificare il percorso completo del file del certificato della Certificate Authority (CA) utilizzato per verificare il certificato del server.
TLS key file	Specificare il percorso completo del file della chiave privata del client corrispondente al certificato del client.
TLS certificate file	Specificare il percorso completo del file del certificato del client se è richiesta l'autenticazione TLS reciproca.
Server TLS certificate issuer	Specificare un Distinguished Name (DN) dell'emittente da confrontare con il certificato del server.
Server TLS certificate subject	Specificare un Distinguished Name (DN) del soggetto da confrontare con il certificato del server.

Nelle installazioni esistenti, modificare i seguenti campi in `zabbix.conf.php`:

```
$ZBX_SERVER_TLS['ACTIVE'] = '1';
$ZBX_SERVER_TLS['CA_FILE'] = '/etc/zabbix/ssl/ca.crt';
$ZBX_SERVER_TLS['KEY_FILE'] = '/etc/zabbix/ssl/frontend.key';
$ZBX_SERVER_TLS['CERT_FILE'] = '/etc/zabbix/ssl/frontend.crt';
// Facoltativamente:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Verificare la crittografia confermando che non vi siano messaggi di errore in Zabbix frontend o nel file di log di Zabbix server:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 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 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 cronologia, trend e log di audit 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:

- Sovrascrivi il periodo della cronologia degli item
- Sovrascrivi il periodo dei trend degli item

Per utilizzare l'housekeeping partizionato per cronologia e trend, entrambe queste opzioni devono essere abilitate. È anche possibile abilitare la sovrascrittura singolarmente, solo per la cronologia oppure solo per i 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 di 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.

Si noti 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 acquisire familiarità con la compressione nella documentazione di TimescaleDB prima di utilizzare la compressione.

Si noti che esistono alcune limitazioni imposte dalla compressione, in particolare:

- Non sono consentite modifiche ai chunk compressi (insert, delete, update)
- Non sono consentite modifiche allo schema per le tabelle compresse.

Le impostazioni di compressione possono essere modificate nel blocco *Compressione di cronologia, trend e log di audit* nella sezione *Amministrazione* > *Housekeeping* del frontend di Zabbix.

Parametro	Predefinito	Commenti
Abilita compressione	Abilitata	Selezionare o deselezionare la casella di controllo non attiva/disattiva immediatamente la compressione. Poiché la compressione è gestita da Housekeeper, le modifiche avranno effetto entro un massimo di 2 volte `HousekeepingFrequency` ore (impostato in zabbix_server.conf) Dopo aver disabilitato la compressione, i nuovi chunk che rientrano nel periodo di compressione non verranno compressi. Tuttavia, tutti i dati precedentemente compressi rimarranno compressi. Per decomprimere i chunk compressi in precedenza, 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 tardivi (ad esempio dati ritardati da un proxy) più vecchi di questo valore verranno scartati.

Per ottenere prestazioni migliori nell'aggiornamento dei trend, è possibile ridurre "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 dei chunk rimanga entro le risorse disponibili del sistema.

8 Configurazione di Elasticsearch

Zabbix può memorizzare i dati di cronologia in Elasticsearch come alternativa a un database relazionale.

Attention:
Il supporto di Elasticsearch è attualmente sperimentale.

Questa guida descrive la configurazione per Elasticsearch 7.X. Se si utilizza una versione diversa, alcune funzionalità potrebbero non funzionare come previsto.

La configurazione prevede la creazione di una posizione di archiviazione Elasticsearch per ciascun tipo di valore, l'impostazione del preprocessing (se necessario) e la connessione di Zabbix alla propria istanza Elasticsearch.

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 (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 importanti

- Elasticsearch richiede libcurl. Per i dettagli, vedere i requisiti.
- L'housekeeper non elimina i dati da Elasticsearch.
- Se tutti i dati di history sono archiviati in Elasticsearch, i trend **non** vengono calcolati né archiviati nel database. Valutare l'estensione del periodo di archiviazione della history.
- Quando viene utilizzato Elasticsearch, le query di intervallo che recuperano valori dal database sono limitate dal timestamp del periodo di archiviazione dei dati.
- Elasticsearch non è supportato per Zabbix proxy; utilizzare invece SQLite.

Se Elasticsearch non è ancora installato, fare riferimento alla 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}
```

Memorizzazione della cronologia in indici basati sulla data

Invece di scrivere tutti i dati della cronologia in un singolo 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 rende più semplice gestire il volume dei dati e la conservazione nel tempo.

Per abilitare questa funzionalità, è necessario:

- Creare un template di indice per ogni tipo di valore che si desidera memorizzare: questo indica a Elasticsearch quale mapping applicare quando crea automaticamente un nuovo indice basato sulla data.
- Creare una pipeline di ingest per ogni tipo di valore: elabora ogni valore in ingresso e lo instrada verso il corretto indice basato sulla data.
- Configurare il parametro `HistoryStorageDateIndex` nel file di configurazione del server Zabbix: questo abilita la memorizzazione dei valori in più indici basati 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 i seguenti parametri:

- `HistoryStorageURL` - l'URL della propria istanza Elasticsearch.
- `HistoryStorageTypes` - elenco separato da virgole dei tipi di valore da memorizzare in Elasticsearch.

Ad esempio, per memorizzare in Elasticsearch i valori di tipo *Character*, *Log*, *Text* e *JSON* (mantenendo i valori *Numeric* in un database):

```
HistoryStorageURL=http://localhost:9200
HistoryStorageTypes=str,log,text,json
```

Se si utilizzano indici basati sulla data per **tutti** i valori memorizzati in Elasticsearch, impostare anche il parametro `HistoryStorageDateInde`

```
HistoryStorageDateIndex=1
```

Dopo aver apportato le modifiche, riavviare Zabbix server:

```
systemctl restart zabbix-server
```

Configurazione del frontend di Zabbix

Nel file di configurazione del frontend di Zabbix (`zabbix.conf.php`), dichiarare `$HISTORY` come variabile globale e impostare i valori `url` e `types` in modo che corrispondano alla configurazione del server:

```
// File di configurazione della GUI di Zabbix.
global $DB, $HISTORY;

$HISTORY['url']   = 'http://localhost:9200';
$HISTORY['types'] = ['str', 'log', 'text', 'json'];
```

Risoluzione dei problemi

I seguenti passaggi possono aiutarti a risolvere i problemi con la tua configurazione di Elasticsearch:

1. 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": {} }
```

2. Verifica che mapping, template e pipeline di ingest 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 quelle previste nella documentazione API di Elasticsearch.

3. Controlla se alcuni shard si trovano in uno stato di errore; il riavvio di Elasticsearch potrebbe risolvere il problema.
4. Verifica che la configurazione di Elasticsearch consenta l'accesso da Zabbix server e Zabbix frontend.
5. Usa il parametro di configurazione di Zabbix server LogSlowQueries per identificare le query lente.
6. Controlla i log di Elasticsearch per eventuali errori.
7. Se devi reimpostare la tua configurazione di Elasticsearch e ricominciare da capo, puoi eliminare tutti gli indici, i template e le pipeline di ingest:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

9 Note specifiche della 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
```

10 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 viene eseguito Zabbix agent tramite il file `zabbix_agentd.conf`, perché agent ignorerà questa configurazione e verrà eseguito 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 sovrascrivere l'utente e il gruppo predefiniti per Zabbix agent, eseguire:

```
systemctl edit zabbix-agent
```

Quindi, aggiungere il seguente contenuto:

```
[Service]
User=root
Group=root
```

Ricaricare i daemon e riavviare il servizio zabbix-agent:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Per **Zabbix agent** questo riabilita la funzionalità di configurazione dell'utente nel file `zabbix_agentd.conf`. Ora è necessario impostare i parametri di configurazione `User=root` e `AllowRoot=1` nel file di configurazione dell'agent.

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.

11 Zabbix agent su Microsoft Windows

Configurazione di 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 di Zabbix agent o più istanze di agent su un host Microsoft Windows. Una singola istanza può utilizzare:

- il file di configurazione predefinito, situato nella stessa directory del file binario dell'agent;
- un file di configurazione specificato nella riga di comando.

Nel caso di più istanze, ogni istanza di 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 i dettagli sulla configurazione di Zabbix Windows agent, vedere le opzioni del file di configurazione.

Parametro Hostname

Per eseguire i controlli attivi su un host, Zabbix agent deve avere il nome host definito. Inoltre, il valore del nome host impostato lato agent deve corrispondere esattamente al "Nome host" configurato per l'host nel frontend.

Il valore del nome host lato agent può essere definito tramite il parametro **Hostname** o **HostnameItem** nel file di configurazione dell'agent, oppure vengono utilizzati i valori predefiniti se uno di questi parametri non è specificato.

Il valore predefinito del parametro **HostnameItem** è il valore restituito dalla chiave agent "system.hostname". In Windows, restituisce il risultato della funzione gethostname(), che interroga i provider dello spazio dei nomi per determinare il nome del 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. Quindi, in pratica, se entrambi questi parametri non sono specificati, il nome host effettivo sarà il nome NetBIOS del host; Zabbix agent utilizzerà il nome host NetBIOS per recuperare l'elenco dei controlli attivi da Zabbix server 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, che è limitato a 15 caratteri ed è solo in MAIUSCOLO;
- *host* - con distinzione tra maiuscole e minuscole, restituisce il nome host Windows completo e reale (senza dominio);
- *shorthost* - restituisce la parte del nome host prima del primo punto. Restituirà la stringa completa se il nome non contiene un punto.
- *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) - usa le maiuscole/minuscole originali;
- *lower* - converte il testo in minuscolo.

Quindi, per semplificare la configurazione del file zabbix_agentd.conf e renderla uniforme, si possono utilizzare tre diversi approcci:

1. Lasciare non definiti i parametri **Hostname** o **HostnameItem** e Zabbix agent utilizzerà il nome host NetBIOS come nome host.
2. Lasciare non definito il parametro **Hostname** e definire **HostnameItem** in questo modo:
   HostnameItem=system.hostname[host] - affinché Zabbix agent utilizzi il nome host Windows completo e reale (con distinzione tra maiuscole e minuscole) come nome host
   HostnameItem=system.hostname[shorthost,lower] - affinché Zabbix agent utilizzi solo la parte del nome host prima del primo punto, convertita in minuscolo.
   HostnameItem=system.hostname[fqdn] - affinché Zabbix agent utilizzi il Fully Qualified Domain Name come nome host.

Il nome host viene utilizzato anche come parte del nome del servizio Windows, che viene usato per installare, avviare, arrestare e disinstallare il servizio Windows. Ad esempio, se il file di configurazione di Zabbix agent specifica `Hostname=Windows_db_server`, allora l'agent verrà installato come servizio Windows "`Zabbix Agent [Windows_db_server]`". Pertanto, per avere un nome di servizio Windows diverso per ogni istanza di Zabbix agent, 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, è richiesta una versione a 64 bit di Zabbix agent affinché tutti i controlli relativi ai processi a 64 bit in esecuzione funzionino correttamente.

Se si desidera utilizzare 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
```

Deve essere specificato il percorso completo del file di configurazione.

Più istanze di Zabbix agent possono essere installate 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 dell'agent

Per avviare il servizio agent, puoi usare il Pannello di controllo oppure farlo dalla 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 molteplici istanze di Zabbix agent:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Arresto dell'agent

Per arrestare il servizio agent, puoi usare il Pannello di controllo oppure farlo dalla 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 utilizzando il file di configurazione predefinito:

```
zabbix_agentd.exe --uninstall
```

Per disinstallare una singola istanza di Zabbix agent utilizzando 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

Zabbix agent 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.

12 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.

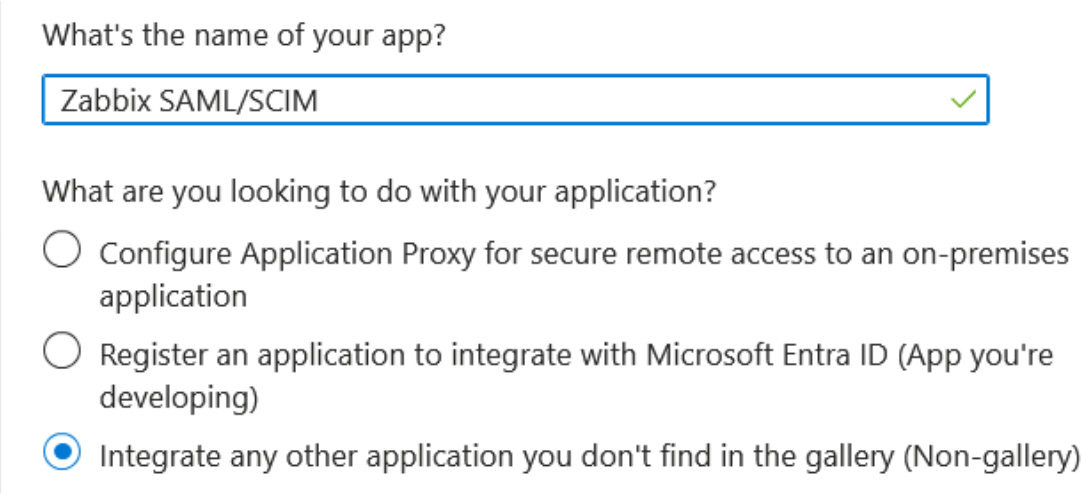
Configurazione di Microsoft Entra ID

Creazione dell'applicazione

1. Accedere al Microsoft Entra admin center in Microsoft Entra ID. A scopo di test, è possibile creare un account di prova gratuito in Microsoft Entra ID.

2. Nel Microsoft Entra admin center selezionare *Applications -> Enterprise applications -> New application -> Create your own application*.

3. Aggiungere il nome dell'app e selezionare l'opzione *Integrate any other application...*. Successivamente, fare clic su *Create*.

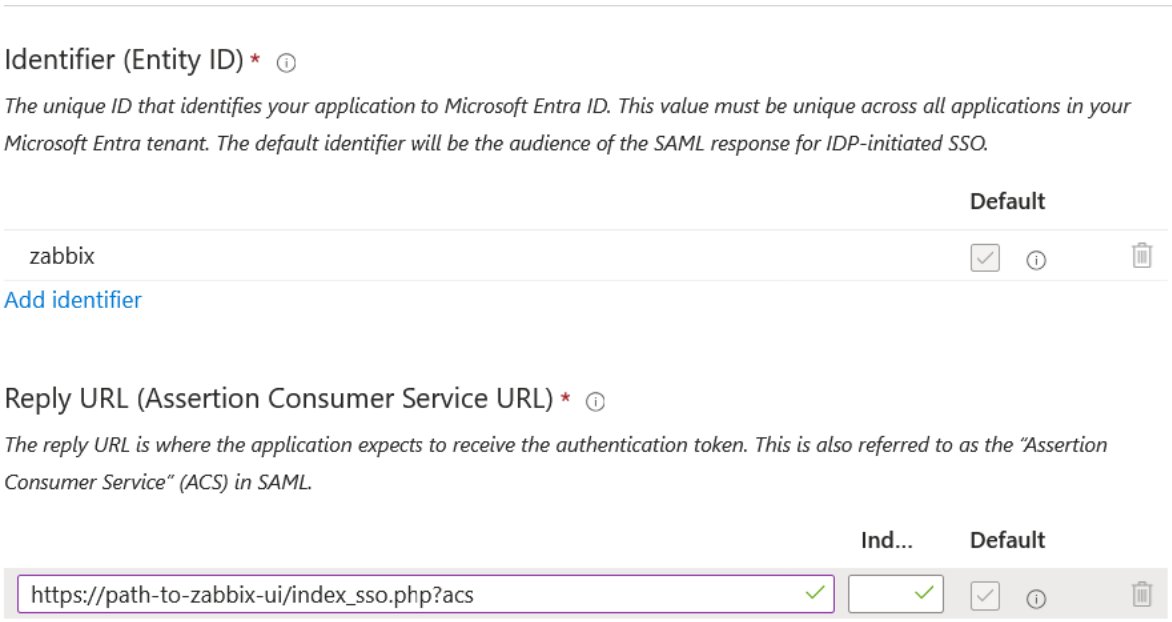


Configurazione del single sign-on

1. Nella pagina della tua applicazione, vai a *Set up single sign on* e fai clic su *Get started*. Quindi seleziona *SAML*.

2. Modifica *Basic SAML Configuration*:

   - In *Identifier (Entity ID)* imposta un nome univoco per identificare la tua app in Microsoft Entra ID, ad esempio `zabbix`;
   - In *Reply URL (Assertion Consumer Service URL)* imposta l'endpoint di single sign-on di Zabbix: `https://<path-to-zabbix-ui>/inde`



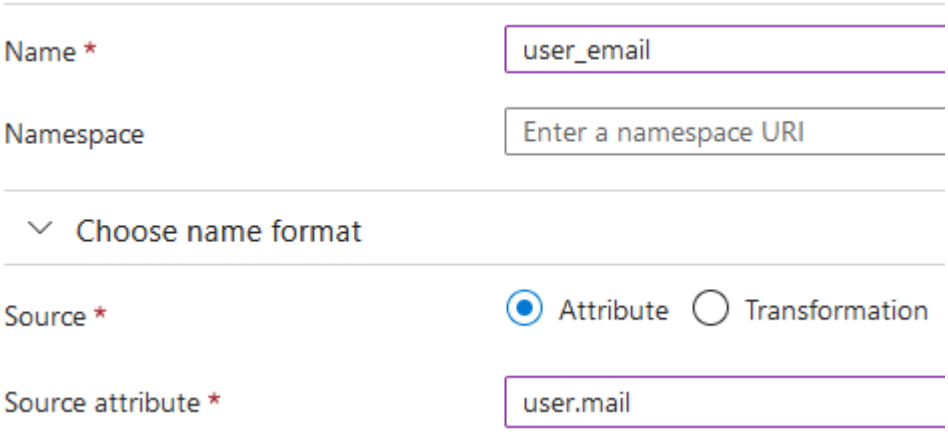
Nota che è richiesto "https". Per farlo funzionare con Zabbix, è necessario aggiungere la seguente riga a `conf/zabbix.conf.php`:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

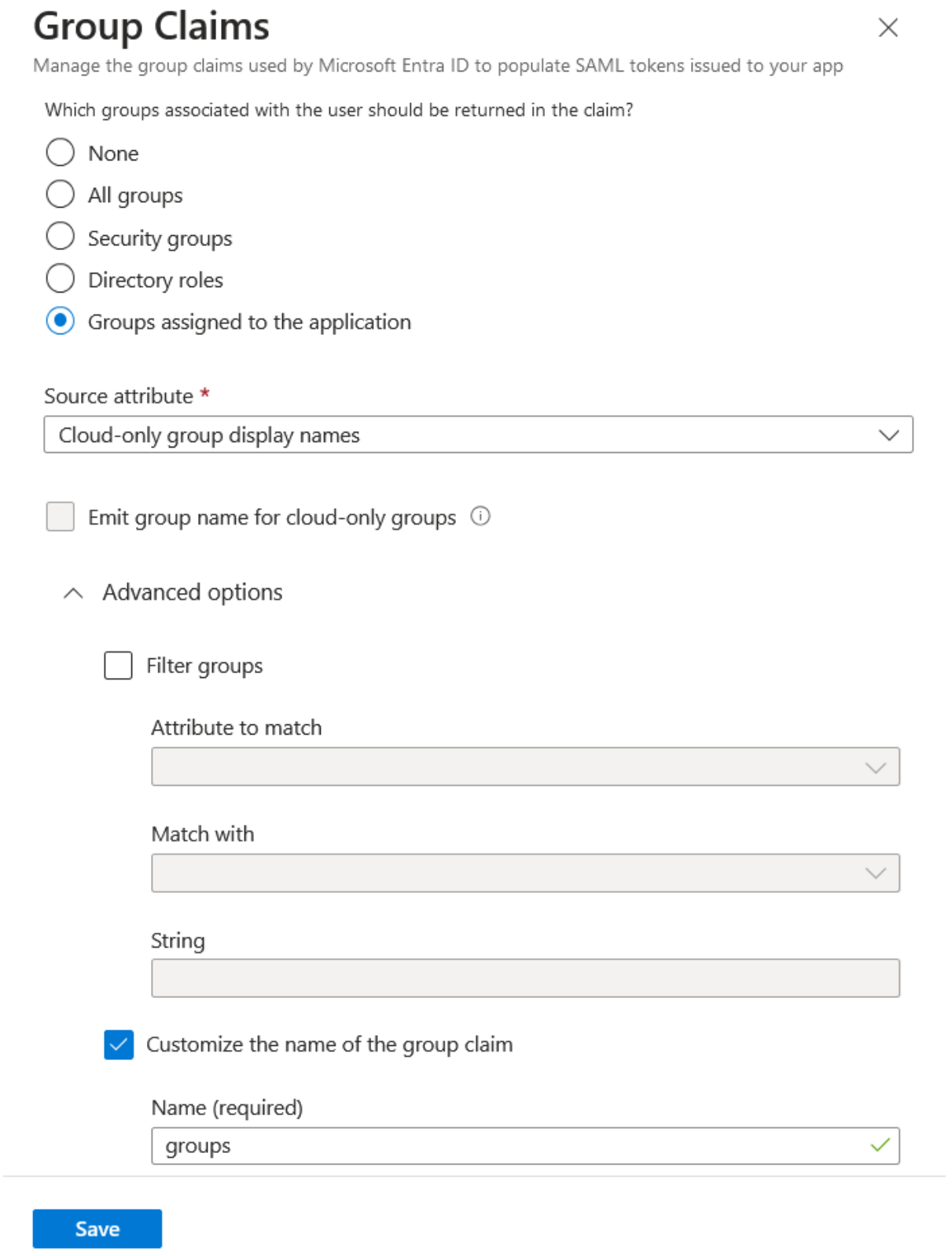
3. Modifica *Attributes & Claims*. Devi aggiungere tutti gli attributi che vuoi passare a Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

I nomi degli attributi sono arbitrari. Possono essere utilizzati nomi di attributi diversi, tuttavia è necessario che corrispondano al valore del rispettivo campo nelle impostazioni SAML di Zabbix.

- Fai clic su *Add new claim* per aggiungere un attributo:



- Fai clic su *Add a group claim* per aggiungere un attributo per passare i gruppi a Zabbix:



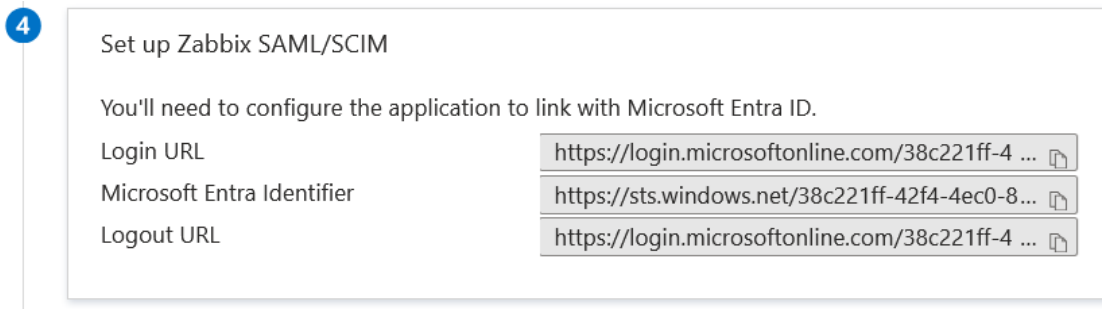
In questa claim è importante che i nomi dei gruppi (anziché gli ID dei gruppi) vengano passati a Zabbix tramite il *Source attribute* selezionato. Altrimenti il provisioning JIT degli utenti non funzionerà correttamente.

4. In *SAML Certificates* scarica il certificato Base64 fornito da Entra ID e inseriscilo in `conf/certs` dell'installazione del frontend di Zabbix.

Imposta i permessi 644 eseguendo:

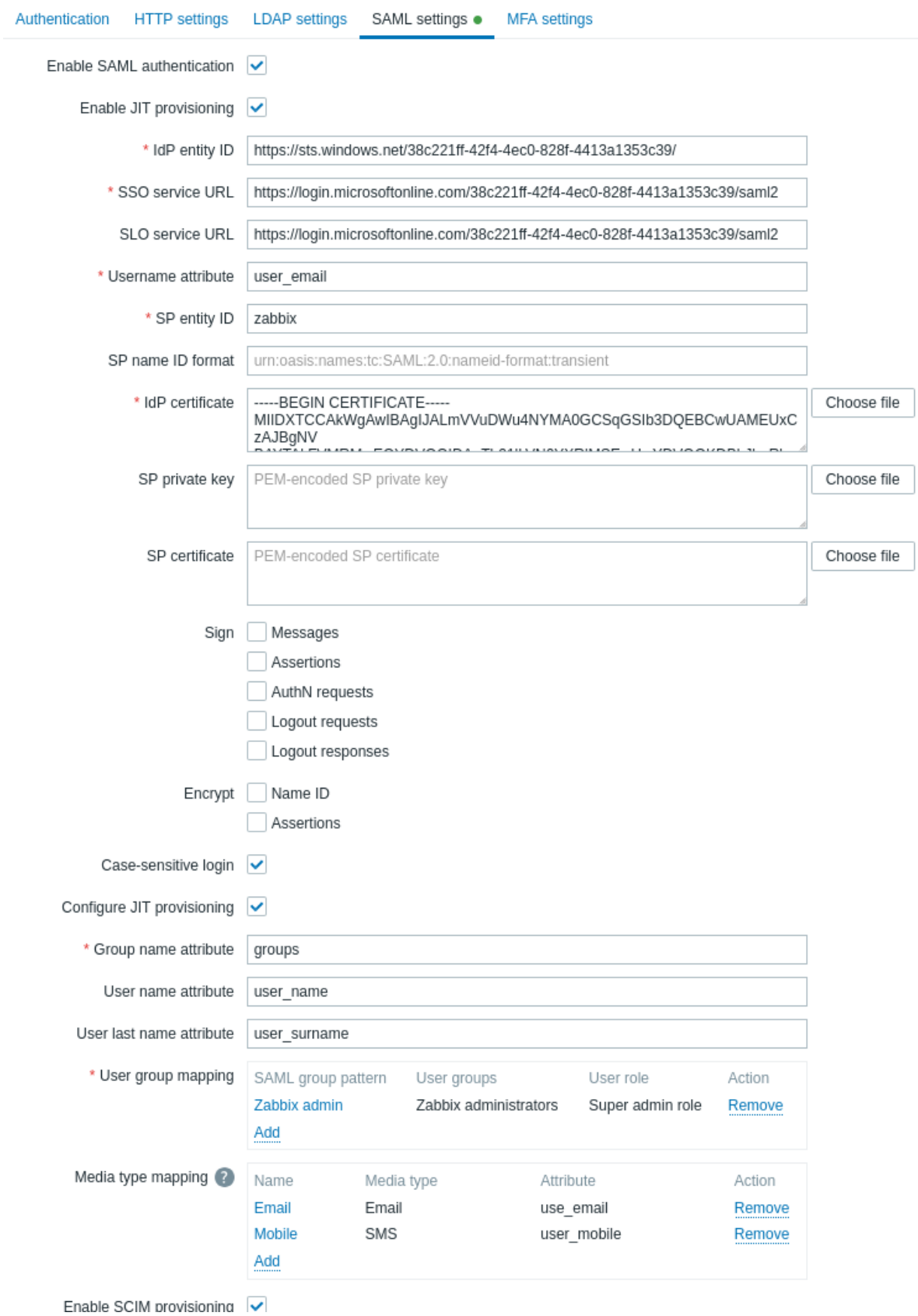
```
chmod 644 entra.cer
```

5. Usa i valori di *Set up <your app name>* in Entra ID per configurare l'autenticazione SAML di Zabbix (vedi la sezione successiva):



Configurazione di Zabbix

1. In Zabbix, andare alle impostazioni SAML e compilare le opzioni di configurazione in base alla configurazione di Entra ID:



Campo di 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 nome utente	Attributo personalizzato (claim)	`user_email`
Attributo nome gruppo	Attributo personalizzato (claim)	`groups`
Attributo nome	Attributo personalizzato (claim)	`user_name`

Campo di Zabbix	Campo di configurazione in Entra ID	Valore di esempio
Attributo cognome	Attributo personalizzato (claim)	`user_lastname`

2. Aggiungere il certificato Base64 fornito da Entra ID.

Se in 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 richiesti file nel filesystem.

Se in zabbix.conf.php è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere reso disponibile nel filesystem (per impostazione predefinita in ui/conf/certs o nel percorso configurato in zabbix.conf.php) e il frontend non memorizzerà i certificati nel database.
Si noti che se `$SSO['CERT_STORAGE']` non è impostato o è commentato, si presume l'archiviazione su file e i certificati vengono letti da ui/conf/certs.

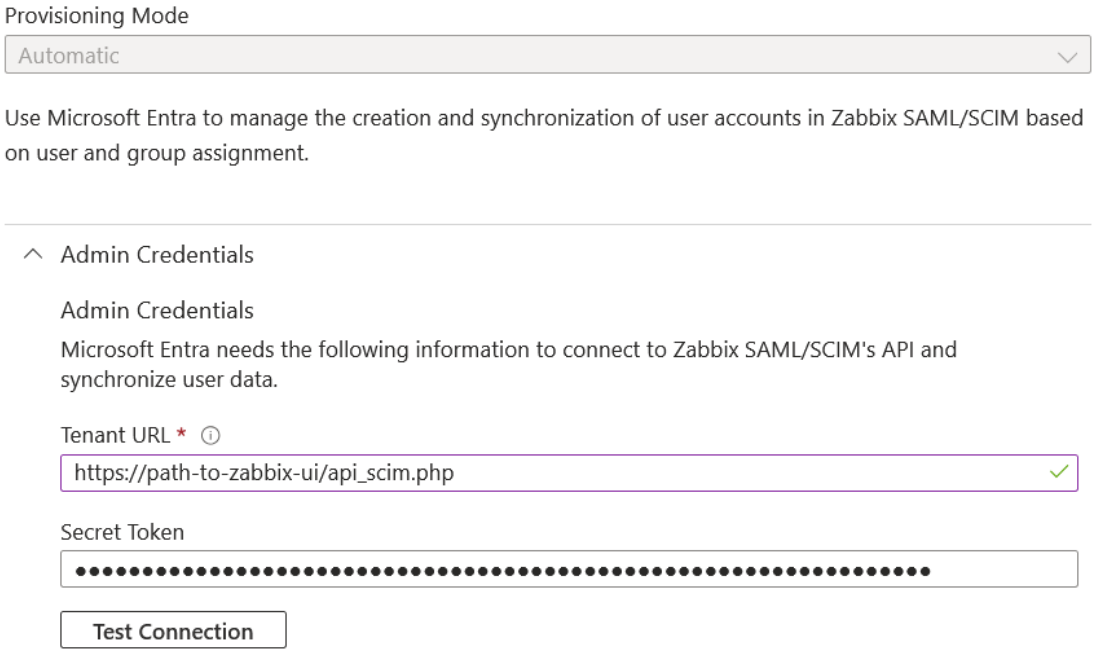
È inoltre necessario configurare la mappatura dei gruppi utenti.
La mappatura dei supporti è facoltativa.

3. Premere il pulsante *Update* per salvare queste impostazioni.

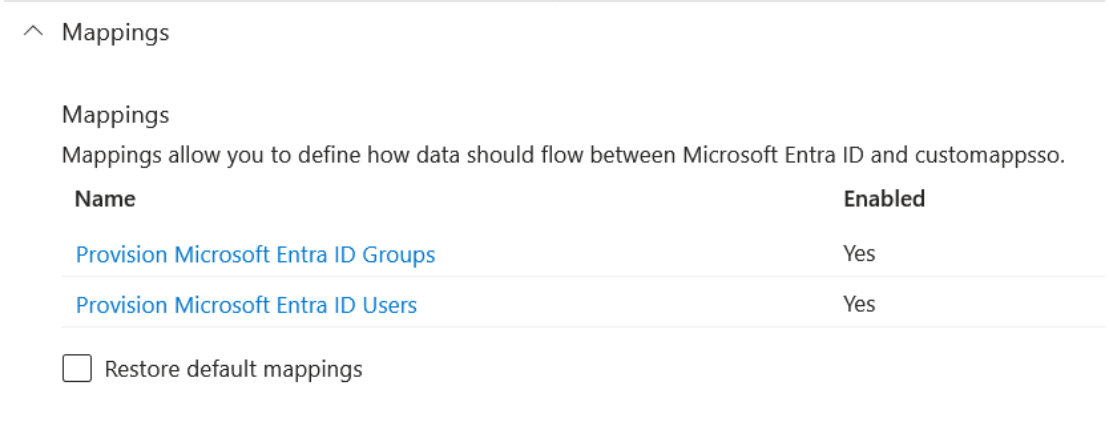
Provisioning utenti SCIM

1. Nella pagina dell'applicazione Entra ID, dal menu principale apri la pagina Provisioning. Fai clic su *Get started* e quindi seleziona la modalità di provisioning automatico:

- 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 di Super admin.
- Fai clic su *Test connection* per verificare se la connessione è stata stabilita.

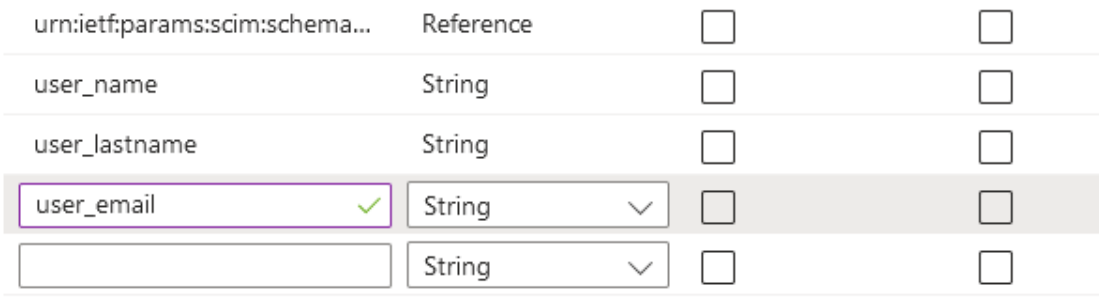


2. Ora puoi aggiungere tutti gli attributi che verranno passati con SCIM a Zabbix. Per farlo, fai clic su *Mappings* e quindi su *Provision Microsoft Entra ID Users*.



Nella parte inferiore dell'elenco Attribute Mapping, abilita *Show advanced options*, quindi fai clic su *Edit attribute list for custom-appsso*.

Nella parte inferiore dell'elenco degli attributi, aggiungi i tuoi attributi con tipo 'String':



Salva l'elenco.

3. Ora puoi aggiungere le mappature per gli attributi aggiunti. Nella parte inferiore dell'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 a *Microsoft Entra admin center* e quindi 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.

Tieni presente che la richiesta PATCH Users in Entra ID non supporta le modifiche ai media.

Firma delle richieste di autenticazione

È possibile configurare Entra ID per convalidare la firma delle richieste di autenticazione firmate.

Per farlo, creare chiavi pubblica/privata:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Assegnare i permessi:

```
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 usando 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 (*zabbix.conf.php*).

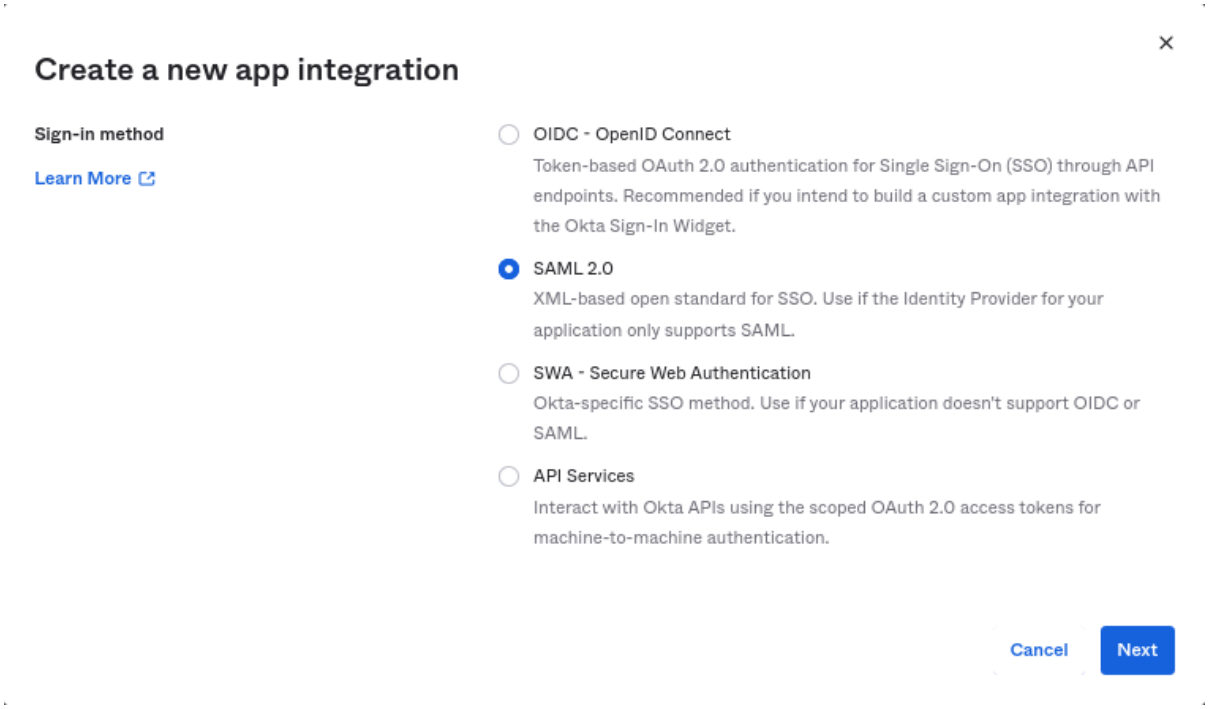
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 Configurazione di SAML con Okta

Questa sezione fornisce linee guida per configurare Okta al fine di abilitare l'autenticazione SAML 2.0 e il provisioning degli utenti per Zabbix.

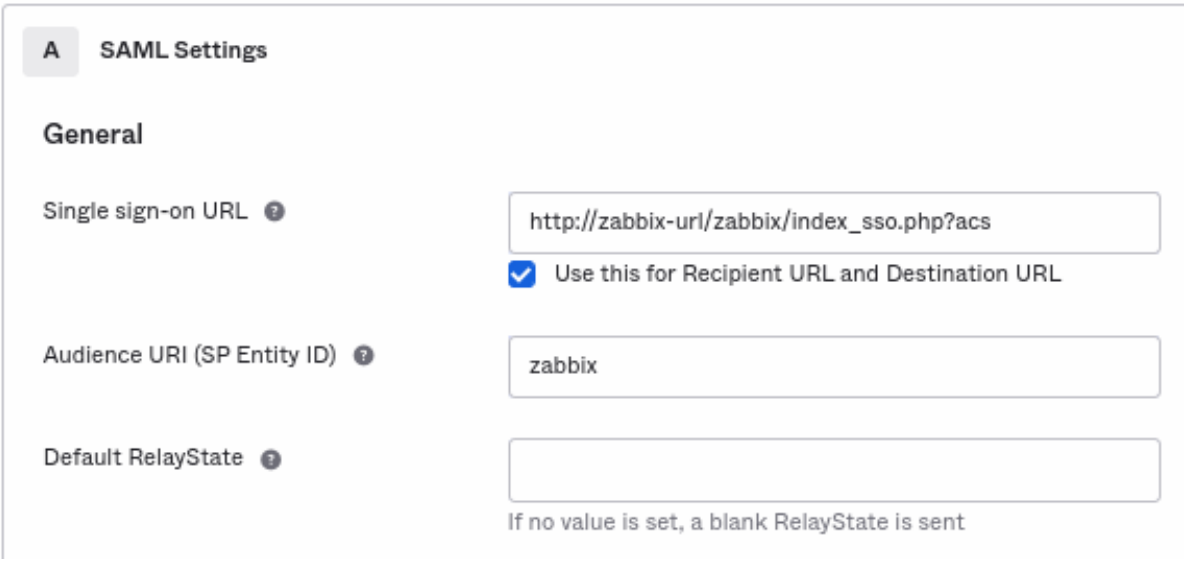
Configurazione di Okta

1. Vai a https://developer.okta.com/signup/ e registrati/accedi al tuo account.
2. Nell'interfaccia web di Okta, vai a *Applications* → *Applications*.
3. Fai clic su *Create App Integration*.

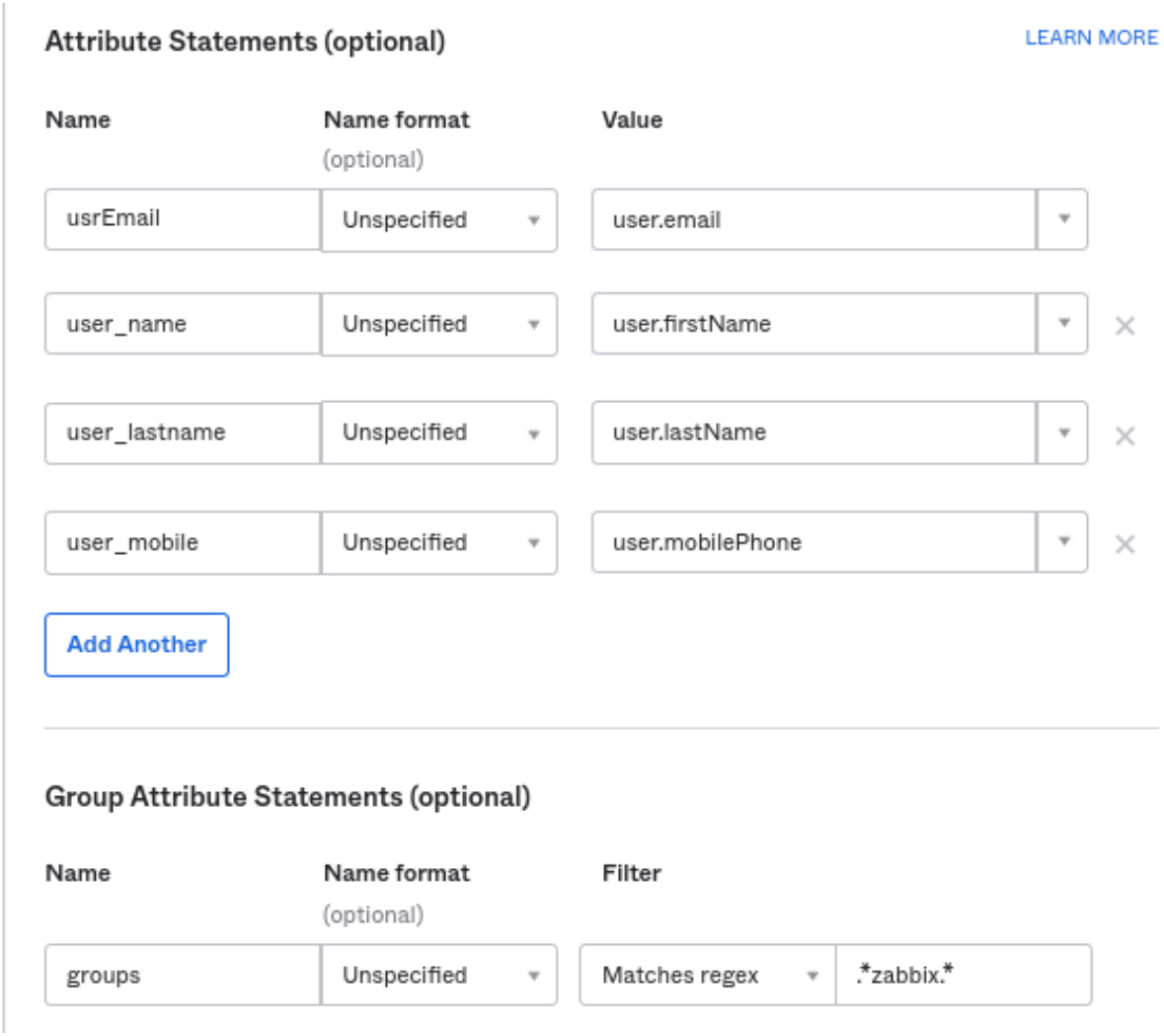


Seleziona "SAML 2.0" come metodo di accesso e fai clic su *Next*.

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`
    Nota l'uso di "http" e non di "https", in modo che il parametro `acs` non venga rimosso dalla richiesta. Anche la casella di controllo *Use this for Recipient URL and Destination URL* deve essere selezionata.
  - *Audience URI (SP Entity ID)*: `zabbix`
    Nota che questo valore verrà utilizzato 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 richiesto 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 istruzioni sugli attributi vengono inserite nelle asserzioni SAML condivise con Zabbix.

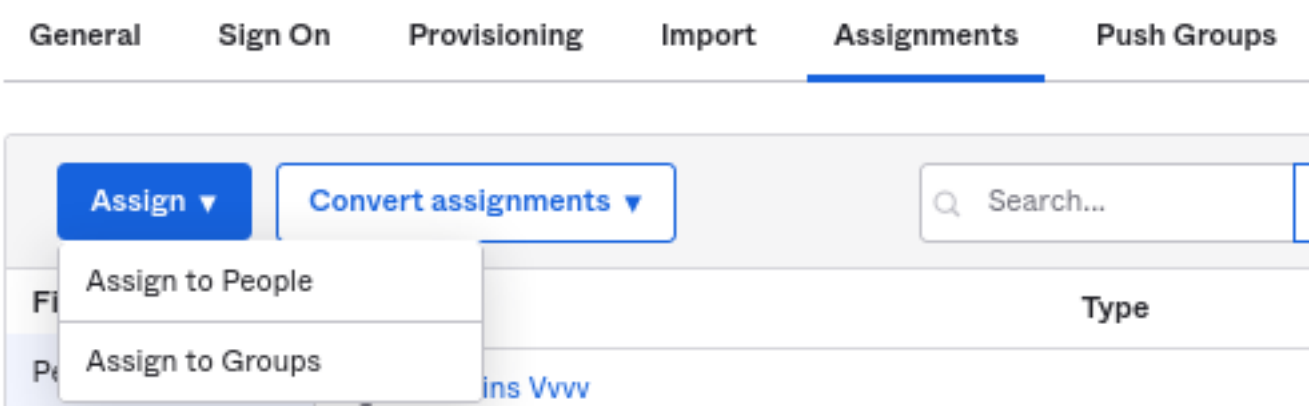
I nomi degli attributi utilizzati qui sono esempi arbitrari. Puoi usare nomi di attributi diversi, tuttavia è necessario che corrispondano al valore del rispettivo campo nelle impostazioni SAML di Zabbix.

Se desideri configurare l'accesso SAML a Zabbix *senza* provisioning utenti JIT, allora è richiesto solo l'attributo email.

Note:
Se prevedi di utilizzare una connessione crittografata, genera i certificati di crittografia privato e pubblico, quindi carica il certificato pubblico in Okta. Il modulo di 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" dell'applicazione appena creata e fai clic sul pulsante *Assign*, quindi seleziona "Assign to People" dal menu a discesa.



8. Nella finestra popup visualizzata, 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:

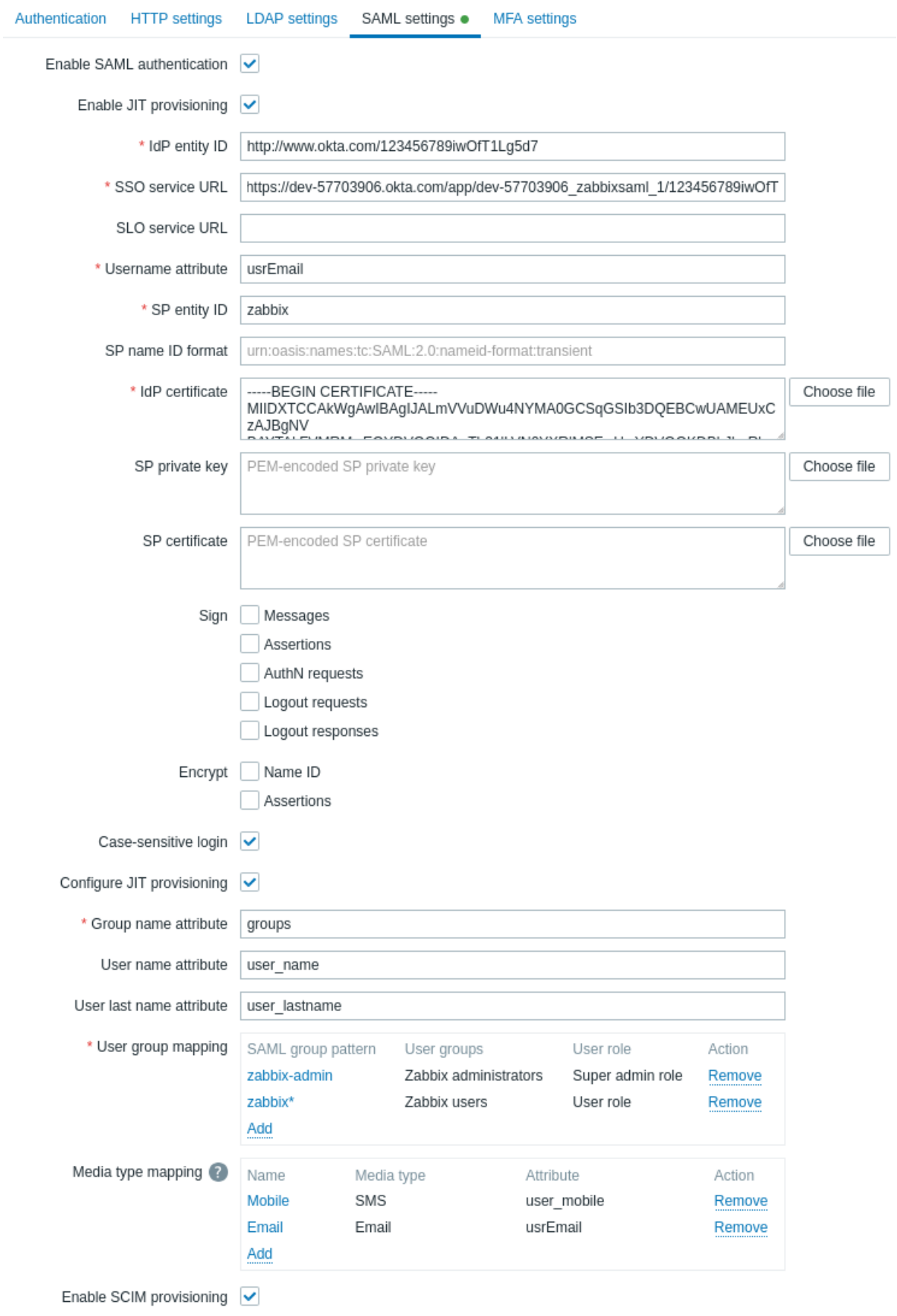
```
chmod 644 idp.crt
```

10. Vai alla scheda "Sign On" e fai clic sul pulsante *View Setup Instructions*.

Le **istruzioni** di configurazione si apriranno in una nuova scheda; tieni questa scheda aperta durante la configurazione di Zabbix.

Configurazione di Zabbix

1. In Zabbix, andare 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
ID entità IdP	Identity Provider Issuer	
URL del servizio SSO	Identity Provider Single Sign-On URL	
Attributo nome utente	Attribute name	`usrEmail`
ID entità SP	Audience URI	`zabbix`
Attributo nome gruppo	Attribute name	`groups`

Campo di Zabbix	Campo di configurazione in Okta	Valore di esempio
Attributo nome utente	Attribute name	`user_name`
Attributo cognome utente	Attribute name	`user_lastname`

È inoltre necessario configurare la mappatura dei gruppi utente e dei supporti.

2. Aggiungere il certificato Base64 fornito nelle istruzioni di configurazione SAML di Okta.

Se in 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 richiesti file nel filesystem.

Se in zabbix.conf.php è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere reso disponibile nel filesystem (per impostazione predefinita in ui/conf/certs o nel percorso configurato in zabbix.conf.php) e il frontend non memorizzerà i certificati nel database. Si noti che se `$SSO['CERT_STORAGE']` non è impostato o è commentato, viene presunto l'archiviazione su file e i certificati vengono letti da ui/conf/certs.

3. Se in Okta *Assertion Encryption* è stato impostato su "Encrypted", anche in Zabbix deve essere selezionata la casella "Assertions" del parametro *Encrypt*.

4. Premere il pulsante *Update* per salvare queste impostazioni.

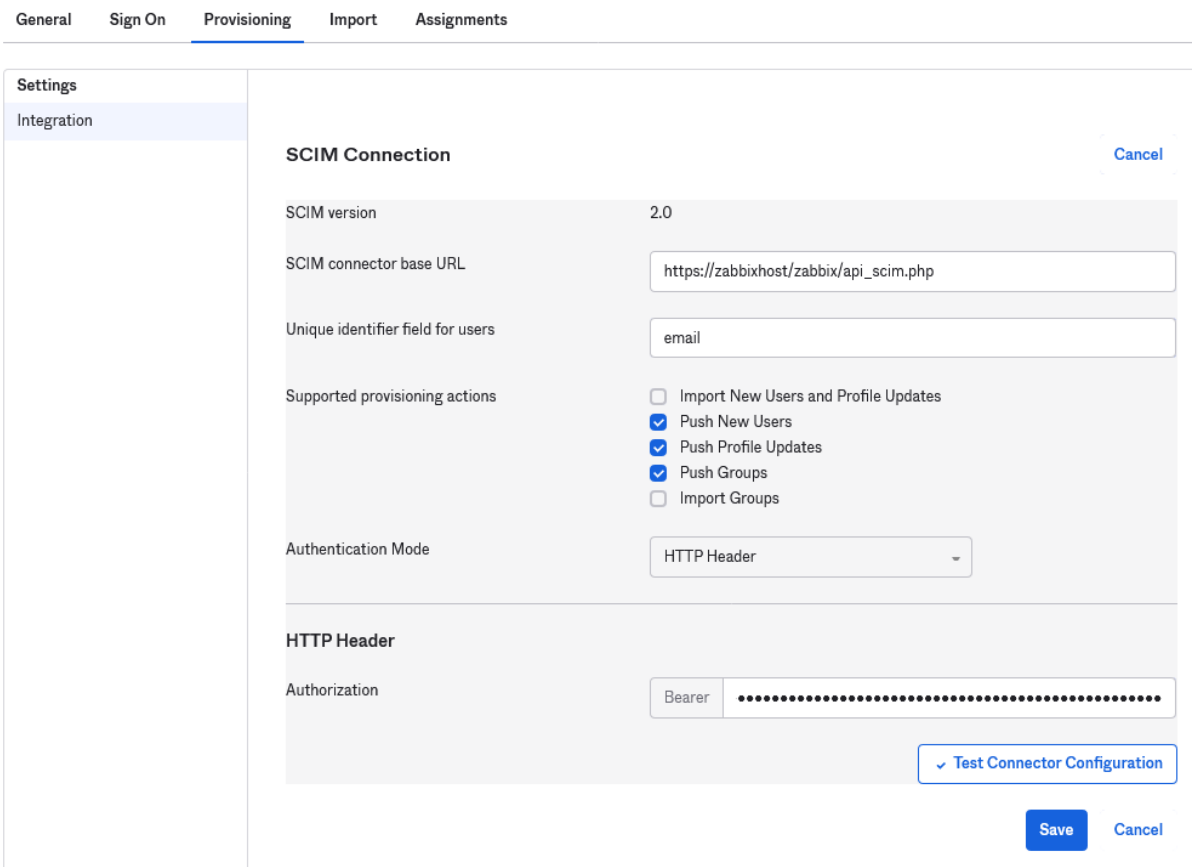
Provisioning SCIM

1. Per attivare il provisioning SCIM, andare su "General" -> "App Settings" dell'applicazione in Okta.

Selezionare la casella di controllo *Enable SCIM provisioning*. Di conseguenza, verrà visualizzata una nuova scheda *Provisioning*.

2. Andare alla scheda "Provisioning" per configurare una connessione SCIM:

- In *SCIM connector base URL* specificare il percorso del frontend di Zabbix e aggiungere `api_scim.php`, ad esempio: `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:
Se si riscontrano problemi di autenticazione, vedere Authorization header forwarding.

3. Fare clic su *Test Connector Configuration* per testare la connessione. Se tutto è corretto, verrà visualizzato un messaggio di successo.

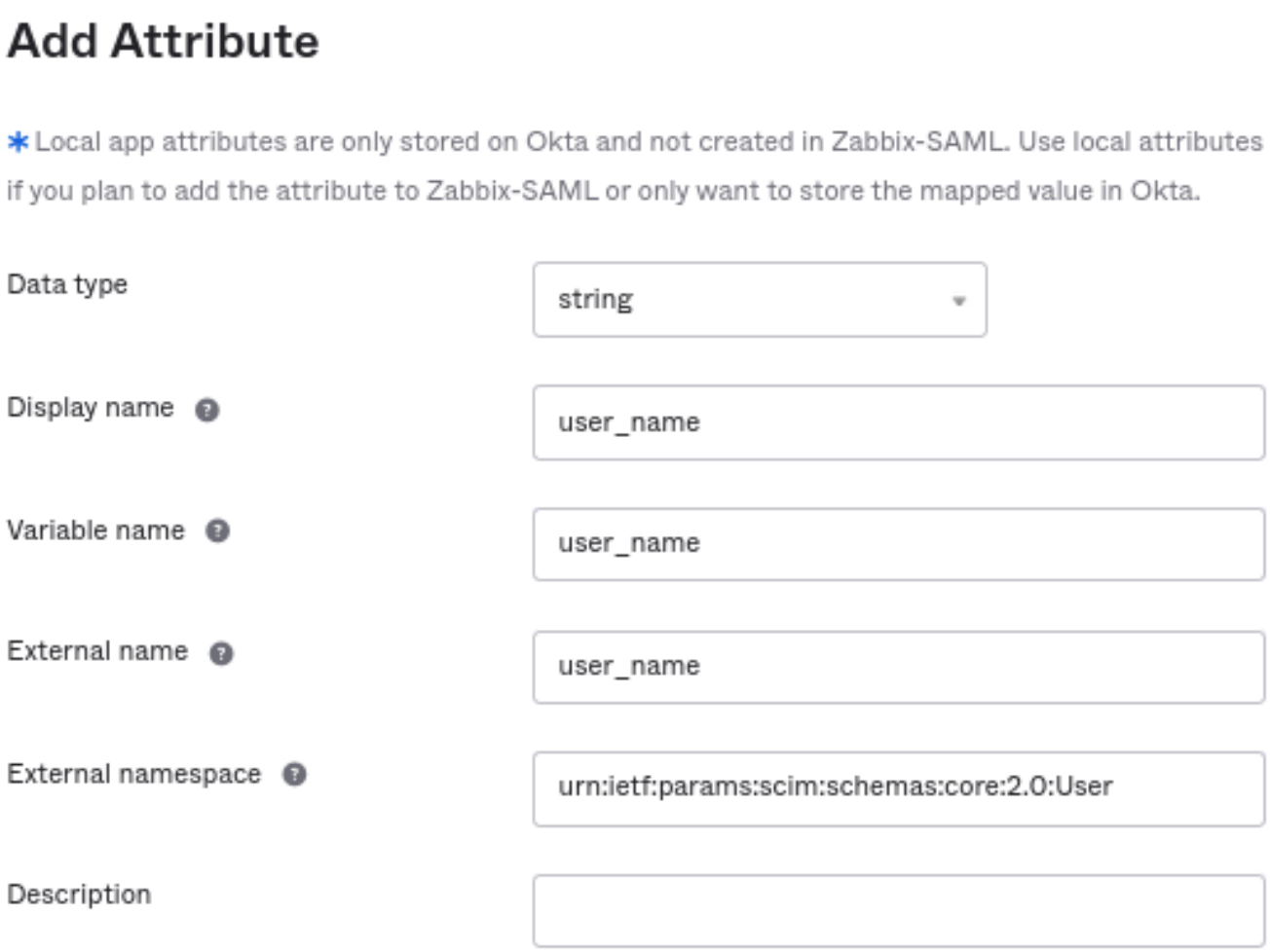
4. In ”Provisioning” -> ”To App”, assicurarsi di selezionare le seguenti caselle di controllo:

- Create Users
- Update User Attributes
- Deactivate Users

Questo garantirà che questi tipi di richiesta vengano inviati a Zabbix.

5. Assicurarsi che tutti gli attributi definiti in SAML siano definiti anche in SCIM. È possibile accedere all’editor del profilo per l’app in ”Provisioning” -> ”To App”, facendo clic su *Go to Profile Editor*.

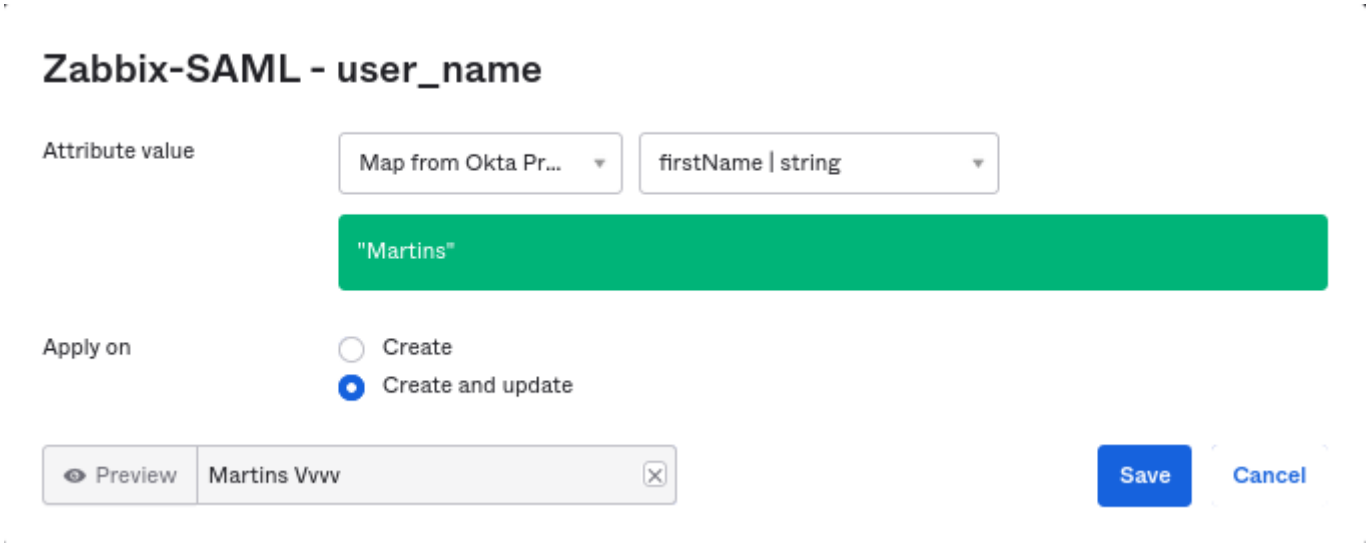
Fare clic su *Add Attribute*. Compilare i valori di *Display name*, *Variable name*, *External name* con il nome dell’attributo SAML, ad esempio `user_name`.



External namespace deve essere uguale allo schema utente: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Andare su ”Provisioning” -> ”To App” -> ”Attribute Mappings” dell’applicazione. Fare clic su *Show Unmapped Attributes* in basso. Verranno visualizzati gli attributi aggiunti di recente.

7. Mappare ciascun attributo aggiunto.



8. Aggiungere 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 gruppi nella scheda ”Push Groups”. Il modello di mappatura dei gruppi utenti nelle impostazioni SAML di Zabbix deve corrispondere a un gruppo specificato qui. Se non vi è alcuna corrispondenza, l’utente non può essere creato in Zabbix.

Le informazioni sui membri del gruppo vengono inviate ogni volta che viene apportata una modifica.

14 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 OneLogin. A scopo di test, puoi creare un account sviluppatore gratuito in OneLogin.

2. Nell’interfaccia web di OneLogin, vai a *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)*.

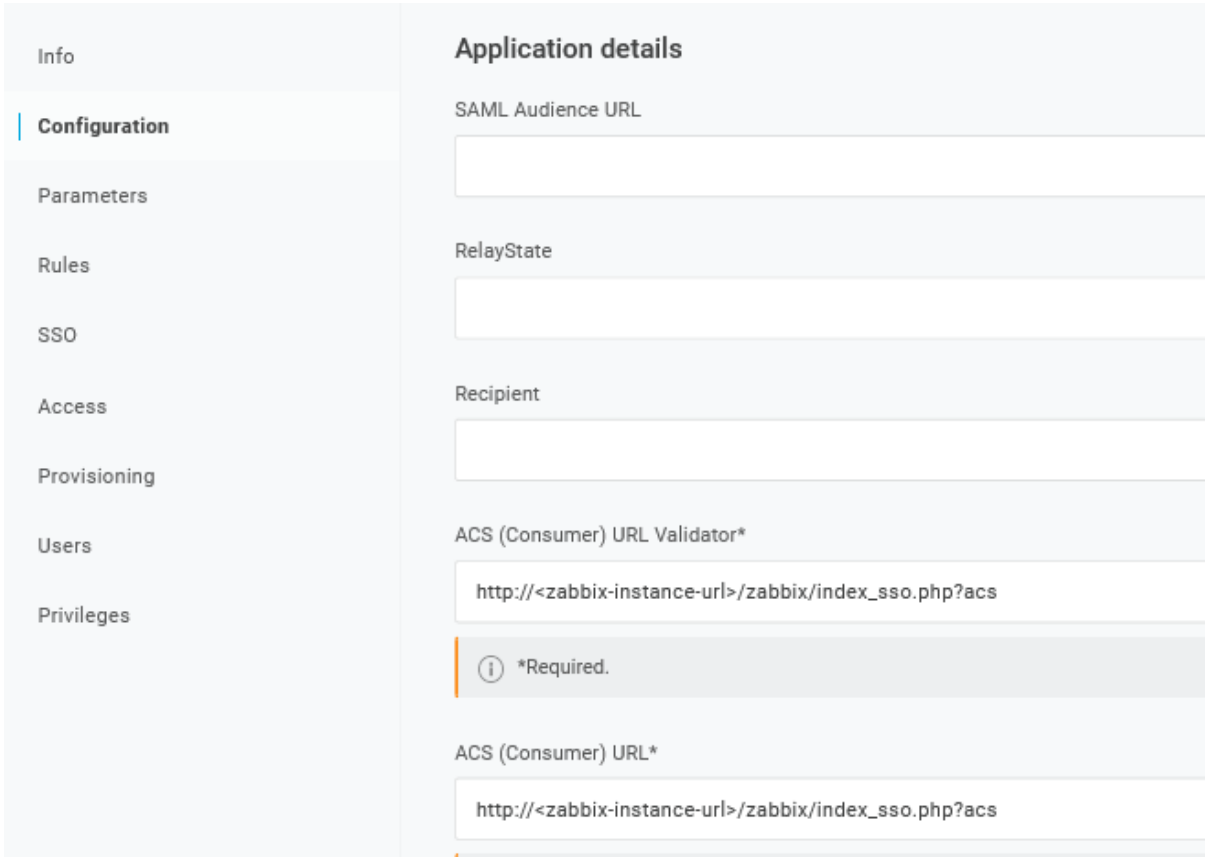
4. Per iniziare, potresti voler personalizzare il nome visualizzato della tua app. Potresti anche voler aggiungere l’icona e i dettagli dell’app. Dopodiché, fai clic su *Save*.

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”, in modo che il parametro `acs` non venga rimosso dalla richiesta.



È anche possibile usare "https". Per farlo funzionare con Zabbix, è necessario aggiungere la seguente riga a `conf/zabbix.conf.php`:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Lasciare le altre opzioni ai 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`:

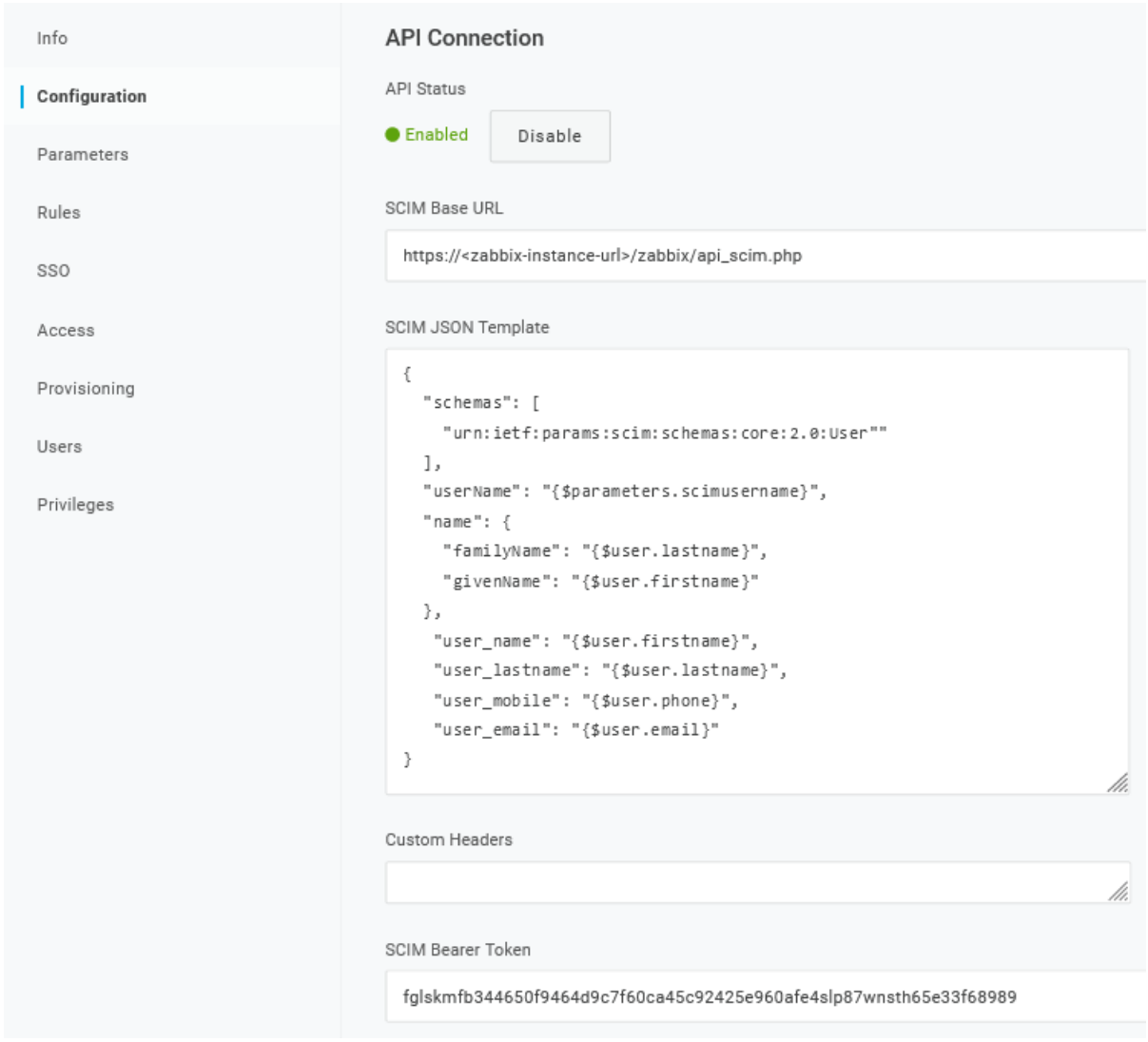
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

I nomi degli attributi sono arbitrari. Possono essere usati nomi di attributi diversi, tuttavia è necessario che corrispondano al rispettivo valore di campo nelle impostazioni SAML di Zabbix.

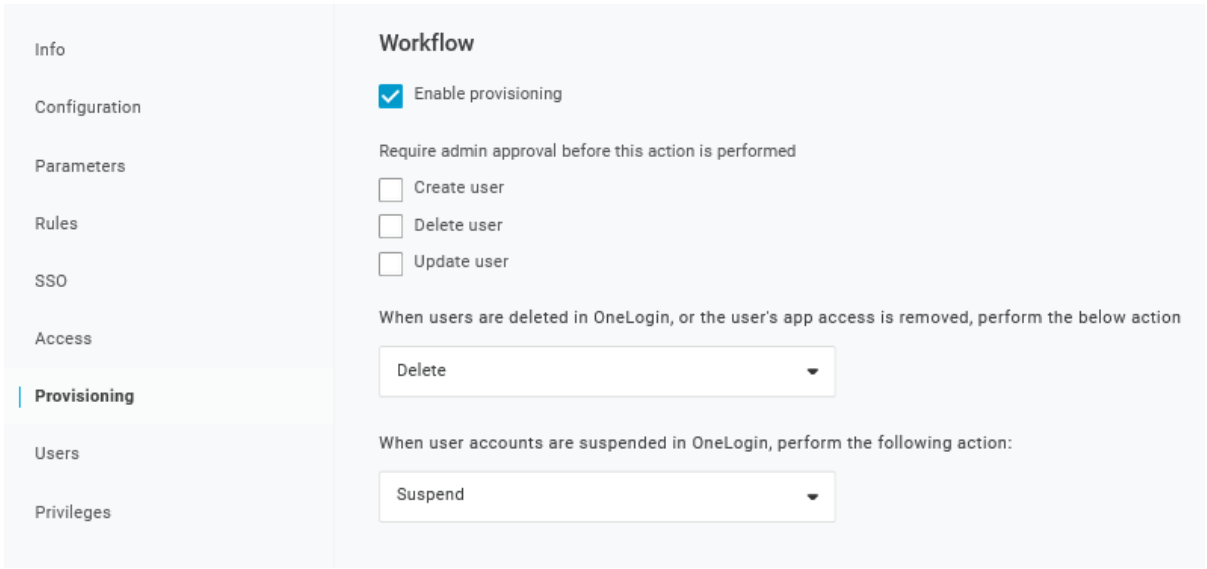
Si noti che, affinché il provisioning degli utenti funzioni, OneLogin deve ricevere in risposta un attributo 'name' con 'givenName' e 'familyName', anche se non era richiesto dal service provider. Pertanto è necessario specificarlo nello schema nella parte di configurazione dell'applicazione.

- *SCIM Bearer Token*: inserire un token API di Zabbix con permessi di Super admin.

Fare clic su *Enable* per attivare la connessione.

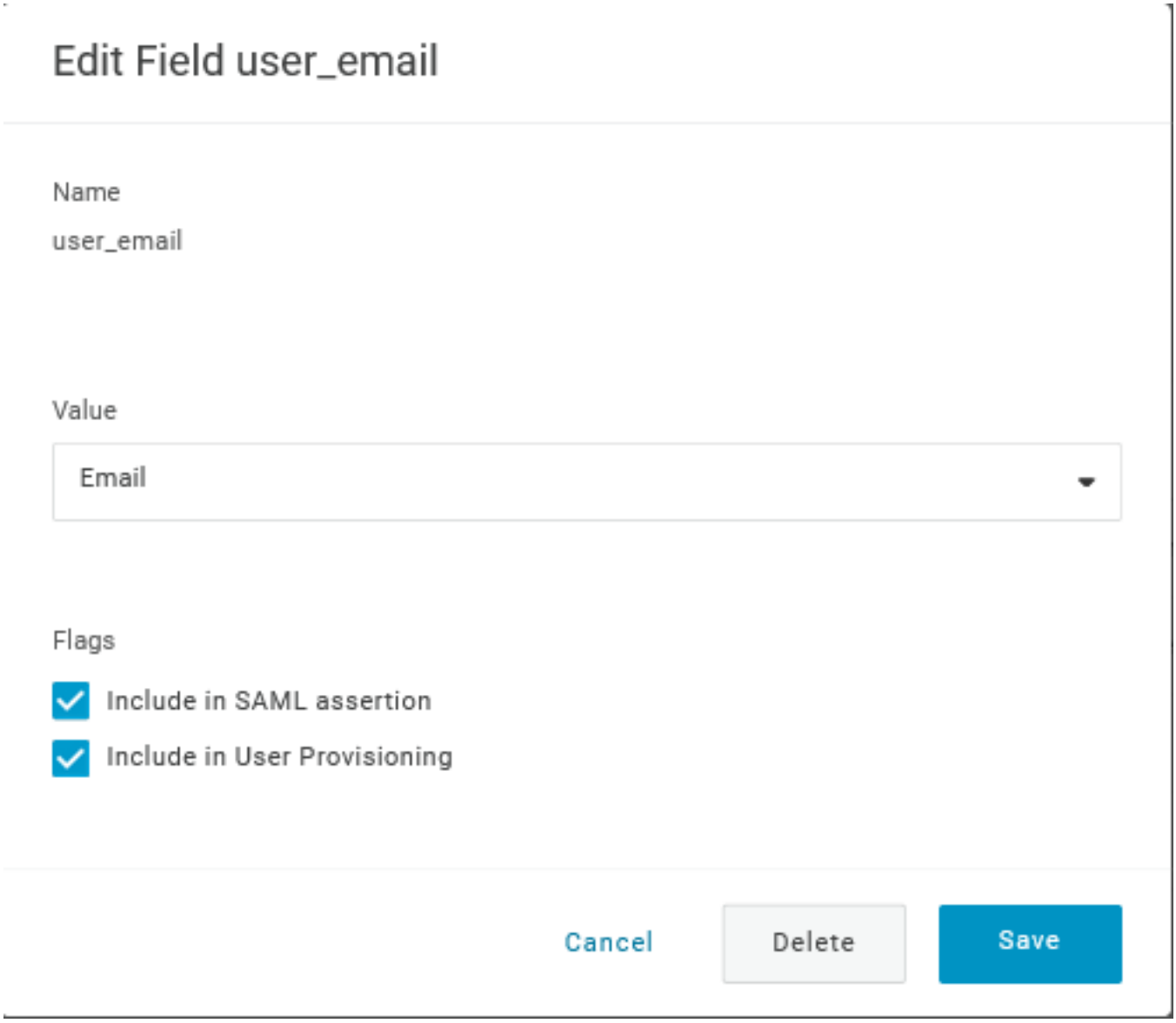


3. Nella pagina *Provisioning*, abilitare l'opzione Provisioning:



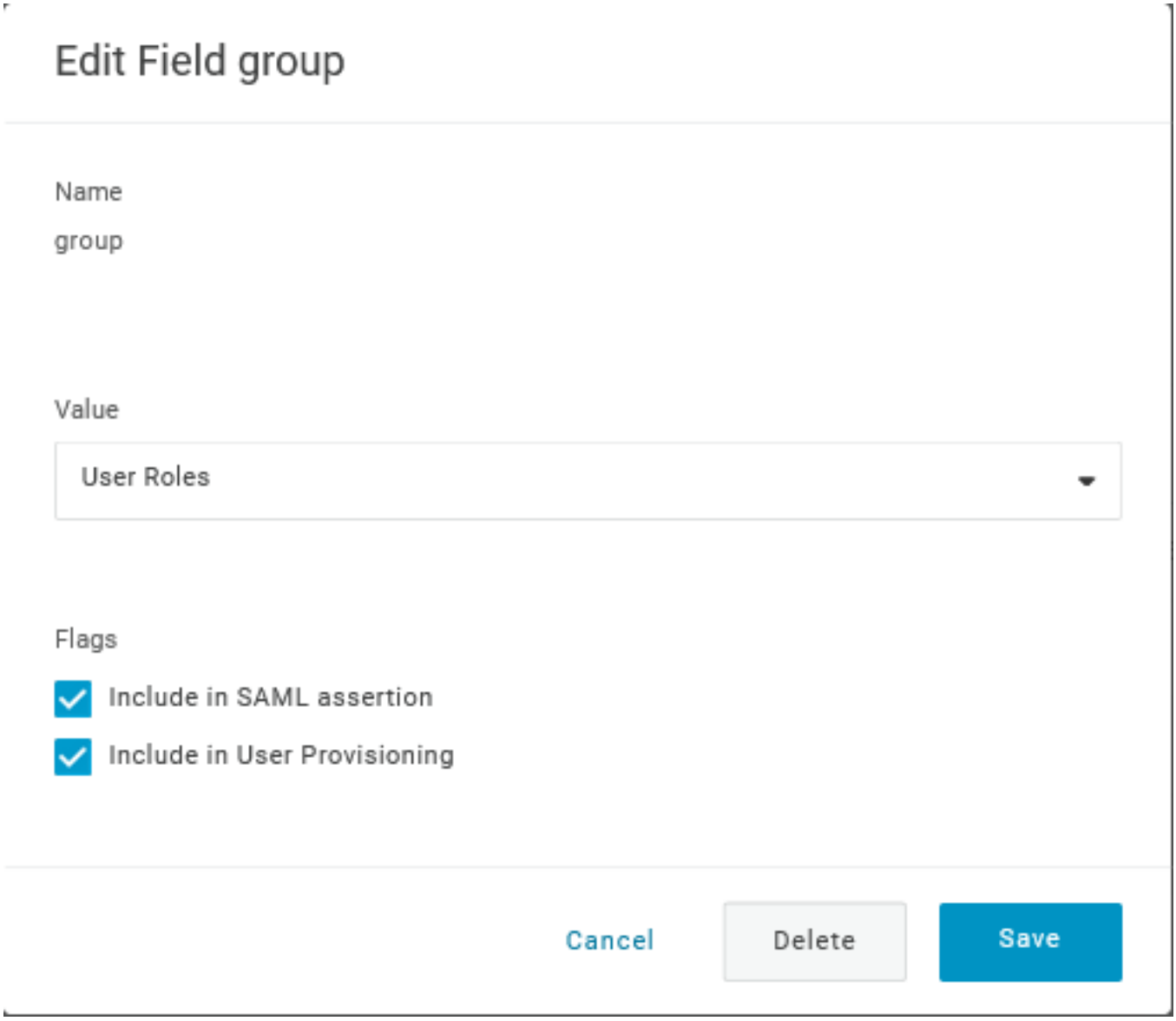
4. La pagina *Parameters* contiene un elenco di parametri predefiniti:

- Assicurarsi che 'scimusername' corrisponda al valore di login dell'utente in OneLogin (ad esempio email);
- 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`:

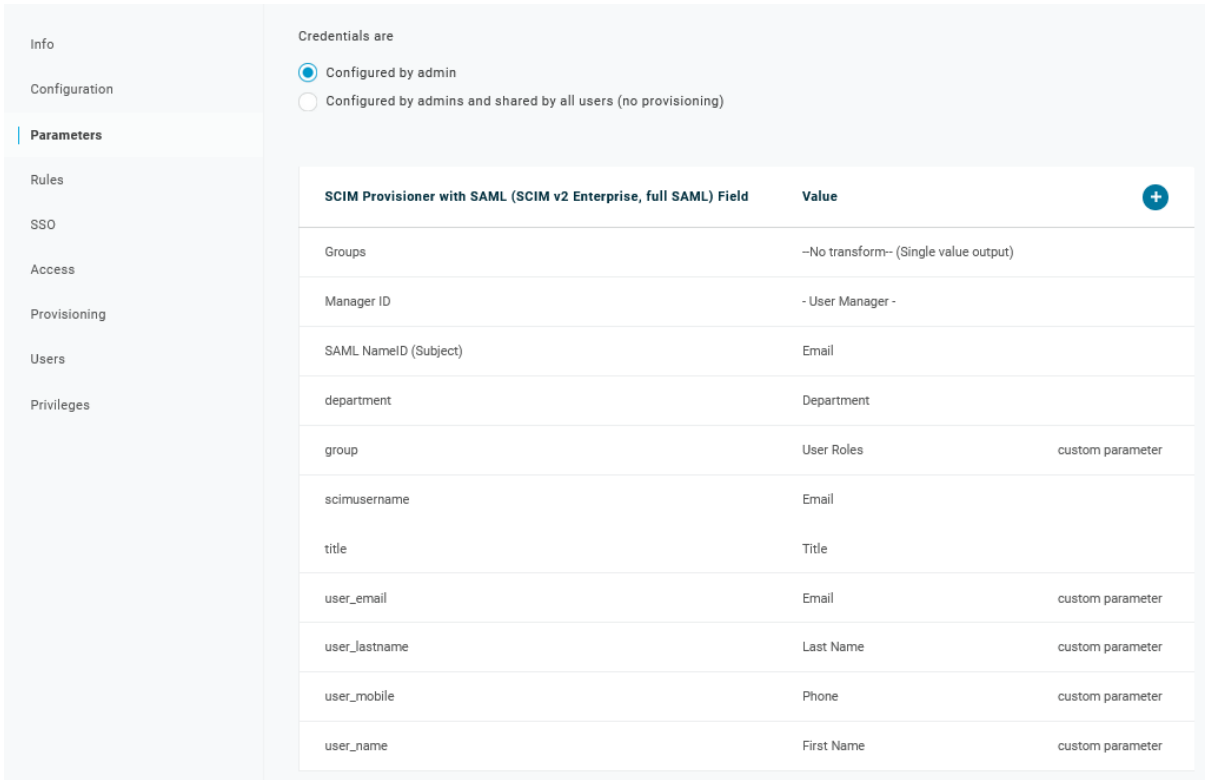


Quando si aggiunge un parametro, assicurarsi di selezionare sia l'opzione *Include in SAML assertion* sia l'opzione *Include in User Provisioning*.

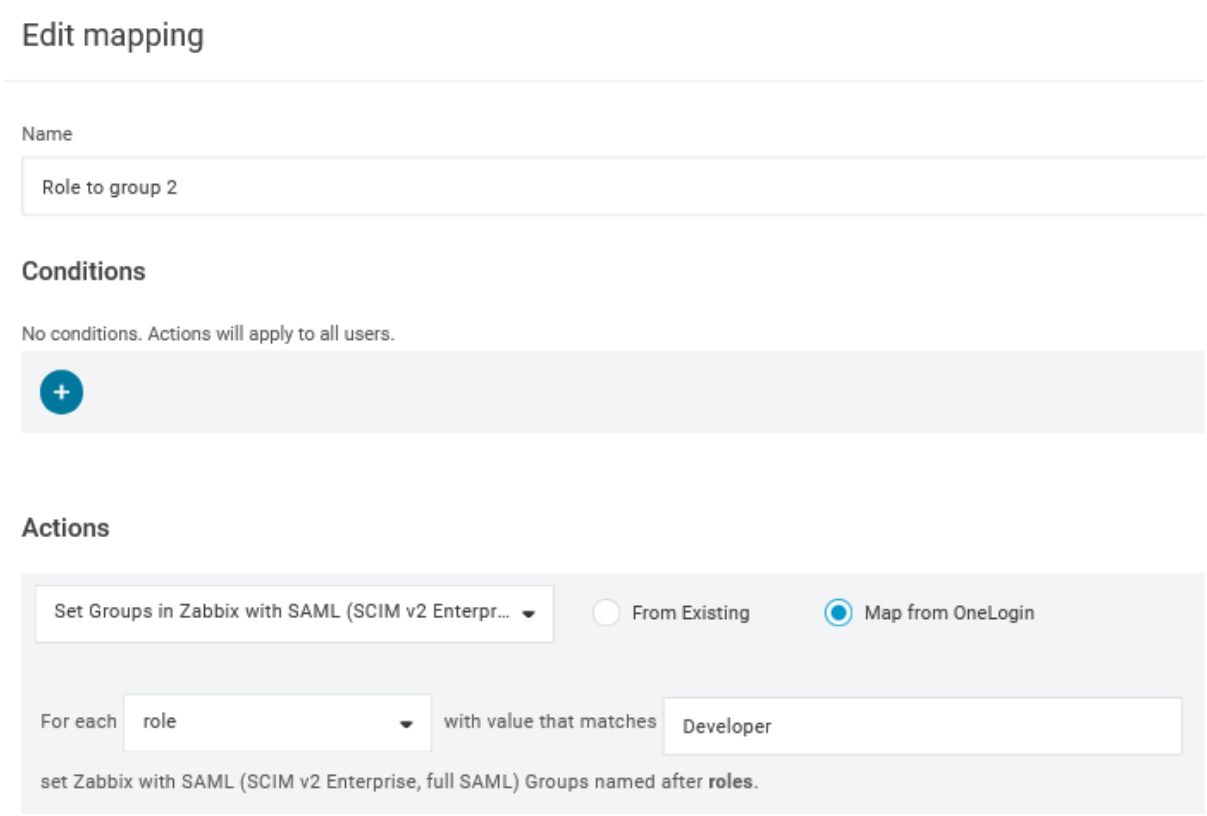
- Aggiungere un parametro 'group' che corrisponda ai ruoli utente in OneLogin. I ruoli utente verranno passati come stringa, separati da un punto e virgola `;`. I ruoli utente di OneLogin saranno utilizzati per creare gruppi utenti in Zabbix:



Verificare l’elenco dei parametri:



5. Nella pagina *Rules*, creare le mappature dei ruoli utente per il 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, andare alle impostazioni SAML e compilare le opzioni di configurazione in base alla configurazione di OneLogin:



Campo di Zabbix	Campo di configurazione in OneLogin	Valore di esempio
ID entità IdP	URL issuer (vedere la scheda SSO della propria applicazione in OneLogin)	

Campo di Zabbix	Campo di configurazione in OneLogin	Valore di esempio
URL del servizio SSO	Endpoint SAML 2.0 (HTTP) (vedere la scheda SSO della propria applicazione in OneLogin)	
URL del servizio SLO	Endpoint SLO (HTTP) (vedere la scheda SSO della propria applicazione in OneLogin)	
Attributo nome utente	Parametro personalizzato	`user_email`
Attributo nome gruppo	Parametro personalizzato	`group`
Attributo nome dell'utente	Parametro personalizzato	`user_name`
Attributo cognome dell'utente	Parametro personalizzato	`user_lastname`

È inoltre necessario configurare la mappatura dei gruppi utenti. La mappatura dei media è facoltativa. Fare clic su *Aggiorna* per salvare queste impostazioni.

2. Aggiungere il certificato Base64 fornito da OneLogin.

Se in 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 richiesti file nel filesystem.

Se in zabbix.conf.php è impostato `$SSO['CERT_STORAGE'] = 'file'`, il certificato deve essere reso disponibile nel filesystem (per impostazione predefinita in ui/conf/certs o nel percorso configurato in zabbix.conf.php) e il frontend non memorizzerà i certificati nel database. Si noti che se `$SSO['CERT_STORAGE']` non è impostato o è commentato, si presume l'archiviazione su file e i certificati vengono letti da ui/conf/certs.

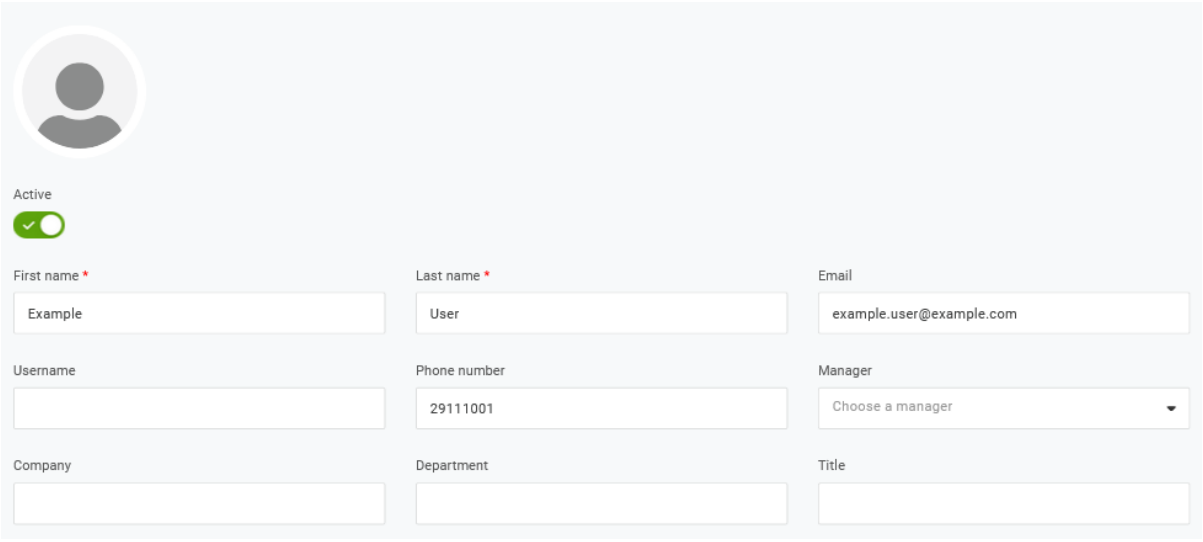
È possibile accedere al download del certificato in OneLogin in *Applications* -> *SSO* -> facendo clic su *View details* sotto il certificato corrente.

3. Premere il pulsante *Aggiorna* per salvare queste impostazioni.

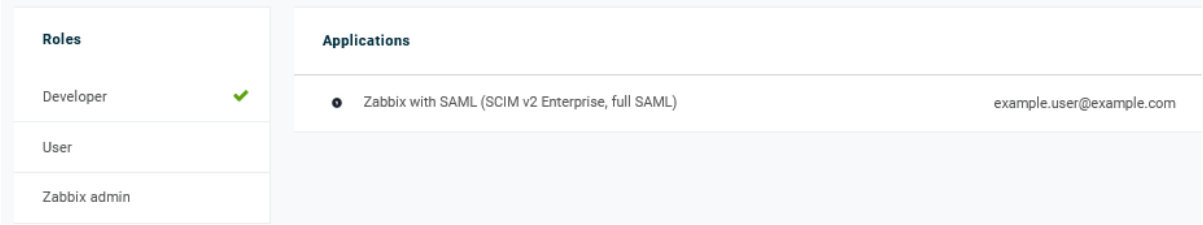
Provisioning utenti SCIM

Con il provisioning utenti abilitato, ora è possibile aggiungere/aggiornare gli utenti e i relativi ruoli in OneLogin e farne eseguire immediatamente il provisioning in Zabbix.

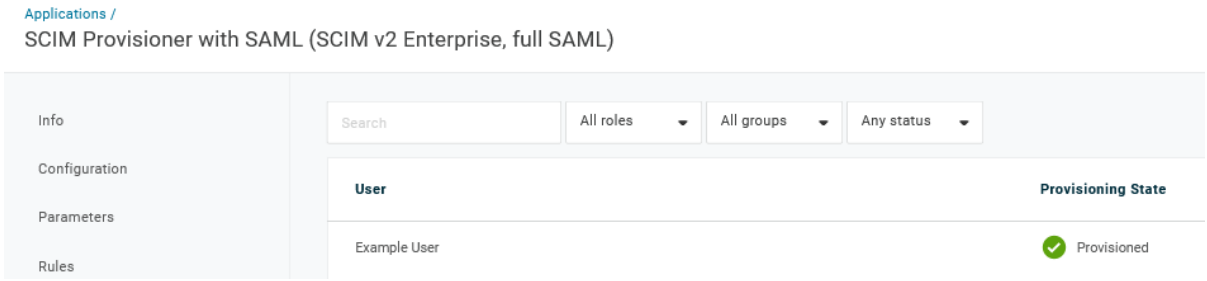
Ad esempio, è possibile creare un nuovo utente:



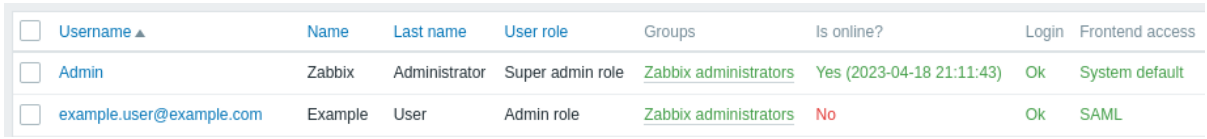
Aggiungerlo a un ruolo utente e all'applicazione che eseguirà il provisioning dell'utente:



Quando si salva l'utente, verrà eseguito il provisioning in Zabbix. In Application -> Users è possibile controllare lo stato del provisioning degli utenti correnti dell'applicazione:



Se il provisioning è stato eseguito correttamente, 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

15 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 Installazione del servizio web di Zabbix.

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 il file di configurazione del server e i parametri di configurazione del frontend siano configurati correttamente.

Zabbix server

I seguenti parametri nel file di configurazione di Zabbix server devono essere aggiornati: *WebServiceURL* e *StartReportWriters*.

WebServiceURL

Questo parametro è necessario per abilitare la comunicazione con il servizio web. L'URL deve essere nel formato `http[s]://host:port/rep`

- 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.
- È obbligatorio specificare il percorso `/report` (il percorso è codificato in modo fisso 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 comportare accessi non autorizzati.

StartReportWriters

Questo parametro determina quanti processi di scrittura 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 scrittura dei report.

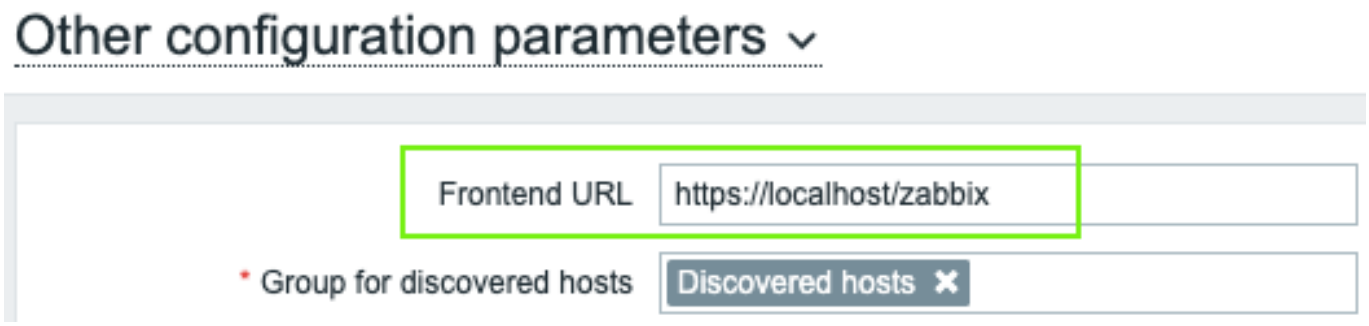
Esempio:

```
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.

16 Lingue aggiuntive del frontend

Panoramica

Per utilizzare una lingua diversa dall'inglese nell'interfaccia web di Zabbix, la relativa impostazione locale deve essere installata sul server web. Inoltre, è necessaria l'estensione PHP gettext affinché le traduzioni funzionino.

Installing locales

To list all installed languages, run:

```
locale -a
```

If some languages that are needed are not listed, open the */etc/locale.gen* file and uncomment the required locales. Since Zabbix uses UTF-8 encoding, you need to select locales with UTF-8 charset.

Now, run:

```
locale-gen
```

Restart the web server.

The locales should now be installed. It may be required to reload Zabbix frontend page in browser using Ctrl + F5 for new languages to appear.

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
```

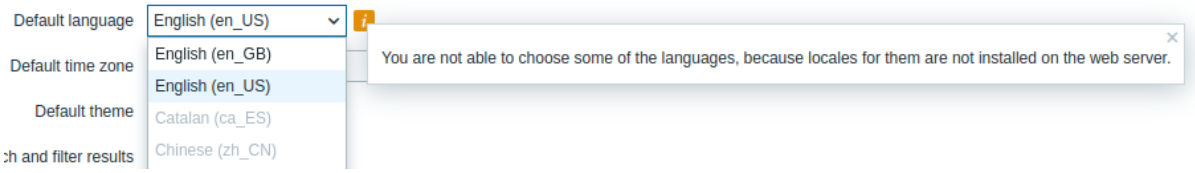
Questo passaggio non è necessario quando si installa Zabbix da pacchetti o da file sorgente tar.gz.

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 di sistema.
- Dopo l'installazione, la lingua predefinita di sistema può essere modificata nella sezione menu *Administration→General→GUI*.
- La lingua per un utente specifico 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. Accanto al selettore della lingua viene visualizzata un'icona arancione se manca almeno una locale. Premendo questa icona verrà visualizzato il seguente messaggio: "Non è possibile scegliere alcune delle lingue, perché le relative locale non sono installate sul server web."



2 Configurazione del processo

Utilizzare la barra laterale per accedere ai contenuti di questa sezione.

1 server Zabbix

Panoramica

I parametri supportati dal file di configurazione del server Zabbix (zabbix_server.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare i dettagli completi.

Parameter	Description
AlertScriptsPath	Il percorso degli script di avviso personalizzati.
AllowRoot	Consente al server di essere eseguito come 'root'.
AllowSoftwareUpdateCheck	Consente alla UI di Zabbix di ricevere informazioni sugli aggiornamenti software da zabbix.com.
AllowUnsupportedDBVersions	Consente al server di funzionare con versioni di database non supportate.
CacheSize	La dimensione della cache di configurazione.
CacheUpdateFrequency	Questo parametro determina ogni quanti secondi Zabbix eseguirà l'aggiornamento della cache di configurazione.
DBHost	Il nome host del database.
DBName	Il nome del database.
DBPassword	La password del database.
DBPort	La porta del database quando non si utilizza un socket locale.
DBSchema	Il nome dello schema del database. Utilizzato per PostgreSQL.
DBSocket	Il percorso del file socket MySQL.
DBUser	L'utente del database.
DBTLSConnect	Impostando questa opzione al valore specificato si impone l'uso di una connessione TLS al database.
DBTLSCAFile	Il percorso completo di un file contenente i certificati della CA di livello superiore per la verifica del certificato del database.
DBTLSCertFile	Il percorso completo di un file contenente il certificato del server Zabbix per l'autenticazione al database.
DBTLSKeyFile	Il percorso completo di un file contenente la chiave privata per l'autenticazione al database.
DBTLSCipher	L'elenco dei cifrari di crittografia consentiti dal server Zabbix per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.
DBTLSCipher13	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	Specifica il livello di debug.
EnableGlobalScripts	Abilita gli script globali sul server Zabbix.
ExportDir	La directory per l'esportazione in tempo reale di eventi, cronologia e trend in formato JSON delimitato da newline. Se impostato, abilita l'esportazione in tempo reale.
ExportFileSize	La dimensione massima per file di esportazione in byte.
ExportType	L'elenco di tipi di entità separati da virgole (events, history, trends) per l'esportazione in tempo reale (tutti i tipi per impostazione predefinita).
ExternalScripts	Il percorso degli script esterni.
Fping6Location	Il percorso di fping6.
FpingLocation	Il percorso di fping.
FrontendAllowedIP	Un elenco di indirizzi IP o intervalli CIDR separati da virgole autorizzati a connettersi dal frontend.
HANodeName	Il nome del nodo del cluster ad alta disponibilità.

Parameter	Description
HistoryCacheSize	La dimensione della cache della cronologia.
HistoryIndexCacheSize	La dimensione della cache dell'indice della cronologia.
HistoryStorageDateIndex	Abilita la pre-elaborazione dei valori della cronologia nell'archiviazione della cronologia per memorizzare i valori in indici diversi in base alla data.
HistoryStorageURL	L'URL HTTP[S] dell'archiviazione della cronologia.
HistoryStorageTypes	Un elenco separato da virgole di tipi di valore da inviare all'archiviazione della cronologia.
HousekeepingFrequency	Questo parametro determina ogni quante ore Zabbix eseguirà la procedura di housekeeping.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
JavaGateway	L'indirizzo IP (o hostname) di Zabbix Java gateway.
JavaGatewayPort	La porta su cui è in ascolto Zabbix Java gateway.
ListenBacklog	Il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Un elenco di indirizzi IP separati da virgole su cui il trapper deve mettersi in ascolto.
ListenPort	La porta di ascolto per il trapper.
LoadModule	Il modulo da caricare all'avvio del server.
LoadModulePath	Il percorso completo della posizione dei moduli del server.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogSlowQueries	Determina quanto tempo può impiegare una query del database prima di essere registrata nel log, in millisecondi.
LogType	Il tipo di output del log.
MaxConcurrentChecksPerPoller	Il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, poller agent o poller SNMP.
MaxHousekeeperDelete	Non verranno eliminate più di 'MaxHousekeeperDelete' righe (corrispondenti a [tablename], [field], [value]) per singola attività in un ciclo di housekeeping.
NodeAddress	L'IP o hostname con porta opzionale per sovrascrivere il modo in cui il frontend deve connettersi al server.
PidFile	Il nome del file PID.
ProblemHousekeepingFrequency	Determina con quale frequenza Zabbix eliminerà i problemi per i trigger eliminati.
ProxyConfigFrequency	Determina con quale frequenza il server Zabbix invia i dati di configurazione a un proxy Zabbix.
ProxyDataFrequency	Determina con quale frequenza il server Zabbix richiede i dati della cronologia da un proxy Zabbix.
ServiceManagerSyncFrequency	Determina con quale frequenza Zabbix sincronizzerà la configurazione di un gestore di servizi.
SMSDevices	Un elenco di file modem separati da virgole autorizzati all'uso da parte del server Zabbix.
SNMPTrapperFile	Il file temporaneo utilizzato per trasferire i dati dal demone SNMP trap al server.
SocketDir	La directory in cui memorizzare i socket IPC utilizzati dai servizi interni di Zabbix.
SourceIP	L'indirizzo IP sorgente.
SSHKeyLocation	Il percorso delle chiavi pubbliche e private per i controlli SSH e le azioni.
SSLCertLocation	Il percorso dei file di certificato client SSL per l'autenticazione del client.
SSLKeyLocation	Il percorso dei file di chiave privata SSL per l'autenticazione del client.
SSLCALocation	Sovrascrive il percorso dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server.
StartAgentPollers	Il numero di istanze pre-fork di poller asincroni di Zabbix agent.
StartAlerters	Il numero di istanze pre-fork di alerter.
StartBrowserPollers	Il numero di istanze pre-fork di poller di item del browser.
StartConnectors	Il numero di istanze pre-fork di worker connector.
StartDBSyncers	Il numero di istanze pre-fork di sincronizzatori della cronologia.
StartDiscoverers	Il numero di istanze pre-fork di worker di discovery.
StartEscalators	Il numero di istanze pre-fork di escalator.
StartHistoryPollers	Il numero di istanze pre-fork di poller della cronologia.
StartHTTPAgentPollers	Il numero di istanze pre-fork di poller HTTP agent asincroni.
StartHTTPPollers	Il numero di istanze pre-fork di poller HTTP.
StartIPMIPollers	Il numero di istanze pre-fork di poller IPMI.
StartJavaPollers	Il numero di istanze pre-fork di poller Java.
StartLLDProcessors	Il numero di istanze pre-fork di worker di low-level discovery (LLD).
StartODBCPollers	Il numero di istanze pre-fork di poller ODBC.
StartPingers	Il numero di istanze pre-fork di pinger ICMP.
StartPollersUnreachable	Il numero di istanze pre-fork di poller per host irraggiungibili (inclusi IPMI e Java).
StartPollers	Il numero di istanze pre-fork di poller.
StartPreprocessors	Il numero di istanze pre-avviate di worker di pre-elaborazione.
StartProxyPollers	Il numero di istanze pre-fork di poller per proxy passivi.
StartReportWriters	Il numero di istanze pre-fork di writer di report.

Parameter	Description
StartSNMPPollers	Il numero di istanze pre-fork di poller SNMP asincroni.
StartSNMPTrapper	Se impostato a 1, verrà avviato un processo SNMP trapper.
StartTimers	Il numero di istanze pre-fork di timer.
StartTrappers	Il numero di istanze pre-fork di trapper.
StartVMwareCollectors	Il numero di istanze pre-fork di collector VMware.
StatsAllowedIP	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o nomi DNS di istanze Zabbix esterne. La richiesta di statistiche sarà accettata solo dagli indirizzi elencati qui.
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con proxy Zabbix, agent, servizio web e per i controlli SNMP (eccetto gli item SNMP `walk[OID]` e `get[OID]`).
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 cifrate tra componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato del server o la catena di certificati, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCipherAll	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 certificato e PSK.
TLSCipherAll13	La stringa di cifratura per OpenSSL 1.1.1 o successivo in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	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 certificato.
TLSCipherCert13	La stringa di cifratura per OpenSSL 1.1.1 o successivo in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	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.
TLSCipherPSK13	La stringa di cifratura per OpenSSL 1.1.1 o successivo in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSCRLFile	Il percorso completo di un file contenente certificati revocati. Questo parametro è utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSFrontendAccept	Quali connessioni in ingresso accettare dal frontend.
TLSFrontendCertIssuer	L'emittente del certificato frontend consentito.
TLSFrontendCertSubject	Il soggetto del certificato frontend consentito.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata del server, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSListen	Controlla TLS sul socket trapper.
TmpDir	La directory temporanea.
TrapperTimeout	Specifica il timeout in secondi per: - il recupero dei dati storici da proxy Zabbix; - l'invio dei dati di configurazione a proxy Zabbix; - l'esecuzione di script globali o di comandi remoti sul server Zabbix.
TrendCacheSize	La dimensione della cache dei trend.
TrendFunctionCacheSize	La dimensione della cache delle funzioni di trend.
UnavailableDelay	Determina con quale frequenza l'host viene controllato per la disponibilità durante il periodo di indisponibilità.
UnreachableDelay	Determina con quale frequenza l'host viene controllato per la disponibilità durante il periodo di irraggiungibilità.
UnreachablePeriod	Determina dopo quanti secondi di irraggiungibilità un host viene considerato non disponibile.
User	Abbandona i privilegi a favore di uno specifico utente esistente nel sistema.
ValueCacheSize	La dimensione della cache dei valori della cronologia.
Vault	Specifica il provider del vault.
VaultDBPath	Specifica un percorso da cui recuperare le credenziali del database tramite chiavi.
VaultPrefix	Prefisso personalizzato per il percorso o la query del vault.
VaultTLSCertFile	Il nome del file del certificato SSL utilizzato per l'autenticazione del client.
VaultTLSKeyFile	Il nome del file della chiave privata SSL utilizzato per l'autenticazione del client.
VaultToken	Il token di autenticazione HashiCorp vault.
VaultURL	L'URL HTTP[S] del server vault.
VMwareCacheSize	La dimensione della memoria condivisa per memorizzare i dati VMware.
VMwareFrequency	Il ritardo in secondi tra la raccolta dei dati da un singolo servizio VMware.
VMwarePerfFrequency	Il ritardo in secondi tra il recupero delle statistiche dei contatori delle prestazioni da un singolo servizio VMware.
VMwareTimeout	Il numero massimo di secondi per cui un collector vmware attenderà una risposta dal servizio VMware.

Parameter	Description
WebDriverURL	L'URL HTTP[S] dell'interfaccia WebDriver.
WebServiceURL	URL HTTP[S] del servizio web Zabbix nel formato <host:port>/report.

Tutti i parametri sono facoltativi, salvo ove esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- 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 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

AlertScriptsPath

La posizione degli script di avviso personalizzati (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 alcun effetto se avviato con 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 di database non supportate.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

CacheSize

La dimensione della cache di configurazione, in byte. La dimensione della memoria condivisa per l'archiviazione dei dati di host, item e trigger.

Predefinito: 32M
 Intervallo: 128K-64G

CacheUpdateFrequency

Questo parametro determina con quale frequenza Zabbix eseguirà l'aggiornamento della cache di configurazione, in secondi. Vedi anche le opzioni di runtime control.

Predefinito: 10
 Intervallo: 1-3600

DBHost

Il nome host del database.
Con MySQL, `localhost` o una stringa vuota comportano l'utilizzo di un socket. Con PostgreSQL, se impostato su una stringa vuota, viene utilizzato il socket Unix-domain predefinito; se impostato su un percorso (ad esempio, /var/run/pgbouncer), viene utilizzato il socket Unix-domain specificato.

Predefinito: `localhost`

DBName

Il nome del database.

Obbligatorio: Sì

DBPassword

La password del database. Commentare questa riga se non viene utilizzata alcuna password.

DBPort

La porta del database quando non si utilizza il socket Unix-domain predefinito.[3]

Predefinito per MySQL: 3306

Predefinito per PostgreSQL: 5432

Intervallo: 1024-65535

DBSchema

Il nome dello schema del database. Utilizzato per PostgreSQL.

DBSocket

Il percorso del file socket MySQL.[3]

DBUser

L'utente del database.

DBTLSConnect

Impostando questa opzione ai seguenti valori si impone l'uso di 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 con 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 impostata alcuna opzione e il comportamento dipende dalla configurazione del database.

DBTLSCAFile

Il percorso completo di un file contenente i certificati della/e 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

Il percorso completo di un file contenente il certificato del server Zabbix per l'autenticazione al database.

DBTLSKeyFile

Il percorso completo di un file contenente la chiave privata per l'autenticazione al database.

DBTLSCipher

L'elenco dei cifrari di crittografia che il server Zabbix consente per i protocolli TLS fino alla versione TLS v1.2 inclusa. Supportato solo per MySQL.

DBTLSCipher13

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

Specifica il livello di debug:
0 - informazioni di base sull'avvio e l'arresto dei processi di Zabbix
1 - informazioni critiche;
2 - informazioni sugli errori;
3 - avvisi;
4 - per il debug (produce molte informazioni);
5 - debug esteso (produce ancora più informazioni).
Vedere 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

ExportDir

La directory per l'esportazione in tempo reale di eventi, cronologia e trend nel formato JSON delimitato da newline. Se impostata, abilita l'esportazione in tempo reale.

ExportFileSize

La dimensione massima per file di esportazione in byte. Utilizzata per la rotazione se `ExportDir` è impostato.

Predefinito: 1G
 Intervallo: 1M-1G

ExportType

L'elenco dei tipi di entità delimitati da virgole (eventi, cronologia, trend) per l'esportazione in tempo reale (tutti i tipi per impostazione predefinita). Valido solo se ExportDir è impostato.
Nota che se ExportType è specificato, ma ExportDir non lo è, allora si tratta di un errore di configurazione e il server non si avvierà.

Esempio per l'esportazione di cronologia e trend:

```
ExportType=history,trends
```

Esempio solo per l'esportazione degli eventi:

```
ExportType=events
```

ExternalScripts

La posizione degli script esterni (dipende dalla variabile di installazione in fase di compilazione `datadir`).

Predefinito: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Il percorso di fping6. Assicurati che il binario fping6 abbia root come proprietario e il flag SUID impostato. Lascia vuoto ("Fping6Location=") se la tua utility fping è in grado di elaborare indirizzi IPv6.

Predefinito: `/usr/sbin/fping6`

FpingLocation

Il percorso di fping. Assicurati che il binario fping abbia come proprietario root e il flag SUID impostato.

Predefinito: `/usr/sbin/fping`

FrontendAllowedIP

Un elenco di indirizzi IP o intervalli CIDR separati da virgole a cui è consentito connettersi dal frontend. La connessione del frontend sarà accettata solo dagli indirizzi elencati qui se questo parametro è impostato. 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

Il nome del nodo del cluster ad alta disponibilità. Se vuoto, il server funziona in modalità standalone e viene creato un nodo con nome vuoto.

HistoryCacheSize

La dimensione della cache della cronologia, in byte. La dimensione della memoria condivisa per l'archiviazione dei dati della cronologia.

Predefinito: 16M
 Intervallo: 128K-16G

HistoryIndexCacheSize

La dimensione della cache dell'indice della cronologia, in byte. La dimensione della memoria condivisa per l'indicizzazione dei dati della cronologia memorizzati nella cache della cronologia. La dimensione della cache dell'indice richiede circa 100 byte per memorizzare nella cache un item.

Predefinito: 4M
 Intervallo: 128K-16G

HistoryStorageDateIndex

Abilita la pre-elaborazione dei valori dello storico nell'archiviazione dello storico per memorizzare i valori in indici diversi in base alla data.

Predefinito: 0
 Valori: 0 - disabilita; 1 - abilita

HistoryStorageURL

L'URL HTTP[S] dello storage della cronologia. Questo parametro viene utilizzato per la configurazione di Elasticsearch.

HistoryStorageTypes

Un elenco di tipi di valore separati da virgole da inviare allo storage della cronologia. Questo parametro viene utilizzato per la configurazione di Elasticsearch.

Predefinito: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Questo parametro determina con quale frequenza Zabbix eseguirà la procedura di housekeeping, in ore. L'housekeeping consiste nella rimozione delle informazioni obsolete dal database.
Nota: Per evitare che il processo di housekeeping venga sovraccaricato (ad esempio, quando i periodi di storico e trend vengono notevolmente ridotti), in un singolo ciclo di housekeeping non vengono eliminate più di 4 volte HousekeepingFrequency ore di informazioni obsolete, per ciascun item. Pertanto, se HousekeepingFrequency è 1, per ciclo non verranno eliminate 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. Pertanto, 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 ritardo di un'ora.
È possibile disabilitare l'housekeeping automatico impostando HousekeepingFrequency a 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.
Vedere anche le opzioni di runtime control e i dettagli sulla procedura di housekeeping.

Predefinito: 1
 Intervallo: 0-24

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione. Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli. Vedere le note speciali sulle limitazioni.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

L'indirizzo IP (o hostname) dello Zabbix Java gateway. Richiesto solo se vengono avviati i Java poller.

JavaGatewayPort

La porta su cui è in ascolto Zabbix Java gateway.

Predefinito: 10052
 Intervallo: 1024-32767

ListenBacklog

Il numero massimo di connessioni in attesa 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

Un elenco di indirizzi IP separati da virgole sui quali il trapper deve restare in ascolto.
Il trapper resterà in ascolto su tutte le interfacce di rete se questo parametro non è specificato.

Predefinito: `0.0.0.0`

ListenPort

La porta di ascolto per il trapper.

Predefinito: 10051
 Intervallo: 1024-32767

LoadModule

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 '/'), allora LoadModulePath viene ignorato.
Formati:
LoadModule=<module.so>
LoadModule=<path/module.s
consentito includere più parametri LoadModule.

LoadModulePath

Il percorso completo della posizione dei moduli del server. Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Il nome del file di registro.

Obbligatorio: Sì, se LogType è impostato su *file*; altrimenti no

LogFileSize

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 ricominciato da capo.

Predefinito: 1
 Intervallo: 0-1024
 Obbligatorio: Sì, se LogType è impostato su *file*; altrimenti no

LogSlowQueries

Determina per quanto tempo una query del database può essere eseguita prima di essere registrata, in millisecondi.
0 - non registrare le query lente.
Questa opzione viene abilitata a partire da DebugLevel=3.

Predefinito: 0
 Intervallo: 0-3600000

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`

MaxConcurrentChecksPerPoller

Il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, poller agent o poller SNMP. Vedere StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Predefinito: 1000
 Intervallo: 1-1000

MaxHousekeeperDelete

La tabella "housekeeper" contiene "task" per la procedura di housekeeping nel formato: `[housekeeperid]`, `[object]`, `[objectid]`.
 Non verranno eliminate più di `MaxHousekeeperDelete` righe dalle tabelle `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` e `problem` per ogni task in un ciclo di housekeeping.
 Se impostato su 0, non viene applicato alcun limite. In questo caso è necessario sapere bene cosa si sta facendo, per non sovraccaricare il database.[2]
Questo parametro si applica solo all'eliminazione dei dati residui lasciati da item già eliminati.
 Vedere anche i dettagli sulla procedura di housekeeping.

Predefinito: 5000
 Intervallo: 0-1000000

NodeAddress

IP o hostname con porta opzionale per sovrascrivere il modo in cui il frontend deve connettersi al server.
Formato: <address>[:<port>]

Se IP o hostname non sono impostati, verrà utilizzato il valore di ListenIP. Se ListenIP non è impostato, verrà utilizzato il valore `localhost`.
Se la porta non è impostata, verrà utilizzato il valore di ListenPort. Se ListenPort non è impostato, verrà utilizzato il valore 10051.

Questa opzione può essere sovrascritta dall'indirizzo specificato nella configurazione del frontend.

Vedi anche: parametro HANodeName; Abilitazione dell'alta disponibilità.

Predefinito: `localhost:10051`

PidFile

Nome del file PID.

Predefinito: `/tmp/zabbix_server.pid`

ProblemHousekeepingFrequency

Determina con quale frequenza Zabbix eliminerà i problemi relativi ai trigger eliminati, in secondi.
 Vedere anche i dettagli sulla procedura di housekeeping.

Predefinito: 60
 Intervallo: 1-3600

ProxyConfigFrequency

Determina con quale frequenza Zabbix server invia i dati di configurazione a uno Zabbix proxy, in secondi. Utilizzato solo per i proxy in modalità passiva.

Predefinito: 10
 Intervallo: 1-604800

ProxyDataFrequency

Determina con quale frequenza il server Zabbix richiede i dati di cronologia da un proxy Zabbix, in secondi. Utilizzato solo per i proxy in modalità passiva.

Predefinito: 1
 Intervallo: 1-3600

ServiceManagerSyncFrequency

Determina con quale frequenza Zabbix sincronizzerà la configurazione di un service manager, in secondi.

Predefinito: 60
 Intervallo: 1-3600

SMSDevices

Un elenco di file modem separati da virgole autorizzati all'uso da parte di Zabbix server.
L'invio di SMS non è possibile se questo parametro non è impostato.

Esempio:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

File temporaneo utilizzato per trasferire i dati dal demone SNMP trap al server.
Deve essere lo stesso del file di configurazione di zabbix_trap_receiver.pl o di SNMPTT.

Predefinito: `/tmp/zabbix_traps.tmp`

SocketDir

Directory in cui memorizzare i socket IPC utilizzati dai servizi interni di Zabbix.

Predefinito: `/tmp`

SourceIP

Indirizzo IP di origine per:

- connessioni in uscita verso Zabbix proxy e Zabbix agent
- connessioni senza agent (VMware, SSH, JMX, SNMP, Telnet e controlli semplici)
- connessioni HTTP agent
- richieste HTTP JavaScript degli item script
- richieste HTTP JavaScript del preprocessing
- invio di email di notifica (connessioni al server SMTP)
- notifiche webhook (connessioni HTTP JavaScript)
- connessioni al Vault

SSHKeyLocation

Posizione delle chiavi pubbliche e private per i controlli SSH e le azioni.

SSLCertLocation

Percorso dei file del certificato client SSL per l'autenticazione del client.
Questo parametro è utilizzato solo nel monitoraggio web.

SSLKeyLocation

Percorso dei file della chiave privata SSL per l'autenticazione del client.
Questo parametro è utilizzato solo nel monitoraggio web.

SSLCALocation

Sovrascrive il percorso dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server. Se non impostato, verrà utilizzata la directory di sistema predefinita.
Si noti che il valore di questo parametro verrà impostato come opzione libcurl CURLOPT_CAPATH. Per le versioni di libcurl precedenti alla 7.42.0, questo 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 e nell'autenticazione SMTP.

StartAgentPollers

Il numero di istanze pre-fork di poller dell'agent Zabbix. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartAlerters

Il numero di istanze pre-fork di alerters.

Predefinito: 3
 Intervallo: 1-100

StartBrowserPollers

Il numero di istanze pre-forked dei poller degli item del browser.

Predefinito: 1
 Intervallo: 0-1000

StartConnectors

Il numero di istanze pre-fork di worker del connettore. Il processo di gestione del connettore viene avviato automaticamente quando viene avviato un worker del connettore.

Predefinito: 0
 Intervallo: 0-1000

StartDBSyncers

Il numero di istanze pre-fork di history syncers.
Nota: Prestare attenzione quando si modifica questo valore; aumentarlo potrebbe causare più danni che benefici. In linea generale, il valore predefinito dovrebbe essere sufficiente per gestire fino a 4000 NVPS.

Predefinito: 4
 Intervallo: 1-100

StartDiscoverers

Il numero di istanze pre-fork di discovery workers[1].

Predefinito: 5
 Intervallo: 0-1000

StartEscalators

Il numero di istanze pre-fork di escalators.

Predefinito: 1
 Intervallo: 1-100

StartHistoryPollers

Il numero di istanze pre-fork di history pollers.
Richiesto solo per i controlli calcolati.

Predefinito: 5
 Intervallo: 0-1000

StartHTTPAgentPollers

Il numero di istanze pre-fork di poller dell'agent HTTP. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartHTTPPollers

Il numero di istanze pre-fork di HTTP pollers[1].

Predefinito: 1
 Intervallo: 0-1000

StartIPMIPollers

Il numero di istanze pre-fork di poller IPMI.

Predefinito: 0
 Intervallo: 0-1000

StartJavaPollers

Il numero di istanze pre-fork di Java pollers[1].

Predefinito: 0
 Intervallo: 0-1000

StartLLDProcessors

Il numero di istanze pre-fork di worker di low-level discovery (LLD)[1].
Il processo manager LLD viene avviato automaticamente quando viene avviato un worker LLD.

Predefinito: 2
 Intervallo: 1-100

StartODBCPollers

Il numero di istanze pre-fork di ODBC poller[1].

Predefinito: 1
 Intervallo: 0-1000

StartPingers

Il numero di istanze pre-fork di ICMP pingers[1].

Predefinito: 1
 Intervallo: 0-1000

StartPollersUnreachable

Il numero di istanze pre-fork di poller per host non raggiungibili (inclusi IPMI e Java)[1].
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

Il numero di istanze pre-fork di pollers[1].

Predefinito: 5
 Intervallo: 0-1000

StartPreprocessors

Il numero di istanze pre-avviate dei thread worker[1] di preprocessing dovrebbe essere impostato a un valore non inferiore al numero di core CPU disponibili. È opportuno impostare un numero maggiore di worker se il preprocessing non è limitato dalla CPU e comporta molte richieste di rete.

Predefinito: 16
 Intervallo: 1-1000

StartProxyPollers

Il numero di istanze pre-fork dei poller per proxy passivi[1].

Predefinito: 1
 Intervallo: 0-250

StartReportWriters

Il numero di istanze pre-fork dei report writer.
Se impostato a 0, la generazione pianificata dei report è disabilitata.
Il processo report manager viene avviato automaticamente quando viene avviato un report writer.

Predefinito: 0
 Intervallo: 0-100

StartSNMPPollers

Il numero di istanze pre-fork di poller SNMP. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartSNMPTrapper

Se impostato a 1, verrà avviato un processo SNMP trapper.

Predefinito: 0
 Intervallo: 0-1

StartTimers

Il numero di istanze pre-fork di timers.
I timers elaborano i periodi di manutenzione.

Predefinito: 1
 Intervallo: 1-1000

StartTrappers

Il numero di istanze pre-fork di trapper[1].
I trapper accettano connessioni in ingresso da Zabbix sender, agent attivi e proxy attivi.

Predefinito: 5
 Intervallo: 0-1000

StartVMwareCollectors

Il numero di istanze VMware collector pre-fork.

Predefinito: 0
 Intervallo: 0-250

StatsAllowedIP

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di istanze Zabbix esterne. Le richieste di statistiche saranno accettate solo dagli indirizzi elencati qui. Se questo parametro non è impostato, nessuna richiesta di statistiche sarà 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
```

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy, agent, servizio web e controlli SNMP legacy (numero OID singolo o stringa).

Questo parametro definisce la durata per varie operazioni di comunicazione:

- esecuzione di comandi remoti su Zabbix agent
- esecuzione di comandi SSH/Telnet
- richieste al servizio web di Zabbix
- timeout di comunicazione per la richiesta di test del tipo di supporto e il metodo `history.push`
- riprogrammazione degli item quando un'interfaccia IPMI diventa non disponibile

- invio della risposta a Zabbix proxy quando lo scambio di dati non riesce a causa di problemi di autorizzazione o crittografia
- scadenza per socket asincroni IPC e 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 heartbeat dei controlli attivi
- recupero dei dati (valori) di Zabbix agent dagli agent attivi
- recupero dei dati da Zabbix sender
- risposta quando trapper non riesce a elaborare una richiesta
- invio dell'elenco dei controlli attivi a Zabbix agent

Questo timeout **non** verrà utilizzato per quei controlli che hanno configurate impostazioni di timeout flessibile nel frontend (a livello globale, di proxy o di singolo item). Ad esempio, gli item SNMP `walk[OID]` e `get[OID]` utilizzano il timeout configurato nel frontend; i controlli SNMP legacy continuano invece a utilizzare il valore di timeout del server.

Predefinito: 3
 Intervallo: 1-30

TLSCAFile

Il percorso completo di un file contenente i certificati della CA di primo livello per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSCertFile

Il percorso completo di un file contenente il certificato del server o la catena di certificati, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCipherAll

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 certificato e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato 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

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 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

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.

TLSCipherPSK

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

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Il percorso completo di un file contenente certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSFrontendAccept

Quali connessioni in ingresso accettare dal frontend.

È possibile specificare più valori, separati da una virgola:

- unencrypted - accetta connessioni senza crittografia.
- cert - accetta connessioni protette con TLS e un certificato.

Predefinito: `unencrypted`

TLSFrontendCertIssuer

Autorità emittente del certificato frontend consentita.

TLSFrontendCertSubject

Soggetto del certificato frontend consentito.

TLSKeyFile

Il percorso completo di un file contenente la chiave privata del server, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSListen

Controlla il TLS sul socket trapper.

Valori supportati:

- required - accetta solo connessioni TLS

TmpDir

La directory temporanea.

Predefinito: `/tmp`

TrapperTimeout

Specifica il timeout 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 di comandi remoti su Zabbix server

Predefinito: 300
 Intervallo: 1-300

TrendCacheSize

La dimensione della cache dei trend, in byte.
La dimensione della memoria condivisa per memorizzare i dati dei trend.

Predefinito: `4M`
 Intervallo: 128K-16G

TrendFunctionCacheSize

La dimensione della cache delle funzioni di trend, in byte.
La dimensione della memoria condivisa per la cache dei dati calcolati delle funzioni di trend.

Predefinito: `4M`
 Intervallo: 128K-2G

UnavailableDelay

Determina con quale frequenza l'host viene controllato per la disponibilità durante il periodo di indisponibilità, in secondi.

Predefinito: 60
 Intervallo: 1-3600

UnreachableDelay

Determina con quale frequenza viene controllata la disponibilità dell'host durante il periodo di irraggiungibilità, in secondi.

Predefinito: 15
 Intervallo: 1-3600

UnreachablePeriod

Determina dopo quanti secondi di irraggiungibilità un host viene considerato non disponibile.

Predefinito: 45
 Intervallo: 1-3600

Utente

Abbandona i privilegi a favore di uno specifico utente esistente nel sistema.
Ha effetto solo se eseguito come 'root' e se AllowRoot è disabilitato.

Predefinito: `zabbix`

ValueCacheSize

La dimensione della cache dei valori della cronologia, in byte.
La dimensione della memoria condivisa per la memorizzazione nella cache delle richieste dei dati della cronologia degli item.
Impostando il valore su 0 si disabilita la cache dei valori (scelta non consigliata).
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

Specifica 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`

VaultDBPath

Percorso o query del Vault, a seconda del Vault, da cui verranno recuperate tramite chiavi le credenziali per il database.

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

Un prefisso personalizzato per il percorso o la query di Vault, a seconda del Vault. Se non specificato, verranno utilizzati i valori predefiniti più adatti.
Si noti che per HashiCorp, se VaultPrefix non è specificato, 'data' viene aggiunto automaticamente dopo il mountpoint.

Esempio di prefisso per Hashicorp:

```
v1/secret/data/zabbix/
```

Esempio di prefisso per Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

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, lasciare vuoto il campo del file della chiave SSL.
 La directory

che contiene questo file è specificata dal parametro di configurazione SSLCertLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultTLSKeyFile

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 che contiene questo file è specificata dal parametro di configurazione SSLKeyLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultToken

Il token di autenticazione di HashiCorp Vault che deve essere stato generato esclusivamente per Zabbix server con permessi di sola lettura per i percorsi specificati in Vault macros e permessi di sola lettura per il percorso specificato nel parametro di configurazione opzionale VaultDBPath.
È un errore se VaultToken e la variabile d'ambiente VAULT_TOKEN sono definiti contemporaneamente.

Obbligatorio: Sì, se Vault è impostato su *HashiCorp*; altrimenti no

VaultURL

L'URL HTTP[S] del server vault. Verrà utilizzata la directory dei certificati CA a livello di sistema se SSLCALocation non è specificato.

Predefinito: `https://127.0.0.1:8200`

VMwareCacheSize

La dimensione della memoria condivisa per memorizzare i dati VMware.
È possibile utilizzare un controllo interno VMware `zabbix[vmware,buffer,...]` per monitorare l'utilizzo della cache VMware (vedere Controlli interni).
Si noti che la memoria condivisa non viene allocata se non sono configurate istanze del collector vmware da avviare.

Predefinito: 8M
 Intervallo: 256K-2G

VMwareFrequency

Il ritardo in secondi tra le raccolte dati da un singolo servizio VMware.
Questo ritardo deve essere impostato all'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwarePerfFrequency

Il ritardo in secondi tra i recuperi delle statistiche dei contatori delle prestazioni da un singolo servizio VMware. Questo ritardo deve essere impostato all'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware che utilizza i contatori delle prestazioni VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwareTimeout

Il numero massimo di secondi per cui un collector vmware attenderà una risposta dal servizio VMware (vCenter o hypervisor ESX).

Predefinito: 10
 Intervallo: 1-300

WebServiceURL

L'URL HTTP[S] del servizio web di Zabbix nel formato `http[s]://host:port/report`.

Esempio:

```
WebServiceURL=http://localhost:10053/report
```

Nota: lo schema (`http://`) può essere omesso solo per connessioni non TLS (HTTP); se è configurato TLS, è necessario usare `https://`.

WebDriverURL

URL HTTP[S] dell'interfaccia WebDriver.

Esempio (utilizzato con il server standalone Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

Note a piè di pagina

[1] Si noti che un numero eccessivo di processi di raccolta dati (poller, poller irraggiungibili, poller ODBC, poller HTTP, poller Java, pinger, trapper, proxypoller), insieme a IPMI manager, SNMP trapper, worker di preprocessing e worker di discovery, può esaurire il limite di file descriptor per processo del preprocessing manager.

Warning:
L'esaurimento del limite di file descriptor causerà l'arresto di Zabbix server, in genere poco dopo l'avvio, ma talvolta anche più tardi. Per evitare tali problemi, controllare il file di configurazione di Zabbix server per ottimizzare il numero di controlli concorrenti e di processi. Inoltre, se necessario, assicurarsi che il limite di file descriptor sia impostato su un valore sufficientemente alto verificando e adeguando i limiti di sistema.

[2] Quando viene eliminato un gran numero di item, il carico sul database aumenta, perché l'housekeeper dovrà rimuovere tutti i dati storici associati a tali item. 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 dovranno essere rimossi 5000 item (1*50*100). Se per MaxHousekeeperDelete è impostato 500 (MaxHousekeeperDelete=500), il processo housekeeper dovrà rimuovere fino a 2500000 valori (5000*500) per gli item eliminati dalle tabelle history e trends in un ciclo.

[3] DBSocket e DBPort si escludono a vicenda nella configurazione del server. Specificarne solo uno, oppure lasciare entrambi non definiti.

2 Proxy Zabbix

Panoramica

I parametri supportati dal file di configurazione del proxy Zabbix (zabbix_proxy.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare i dettagli completi.

Parametro	Descrizione
AllowRoot	Consente al proxy di essere eseguito come 'root'.
AllowUnsupportedDBVersions	Consente al proxy di funzionare con versioni di database non supportate.
CacheSize	La dimensione della cache di configurazione.
ConfigFrequency	Questo parametro è **deprecato** (utilizzare invece ProxyConfigFrequency). Con quale frequenza il proxy recupera i dati di configurazione dal server Zabbix, in secondi.
DataSenderFrequency	Il proxy invierà i dati raccolti al server ogni N secondi.
DBHost	Il nome host del database.
DBName	Il nome del database o il percorso al file del database per SQLite3.
DBPassword	La password del database.
DBPort	La porta del database quando non si utilizza un socket locale.
DBSchema	Il nome dello schema del database. Utilizzato per PostgreSQL.
DBSocket	Il percorso al file socket MySQL.
DBUser	L'utente del database.
DBTLSConnect	Impostando questa opzione al valore specificato si impone l'uso di una connessione TLS al database.
DBTLSCAFile	Il percorso completo di un file contenente i certificati della CA di livello superiore per la verifica del certificato del database.
DBTLSCertFile	Il percorso completo di un file contenente il certificato del proxy Zabbix per l'autenticazione al database.
DBTLSKeyFile	Il percorso completo di un file contenente la chiave privata per l'autenticazione al database.
DBTLSCipher	L'elenco dei cifrari di crittografia consentiti dal proxy Zabbix per i protocolli TLS fino a TLS v1.2. Supportato solo per MySQL.
DBTLSCipher13	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	Il livello di debug.
EnableRemoteCommands	Indica se i comandi remoti dal server Zabbix sono consentiti.
ExternalScripts	La posizione degli script esterni.
Fping6Location	La posizione di fping6.
FpingLocation	La posizione di fping.
HistoryCacheSize	La dimensione della cache della cronologia.
HistoryIndexCacheSize	La dimensione della cache dell'indice della cronologia.
Hostname	Un nome proxy univoco, con distinzione tra maiuscole e minuscole.
HostnameItem	L'item utilizzato per impostare Hostname se non è definito.
HousekeepingFrequency	Con quale frequenza Zabbix eseguirà la procedura di housekeeping, in ore.
Include	È possibile includere singoli file o tutti i file in una directory nel file di configurazione.
JavaGateway	L'indirizzo IP (o hostname) del Java gateway Zabbix.
JavaGatewayPort	La porta su cui il Java gateway Zabbix è in ascolto.
ListenBacklog	Il numero massimo di connessioni in attesa nella coda TCP.

Parametro	Descrizione
ListenIP	Un elenco di indirizzi IP separati da virgole su cui il trapper deve essere in ascolto.
ListenPort	La porta di ascolto per il trapper.
LoadModule	Il modulo da caricare all'avvio del proxy.
LoadModulePath	Il percorso completo della posizione dei moduli del proxy.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
LogSlowQueries	Quanto tempo può impiegare una query del database prima di essere registrata nel log.
LogType	Il tipo di output del log.
MaxConcurrentChecksPerPoller	Il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, poller agent o poller SNMP.
PidFile	Il nome del file PID.
ProxyBufferMode	Specifica il meccanismo di archiviazione dei dati di cronologia, discovery e autoregistrazione (disco/memoria/ibrido).
ProxyConfigFrequency	Con quale frequenza il proxy recupera i dati di configurazione dal server Zabbix, in secondi.
ProxyLocalBuffer	Il proxy manterrà i dati localmente per N ore, anche se i dati sono già stati sincronizzati con il server.
ProxyMemoryBufferAge	L'età massima dei dati nel buffer di memoria del proxy, in secondi.
ProxyMemoryBufferSize	La dimensione della cache di memoria condivisa per i dati raccolti di cronologia, discovery e registrazione automatica.
ProxyMode	La modalità operativa del proxy (attiva/passiva).
ProxyOfflineBuffer	Il proxy manterrà i dati per N ore in caso di assenza di connettività con il server Zabbix.
Server	Se ProxyMode è impostato sulla modalità attiva: indirizzo IP o nome DNS del server Zabbix (indirizzo:porta) oppure cluster (indirizzo:porta;indirizzo2:porta) da cui ottenere i dati di configurazione e a cui inviare i dati. Se ProxyMode è impostato sulla modalità passiva: elenco di indirizzi IP del server Zabbix separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS.
SNMPTrapperFile	Il file temporaneo utilizzato per trasferire i dati dal demone trap SNMP al proxy.
SocketDir	La directory in cui memorizzare i socket IPC utilizzati dai servizi interni di Zabbix.
SourceIP	L'indirizzo IP sorgente.
SSHKeyLocation	La posizione delle chiavi pubbliche e private per i controlli SSH e le azioni.
SSLCertLocation	La posizione dei file di certificato client SSL per l'autenticazione del client.
SSLKeyLocation	La posizione dei file di chiave privata SSL per l'autenticazione del client.
SSLCALocation	Sovrascrive la posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server.
StartAgentPollers	Il numero di istanze pre-fork di poller asincroni Zabbix agent.
StartBrowserPollers	Il numero di istanze pre-fork di poller degli item del browser.
StartDBSyncers	Il numero di istanze pre-fork di sincronizzatori della cronologia.
StartDiscoverers	Il numero di istanze pre-fork di worker di discovery.
StartHTTPAgentPollers	Il numero di istanze pre-fork di poller asincroni HTTP agent.
StartHTTPPollers	Il numero di istanze pre-fork di poller HTTP.
StartIPMIPollers	Il numero di istanze pre-fork di poller IPMI.
StartJavaPollers	Il numero di istanze pre-fork di poller Java.
StartODBCPollers	Il numero di istanze pre-fork di poller ODBC.
StartPingers	Il numero di istanze pre-fork di pinger ICMP.
StartPollersUnreachable	Il numero di istanze pre-fork di poller per host non raggiungibili (inclusi IPMI e Java).
StartPollers	Il numero di istanze pre-fork di poller.
StartPreprocessors	Il numero di istanze pre-avviate di worker di preprocessing.
StartSNMPPollers	Il numero di istanze pre-fork di poller SNMP asincroni.
StartSNMPTrapper	Se impostato a 1, verrà avviato un processo trapper SNMP.
StartTrappers	Il numero di istanze pre-fork di trapper.
StartVMwareCollectors	Il numero di istanze collector VMware pre-fork.
StatsAllowedIP	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di istanze Zabbix esterne. La richiesta di statistiche sarà accettata solo dagli indirizzi elencati qui.
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con proxy Zabbix, agent, servizio web e per i controlli SNMP (eccetto gli item SNMP `walk[OID]` e `get[OID]`).
TLSAccept	Quali connessioni in ingresso accettare dal server Zabbix.
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 cifrate tra componenti Zabbix.

Parametro	Descrizione
TLSCertFile	Il percorso completo di un file contenente il certificato del server o la catena di certificati, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCipherAll	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 certificato e PSK.
TLSCipherAll13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.
TLSCipherCert	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 certificato.
TLSCipherCert13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.
TLSCipherPSK	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.
TLSCipherPSK13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.
TLSConnect	Come il proxy deve connettersi al server Zabbix.
TLSCRLFile	Il percorso completo di un file contenente certificati revocati. Questo parametro è utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata del proxy, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSListen	Controlla TLS sul socket del trapper.
TLSPSKFile	Il percorso completo di un file contenente la chiave precondivisa del proxy, utilizzato per le comunicazioni cifrate con il server Zabbix.
TLSPSKIdentity	La stringa identificativa della chiave precondivisa, utilizzata per le comunicazioni cifrate con il server Zabbix.
TLSServerCertIssuer	L'emittente consentito del certificato del server.
TLSServerCertSubject	Il soggetto consentito del certificato del server.
TmpDir	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	Con quale frequenza un host viene controllato per verificarne la disponibilità durante il periodo di indisponibilità.
UnreachableDelay	Con quale frequenza un host viene controllato per verificarne la disponibilità durante il periodo di irraggiungibilità.
UnreachablePeriod	Dopo quanti secondi di irraggiungibilità l'host deve essere considerato non disponibile.
User	Abbandona i privilegi a favore di uno specifico utente esistente nel sistema.
Vault	Il provider del vault.
VaultDBPath	La posizione da cui recuperare le credenziali del database tramite chiavi.
VaultPrefix	Prefisso personalizzato per il percorso o la query del vault.
VaultTLSCertFile	Il nome del file del certificato SSL utilizzato per l'autenticazione del client.
VaultTLSKeyFile	Il nome del file della chiave privata SSL utilizzato per l'autenticazione del client.
VaultToken	Il token di autenticazione HashiCorp vault.
VaultURL	L'URL HTTP[S] del server vault.
VMwareCacheSize	La dimensione della memoria condivisa per l'archiviazione dei dati VMware.
VMwareFrequency	Il ritardo in secondi tra le raccolte dati da un singolo servizio VMware.
VMwarePerfFrequency	Il ritardo in secondi tra i recuperi delle statistiche dei contatori di prestazioni da un singolo servizio VMware.
VMwareTimeout	Il numero massimo di secondi per cui un collector vmware attenderà una risposta dal servizio VMware.
WebDriverURL	L'URL HTTP[S] dell'interfaccia WebDriver.

Tutti i parametri sono facoltativi, salvo ove esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione distribuiti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta 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

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 alcun effetto se avviato con un utente normale.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

AllowUnsupportedDBVersions

Consente al proxy di funzionare con versioni di database non supportate.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

CacheSize

La dimensione della cache di configurazione, in byte. La dimensione della memoria condivisa per l'archiviazione dei dati di host e item.

Predefinito: 32M
 Intervallo: 128K-64G

ConfigFrequency

Questo parametro è **deprecato** (utilizzare invece ProxyConfigFrequency).
Indica con quale frequenza il proxy recupera i dati di configurazione dal Zabbix server, in secondi.
Parametro del proxy attivo. Ignorato per i proxy passivi (vedere il parametro ProxyMode).

Predefinito: 3600
 Intervallo: 1-604800

DataSenderFrequency

Il proxy invierà i dati raccolti al server ogni N secondi. 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

Il nome dell'host del database.
Con MySQL, `localhost` o una stringa vuota comportano l'uso di un socket. Con PostgreSQL, solo una stringa vuota comporta il tentativo di usare un socket.

Predefinito: `localhost`

DBName

Il nome del database o il percorso al file del database per SQLite3 (l'architettura multi-processo di Zabbix non consente di utilizzare un database in memoria, ad esempio `:memory:`, `file::memory:?cache=shared` oppure `file:memdb1?mode=memory&cache=shared`). Non tentare di utilizzare lo stesso database utilizzato dal server Zabbix.

Obbligatorio: Sì

DBPassword

La password del database. Commentare questa riga se non viene utilizzata alcuna password. Ignorato per SQLite.

DBPort

La porta del database quando non si utilizza un socket locale.[1] Ignorato per SQLite.

Predefinito per MySQL: 3306

Predefinito per PostgreSQL: 5432

Intervallo: 1024-65535

DBSchema

Il nome dello schema del database. Utilizzato per PostgreSQL.

DBSocket

Il percorso del file socket MySQL.[1]
La porta del database quando non si utilizza il socket locale. Ignorato per SQLite.

Predefinito: 3306

DBUser

L'utente del database. Ignorato per SQLite.

DBTLSConnect

L’impostazione di questa opzione impone l’uso di 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
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 impostata alcuna opzione e il comportamento dipende dalla configurazione del database.

DBTLSCAFile

Il percorso completo di un file contenente i certificati della 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

Il percorso completo di un file contenente il certificato proxy Zabbix per l’autenticazione al database.

DBTLSKeyFile

Il percorso completo di un file contenente la chiave privata per l’autenticazione al database.

DBTLSCipher

L’elenco dei cifrari di crittografia che Zabbix proxy consente per i protocolli TLS fino a TLS v1.2 incluso. Supportato solo per MySQL.

DBTLSCipher13

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

Specifica il livello di debug:
0 - informazioni di base sull’avvio e l’arresto dei processi di 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

Se i comandi remoti da Zabbix server sono consentiti.

Predefinito: 0
 Valori: 0 - non consentiti; 1 - consentiti

ExternalScripts

La posizione degli script esterni (dipende dalla variabile di installazione in fase di compilazione `datadir`).

Predefinito: `/usr/local/share/zabbix/externalscripts`

Fping6Location

Il percorso di fping6. Assicurarsi che il binario fping6 abbia root come proprietario e il flag SUID impostato. Lasciare vuoto (”Fping6Location=”) se l’utilità fping in uso è in grado di elaborare indirizzi IPv6.

Predefinito: `/usr/sbin/fping6`

FpingLocation

Il percorso di fping. Assicurati che il binario fping abbia root come proprietario e il flag SUID impostato.

Predefinito: `/usr/sbin/fping`

HistoryCacheSize

La dimensione della cache della cronologia, in byte. La dimensione della memoria condivisa per l’archiviazione dei dati della cronologia.

Predefinito: 16M
 Intervallo: 128K-16G

HistoryIndexCacheSize

La dimensione della cache dell’indice della cronologia, in byte. La dimensione della memoria condivisa per l’indicizzazione dei dati della cronologia memorizzati nella cache della cronologia. La dimensione della cache dell’indice richiede circa 100 byte per memorizzare nella cache un item.

Predefinito: 4M
 Intervallo: 128K-16G

Hostname

Un nome univoco del proxy, con distinzione tra maiuscole e minuscole. Assicurarsi che il nome del proxy sia noto al server.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128

Predefinito: impostato da HostnameItem

HostnameItem

L'item utilizzato per impostare Hostname se non è definito (questo verrà eseguito sul proxy in modo analogo a quanto avviene su un agent). Ignorato se Hostname è impostato.
Non supporta UserParameters, i contatori delle prestazioni o gli alias, ma supporta system.run[].

Predefinito: system.hostname

HousekeepingFrequency

Con quale frequenza Zabbix eseguirà la 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 a 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 procedura di housekeeping.

Predefinito: 1
 Intervallo: 0-24

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione.
Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali sulle limitazioni.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

L'indirizzo IP (o hostname) del Java gateway di Zabbix. Richiesto solo se i Java poller sono avviati.

JavaGatewayPort

La porta su cui è in ascolto Zabbix Java gateway.

Predefinito: 10052
 Intervallo: 1024-32767

ListenBacklog

Il numero massimo di connessioni in attesa 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

Un elenco di indirizzi IP separati da virgole sui quali il trapper deve mettersi in ascolto.
Il trapper si metterà in ascolto su tutte le interfacce di rete se questo parametro è assente.

Predefinito: 0.0.0.0

ListenPort

La porta di ascolto per il trapper.

Predefinito: 10051
 Intervallo: 1024-32767

LoadModule

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 '/'), allora LoadModulePath viene ignorato.
Formati:
LoadModule=<module.so>
LoadModule=<path/module.s
consentito includere più parametri LoadModule.

LoadModulePath

Il percorso completo della posizione dei moduli del proxy. Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Il nome del file di 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 ricominciato da capo.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi.

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

LogType

Il tipo dell'output di 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`

LogSlowQueries

Per quanto tempo una query del database può durare prima di essere registrata nel log (in millisecondi).
0 - non registrare le query lente.
Questa opzione viene abilitata a partire da DebugLevel=3.

Predefinito: 0
 Intervallo: 0-3600000

MaxConcurrentChecksPerPoller

Il numero massimo di controlli asincroni che possono essere eseguiti contemporaneamente da ciascun poller HTTP agent, poller agent o poller SNMP. Vedere StartHTTPAgentPollers, StartAgentPollers e StartSNMPPollers.

Predefinito: 1000
 Intervallo: 1-1000

PidFile

Il nome del file PID.

Predefinito: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Specifica il meccanismo di archiviazione dei dati di cronologia, di network discovery e di autoregistrazione: *disk* - i dati vengono memorizzati nel database e caricati dal database; *memory* - i dati vengono memorizzati in memoria e caricati dalla memoria. Se il buffer esaurisce la memoria, i dati meno recenti verranno scartati. All'arresto il buffer viene scartato. *hybrid* - il buffer del proxy normalmente funziona come in modalità memory finché non esaurisce la memoria oppure finché il record più vecchio non supera l'età configurata. Se ciò accade, il buffer viene scaricato nel database e funziona come in modalità disk finché tutti i dati non sono stati caricati, dopodiché ricomincia a funzionare in memoria. All'arresto il buffer in memoria viene scaricato nel database.

Vedere anche: Buffer di memoria del proxy.

Predefinito: `disk`
 Valori: disk; memory; hybrid

ProxyConfigFrequency

Con quale frequenza il proxy recupera i 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

Il proxy manterrà i dati localmente per N ore, anche se i dati sono già stati sincronizzati con il server.
Questo parametro può essere utilizzato se i dati locali verranno usati da applicazioni di terze parti.

Predefinito: 0
 Intervallo: 0-720

ProxyMemoryBufferAge

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 finché 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

La dimensione della cache di memoria condivisa per i dati raccolti di cronologia, discovery e autoregistrazione, in byte. Se abilitato (diverso da zero), il proxy manterrà in memoria i dati di cronologia, discovery e autoregistrazione, a meno che la cache non sia piena o che i record memorizzati siano più vecchi del valore ProxyMemoryBufferAge definito. Questo parametro non può essere utilizzato insieme al parametro ProxyLocalBuffer.

Predefinito: 0
 Intervallo: 0;128K-2G

ProxyMode

La modalità operativa del proxy.
0 - proxy in modalità attiva
1 - proxy in modalità passiva
Nota che i dati di configurazione (sensibili) del proxy 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 fingersi un proxy attivo e richiedere i dati di configurazione; l'autenticazione non avviene.

Predefinito: 0
 Intervallo: 0-1

ProxyOfflineBuffer

Il proxy conserverà i dati per N ore in caso di assenza di connettività con Zabbix server.
I dati meno recenti andranno persi.

Predefinito: 1
 Intervallo: 1-720

Server

Se ProxyMode è impostato su *modalità attiva*:
Indirizzo IP o nome DNS del server Zabbix (indirizzo:porta) oppure cluster (indirizzo:porta;indirizzo2:porta) 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 *modalità passiva*:
Elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS del server Zabbix. Le connessioni in ingresso saranno accettate solo dagli indirizzi 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.
'::/0' consentirà qualsiasi indirizzo IPv4 o IPv6. '0.0.0.0/0' può essere utilizzato 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

Un file temporaneo utilizzato per trasferire i dati dal demone SNMP trap al proxy.
Deve essere lo stesso del file di configurazione di zabbix_trap_receiver.pl o di SNMPTT.

Predefinito: `/tmp/zabbix_traps.tmp`

SocketDir

La directory in cui memorizzare i socket IPC utilizzati dai servizi interni di Zabbix.

Predefinito: `/tmp`

SourceIP

L'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server
- le connessioni senza agent (VMware, SSH, JMX, SNMP, Telnet e controlli semplici)
- le connessioni dell'agent HTTP
- le richieste HTTP JavaScript degli item script
- le richieste HTTP JavaScript del preprocessing
- le connessioni al Vault

SSHKeyLocation

La posizione delle chiavi pubbliche e private per i controlli SSH e le azioni.

SSLCertLocation

La posizione dei file del certificato client SSL per l'autenticazione del client.
Questo parametro viene utilizzato solo nel monitoraggio web.

SSLKeyLocation

La posizione dei file della chiave privata SSL per l'autenticazione del client.
Questo parametro viene utilizzato solo nel monitoraggio web.

SSLCALocation

La posizione dei file dell'autorità di certificazione (CA) per la verifica del certificato SSL del server.
Si noti che il valore di questo parametro verrà impostato come opzione libcurl CURLOPT_CAPATH. Per le versioni di libcurl precedenti alla 7.42.0, questo 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 e nell'autenticazione SMTP.

StartAgentPollers

Il numero di istanze pre-fork di poller dell'agent Zabbix. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartBrowserPollers

Il numero di istanze pre-fork dei poller degli item del browser.

Predefinito: 1
 Intervallo: 0-1000

StartDBSyncers

Il numero di istanze pre-fork dei history syncers.
Nota: Prestare attenzione quando si modifica questo valore; aumentarlo potrebbe causare più danni che benefici.

Predefinito: 4
 Intervallo: 1-100

StartDiscoverers

Il numero di istanze pre-fork dei discovery workers.

Predefinito: 5
 Intervallo: 0-1000

StartHTTPAgentPollers

Il numero di istanze pre-fork di poller dell'agent HTTP. Vedi MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartHTTPPollers

Il numero di istanze pre-forked di HTTP pollers.

Predefinito: 1
 Intervallo: 0-1000

StartIPMIPollers

Il numero di istanze pre-fork di IPMI pollers.

Predefinito: 0
 Intervallo: 0-1000

StartJavaPollers

Il numero di istanze pre-fork di Java pollers.

Predefinito: 0
 Intervallo: 0-1000

StartODBCPollers

Il numero di istanze pre-fork di ODBC pollers.

Predefinito: 1
 Intervallo: 0-1000

StartPingers

Il numero di istanze pre-fork di ICMP pingers.

Predefinito: 1
 Intervallo: 0-1000

StartPollersUnreachable

Il numero di istanze pre-fork di poller 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

Il numero di istanze pre-fork dei poller.

Predefinito: 5
 Intervallo: 0-1000

StartPreprocessors

Il numero di istanze pre-avviate dei thread worker di preprocessing deve essere impostato a un valore non inferiore al numero di core CPU disponibili. È necessario impostare più worker se il preprocessing non è limitato dalla CPU e comporta molte richieste di rete.

Predefinito: 16
 Intervallo: 1-1000

StartSNMPPollers

Il numero di istanze pre-fork di poller SNMP. Vedere MaxConcurrentChecksPerPoller.

Predefinito: 1
 Intervallo: 0-1000

StartSNMPTrapper

Se impostato a 1, verrà avviato un processo di SNMP trapper.

Predefinito: 0
 Intervallo: 0-1

StartTrappers

Il numero di istanze pre-fork dei trapper.
I trapper accettano connessioni in ingresso da Zabbix sender e dagli agent attivi.

Predefinito: 5
 Intervallo: 0-1000

StartVMwareCollectors

Il numero di istanze VMware collector pre-fork.

Predefinito: 0
 Intervallo: 0-250

StatsAllowedIP

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di istanze Zabbix esterne. La richiesta di statistiche sarà accettata solo dagli indirizzi elencati qui. Se questo parametro non è impostato, nessuna richiesta di statistiche sarà 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
```

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix server, agent, servizio web e 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 lo scambio di dati non riesce a causa di problemi di autorizzazioni o crittografia
- Scadenza per socket asincroni IPC e opzioni di controllo runtime
- Richieste DNS del poller asincrono
- Risposta per 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 utilizzato per quei controlli che hanno configurate impostazioni di timeout flessibile nel frontend (a livello globale, proxy o per-item).
Ad esempio, gli item SNMP `walk[OID]` e `get[OID]` utilizzano il timeout configurato nel frontend; i controlli SNMP legacy continuano a utilizzare il valore di timeout del server.

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Quali connessioni in ingresso accettare dal server Zabbix. Utilizzato per un proxy passivo, ignorato su un proxy attivo. È possibile specificare più valori, separati da una 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 per un proxy passivo, se sono definiti i parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCAFile

Il percorso completo di un file contenente i certificati della/e CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSCertFile

Il percorso completo di un file contenente il certificato del proxy o la catena di certificati, utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSCipherAll

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 certificato e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione delle 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

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 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

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.

TLSCipherPSK

La stringa di priorità GnuTLS o la stringa dei cifrari 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

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Come il proxy deve connettersi al server Zabbix. Utilizzato 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 i parametri del certificato TLS o della PSK (anche per una connessione *unencrypted*); altrimenti no

TLSCRLFile

Il percorso completo di un file contenente certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSKeyFile

Il percorso completo di un file contenente la chiave privata del proxy, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSListen

Controlla TLS sul socket trapper.

Valori supportati:

- required - accetta solo connessioni TLS

TLSPSKFile

Il percorso completo di un file contenente la chiave precondivisa del proxy, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSPSKIdentity

La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSServerCertIssuer

L'autorità emittente consentita del certificato del server.

TLSServerCertSubject

Il subject del certificato del server consentito.

TmpDir

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 di comandi remoti.

Predefinito: 300
 Intervallo: 1-300

UnavailableDelay

Con quale frequenza viene controllata la disponibilità di un host durante il periodo di indisponibilità, in secondi.

Predefinito: 60
 Intervallo: 1-3600

UnreachableDelay

Con quale frequenza viene controllata la disponibilità di un host durante il periodo di irraggiungibilità, in secondi.

Predefinito: 15
 Intervallo: 1-3600

UnreachablePeriod

Dopo quanti secondi di irraggiungibilità un host deve essere considerato non disponibile.

Predefinito: 45
 Intervallo: 1-3600

Utente

Abbandona i privilegi a favore di uno specifico utente esistente nel sistema.
Ha effetto solo se eseguito come 'root' e AllowRoot è disabilitato.

Predefinito: `zabbix`

Vault

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`

VaultDBPath

Percorso o query di Vault, a seconda del Vault, da cui verranno recuperate tramite chiavi le credenziali per il database.

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_proxy_database
```

Questa opzione può essere utilizzata solo se DBUser e DBPassword non sono specificati.

VaultPrefix

Un prefisso personalizzato per il percorso o la query di Vault, a seconda del Vault. Se non specificato, verranno utilizzati i valori predefiniti più adatti.
Si noti che per HashiCorp, se VaultPrefix non è specificato, 'data' viene aggiunto automaticamente dopo il mountpoint.
Si noti che per HashiCorp, se VaultPrefix non è specificato, 'data' viene aggiunto automaticamente dopo il mountpoint.

Prefisso di esempio per Hashicorp:

```
v1/secret/data/zabbix/
```

Prefisso di esempio per Cyberark:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

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, lasciare vuoto il campo del file della chiave SSL.
La directory che contiene questo file è specificata dal parametro di configurazione SSLCertLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultTLSKeyFile

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 che contiene questo file è specificata dal parametro di configurazione SSLKeyLocation.
Questa opzione può essere omessa, ma è consigliata per il vault CyberArkCCP.

VaultToken

Il token di autenticazione di HashiCorp Vault che deve essere stato generato esclusivamente per il proxy Zabbix con permessi di sola lettura per il percorso specificato nel parametro di configurazione opzionale VaultDBPath.
È un errore se VaultToken e la variabile d'ambiente VAULT_TOKEN sono definiti contemporaneamente.

Obbligatorio: Sì, se Vault è impostato su *HashiCorp*; altrimenti no

VaultURL

L'URL HTTP[S] 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

La dimensione della memoria condivisa per memorizzare i dati VMware.
È possibile utilizzare un controllo interno VMware zabbix[vmware,buffer,...] per monitorare l'utilizzo della cache VMware (vedere Controlli interni).
Si noti che la memoria condivisa non viene allocata se non sono configurate istanze del collector VMware da avviare.

Predefinito: 8M
 Intervallo: 256K-2G

VMwareFrequency

Il ritardo in secondi tra le raccolte dei dati da un singolo servizio VMware.
Questo ritardo deve essere impostato all'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwarePerfFrequency

Il ritardo in secondi tra i recuperi delle statistiche dei contatori di prestazione da un singolo servizio VMware.
Questo ritardo deve essere impostato all'intervallo di aggiornamento minimo di qualsiasi item di monitoraggio VMware che utilizza i contatori di prestazione VMware.

Predefinito: 60
 Intervallo: 10-86400

VMwareTimeout

Il numero massimo di secondi per cui un collector vmware attenderà una risposta dal servizio VMware (vCenter o hypervisor ESX).

Predefinito: 10
 Intervallo: 1-300

WebDriverURL

URL HTTP[S] dell'interfaccia WebDriver.

Esempio (utilizzato con il server standalone Selenium WebDriver):

```
WebDriverURL=http://localhost:4444
```

Note a piè di pagina

[1] DBSocket e DBPort si escludono a vicenda nella configurazione del proxy. Specificarne solo uno oppure lasciare entrambi non definiti.

3 Agente Zabbix (UNIX)

Panoramica

I parametri supportati dal file di configurazione di Zabbix agent (zabbix_agentd.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare tutti i dettagli.

Parameter	Description
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un modello.
AllowRoot	Consente all'agent di essere eseguito come 'root'.
BufferSend	Non mantenere i dati nel buffer per più di N secondi.
BufferSize	Il numero massimo di valori nel buffer di memoria.
DebugLevel	Il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un modello.
EnableRemoteCommands	Indica se i comandi remoti da Zabbix server sono consentiti.
HeartbeatFrequency	La frequenza dei messaggi heartbeat in secondi.
HostInterface	Un parametro facoltativo che definisce l'interfaccia host.
HostInterfaceItem	Un parametro facoltativo che definisce un item usato per ottenere l'interfaccia host.
HostMetadata	Un parametro facoltativo che definisce i metadati dell'host.
HostMetadataItem	Un parametro facoltativo che definisce un item di Zabbix agent usato per ottenere i metadati dell'host.
Hostname	Un parametro facoltativo che definisce il nome host.
HostnameItem	Un parametro facoltativo che definisce un item di Zabbix agent usato per ottenere il nome host.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
ListenBacklog	Il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Un elenco di indirizzi IP separati da virgole sui quali l'agent deve restare in ascolto.
ListenPort	L'agent resterà in ascolto su questa porta per le connessioni dal server.
LoadModule	Il modulo da caricare all'avvio dell'agent.
LoadModulePath	Il percorso completo della posizione dei moduli dell'agent.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione dei comandi shell eseguiti come avvisi.
LogType	Il tipo di output del log.
MaxLinesPerSecond	Il numero massimo di nuove righe che l'agent invierà al Zabbix server o proxy al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'.
PidFile	Il nome del file PID.
RefreshActiveChecks	Con quale frequenza viene aggiornato l'elenco dei controlli attivi.
Server	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server Zabbix e proxy Zabbix.
ServerActive	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	L'indirizzo IP sorgente.

Parameter	Description
StartAgents	Il numero di istanze pre-fork di zabbix_agentd che elaborano i controlli passivi.
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con Zabbix proxy o server.
TLSAccept	Quali connessioni in ingresso accettare.
TLSCAFile	Il percorso completo di un file contenente i certificati della/e CA di livello superiore per la verifica del certificato del peer, usato per le comunicazioni cifrate tra componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato dell'agent o la catena di certificati, usato per le comunicazioni cifrate tra componenti Zabbix.
TLSCipherAll	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 cifratura basata su certificato e PSK.
TLSCipherAll13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato e PSK.
TLSCipherCert	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 cifratura basata su certificato.
TLSCipherCert13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato.
TLSCipherPSK	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 cifratura basata su PSK.
TLSCipherPSK13	La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su PSK.
TLSConnect	Come l'agent deve connettersi a Zabbix server o proxy.
TLSCRLFile	Il percorso completo di un file contenente certificati revocati. Questo parametro è usato per le comunicazioni cifrate tra componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata dell'agent, usato per le comunicazioni cifrate tra componenti Zabbix.
TLSPSKFile	Il percorso completo di un file contenente la chiave precondivisa dell'agent, usato per le comunicazioni cifrate con Zabbix server.
TLSPSKIdentity	La stringa identificativa della chiave precondivisa, usata per le comunicazioni cifrate con Zabbix server.
TLSServerCertIssuer	L'emittente consentito del certificato del server (proxy).
TLSServerCertSubject	Il soggetto consentito del certificato del server (proxy).
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti ai parametri definiti dall'utente.
User	Abbandona i privilegi a favore di uno specifico utente esistente nel sistema.
UserParameter	Un parametro definito dall'utente da monitorare.
UserParameterDir	Il percorso di ricerca predefinito per i comandi UserParameter.

Tutti i parametri sono facoltativi, salvo ove esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione distribuiti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta 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

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più corta 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 la chiave abbreviata **zabbix.userid** può essere utilizzata per recuperare i dati.

Esempio 2: ottenimento 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, nonché di usare **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]**, ecc.

AllowKey

Consente l'esecuzione di quelle chiavi item che corrispondono a un modello. Il modello della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con DenyKey. I parametri vengono elaborati uno per uno in base al loro ordine di apparizione. Vedi anche: Limitazione dei controlli agent.

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 alcun effetto se avviato con un utente normale.

Predefinito: 0
 Valori: 0 - non consentire; 1 - consentire

BufferSend

Non mantenere i dati nel buffer per più di N secondi.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Il numero massimo di valori nel buffer di memoria. L'agent invierà tutti i dati raccolti al server o al proxy Zabbix se il buffer è pieno.

Predefinito: 100
 Intervallo: 2-65535

DebugLevel

Specifica il livello di debug:
0 - informazioni di base sull'avvio e l'arresto dei processi di 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 di quelle chiavi item che corrispondono a un modello. Il modello della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con AllowKey. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. Vedi anche: Limitazione dei controlli agent.

EnableRemoteCommands

Se i comandi remoti da Zabbix server sono consentiti. Questo parametro è **deprecato**; utilizzare invece AllowKey=system.run[*] oppure DenyKey=system.run[*].
Si tratta di un alias interno per i parametri AllowKey/DenyKey a seconda del valore:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

HeartbeatFrequency

La frequenza dei messaggi heartbeat in secondi. Utilizzata per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Un parametro opzionale che definisce l'interfaccia del host (indirizzo IP o nome DNS) utilizzata durante il processo di autoregistration del host. Questo valore verrà utilizzato per popolare l'interfaccia nel host appena creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per maggiori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent genererà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Un parametro opzionale che definisce un item utilizzato per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante il processo di autoregistration del host. Questo valore viene utilizzato solo se HostInterface non è definito. Per maggiori dettagli, vedere Using DNS as default interface.

Durante una richiesta di autoregistration, 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 di AllowKey/DenyKey.

HostMetadata

Un parametro opzionale che definisce i metadati utilizzati per identificare o distinguere l'host durante il processo di autoregistration dell'host (agent attivo). HostMetadata consente di distinguere tra 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 è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno rifiutati e segnalati come non validi.

Intervallo: 0-2034 byte

HostMetadataItem

Un parametro opzionale che definisce un item di Zabbix agent utilizzato 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 nel processo di autoregistrazione dell'host (agent attivo). HostMetadataItem consente di distinguere tra host oltre il nome host.

I parametri utente e gli alias sono supportati. L'item system.run[] è supportato indipendentemente dalle impostazioni di AllowKey/DenyKey.

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, altrimenti verrà ignorato. Quando un parametro prevede un indirizzo IP o un nome DNS, i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno anch'essi rifiutati e segnalati come non validi.

Hostname

Un elenco di nomi host univoci, con distinzione tra maiuscole e minuscole, separati da virgole. Obbligatorio per i controlli attivi e deve corrispondere ai nomi host come configurati sul server. Il valore viene acquisito da HostnameItem se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per nome host, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Un parametro opzionale che definisce un item di Zabbix agent utilizzato per ottenere il nome del 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/DenyKey.

Predefinito: `system.hostname`

Include

È possibile includere singoli file o tutti i file di una directory nel file di configurazione. Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali sulle limitazioni.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Il numero massimo di connessioni in attesa 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

Un elenco di indirizzi IP separati da virgole sui quali l'agent deve restare in ascolto.

Predefinito: `0.0.0.0`

ListenPort

L'agent rimarrà in ascolto su questa porta per le connessioni dal server.

Predefinito: 10050
 Intervallo: 1024-32767

LoadModule

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 che precede il nome è assoluto (inizia con '/'), LoadModulePath viene ignorato.
Formati:
LoadModule=<module.so>
LoadModule=<path/modu
consentito includere più parametri LoadModule.

LoadModulePath

Il percorso completo della posizione dei moduli dell'agent. Il valore predefinito dipende dalle opzioni di compilazione.

LogFile

Il nome del file di 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 ricominciato da capo.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione dei comandi shell eseguiti come avvisi. I comandi verranno registrati solo se eseguiti da 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

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`

MaxLinesPerSecond

Il numero massimo di nuove righe che l'agent invierà al server o al proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'. Il valore fornito verrà sovrascritto dal parametro 'maxlines', specificato nella chiave 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

Il nome del file PID.

Predefinito: `/tmp/zabbix_agentd.pid`

RefreshActiveChecks

Con quale frequenza viene aggiornato l'elenco dei controlli attivi, in secondi. Si noti che, dopo un tentativo non riuscito di aggiornare i controlli attivi, il successivo tentativo di aggiornamento verrà effettuato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server e 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. Si noti che gli "indirizzi IPv6 compatibili con IPv4" (prefisso 0000::/96) sono supportati ma deprecati da RFC4291. Gli spazi sono consentiti.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obbligatorio: sì, se StartAgents non è esplicitamente impostato su 0

ServerActive

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 opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo 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 dovrebbe essere specificato più di un proxy Zabbix per ciascun server/cluster Zabbix. Se viene specificato un proxy Zabbix, il server/cluster Zabbix per quel proxy non deve essere specificato.
È possibile fornire più indirizzi separati da virgola per utilizzare in parallelo diversi server Zabbix indipendenti. Gli spazi sono consentiti.
Se la porta non è specificata, viene utilizzata la porta predefinita.
Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se per quel 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

L'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;
- 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 a 0, i controlli passivi vengono disabilitati e l'agent non resterà in ascolto su alcuna porta TCP.

Predefinito: 10
 Intervallo: 0-100

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire una connessione e scambiare dati con Zabbix proxy o server.

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 nei controlli attivi;
- il recupero dei dati di log tramite logfile;
- l'invio di messaggi heartbeat;
- la durata massima per i controlli `vfs.*`;
- l'utilizzo da parte dei moduli Zabbix agent;
- l'uso come fallback negli scenari in cui un server o proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli agent che hanno impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o di singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Quali connessioni in ingresso accettare. Utilizzato per i controlli passivi. È possibile specificare più valori, separati da una 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 del PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCAFile

Il percorso completo del file contenente i certificati della/e CA di primo livello per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSCertFile

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

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 certificato e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

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 certificato 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

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 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

La stringa di cifratura per OpenSSL 1.1.1 o versioni successive in TLS 1.3. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.

TLSCipherPSK

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

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 PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Come l'agent deve connettersi al server o al proxy Zabbix. Utilizzato 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 i parametri del certificato TLS o PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCRLFile

Il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni crittografate tra i componenti di Zabbix.

TLSKeyFile

Il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni cifrate tra i componenti di Zabbix.

TLSPSKFile

Il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSPSKIdentity

La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con Zabbix server.

TLSServerCertIssuer

L'autorità emittente del certificato del server (proxy) consentita.

TLSServerCertSubject

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

Abbandona i privilegi a favore di un utente specifico, già esistente nel sistema.
Ha effetto solo se eseguito come 'root' e se AllowRoot è disabilitato.

Predefinito: `zabbix`

UserParameter

Un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: UserParameter=<key>,<shell command>
Si noti 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

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

4 Agente Zabbix 2 (UNIX)

Panoramica

Zabbix agent 2 è una nuova generazione di Zabbix agent e può essere utilizzato al posto di Zabbix agent.

I parametri supportati dal file di configurazione di Zabbix agent 2 (zabbix_agent2.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare tutti i dettagli.

Parametro	Descrizione
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un modello.
BufferSend	Non mantenere i dati nel buffer per più di N secondi.
BufferSize	Il numero massimo di valori nel buffer di memoria.
ControlSocket	Il socket di controllo, utilizzato per inviare comandi di runtime con l'opzione '-R'.
DebugLevel	Il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un modello.
EnablePersistentBuffer	Abilita l'uso dell'archiviazione persistente locale per gli item attivi.
ForceActiveChecksOnStart	Esegue i controlli attivi immediatamente dopo il riavvio per la prima configurazione ricevuta.
HeartbeatFrequency	La frequenza dei messaggi heartbeat in secondi.
HostInterface	Parametro facoltativo che definisce l'interfaccia host.
HostInterfaceItem	Parametro facoltativo che definisce un item utilizzato per ottenere l'interfaccia host.
HostMetadata	Parametro facoltativo che definisce i metadati dell'host.
HostMetadataItem	Parametro facoltativo che definisce un item di Zabbix agent utilizzato per ottenere i metadati dell'host.
Hostname	Parametro facoltativo che definisce il nome host.
HostnameItem	Parametro facoltativo che definisce un item di Zabbix agent utilizzato per ottenere il nome host.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
ListenIP	Un elenco di indirizzi IP separati da virgole sui quali l'agent deve restare in ascolto.
ListenPort	L'agent resterà in ascolto su questa porta per 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.
PersistentBufferFile	Il file in cui Zabbix agent 2 deve mantenere il database SQLite.
PersistentBufferPeriod	Il periodo di tempo per il quale i dati devono essere archiviati quando non c'è connessione al server o al proxy.
PidFile	Il nome del file PID.
Plugins.<PluginName>.System.Capacity	Il limite di controlli per plugin che possono essere eseguiti contemporaneamente.
Plugins.Log.MaxLinesPerSecond	Il numero massimo di nuove righe che l'agent invierà al server o proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Abilita la registrazione come avvisi dei comandi shell eseguiti.
PluginSocket	Il percorso del socket UNIX per le comunicazioni con i plugin caricabili.
PluginTimeout	Il timeout per le connessioni con i plugin caricabili, in secondi.
RefreshActiveChecks	Con quale frequenza viene aggiornato l'elenco dei controlli attivi.
Server	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di server Zabbix e proxy Zabbix.
ServerActive	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	L'indirizzo IP di origine.
StatusPort	Se impostato, l'agent resterà in ascolto su questa porta per richieste di stato HTTP (http://localhost:<port>/status).
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con il proxy o server Zabbix.
TLSAccept	Quali connessioni in ingresso accettare.
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 cifrate tra componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCipherAll	La stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato e PSK.
TLSCipherAll13	La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato e PSK.
TLSCipherCert	La stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato.

Parametro	Descrizione
TLSCipherCert13	La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato.
TLSCipherPSK	La stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su PSK.
TLSCipherPSK13	La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su PSK.
TLSConnect	Come l'agent deve connettersi al server o proxy Zabbix.
TLSCRLFile	Il percorso completo di un file contenente certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata dell'agent, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSPSKFile	Il percorso completo di un file contenente la chiave precondivisa dell'agent, utilizzato per le comunicazioni cifrate con il server Zabbix.
TLSPSKIdentity	La stringa identificativa della chiave precondivisa, utilizzata per le comunicazioni cifrate con il server Zabbix.
TLSServerCertIssuer	L'emittente del certificato del server (proxy) consentito.
TLSServerCertSubject	Il soggetto del certificato del server (proxy) consentito.
UnsafeUserParameters	Consente che tutti i caratteri vengano passati negli argomenti ai parametri definiti dall'utente.
UserParameter	Un parametro definito dall'utente da monitorare.
UserParameterDir	Il percorso di ricerca predefinito per i comandi UserParameter.

Tutti i parametri sono facoltativi, a meno che non sia 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 distribuiti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta 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

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più corta 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 la chiave abbreviata **zabbix.userid** può essere utilizzata per recuperare i dati.

Esempio 2: ottenimento 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, nonché di usare **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]**, ecc.

AllowKey

Consente l'esecuzione di quelle chiavi item che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con DenyKey. I parametri vengono elaborati uno per uno secondo il loro ordine di apparizione. Vedi anche: Limitazione dei controlli agent.

BufferSend

L'intervallo di tempo in secondi che determina con quale frequenza i valori vengono inviati dal buffer a Zabbix server. Si noti che, se il buffer è pieno, i dati verranno inviati prima.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Il numero massimo di valori nel buffer di memoria. L'agent invierà tutti i dati raccolti al server o al proxy Zabbix se il buffer è pieno. Questo parametro deve essere utilizzato solo se il buffer persistente è disabilitato (*EnablePersistentBuffer=0*).

Predefinito: 1000
 Intervallo: 2-65535

ControlSocket

Il socket di controllo, utilizzato per inviare comandi di runtime con l'opzione '-R'.

Predefinito: `/tmp/agent.sock`

DebugLevel

Specifica il livello di debug:
0 - informazioni di base sull'avvio e l'arresto dei processi di 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 di quelle chiavi item che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con AllowKey. I parametri vengono elaborati uno per uno in base al loro ordine di apparizione. Vedi anche: Limitazione dei controlli agent.

EnablePersistentBuffer

Abilita l'utilizzo dell'archiviazione persistente locale per gli item attivi. Se l'archiviazione persistente è disabilitata, verrà utilizzato il buffer di memoria.

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 plugin, ad esempio: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

HeartbeatFrequency

La frequenza dei messaggi heartbeat in secondi. Utilizzata per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Un parametro opzionale che definisce l'interfaccia del host (indirizzo IP o nome DNS) utilizzata durante il processo di autoregistration del host. Questo valore verrà utilizzato per popolare l'interfaccia nel host appena creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per maggiori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent genererà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Un parametro opzionale che definisce un item utilizzato per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante il processo di autoregistration del host. Questo valore viene utilizzato solo se HostInterface non è definito. Per maggiori dettagli, vedere Using DNS as default interface.

Durante una richiesta di autoregistration, 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 di AllowKey/DenyKey.

HostMetadata

Un parametro facoltativo che definisce i metadati utilizzati per identificare o distinguere l'host durante il processo di autoregistration dell'host. HostMetadata consente di distinguere tra host oltre il nome host.

Se non è definito, il valore verrà acquisito da HostMetadataItem.

L'agent segnalerà un errore e non si avvierà se il valore specificato supera il limite di 2034 byte o è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno rifiutati e segnalati come non validi.

Intervallo: 0-2034 byte

HostMetadataItem

Un parametro facoltativo che definisce un item di Zabbix agent utilizzato 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 ed è utilizzato solo nel processo di autoregistrazione dell'host. HostMetadataItem consente di distinguere tra host oltre il nome host.

Sono supportati i parametri utente e gli alias. L'item system.run[] è supportato indipendentemente dalle impostazioni di AllowKey/DenyKey.

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, altrimenti verrà ignorato. Quando un parametro si aspetta un indirizzo IP o un nome DNS, i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno anch'essi rifiutati e segnalati come non validi.

Hostname

Un elenco di nomi host univoci, sensibili alle maiuscole/minuscole, delimitati da virgole. Obbligatorio per i controlli attivi e deve corrispondere ai nomi host come configurati sul server. Il valore viene acquisito da HostnameItem se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per nome host, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Un parametro opzionale che definisce un item utilizzato per ottenere il nome del 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/DenyKey.

Predefinito: `system.hostname`

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione. Durante l'installazione Zabbix creerà la directory di inclusione in /usr/local/etc, a meno che non venga modificata in fase di compilazione. Il percorso può essere relativo alla posizione del file *zabbix_agent2.conf*.
Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali relative alle limitazioni.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Un elenco di indirizzi IP separati da virgole sui quali l'agent deve mettersi in ascolto. Il primo indirizzo IP viene inviato al server Zabbix, se si connette ad esso, per recuperare l'elenco dei controlli attivi.

Predefinito: `0.0.0.0`

ListenPort

L'agent resterà in ascolto su questa porta per le connessioni dal server.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Il nome del file di log.

Predefinito: `/tmp/zabbix_agent2.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 ricominciato da capo.

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`

PersistentBufferFile

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

Il periodo di tempo per il quale i dati devono essere memorizzati quando non c'è connessione al server o al proxy. I dati meno recenti 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

Il nome del file PID.

Predefinito: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

Il limite di controlli per il plugin `<PluginName>` che possono essere eseguiti contemporaneamente.

Predefinito: 1000 Intervallo: 1-1000

Plugins.Log.MaxLinesPerSecond

Il numero massimo di nuove righe che l'agent invierà al server o al proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'. Il valore fornito verrà sovrascritto dal parametro 'maxlines', specificato nella chiave item 'log' e '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

Plugins.SystemRun.LogRemoteCommands

Abilita la registrazione come avvisi dei comandi shell eseguiti. I comandi verranno registrati solo se eseguiti da 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

Il percorso del socket UNIX per le comunicazioni con i plugin caricabili.

Predefinito: `/tmp/agent.plugin.sock`

PluginTimeout

Il timeout per le connessioni con i plugin caricabili, in secondi.

Predefinito: `Timeout`
 Intervallo: 1-30

RefreshActiveChecks

Con quale frequenza viene aggiornato l'elenco dei controlli attivi, in secondi. Si noti che, dopo un tentativo non riuscito di aggiornare i controlli attivi, il successivo tentativo di aggiornamento verrà effettuato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server Zabbix o 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. Gli spazi sono consentiti. Se questo parametro non è specificato, i controlli passivi vengono disabilitati e l'agent non resterà in ascolto su alcuna porta TCP.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

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 opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo 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 dovrebbe essere specificato più di un proxy Zabbix per ciascun server/cluster Zabbix. Se viene specificato un proxy Zabbix, il server/cluster Zabbix per quel proxy non deve essere specificato.
È possibile fornire più indirizzi delimitati da virgola per utilizzare in parallelo diversi server Zabbix indipendenti. Gli spazi sono consentiti.
Se la porta non è specificata, viene utilizzata 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

L'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy.
- stabilire connessioni durante l'esecuzione di alcuni item (web.page.get, net.tcp.port, ecc.).

StatusPort

Se impostato, l'agent resterà in ascolto su questa porta per le richieste di stato HTTP (http://localhost:<port>/status).

Intervallo: 1024-32767

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire una connessione e scambiare dati con Zabbix proxy o server.

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 nei controlli attivi;
- il recupero dei dati di log tramite logfile;
- l'invio di messaggi heartbeat;
- la durata massima per i controlli `vfs.*`;
- l'utilizzo come fallback negli scenari in cui un server o proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli agent che hanno impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o di singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Le connessioni in ingresso da accettare. Utilizzato 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 PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCAFile

Il percorso completo del file contenente i certificati della/e CA di primo livello per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSCertFile

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

La stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato e PSK.

Esempio:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificato e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

La stringa di cifratura OpenSSL (TLS 1.2). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.

Esempio:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Si noti che questo parametro non può essere utilizzato insieme a `TLSAccept=cert,psk`; per la connessione con certificato (`TLSConnect=cert`) utilizzare invece TLSCipherAll.

TLSCipherCert13

La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive i criteri di selezione predefiniti della suite di cifratura per la crittografia basata su certificato.

Si noti che questo parametro non può essere utilizzato insieme a `TLSAccept=cert,psk`; per la connessione con certificato (`TLSConnect=cert`) utilizzare invece TLSCipherAll13.

TLSCipherPSK

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:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

La stringa di cifratura OpenSSL (TLS 1.3) per OpenSSL 1.1.1 o versioni successive. Sovrascrive 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

Come l'agent deve connettersi al server o al proxy Zabbix. Utilizzato 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 i parametri del certificato TLS o PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCRLFile

Il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSKeyFile

Il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni crittografate con Zabbix server.

TLSPSKIdentity

La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con Zabbix server.

TLSServerCertIssuer

L'autorità emittente del certificato del server (proxy) consentita.

TLSServerCertSubject

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

Un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: UserParameter=<key>,<shell command>
Si noti 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

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
```

####5 Agente Zabbix (Windows) {#manual-appendix-config-zabbix_agentd_win}

Panoramica

I parametri supportati dal file di configurazione dell'agent Zabbix per Windows (zabbix_agentd.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare tutti i dettagli.

Parameter	Description
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un modello.
BufferSend	Non mantenere i dati nel buffer per più di N secondi.
BufferSize	Il numero massimo di valori nel buffer di memoria.
DebugLevel	Il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un modello.
EnableRemoteCommands	Indica se i comandi remoti dal server Zabbix sono consentiti.
HeartbeatFrequency	La frequenza dei messaggi heartbeat in secondi.
HostInterface	Un parametro facoltativo che definisce l'interfaccia host.
HostInterfaceItem	Un parametro facoltativo che definisce un item utilizzato per ottenere l'interfaccia host.
HostMetadata	Un parametro facoltativo che definisce i metadati dell'host.
HostMetadataItem	Un parametro facoltativo che definisce un item dell'agent Zabbix utilizzato per ottenere i metadati dell'host.

Parameter	Description
Hostname	Un parametro facoltativo che definisce il nome host.
HostnameItem	Un parametro facoltativo che definisce un item dell'agent Zabbix utilizzato per ottenere il nome host.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
ListenBacklog	Il numero massimo di connessioni in attesa nella coda TCP.
ListenIP	Un elenco di indirizzi IP separati da virgole sui quali l'agent deve restare in ascolto.
ListenPort	L'agent resterà in ascolto su questa porta per le connessioni dal server.
LogFile	Il nome del file di log.
LogFileSize	La dimensione massima del file di log.
LogRemoteCommands	Abilita la registrazione come avvisi dei comandi shell eseguiti.
LogType	Il tipo di output del log.
MaxLinesPerSecond	Il numero massimo di nuove righe che l'agent invierà al server o proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'.
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 perfcounter devono essere in inglese.
RefreshActiveChecks	Con quale frequenza viene aggiornato l'elenco dei controlli attivi.
Server	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure nomi DNS di server Zabbix e proxy Zabbix.
ServerActive	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	L'indirizzo IP sorgente.
StartAgents	Il numero di istanze pre-fork di zabbix_agentd che elaborano i controlli passivi.
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con proxy o server Zabbix.
TLSAccept	Quali connessioni in ingresso accettare.
TLSCAFile	Il percorso completo di un file contenente i certificati della/e CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSConnect	Come l'agent deve connettersi al server o proxy Zabbix.
TLSCRLFile	Il percorso completo di un file contenente i certificati revocati. Questo parametro è utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata dell'agent, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSPSKFile	Il percorso completo di un file contenente la chiave precondivisa dell'agent, utilizzato per le comunicazioni cifrate con il server Zabbix.
TLSPSKIdentity	La stringa identificativa della chiave precondivisa, utilizzata per le comunicazioni cifrate con il server Zabbix.
TLSServerCertIssuer	L'emittente consentito del certificato del server (proxy).
TLSServerCertSubject	Il soggetto consentito del certificato del server (proxy).
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti ai parametri definiti dall'utente.
UserParameter	Un parametro definito dall'utente da monitorare.
UserParameterDir	Il percorso di ricerca predefinito per i comandi UserParameter.

Tutti i parametri sono facoltativi, salvo nei casi in cui sia esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del demone, non i valori presenti nei file di configurazione distribuiti.
- I valori supportano le variabili d'ambiente.
- Zabbix supporta 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

Alias

Imposta un alias per una chiave item. Può essere utilizzato per sostituire una chiave item lunga e complessa con una più corta 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* 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 la chiave abbreviata **pg_usage** può essere utilizzata per recuperare i dati.

Esempio 2: ottenimento 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, nonché di usare **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]**, ecc.

AllowKey

Consente l'esecuzione di quelle chiavi di item che corrispondono a un modello. Il modello della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con DenyKey. I parametri vengono elaborati uno per uno in base al loro ordine di apparizione. Vedi anche: Limitazione dei controlli dell'agent.

BufferSend

Non conservare i dati nel buffer per più di N secondi.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Il numero massimo di valori nel buffer di memoria. L'agent invierà tutti i dati raccolti al server o al proxy Zabbix se il buffer è pieno.

Predefinito: 100
 Intervallo: 2-65535

DebugLevel

Specifica il livello di debug:
0 - informazioni di base sull'avvio e l'arresto dei processi di 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 di quelle chiavi item che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con AllowKey. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. Vedi anche: Limitazione dei controlli agent.

EnableRemoteCommands

Se i comandi remoti da Zabbix server sono consentiti. Questo parametro è **deprecato**; utilizzare invece AllowKey=system.run[*] o DenyKey=system.run[*].
È un alias interno per i parametri AllowKey/DenyKey a seconda del valore:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Predefinito: 0
 Valori: 0 - non consentire, 1 - consentire

HeartbeatFrequency

La frequenza dei messaggi heartbeat in secondi. Utilizzata per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Un parametro opzionale che definisce l'interfaccia del host (indirizzo IP o nome DNS) utilizzata durante il processo di autoregistration del host. Questo valore verrà usato per popolare l'interfaccia nel host appena creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per maggiori 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

Un parametro facoltativo che definisce un item utilizzato per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante il processo di autoregistration. Questo valore viene utilizzato solo se HostInterface non è definito. Per maggiori dettagli, vedere Using DNS as default interface.

Durante una richiesta di autoregistration, 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 di AllowKey/DenyKey.

HostMetadata

Un parametro opzionale che definisce i metadati utilizzati per identificare o distinguere l'host durante il processo di autoregistration dell'host (agent attivo). HostMetadata consente di distinguere tra 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 è una stringa non UTF-8. Quando un parametro prevede un indirizzo IP o un nome DNS, anche i valori che sono UTF-8 validi ma non sono indirizzi 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

Un parametro facoltativo che definisce un item dell'agent Zabbix utilizzato per ottenere i metadati del 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 nel processo di autoregistrazione del host (agent attivo). HostMetadataItem consente di distinguere tra 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 di AllowKey/DenyKey.

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, altrimenti verrà ignorato. Quando un parametro prevede un indirizzo IP o un nome DNS, i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno anch'essi rifiutati e segnalati come non validi.

I metadati su più righe non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Hostname

Un elenco di nomi host univoci, con distinzione tra maiuscole e minuscole, separati da virgole. Obbligatorio per i controlli attivi e deve corrispondere ai nomi host come configurati sul server. Il valore viene acquisito da HostnameItem se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per nome host, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Un parametro opzionale che definisce un item di Zabbix agent utilizzato per ottenere il nome del host. Questa opzione viene utilizzata solo quando Hostname non è definito. I parametri utente, i contatori delle prestazioni o gli alias non sono supportati, ma l'item system.run[] è supportato indipendentemente dai valori di AllowKey/DenyKey.
Vedi anche una descrizione più dettagliata.

Predefinito: `system.hostname`

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione (situato in `C:\Program Files\Zabbix Agent` per impostazione predefinita se Zabbix agent è installato utilizzando i pacchetti di installazione Windows MSI; situato nella cartella specificata durante l'installazione se Zabbix agent è installato come archivio zip). Tutti i file inclusi devono avere una

sintassi corretta, altrimenti agent non si avvierà.
Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali sulle limitazioni.

Esempio:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Il numero massimo di connessioni in attesa 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

Un elenco di indirizzi IP delimitati da virgole sui quali l'agent deve restare in ascolto.

Predefinito: 0.0.0.0

ListenPort

L'agent rimarrà in ascolto su questa porta per le connessioni dal server.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Il nome del file di log dell'agent.

Predefinito: c:\zabbix_agentd.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 ricominciato da capo.

Predefinito: 1
 Intervallo: 0-1024

LogRemoteCommands

Abilita la registrazione come avvisi dei comandi shell eseguiti. I comandi verranno registrati solo se eseguiti da 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

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

Il numero massimo di nuove righe che l'agent invierà al server o al proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log', 'logrt' ed 'eventlog'. Il valore fornito verrà sovrascritto dal parametro 'maxlines', specificato nella chiave 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

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).
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:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Si notino le virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere utilizzato come chiave dell'item durante la creazione di un item. I campioni per il calcolo del valore medio verranno acquisiti ogni secondo.
È possibile eseguire "typeperf -qx" per ottenere l'elenco di tutti i contatori delle prestazioni disponibili in Windows.

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. 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 notino le virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere utilizzato come chiave dell'item durante la creazione di un item. I campioni per il calcolo del valore medio verranno acquisiti ogni secondo.
È possibile trovare l'elenco delle stringhe in inglese visualizzando la seguente chiave di registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Con quale frequenza viene aggiornato l'elenco dei controlli attivi, in secondi. Si noti che, dopo un tentativo non riuscito di aggiornare i controlli attivi, il successivo tentativo di aggiornamento verrà effettuato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server Zabbix o 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. Si noti che gli "indirizzi IPv6 compatibili con IPv4" (prefisso 0000::/96) sono supportati ma deprecati da RFC4291. Gli spazi sono consentiti.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obbligatorio: sì, se StartAgents non è esplicitamente impostato su 0

ServerActive

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 opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo 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 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 virgola per utilizzare in parallelo diversi server Zabbix indipendenti. Gli spazi sono consentiti.
Se la porta non è specificata, viene utilizzata la porta predefinita.
Gli indirizzi IPv6 devono essere racchiusi tra parentesi quadre se per quel 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
```

Intervallo: (*)

SourceIP

L'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;
- l'esecuzione di 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 a 0, i controlli passivi vengono disabilitati e l'agent non resterà in ascolto 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 nei controlli attivi;
- recupero dei dati di log tramite il monitoraggio di logfile o del registro eventi di Windows;
- invio di messaggi heartbeat;
- durata massima per i controlli `vfs.*`;
- utilizzato dai moduli di Zabbix agent;
- usato come fallback negli scenari in cui un server o proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli agent che hanno impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o per-item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Le connessioni in ingresso da accettare. Utilizzato 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 PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCAFile

Il percorso completo del file contenente i certificati della CA o delle CA di primo livello 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 dell'agent o la catena di certificati, utilizzato per le comunicazioni crittografate con i componenti di Zabbix.

TLSConnect

Come l'agent deve connettersi al server o al proxy Zabbix. Utilizzato 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 i parametri del certificato TLS o della PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCRLFile

Il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSKeyFile

Il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni crittografate tra i componenti di Zabbix.

TLSPSKFile

Il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSPSKIdentity

La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSServerCertIssuer

L'autorità emittente del certificato del server (proxy) consentita.

TLSServerCertSubject

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

Un parametro definito dall'utente da monitorare.
Possono esserci diversi parametri definiti dall'utente.
Formato: UserParameter=<key>,<shell command>
Si noti 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,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

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
```

Note:
(*) Il numero di server attivi elencati in ServerActive più il numero di istanze pre-fork per i controlli passivi specificato in StartAgents deve essere inferiore a 64.

6 Agente Zabbix 2 (Windows)

Panoramica

Zabbix agent 2 è una nuova generazione di Zabbix agent e può essere utilizzato al posto di Zabbix agent.

I parametri supportati dal file di configurazione di Zabbix agent 2 per Windows (zabbix_agent2.conf) sono elencati in questa sezione.

I parametri sono elencati senza informazioni aggiuntive. Fare clic sul parametro per visualizzare tutti i dettagli.

Parameter	Description
Alias	Imposta un alias per una chiave item.
AllowKey	Consente l'esecuzione delle chiavi item che corrispondono a un modello.
BufferSend	Non mantenere i dati nel buffer per più di N secondi.
BufferSize	Il numero massimo di valori nel buffer di memoria.
ControlSocket	Il socket di controllo, utilizzato per inviare comandi di runtime con l'opzione '-R'.
DebugLevel	Il livello di debug.
DenyKey	Nega l'esecuzione delle chiavi item che corrispondono a un modello.
EnablePersistentBuffer	Abilita l'uso dell'archiviazione persistente locale per gli item attivi.
ForceActiveChecksOnStart	Esegue i controlli attivi immediatamente dopo il riavvio per la prima configurazione ricevuta.
HeartbeatFrequency	La frequenza dei messaggi heartbeat in secondi.
HostInterface	Un parametro facoltativo che definisce l'interfaccia host.
HostInterfaceItem	Un parametro facoltativo che definisce un item utilizzato per ottenere l'interfaccia host.
HostMetadata	Un parametro facoltativo che definisce i metadati dell'host.
HostMetadataItem	Un parametro facoltativo che definisce un item di Zabbix agent utilizzato per ottenere i metadati dell'host.
Hostname	Un parametro facoltativo che definisce il nome host.
HostnameItem	Un parametro facoltativo che definisce un item di Zabbix agent utilizzato per ottenere il nome host.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
ListenIP	Un elenco di indirizzi IP separati da virgole sui quali l'agent deve restare in ascolto.
ListenPort	L'agent resterà in ascolto su questa porta per le connessioni dal server.
LogFile	Il nome del file di log.

Parameter	Description
LogFileSize	La dimensione massima del file di log.
LogType	Il tipo di output del log.
PersistentBufferFile	Il file in cui Zabbix agent 2 deve conservare il database SQLite.
PersistentBufferPeriod	Il periodo di tempo per il quale i dati devono essere memorizzati quando non c'è connessione al server o al proxy.
Plugins.<PluginName>.System.Capacity	Il limite di controlli per plugin che possono essere eseguiti contemporaneamente.
Plugins.Log.MaxLinesPerSecond	Il numero massimo di nuove righe che l'agent invierà al server o proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log' e 'logrt'.
Plugins.SystemRun.LogRemoteCommands	Abilita la registrazione come avvisi dei comandi shell eseguiti.
PluginSocket	Il percorso del socket UNIX per le comunicazioni con i plugin caricabili.
PluginTimeout	Il timeout per le connessioni con i plugin caricabili, in secondi.
PerfCounter	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).
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 perfcounter devono essere in inglese.
RefreshActiveChecks	Con quale frequenza viene aggiornato l'elenco dei controlli attivi.
Server	Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, o nomi DNS di server Zabbix e proxy Zabbix.
ServerActive	L'indirizzo del server/proxy Zabbix o la configurazione del cluster da cui ottenere i controlli attivi.
SourceIP	L'indirizzo IP sorgente.
StatusPort	Se impostato, l'agent resterà in ascolto su questa porta per le richieste di stato HTTP (http://localhost:<port>/status).
Timeout	Specifica per quanto tempo attendere (in secondi) per stabilire la connessione e scambiare dati con il proxy o server Zabbix.
TLSAccept	Quali connessioni in ingresso accettare.
TLSCAFile	Il percorso completo di un file contenente i certificati della/e CA di livello superiore per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato dell'agent o la catena di certificati, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSCipherAll13	La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato e PSK.
TLSCipherCert13	La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su certificato.
TLSCipherPSK13	La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la cifratura basata su PSK.
TLSConnect	Come l'agent deve connettersi al server o proxy Zabbix.
TLSCRLFile	Il percorso completo di un file contenente certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata dell'agent, utilizzato per le comunicazioni cifrate tra componenti Zabbix.
TLSPSKFile	Il percorso completo di un file contenente la chiave precondivisa dell'agent, utilizzato per le comunicazioni cifrate con il server Zabbix.
TLSPSKIdentity	La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni cifrate con il server Zabbix.
TLSServerCertIssuer	L'emittente del certificato del server (proxy) consentito.
TLSServerCertSubject	Il soggetto del certificato del server (proxy) consentito.
UnsafeUserParameters	Consente il passaggio di tutti i caratteri negli argomenti ai parametri definiti dall'utente.
UserParameter	Un parametro definito dall'utente da monitorare.
UserParameterDir	Il percorso di ricerca predefinito per i comandi UserParameter.

Tutti i parametri sono facoltativi, salvo ove esplicitamente indicato che il parametro è obbligatorio.

Si noti che:

- I valori predefiniti riflettono i valori predefiniti del processo, non i valori nei file di configurazione distribuiti;
- I valori supportano le variabili d'ambiente;
- Zabbix supporta 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

Alias

Imposta un alias per una chiave item. Può essere usato per sostituire una chiave item lunga e complessa con una più corta 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 usati 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 la chiave abbreviata **pg_usage** può essere usata per recuperare i dati.

Esempio 2: ottenimento del carico della CPU con parametri predefiniti e personalizzati.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Questo consente di usare la chiave **cpu.load** per ottenere il carico della CPU con i parametri predefiniti, nonché di usare **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 usando **vfs.fs.discovery** con parametri diversi per ciascuna regola, ad esempio **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]**, ecc.

AllowKey

Consente l'esecuzione di quelle chiavi item che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di qualsiasi carattere.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con DenyKey. I parametri vengono elaborati uno per uno secondo il loro ordine di apparizione. Vedi anche: Restrizione dei controlli agent.

BufferSend

L'intervallo di tempo in secondi che determina con quale frequenza i valori vengono inviati dal buffer a Zabbix server.
Si noti che, se il buffer è pieno, i dati verranno inviati prima.

Predefinito: 5
 Intervallo: 1-3600

BufferSize

Il numero massimo di valori nel buffer di memoria. L'agent invierà tutti i dati raccolti al server o al proxy Zabbix se il buffer è pieno.
Questo parametro deve essere utilizzato solo se il buffer persistente è disabilitato (*EnablePersistentBuffer=0*).

Predefinito: 1000
 Intervallo: 2-65535

ControlSocket

Il socket di controllo, utilizzato per inviare comandi di runtime con l'opzione '-R'.

Predefinito: `\\.\pipe\agent.sock`

DebugLevel

Specifica il livello di debug:
0 - informazioni di base sull'avvio e l'arresto dei processi di 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 di quelle chiavi item che corrispondono a un pattern. Il pattern della chiave è un'espressione con caratteri jolly che supporta il carattere "*" per corrispondere a un numero qualsiasi di caratteri qualsiasi.
È possibile definire più regole di corrispondenza delle chiavi in combinazione con AllowKey. I parametri vengono elaborati uno per uno in base all'ordine in cui compaiono. Vedi anche: Limitazione dei controlli agent.

EnablePersistentBuffer

Abilita l'utilizzo dell'archiviazione persistente locale per gli item attivi. Se l'archiviazione persistente è disabilitata, verrà utilizzato il buffer di memoria.

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 plugin, ad esempio: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Predefinito: 0
 Valori: 0 - disabilitato, 1 - abilitato

HeartbeatFrequency

La frequenza dei messaggi heartbeat in secondi. Utilizzata per monitorare la disponibilità dei controlli attivi.
0 - messaggi heartbeat disabilitati.

Predefinito: 60
 Intervallo: 0-3600

HostInterface

Un parametro opzionale che definisce l'interfaccia del host (indirizzo IP o nome DNS) utilizzata durante il processo di autoregistration del host. Questo valore verrà usato per popolare l'interfaccia nel host appena creato e consente di configurare esplicitamente un indirizzo IP o DNS. Per maggiori dettagli, vedere Using DNS as default interface.

Se non definito, il valore verrà acquisito da HostInterfaceItem.

L'agent genererà un errore e non si avvierà se il valore supera il limite di 255 caratteri.

Intervallo: 0-255 caratteri

HostInterfaceItem

Un parametro opzionale che definisce un item utilizzato per determinare l'interfaccia del host (indirizzo IP o nome DNS) durante il processo di autoregistration del host. Questo valore viene utilizzato solo se HostInterface non è definito. Per maggiori dettagli, vedere Using DNS as default interface.

Durante una richiesta di autoregistration, 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 di AllowKey/DenyKey.

HostMetadata

Un parametro facoltativo che definisce i metadati utilizzati per identificare o distinguere l'host durante il processo di autoregistration dell'host (agent attivo). HostMetadata consente di distinguere tra host oltre il nome host.

Se non è definito, il valore verrà acquisito da HostMetadataItem.

L'agent segnalerà 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 che sono UTF-8 validi ma non sono indirizzi 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

Un parametro opzionale che definisce un item di Zabbix agent utilizzato 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 ed è utilizzato solo nel processo di autoregistrazione dell'host (agent attivo). HostMetadataItem consente di distinguere tra host oltre il nome host.

I parametri utente e gli alias sono supportati. L'item system.run[] è supportato indipendentemente dalle impostazioni di AllowKey/DenyKey.

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, altrimenti verrà ignorato. Quando un parametro prevede un indirizzo IP o un nome DNS, i valori che sono UTF-8 validi ma non sono indirizzi IP o nomi DNS validi verranno anch'essi rifiutati e segnalati come non validi.

I metadati multilinea non sono supportati: l'output verrà troncato alla prima interruzione di riga.

Hostname

Un elenco di nomi host univoci, con distinzione tra maiuscole e minuscole, separati da virgole. Obbligatorio per i controlli attivi e deve corrispondere ai nomi host come configurati sul server. Il valore viene acquisito da HostnameItem se non definito.
Caratteri consentiti: alfanumerici, '.', ' ', '_' e '-'. Lunghezza massima: 128 caratteri per nome host, 2048 caratteri per l'intera riga.

Predefinito: impostato da HostnameItem

HostnameItem

Un parametro opzionale che definisce un item utilizzato per ottenere il nome del 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/DenyKey.

Predefinito: `system.hostname`

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione (situato in `C:\Program Files\Zabbix Agent 2` per impostazione predefinita se Zabbix agent è installato utilizzando i pacchetti di installazione Windows MSI; situato nella cartella specificata durante l'installazione se Zabbix agent è installato come archivio zip). Tutti i file inclusi devono avere una sintassi corretta, altrimenti l'agent non si avvierà. Il percorso può essere relativo alla posizione del file *zabbix_agent2.conf* (ad esempio, `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali sulle limitazioni.

Esempio:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Un elenco di indirizzi IP separati da virgole sui quali l'agent deve mettersi in ascolto. Il primo indirizzo IP viene inviato al server Zabbix, se si connette ad esso, per recuperare l'elenco dei controlli attivi.

Predefinito: `0.0.0.0`

ListenPort

L'agent resterà in ascolto su questa porta per le connessioni dal server.

Predefinito: 10050
 Intervallo: 1024-32767

LogFile

Il nome del file di log dell'agent.

Predefinito: `c:\zabbix_agent2.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 ricominciato da capo.

Predefinito: 1
 Intervallo: 0-1024

LogType

Il tipo di output del log:
file - scrive il log nel file specificato dal parametro LogFile;
console - scrive il log nell'output standard.

Predefinito: `file`

PersistentBufferFile

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

Il periodo di tempo per il quale i dati devono essere memorizzati quando non c'è connessione al server o al proxy. I dati meno recenti 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

Plugins.<PluginName>.System.Capacity

Il limite di controlli per plugin `<PluginName>` che possono essere eseguiti contemporaneamente.

Predefinito: 1000 Intervallo: 1-1000

Plugins.Log.MaxLinesPerSecond

Il numero massimo di nuove righe che l'agent invierà al server o al proxy Zabbix al secondo durante l'elaborazione dei controlli attivi 'log', 'logrt' ed 'eventlog'. Il valore fornito verrà sovrascritto dal parametro 'maxlines', specificato nella chiave 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 come avvisi dei comandi shell eseguiti. I comandi verranno registrati solo se eseguiti da 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

Il percorso del socket UNIX per le comunicazioni dei plugin caricabili.

Predefinito: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Il timeout per le connessioni con i plugin caricabili, in secondi.

Predefinito: `Timeout`
 Intervallo: 1-30

PerfCounter

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).
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:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Si notino le virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere utilizzato come chiave dell'item durante la creazione di un item. I campioni per il calcolo del valore medio verranno acquisiti ogni secondo.
È possibile eseguire "typeperf -qx" per ottenere l'elenco di tutti i contatori delle prestazioni disponibili in Windows.

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 del contatore delle prestazioni 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 notino le virgolette doppie attorno al percorso del contatore delle prestazioni. Il nome del parametro (interrupts) deve essere utilizzato come chiave dell'item durante la creazione di un item. I campioni per il calcolo del valore medio verranno acquisiti ogni secondo.
È possibile trovare l'elenco delle stringhe in inglese visualizzando la seguente chiave di registro: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Con quale frequenza viene aggiornato l'elenco dei controlli attivi, in secondi. Si noti che, dopo un tentativo non riuscito di aggiornare i controlli attivi, il successivo aggiornamento verrà tentato dopo 60 secondi.

Predefinito: 5
 Intervallo: 1-86400

Server

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server Zabbix o 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. Gli spazi sono consentiti. Se questo parametro non è specificato, i controlli passivi vengono disabilitati e l'agent non resterà in ascolto su alcuna porta TCP.

Esempio:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

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 opzionale separata da due punti.
La configurazione del cluster è costituita da uno o più indirizzi dei membri del gruppo 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 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 delimitati da virgole per utilizzare in parallelo diversi server Zabbix indipendenti. Gli spazi sono consentiti.
Se la porta non è specificata, viene utilizzata 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

L'indirizzo IP sorgente per:

- le connessioni in uscita verso Zabbix server o Zabbix proxy;
- stabilire connessioni durante l'esecuzione di alcuni item (web.page.get, net.tcp.port, ecc.).

StatusPort

Se impostato, l'agent resterà in ascolto su questa porta per le richieste di stato HTTP (http://localhost:<port>/status).

Intervallo: 1024-32767

Timeout

Specifica per quanto tempo attendere (in secondi) per stabilire una 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 nei controlli attivi;
- recupero dei dati di log tramite il monitoraggio di logfile 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 proxy precedente alla versione 7.0 invia controlli senza timeout.

Questo timeout **non** verrà utilizzato per quei controlli agent che hanno impostazioni di timeout configurabili nel frontend (a livello globale, di proxy o di singolo item).

Predefinito: 3
 Intervallo: 1-30

TLSAccept

Le connessioni in ingresso da accettare. Utilizzato 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 la connessione *unencrypted*); altrimenti no

TLSCAFile

Il percorso completo del file contenente i certificati della/e CA di primo livello per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSCertFile

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

La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione delle suite di cifratura per la crittografia basata su certificato e PSK.

Esempio:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su certificato.

Si noti che questo parametro non può essere utilizzato insieme a `TLSAccept=cert,psk`; per la connessione con certificato (`TLSConnect=cert`) utilizzare invece TLSCipherAll13.

TLSCipherPSK13

La stringa di cifratura OpenSSL (TLS 1.3). Sovrascrive i criteri predefiniti di selezione della suite di cifratura per la crittografia basata su PSK.

Esempio:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Come l'agent deve connettersi al server o al proxy Zabbix. Utilizzato 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 i parametri del certificato TLS o PSK (anche per la connessione *unencrypted*); altrimenti no

TLSCRLFile

Il percorso completo del file contenente i certificati revocati. Questo parametro viene utilizzato per le comunicazioni cifrate tra i componenti di Zabbix.

TLSKeyFile

Il percorso completo del file contenente la chiave privata dell'agent, utilizzata per le comunicazioni cifrate tra i componenti di Zabbix.

TLSPSKFile

Il percorso completo del file contenente la chiave precondivisa dell'agent, utilizzata per le comunicazioni cifrate con Zabbix server.

TLSPSKIdentity

La stringa di identità della chiave precondivisa, utilizzata per le comunicazioni crittografate con Zabbix server.

TLSServerCertIssuer

L'autorità emittente del certificato del server (proxy) consentita.

TLSServerCertSubject

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

Un parametro definito dall'utente da monitorare. Possono esserci diversi parametri definiti dall'utente.
Formato: UserParameter=<key>,<shell command>
Si noti 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,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

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
```

7 Configuration parameters of plugins

See configuration parameters for Zabbix agent 2 plugins:

- Ceph plugin
- Docker plugin
- Ember+ plugin
- Memcached plugin
- Modbus plugin
- MongoDB plugin
- MQTT plugin
- MSSQL plugin
- MySQL plugin
- NVIDIA GPU plugin
- Oracle plugin
- PostgreSQL plugin
- Redis plugin
- S.M.A.R.T. plugin

Plugin configuration principles

This page describes plugin configuration principles and best practices.

All plugins are configured using *Plugins.** parameter, which can either be part of the Zabbix agent 2 configuration file or a plugin's own configuration file. If a plugin uses a separate configuration file, path to this file should be specified in the Include parameter of Zabbix agent 2 configuration file.

A typical plugin parameter has the following structure:

Plugins.<PluginName>.<Parameter>=<Value>

Additionally, there are two specific groups of parameters:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* used for defining default parameter values.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* used for defining separate sets of parameters for different monitoring targets via named sessions.

All parameter names should adhere to the following requirements:

- it is recommended to capitalize the names of your plugins
- the parameter should be capitalized
- special characters are not allowed
- nesting isn't limited by a maximum level
- the number of parameters is not limited

For example, to perform active checks that do not have *Scheduling* update interval immediately after the agent restart only for the Uptime plugin, set `Plugins.Uptime.System.ForceActiveChecksOnStart=1` in the configuration file. Similarly, to set custom limit for concurrent checks for the CPU plugin, set the `Plugins.CPU.System.Capacity=N` in the configuration file.

Default values

You can set default values for the connection-related parameters (URI, username, password, etc.) in the configuration file in the format:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

For example, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017`, etc.

If a value for such parameter is not provided in an item key or in the named session parameters, the plugin will use the default value. If a default parameter is also undefined, hardcoded defaults will be used.

> **Note:**
> If an item key does not have any parameters, Zabbix agent 2 will attempt to collect the metric using values defined in the default parameters section.

Named sessions

Named sessions represent an additional level of plugin parameters and can be used to specify separate sets of authentication parameters for each of the instances being monitored. Each named session parameter should have the following structure:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

A session name can be used as a connString item key parameter instead of specifying a URI, username, and/or password separately.

In item keys, the first parameter can be either a connString or a URI. If the first key parameter doesn't match any session name, it will be treated as a URI. Note that passing embedded URI credentials in the item key is not supported, use named session parameters instead.

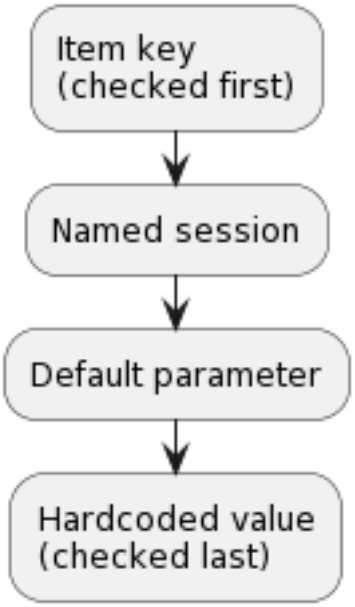
The list of available named session parameters depends on the plugin.

It is possible to override session parameters by specifying new values in the item key parameters (see example).

If a parameter is not defined for the named session, Zabbix agent 2 will use the value defined in the default plugin parameter.

Parameter priority

Zabbix agent 2 plugins search for connection-related parameter values in the following order:



1. The first item key parameter is compared to session names. If no match is found, it is treated as an actual value; in this case, step 3 will be skipped. If a match is found, the parameter value (usually, a URI) must be defined in the named session.
2. Other parameters will be taken from the item key if defined.
3. If an item key parameter (for example, password) is empty, plugin will look for the corresponding named session parameter.
4. If the session parameter is also not specified, the value defined in the corresponding default parameter will be used.
5. If all else fails, the plugin will use the hardcoded default value.

Example 1

Monitoring of two instances "MySQL1" and "MySQL2".

Configuration parameters:

```
Plugins.Mysql.Sessions.MySQL1.Uri=tcp://127.0.0.1:3306
Plugins.Mysql.Sessions.MySQL1.User=mysql1_user
Plugins.Mysql.Sessions.MySQL1.Password=unique_password
Plugins.Mysql.Sessions.MySQL2.Uri=tcp://192.0.2.0:3306
Plugins.Mysql.Sessions.MySQL2.User=mysql2_user
Plugins.Mysql.Sessions.MySQL2.Password=different_password
```

As a result of this configuration, each session name may be used as a connString in an item key, e.g., `mysql.ping[MySQL1]` or `mysql.ping[MySQL2]`.

Example 2

Providing some of the parameters in the item key.

Configuration parameters:

```
Plugins.PostgreSQL.Sessions.Session1.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Sessions.Session1.User=old_username
Plugins.PostgreSQL.Sessions.Session1.Password=session_password
```

Item key: `pgsql.ping[session1,new_username,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the following parameters:

- URI from session parameter: `192.0.2.234:5432`
- Username from the item key: `new_username`
- Password from session parameter (since it is omitted in the item key): `session_password`
- Database name from the item key: `postgres`

Example 3

Collecting a metric using default configuration parameters.

Configuration parameters:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Item key: `pgsql.ping[,,,postgres]`

As a result of this configuration, the agent will connect to PostgreSQL using the parameters:

- Default URI: `192.0.2.234:5432`
- Default username: `zabbix`
- Default password: `password`
- Database name from the item key: `postgres`

Connections

Some plugins support gathering metrics from multiple instances simultaneously. Both local and remote instances can be monitored. TCP and Unix-socket connections are supported.

It is recommended to configure plugins to keep connections to instances in an open state. The benefits are reduced network congestion, latency, and CPU and memory usage due to the lower number of connections. The client library takes care of this.

> **Note:**
> Time period for which unused connections should remain open can be determined by *Plugins.<PluginName>.KeepAlive* parameter. Example: *Plugins.Memcached.KeepAlive*

1 Ceph plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ceph plugin.

It is recommended to specify them in their own configuration file (e.g. `ceph.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with ”#” are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Ceph.Default.ApiKey	no			Default API key for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Mode	no	native / restful	restful (deprecated)	Default mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Default.User	no			Default username for connecting to Ceph; used if no value is specified in an item key or named session.
Plugins.Ceph.Default.Uri	no		https://localhost:8003	Default URI for connecting to Ceph; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify	no	false / true	false	Determines whether an http client should verify the server’s certificate chain and host name. If *true*, TLS accepts any certificate presented by the server and any host name in that certificate. In this mode, TLS is susceptible to man-in-the-middle attacks (should be used only for testing).
Plugins.Ceph.KeepAlive	no	60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	no			Named session API key. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Mode	no	native / restful	restful (deprecated)	Named session mode for connecting to Ceph. The native mode is only supported on Linux and starting with Ceph 16. The restful mode will not work with Ceph version 20 or newer. Note that the user credential set differs for each mode and they are not compatible with each other.
Plugins.Ceph.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Ceph.Sessions.<SessionName>.Uri	no			Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `https` scheme is supported; a scheme can be omitted. A port can be omitted (default=8003). Examples: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

2 Docker plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Docker plugin.

It is recommended to specify them in their own configuration file (e.g. `docker.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Docker.Endpoint			unix:///var/run/docker.sock	Docker daemon unix-socket location. Must contain a scheme (only `unix://` is supported).
Plugins.Docker.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

3 Ember+ plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Ember+ plugin.

It is recommended to specify them in their own configuration file (e.g. `emberplus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The Ember+ plugin is a loadable plugin and is available and fully described in the Ember+ plugin repository.

This plugin is currently only available to be built from the source (for both Unix and Windows).

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.EmberPlus.Default.Uri			tcp://localhost:9998	The default URI to connect. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.EmberPlus.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.EmberPlus.Sessions.<SessionName>.Uri			tcp://localhost:9998	The URI to connect, for the named session. The only supported schema is `tcp://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.EmberPlus.System.Path				Path to the Ember+ plugin executable. Example usage: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix-`
Plugins.EmberPlus.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

4 Memcached plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Memcached plugin.

It is recommended to specify them in their own configuration file (e.g. `memcached.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Memcached.Default.Password				Default password for connecting to Memcached; used if no value is specified in an item key or named session.

Plugins.Memcached.Default.Uri			tcp://localhost:11211	Default URI for connecting to Memcached; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Default.User				Default username for connecting to Memcached; used if no value is specified in an item key or named session.
Plugins.Memcached.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Memcached.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description

Plugins.Memcached.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=11211).
				Examples: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcached.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Memcached.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

5 Modbus plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Modbus plugin.

It is recommended to specify them in their own configuration file (e.g. `modbus.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint is a connection string consisting of a protocol scheme, a host address and a port or serial port name and attributes. **<SessionName>** - define name of a session for using in item keys.
Plugins.Modbus.Sessions.<SessionName>.SlaveID				Slave ID of a named session. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Note* that this named session parameter is checked only if the value provided in the item key slave ID parameter is empty.
Plugins.Modbus.Sessions.<SessionName>.Timeout				Timeout of a named session in seconds. **<SessionName>** - define name of a session for using in item keys. Example: `Plugins.Modbus.Sessions.MB1.Timeout=2`

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

6 MongoDB plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MongoDB plugin.

It is recommended to specify them in their own configuration file (e.g., `mongo.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MongoDB plugin is a loadable plugin and is available and fully described in the MongoDB plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

When connecting to MongoDB, the plugin checks its configuration in a specific order, which determines which value to use. The order, from highest to lowest priority, is as follows:

1. Parameters specified in the connection URI (e.g., query parameters defined after ? in `Plugins.MongoDB.Sessions.<session_name` have the highest priority).
2. Parameters in the Zabbix agent 2 item key (e.g., mongodb.collection.stats).
3. Parameters defined in a named session (e.g., `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Sessions`
4. Default values (e.g., `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

For example:

- If you specify the database for authenticating a connection to the MongoDB server in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=a` then the plugin will consider the `*.<SessionName>.AuthSource` parameter to be set to `admin`, even if the `*.<SessionName>.AuthSource` parameter is set to a different value.
- If you specify the URI in the `*.<SessionName>.Uri` parameter (e.g., `Plugins.MongoDB.Sessions.<session_name>.Uri=mongo` but provide the username and the password in the Zabbix agent 2 item key, the plugin will use the username and password from the item key, even if `Plugins.MongoDB.Sessions.<SessionName>.User` and `Plugins.MongoDB.Sessions.<SessionNam` are configured.

Attention:
If the `mongodb+srv://` scheme is used, the URI may retrieve parameters from a DNS TXT record. Parameters obtained from this TXT record override all other configurations, including URI parameters, Zabbix Agent 2 item key parameters, and session parameters.
 Only one TXT record may exist for the hostname used in the connection string.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.System.Path				Path to the MongoDB plugin executable. Example usage: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-ag`
Plugins.MongoDB.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).
Plugins.MongoDB.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.Uri	No		tcp://127.0.0.1:27017	Connection string of a named session. **<SessionName>** - define name of a session for using in item keys. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`; the `tcp://` scheme is deprecated and kept for backward compatibility with existing configurations. A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Sessions.<SessionName>.User	No			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Password	No			Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.Direct	No		false	Connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. **<SessionName>** - define name of a session for using in item keys. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	No			Authentication mechanism for connecting to MongoDB server. **<SessionName>** - define name of a session for using in item keys. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	No		admin	Database for authenticating connection to MongoDB server. **<SessionName>** - define name of a session for using in item keys. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	No			Replica set name, used for automatic discovery of MongoDB replica set nodes. **<SessionName>** - define name of a session for using in item keys. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference				Order for connecting to MongoDB replica set nodes or fallback. **<SessionName>** - define name of a session for using in item keys. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Sessions.<SessionName>.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: `required` - require TLS connection; `verify\_ca` - verify certificates; `verify\_full` - verify certificates and IP address. Supported since plugin version 1.2.1.
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.MongoDB.Default.Uri			`tcp://127.0.0.1:27017`	Default URI for connecting to MongoDB. Must match the URI format. Supported schemes: `mongodb://` and `mongodb+srv://`. Deprecated scheme: `tcp://` (used as the default for backward compatibility with existing configurations). A port can be omitted (default=27017). Examples: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Default.User				Default username for connecting to MongoDB.
Plugins.MongoDB.Default.Password				Default password for connecting to MongoDB.
Plugins.MongoDB.Default.Direct			`false`	Default connection method used to determine whether automatic discovery of MongoDB replica set nodes should be enabled or disabled. Supported values: `true` - the plugin will connect only to the specified server, ignoring other members of the replica set; `false` - the plugin will automatically discover all nodes in the replica set. If the `tcp://` (deprecated) scheme is used, the parameter is set to `true` by default.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.AuthMechanism				Default authentication mechanism for connecting to MongoDB server. Supported values: `SCRAM-SHA-1`, `SCRAM-SHA-256`, and `MONGODB-X509`. If no value is specified either here, in an item key, in a named session, or in the URI, a SCRAM-based mechanism is used (if a username is provided; otherwise, no authentication will be used).
Plugins.MongoDB.Default.AuthSource			`admin`	Default database for authenticating connection to MongoDB server. If the `MONGODB-X509` authentication mechanism is used, this parameter is automatically set to `$external`; modifying it to any other value will cause an error when launching.
Plugins.MongoDB.Default.ReplicaSet				Default replica set name; used for automatic discovery of MongoDB replica set nodes. If this parameter is used with the `mongodb+srv://` scheme, it will behave as if multiple URLs have been provided; a DNS server may have a TXT record that can override this (or any other) parameter. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.ReadPreference				Default order for connecting to MongoDB replica set nodes or fallback. Supported values: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode`, and `NearestMode`. Unavailable if the `Plugins.MongoDB.Default.Direct` parameter is set to `true`, the agent will fail to start.
Plugins.MongoDB.Default.TLSConnect				Default encryption type for communications between Zabbix agent 2 and monitored databases. Supported values: `required` - requires TLS connection. `verify\_ca` - verifies certificates. `verify\_full` - verifies certificates and IP address. If no value is specified either here, in an item key, in a named session, or in the URI, a TLS connection will not be established.
Plugins.MongoDB.Default.TLSCAFile				Default pathname of a file containing the top-level CA(s) certificates for peer certificate verification. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSCertFile				Default pathname of a file containing the MongoDB client certificate or certificate chain. Used for encrypted communications between Zabbix agent 2 and monitored databases.
Plugins.MongoDB.Default.TLSKeyFile				Default pathname of a file containing the MongoDB client private key used for encrypted communications between Zabbix agent 2 and monitored databases.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

7 MQTT plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MQTT plugin.

It is recommended to specify them in their own configuration file (e.g. `mqtt.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Default.Password				Default password for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Default.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.
Plugins.MQTT.Default.TLSKeyFile				Full pathname of a file containing the MQTT private key for encrypted communications between Zabbix agent 2 and MQTT broker; used if no value is specified in a named session.

Plugins.MQTT.Default.Topic				Default topic for MQTT subscription; used if no value is specified in an item key or named session.
				
				The topic may contain wildcards ("+","#")
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url			tcp://localhost:1883	Default MQTT broker connection string; used if no value is specified in an item key or named session.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Default.User				Default username for connecting to MQTT; used if no value is specified in an item key or named session.
Plugins.MQTT.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile				Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the MQTT private key used for encrypted communications between Zabbix agent 2 and MQTT broker. **<SessionName>** - define name of a session for using in item keys.

Plugins.MQTT.Sessions.<SessionName>.Topic				Named session topic for MQTT subscription.
				<SessionName> - define name of a session for using in item keys.
				
				The topic may contain wildcards ("+","#")
				Examples: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Sessions.<SessionName>.Url				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include query parameters.
				Must match the URL format.
				Supported schemes: `tcp` (default), `ws`, `tls`; a scheme can be omitted.
				A port can be omitted (default=1883).
				Examples: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

If you need to set the request execution timeout (how long to wait for a request to complete before shutting it down), use the item configuration form.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 MSSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MSSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mssql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The MSSQL plugin is a loadable plugin and is available and fully described in the MSSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.CustomQueriesDir			`/usr/local/sbin/zabbix/custom-queries/mssql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Specifies the full path to a directory containing user-defined .sql files with custom queries that the plugin can execute. The plugin loads all available .sql files in the configured directory at startup. This means that any changes to the custom query files will not be reflected until the plugin is restarted. The plugin is started and stopped together with Zabbix agent 2.
Plugins.MSSQL.CustomQueriesEnabled			false	If set, enables the execution of the `mssql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.MSSQL.Default.CACertPath				The default file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server. The certificate must be in PEM format.
Plugins.MSSQL.Default.Database				The default database name to connect to.
Plugins.MSSQL.Default.Encrypt				Specifies the default connection encryption type. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted.
Plugins.MSSQL.Default.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server by default.
Plugins.MSSQL.Default.Password				The password to be sent to a protected MSSQL server by default.
Plugins.MSSQL.Default.TLSMinVersion				The minimum TLS version to use by default. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`.
Plugins.MSSQL.Default.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it by default. Possible values: `true`, `false`.
Plugins.MSSQL.Default.Uri			sqlserver://localhost:1433	The default URI to connect. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored.
Plugins.MSSQL.Default.User				The default username to be sent to a protected MSSQL server.
Plugins.MSSQL.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath				The file path to the public key certificate of the certificate authority (CA) that issued the certificate of the MSSQL server for the named session. The certificate must be in PEM format. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Database				The database name to connect to for the named session. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.MSSQL.Sessions.<SessionName>.Encrypt				Specifies the connection encryption type for the named session. Possible values are: *true* - data sending between plugin and server is encrypted; *false* - data sending between plugin and server is not encrypted beyond the login packet; *strict* - data sending between plugin and server is encrypted E2E using TDS8; *disable* - data sending between plugin and server is not encrypted. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate				The common name (CN) of the certificate of the MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Password				The password to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion				The minimum TLS version to use for the named session. Possible values are: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate				Whether the plugin should trust the server certificate without validating it for the named session. Possible values: `true`, `false`. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.Uri			sqlserver://localhost:1433	The URI to connect, for the named session. The only supported schema is `sqlserver://`. A schema can be omitted. Embedded credentials will be ignored. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.Sessions.<SessionName>.User				The username to be sent to a protected MSSQL server for the named session. **<SessionName>** - define name of a session for using in item keys.
Plugins.MSSQL.System.Path				Path to the MSSQL plugin executable. Global setting for the MSSQL plugin. Applied to all connections. Example usage: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout		1-30	global timeout	The duration, in seconds, to wait for a server to respond when first connecting and on follow-up operations in the session.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

9 MySQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the MySQL plugin.

It is recommended to specify them in their own configuration file (e.g. `mysql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;

- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	no	1-30	global timeout	The maximum amount of time in seconds to wait for a request to be done.
Plugins.Mysql.CustomQueriesPath	no		`/usr/local/share/zabbix/custom-queries/mysql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing `.sql` files with custom queries.
Plugins.Mysql.CustomQueriesEnabled	no		false	If set, enables the execution of the `mysql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Mysql.Default.Password	no			Default password for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.Default.TLSCAFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSCertFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the agent certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.Mysql.Default.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Default.TLSKeyFile	no (yes, if Plugins.Mysql.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the database private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Parameter	Mandatory	Range	Default	Description

Plugins.Mysql.Default.Uri	no		tcp://localhost:3306	Default URI for connecting to MySQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	no			Default username for connecting to MySQL; used if no value is specified in an item key or named session.
Plugins.Mysql.KeepAlive	no	60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Mysql.Sessions.<SessionName>.Password	no			Named session password. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	no (yes, if Plugins.Mysql.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificates for peer certificate verification, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the agent certificate or certificate chain, used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	no			Encryption type for communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - require TLS connection; *verify_ca* - verify certificates; *verify_full* - verify certificates and IP address.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.Mysql.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the database private key used for encrypted communications between Zabbix agent 2 and monitored databases. **<SessionName>** - define name of a session for using in item keys.

Plugins.Mysql.Sessions.<SessionName>.Uri	no			Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=3306).
				Examples: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	no			Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.Mysql.Timeout	no	1-30	global timeout	The maximum time in seconds for waiting when a connection has to be established.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

10 NVIDIA GPU plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the NVIDIA GPU plugin.

It is recommended to specify them in their own configuration file (e.g. `nvidiagpu.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The NVIDIA GPU plugin is a loadable plugin and is available and fully described in the NVIDIA GPU plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.NVIDIA.System.Path				Path to the NVIDIA GPU plugin executable. Example usage: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout		1-30	global timeout	Specifies the maximum time (in seconds) to wait for a server response during connection attempts and subsequent operations in the session. The global item-type timeout or individual item timeout will override this value if greater.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

11 Oracle plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Oracle plugin.

It is recommended to specify them in their own configuration file (e.g. `oracle.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CallTimeout		1-30	global timeout	The maximum wait time in seconds for a request to be completed.
Plugins.Oracle.ConnectTimeout		1-30	global timeout	The maximum wait time in seconds for a connection to be established.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.CustomQueriesPath			/usr/local/share/zabbix/oracle/custom_queries for Unix systems *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle for Windows systems, where * is the drive name from the ProgramFiles environment variable	Full pathname of a directory containing *.sql files with custom queries. Example: /etc/zabbix/oracle/sql
Plugins.Oracle.CustomQueriesEnabled			false	If set, enables the execution of the oracle.custom.query item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.Oracle.Default.Password				Default password for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Service				Default service name for connecting to Oracle (SID is not supported); used if no value is specified in an item key or named session.
Plugins.Oracle.Default.Uri				Default URI for connecting to Oracle; used if no value is specified in an item key or named session. Should not include embedded credentials (they will be ignored). Must match the URI format. Only tcp and tcps schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS key) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS value)
Plugins.Oracle.Default.User				Default username for connecting to Oracle; used if no value is specified in an item key or named session.
Plugins.Oracle.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Oracle.ResolveTNS		true / false	true	The option specifies how to interpret the connection string (ConnString) for connecting to the Oracle server. If it is set to false the default scheme and port will be added to ConnString (if absent). If the option is set to true, the default scheme and port will be omitted (unless explicitly specified in ConnString), and ConnString will be passed to the Oracle client as is. If the Oracle client finds ConnString in the tnsnames.ora file, the connection description found will be used to connect to the Oracle server.
Plugins.Oracle.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define name of a session for using in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Oracle.Sessions.<SessionName>.Service				Named session service name to be used for connection (SID is not supported). **<SessionName>** - define name of a session for using in item keys.
Plugins.Oracle.Sessions.<SessionName>.Uri				Named session connection string for Oracle. **<SessionName>** - define name of a session for using in item keys. Should not include embedded credentials (they will be ignored). Must match the URI format. Only `tcp` and `tcps` schemes are supported; a scheme can be omitted. A port can be omitted (default=1521). It is also possible to specify the TNS key or TNS value as the connection string. The TNS value must be composed without whitespaces. Examples: `tcp://127.0.0.1:1521` `localhost` `zbx_tns_example` (TNS key) `(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR` (TNS value)
Plugins.Oracle.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

12 PostgreSQL plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the PostgreSQL plugin.

It is recommended to specify them in their own configuration file (e.g. `postgresql.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

The PostgreSQL plugin is a loadable plugin and is available and fully described in the PostgreSQL plugin repository.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables;
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Options

Parameter	Description
-V --version	Print the plugin version and license information.
-h --help	Print help information (shorthand).
-t, --test <item key>	Launch plugin for testing (plugin config ignored).

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.CacheMode			prepare	Cache mode for the PostgreSQL connection. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that "describe" is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	global timeout	Maximum wait time (in seconds) for a request to be completed.
Plugins.PostgreSQL.CustomQueriesPath			`/usr/local/share/zabbix/custom_queries/postgresql` for Unix systems `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` for Windows systems, where `*` is the drive name from the `ProgramFiles` environment variable	Full pathname of a directory containing files with custom queries.
Plugins.PostgreSQL.CustomQueriesEnabled			false	If set, enables the execution of the `postgresql.custom.query` item key. If disabled, no queries will be loaded from the custom query directory path.
Plugins.PostgreSQL.Default.Database				Default database for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.Password				Default password for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.Default.TLSCAFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate for peer certificate verification for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSCertFile	(yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL certificate or certificate chain for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.
Plugins.PostgreSQL.Default.TLSConnect				Encryption type for communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Default.TLSKeyFile	No (yes, if Plugins.PostgreSQL.Default.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the PostgreSQL private key for encrypted communications between Zabbix agent 2 and monitored databases; used if no value is specified in a named session.

Plugins.PostgreSQL.Default.Uri	No			Default URI for connecting to PostgreSQL; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	No			Default username for connecting to PostgreSQL; used if no value is specified in an item key or named session.
Plugins.PostgreSQL.KeepAlive	No	60-900	300	Maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.PostgreSQL.Sessions.<SessionName>.CachePreparedMode	No		prepare	Cache mode for the PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *prepare* (default) - will create prepared statements on the PostgreSQL server; *describe* - will use the anonymous prepared statement to describe a statement without creating a statement on the server. Note that ”describe” is primarily useful when the environment does not allow prepared statements such as when running a connection pooler like PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	No			Database for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	No	Must match the password format.		Password for session connection. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	No (yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect is set to *verify_ca* or *verify_full*)			Full pathname of a file containing the top-level CA(s) certificate peer certificate verification. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile is specified			Full pathname of a file containing the PostgreSQL certificate or certificate chain. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	No			Encryption type for PostgreSQL connection. **<SessionName>** - define name of a session for using in item keys. Supported values: *required* - connect using TLS as transport mode without identity checks; *verify_ca* - connect using TLS and verify certificate; *verify_full* - connect using TLS, verify certificate and verify that database identity (CN) specified by DBHost matches its certificate. Undefined encryption type means unencrypted connection.

Parameter	Mandatory	Range	Default	Description
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	yes, if Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile is specified			Full pathname of a file containing the PostgreSQL private key. **<SessionName>** - define name of a session for using in item keys.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri				Connection string of a named session.
				<SessionName> - define name of a session for using in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`.
				Examples: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User				Named session username. **<SessionName>** - define name of a session for using in item keys.
Plugins.PostgreSQL.System.Path	yes			Path to the PostgreSQL plugin executable. Example usage: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout		1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

13 Redis plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the Redis plugin.

It is recommended to specify them in their own configuration file (e.g. `redis.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files.
- Values support environment variables.
- Zabbix supports configuration files only in UTF-8 encoding without BOM.
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password				Default password for connecting to Redis; used if no value is specified in an item key or named session.
Plugins.Redis.Default.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Default.TLSCAFile				Default full pathname of a file containing the top-level CA(s) certificates.
Plugins.Redis.Default.TLSCertFile				Default full pathname of a file containing the agent's certificate.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.TLSKeyFile				Default full pathname of a file containing the agent's private key.

Plugins.Redis.Default.Uri			tcp://localhost:6379	Default URI for connecting to Redis; used if no value is specified in an item key or named session.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	Default user to send to the protected Redis server; used if no value is specified in an item key or named session.
Plugins.Redis.KeepAlive		60-900	300	The maximum time of waiting (in seconds) before unused plugin connections are closed.
Plugins.Redis.Sessions.<SessionName>.Password				Named session password. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSConnect				Client (Zabbix agent 2) TLS verification requirement. **<SessionName>** - define the session name to be used in item keys. Supported values: `required` - encrypted, unverified (only for testing) `verify_ca` - encrypted, server certificate `verify_full` - encrypted, server certificate and server name verified with SAN An empty or unset value is treated as no tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	mandatory (yes, if TLSConnect is `verify_ca` or `verify_full`)			Full pathname of a file containing the top-level CA(s) certificates for Redis server certificate verification. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				Full pathname of a file containing the agent's certificate for client authentication. **<SessionName>** - define the session name to be used in item keys.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				Full pathname of a file containing the agent's private key for client authentication. **<SessionName>** - define the session name to be used in item keys.

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	Connection string of a named session.
				<SessionName> - define the session name to be used in item keys.
				
				Should not include embedded credentials (they will be ignored).
				Must match the URI format.
				Supported schemes: `tcp`, `unix`; a scheme can be omitted.
				A port can be omitted (default=6379).
				Examples: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	User to send to the protected Redis server. **<SessionName>** - define the session name to be used in item keys.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

14 SMART plugin

Overview

The configuration file of Zabbix agent 2 is used to configure plugins. These Zabbix agent 2 configuration parameters are supported for operating the SMART plugin.

It is recommended to specify them in their own configuration file (e.g. `smart.conf`) and then use the Include directive for adding this file to the Zabbix agent 2 configuration.

Note that:

- The default values reflect process defaults, not the values in the shipped configuration files;
- Values support environment variables;
- The path to the `smartctl` executable must be set either by adding its directory to the system's `PATH` environment variable or by configuring `Plugins.Smart.Path`; this applies to both Linux and Windows;
- Zabbix supports configuration files only in UTF-8 encoding without BOM;
- Comments starting with "#" are only supported at the beginning of the line.

Parameters

Parameter	Mandatory	Range	Default	Description
Plugins.Smart.Path	no		smartctl	Path to the smartctl executable.
Plugins.Smart.Timeout	no	1-30	global timeout	Request execution timeout (the duration, in seconds, to wait for a request to complete before shutting it down).

See also:

- Description of general Zabbix agent 2 configuration parameters: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Instructions for configuring plugins

8 Zabbix Java gateway

Se si utilizzano gli script `startup.sh` e `shutdown.sh` per avviare Zabbix Java gateway, è possibile 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) nelle proprietà Java (elencate nella seconda colonna).

Se si avvia Zabbix Java gateway manualmente eseguendo direttamente `java`, le corrispondenti proprietà Java devono essere specificate nella riga di comando.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	Indirizzo IP su cui mettersi in ascolto.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Porta su cui mettersi in ascolto.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	Nome del file PID. Se omesso, Zabbix Java Gateway viene avviato come applicazione console.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	no			Nome del file delle proprietà. Può essere utilizzato per impostare proprietà aggiuntive usando un formato chiave-valore, in modo che non siano visibili nella 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 di lavoro 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.

9 Servizio web Zabbix

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 visualizzarne tutti i dettagli.

Parametro	Descrizione
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 durante l’accesso all’URL del frontend.
Include	È possibile includere singoli file o tutti i file di una directory nel file di configurazione.
ListenPort	Il servizio resterà in ascolto su questa porta per 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 formattare il report PDF di una dashboard.
TLSAccept	Quali connessioni in ingresso accettare.
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 cifrate tra i componenti Zabbix.
TLSCertFile	Il percorso completo di un file contenente il certificato del servizio o la catena di certificati, utilizzato per le comunicazioni cifrate tra i componenti Zabbix.
TLSKeyFile	Il percorso completo di un file contenente la chiave privata del servizio, utilizzato per le comunicazioni cifrate tra i componenti Zabbix.

Tutti i parametri sono facoltativi, salvo ove 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;
- Zabbix supporta 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

AllowedIP

Un elenco di indirizzi IP separati da virgole, facoltativamente in notazione CIDR, oppure di nomi DNS di server Zabbix e 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 di 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 validazione del certificato TLS durante l'accesso all'URL del frontend:
0 - non ignorare gli errori del certificato;
1 - ignorare gli errori del certificato.

Predefinito: 0
 Intervallo: 0-1

Include

È possibile includere singoli file o tutti i file in una directory nel file di configurazione. Durante l'installazione Zabbix creerà la directory di inclusione in /usr/local/etc, a meno che non venga modificata in fase di compilazione. Il percorso può essere relativo alla posizione del file *zabbix_web_service.conf*.
Per includere solo i file pertinenti nella directory specificata, è supportato il carattere jolly asterisco per la corrispondenza dei modelli.
Vedere le note speciali relative alle limitazioni.

Esempio:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Il servizio ascolterà su questa porta le connessioni dal server.

Impostazione predefinita: 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 ricominciato da capo.

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 report PDF 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 primo livello per la verifica del certificato del peer, utilizzato per le comunicazioni cifrate 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.

10 Variabili d'ambiente

Panoramica

Le variabili d'ambiente consentono di configurare i componenti di Zabbix senza codificare rigidamente 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 configurazioni diverse.

Nel caso più semplice, impostando il valore del parametro di configurazione DebugLevel del server Zabbix su una variabile d'ambiente, è quindi 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 d'ambiente che non viene specificata all'avvio del componente, viene utilizzato il valore predefinito del parametro.
- Quando si usano i comandi runtime (ad esempio, per aumentare il livello di log dell'agent), devono essere specificate tutte le variabili d'ambiente usate in precedenza. Questo perché i componenti Zabbix usano il proprio file di configurazione per eseguire i comandi runtime; se le variabili d'ambiente vengono omesse, verranno usati i valori predefiniti dei parametri di configurazione. Vedere Esempi.
- 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.
- Le variabili d'ambiente correnti del processo, utilizzate nei file di configurazione, vengono eliminate dopo l'avvio del componente Zabbix. Questo garantisce che i processi figli (ad esempio, gli script remoti eseguiti da Zabbix) non possano accedere a queste variabili. Tuttavia, si noti che le variabili iniziali del processo possono comunque essere recuperate (ad esempio, tramite il file `/proc/<PID>/environ`).

Sintassi

Le variabili d'ambiente devono utilizzare la seguente sintassi: `${alphanumerics/underscores}`.

Il nome della variabile può includere solo lettere (a-z, A-Z), caratteri di sottolineatura (_), 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 verranno trattate come valori normali, il che può produrre errori.

Sintassi corretta delle variabili:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Sintassi non corretta delle variabili:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
In Windows, i nomi delle variabili d'ambiente non distinguono tra maiuscole e minuscole.

Esempi

Gli esempi seguenti mostrano come configurare e utilizzare le variabili d'ambiente con i componenti di Zabbix.

Esempio 1: configurazione e test di Zabbix agent

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 usate 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`). Questo 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 stai creando e configurando una tua immagine personalizzata per i componenti Zabbix (ad esempio, Zabbix agent), puoi definire i parametri di configurazione usando variabili d'ambiente e quindi avviare il container con tali variabili.

1. Durante la preparazione dell'immagine, imposta 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, avvia 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 utilizzi i comandi runtime (ad esempio, per aumentare il livello di log dell'agent), accedi alla shell del container ed esegui 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 normali.

11 Inclusione

Panoramica

File o directory aggiuntivi possono essere inclusi nella configurazione di server, proxy, agent e servizio web utilizzando il parametro `Include`.

Note sull'inclusione

Se il parametro `Include` viene utilizzato per includere un file, il file deve essere leggibile.

Se il parametro `Include` viene utilizzato per includere una directory:

- Tutti i file nella directory devono essere leggibili.
- Non si deve presumere alcun ordine particolare di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico). Pertanto, non definire un parametro in più file "Include" (ad esempio, per sovrascrivere un'impostazione generale con una specifica).
- Tutti i file nella directory vengono inclusi nella configurazione.
- Fare attenzione alle copie di backup dei file create automaticamente da alcuni editor di testo. Ad esempio, se la modifica del file "include/my_specific.conf" produce una copia di backup "include/my_specific_conf.BAK", verranno inclusi entrambi i file. Spostare "include/my_specific.conf.BAK" fuori dalla directory "Include". Su Linux, il contenuto della directory "Include" può essere controllato con il comando "ls -al" per individuare eventuali file non necessari.

Se il parametro `Include` viene utilizzato per includere file tramite un pattern:

- Tutti i file che corrispondono al pattern devono essere leggibili.
- Non si deve presumere alcun ordine particolare di inclusione (ad esempio, i file non vengono inclusi in ordine alfabetico). Pertanto, non definire un parametro in più file "Include" (ad esempio, per sovrascrivere un'impostazione generale con una specifica).

3 Protocolli

Utilizzare la barra laterale per accedere ai contenuti di questa sezione.

1 Protocollo di scambio dati server-proxy

Panoramica

Lo scambio di dati server - 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 invierà inizialmente 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:		

name	value type	description
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 i dati completi di configurazione; assente - altrimenti (opzionale).
data	*array*	Oggetto dei dati delle tabelle. Assente se la configurazione non è stata modificata (opzionale).
<table>	*object*	Uno o più oggetti con i dati di <table> (opzionale, in base alle modifiche).
fields	*array*	Array di nomi di campo.
-	*string*	Nome del campo.
data	*array*	Array di righe.
-	*array*	Array di colonne.
-	*string,number*	Valore della colonna con tipo dipendente dal tipo della colonna nello schema del database.
macro.secrets	*object*	Informazioni sulle macro segrete, assente se non ci sono modifiche nelle macro del vault (opzionale).
config_revision	*number*	Revisione della cache di configurazione - inviata con i dati di configurazione (opzionale).
del_hostids	*array*	Array degli hostid rimossi (opzionale).
-	*number*	Identificatore host.
del_macro_hostids	*array*	Array degli hostid da cui sono state rimosse tutte le macro (opzionale).
-	*number*	Identificatore 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", "ipmi_password"
"data": [
[10084, "Zabbix server", 0, -1, 2, "", "", "Zabbix server", 1, 1, "", "", "", ""]
]
},
"interface": {
"fields": ["interfaceid", "hostid", "main", "type", "useip", "ip", "dns", "port", "available"],
"data": [
[1, 10084, 1, 1, 1, "127.0.0.1", "", "10053", 1]
```

```
]
},
"interface_snmp": {
"fields": ["interfaceid", "version", "bulk", "community", "securityname", "securitylevel", "authpassphrase
"data": []
},
"host_inventory": {
"fields": ["hostid", "type", "type_full", "name", "alias", "os", "os_full", "os_short", "serialno_a", "ser
"data": [
[10084, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
```

```
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
```

```
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


proxy→server:


```
{
  "response": "success",
  "version": "8.0.0"
}
```


Richiesta dati

La richiesta `proxy data` viene utilizzata per ottenere dal proxy i dati di disponibilità delle interfacce del host, i dati storici, di discovery e di autoregistrazione. Questa richiesta viene inviata ogni `ProxyDataFrequency` secondi (parametro di configurazione del server).

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Token della sessione dati.
interface availability	*array*	*(opzionale)* 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 oppure stringa vuota.
history data	*array*	*(opzionale)* 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*	*(opzionale)* Valore dell'item.
id	*number*	Identificatore del valore (contatore crescente, univoco all'interno di una sessione dati).
timestamp	*number*	*(opzionale)* Timestamp degli item di tipo log.
source	*string*	*(opzionale)* Valore source dell'item eventlog.
severity	*number*	*(opzionale)* Valore severity dell'item eventlog.
eventid	*number*	*(opzionale)* Valore eventid dell'item eventlog.

name	value type	description
state	*string*	*(opzionale)* Stato dell'item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(opzionale)* Ultima dimensione del log per gli item di tipo log.
mtime	*number*	*(opzionale)* Ora di modifica degli item di tipo log.
discovery data	*array*	*(opzionale)* 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 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* - 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 del host.
dns	*string*	Nome DNS del host.
port	*number*	*(opzionale)* Numero di porta del servizio.
key_	*string*	*(opzionale)* Chiave item per il controllo di discovery di tipo **9** *SVC_AGENT*
value	*string*	*(opzionale)* Valore ricevuto dal servizio; può essere vuoto per la maggior parte dei servizi.
status	*number*	*(opzionale)* Stato del servizio: **0**, *DOBJECT_STATUS_UP* - Servizio UP **1**, *DOBJECT_STATUS_DOWN* - Servizio DOWN
auto reg- is- tra- tion	*array*	*(opzionale)* Array di oggetti dati di autoregistrazione.
clock	*number*	Timestamp dei dati di autoregistrazione.
host	*string*	Nome host.
ip	*string*	*(opzionale)* Indirizzo IP del host.
dns	*string*	*(opzionale)* Nome DNS risolto dall'indirizzo IP.
port	*string*	*(opzionale)* Porta del host.
host_metadata	*string*	*(opzionale)* Metadati del host inviati dall'agent (in base al parametro di configurazione dell'agent HostMetadata o HostMetadataItem).
tasks	*array*	*(opzionale)* 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

name	value type	description
error	*string*	*(opzionale)* Messaggio di errore.
parent_taskid	*number*	ID del task padre.
more	*number*	*(opzionale)* 1 - ci sono altri dati storici da inviare.
clock	*number*	*(opzionale)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(opzionale)* Timestamp del trasferimento dati (nanosecondi).
version	*string*	Versione del proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informazione sull'esito della richiesta ('success' oppure 'failed').
tasks	*array*	*(opzionale)* 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 su agent **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - esegui lo script su server **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - esegui lo script su proxy
port	*number*	*(opzionale)* Porta per i comandi Telnet e SSH.
authtype	*number*	*(opzionale)* Tipo di autenticazione per i comandi SSH.
username	*string*	*(opzionale)* Nome utente per i comandi Telnet e SSH.
password	*string*	*(opzionale)* Password per i comandi Telnet e SSH.
publickey	*string*	*(opzionale)* Chiave pubblica per i comandi SSH.
privatekey	*string*	*(opzionale)* Chiave privata per i comandi SSH.
parent_taskid	*number*	ID del task padre.
hostid	*number*	ID del 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 di ricezione dei task dei comandi remoti tra il server e un proxy passivo. Quando il server interroga il proxy per ottenere i risultati dei task, invia una richiesta `proxy tasks` vuota all'intervallo definito da `ProxyDataFrequency`. Il proxy risponde con la propria versione corrente e con eventuali risultati di task in sospeso (inclusi i timestamp). Il server quindi conferma la ricezione restituendo una `response` e può includere nuovi task da eseguire da parte del proxy.

name	value type	description
server→proxy:		
request	*string*	'proxy tasks'
proxy→server:		
version	*string*	Versione del proxy (<major>.<minor>.<build>).
clock	*number*	*(opzionale)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(opzionale)* Timestamp del trasferimento dati (nanosecondi).
server→proxy:		
response	*string*	Informazioni sull'esito della richiesta ('success' o '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 da un proxy attivo per ottenere i dati di configurazione del proxy. Questa richiesta viene inviata ogni `ProxyConfigFrequency` secondi (parametro di configurazione del proxy).

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 di sessione della 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 i dati completi di configurazione, assente altrimenti (opzionale).
data	*array*	Oggetto dei dati della tabella. Assente se la configurazione non è stata modificata (opzionale).
<table>	*object*	Uno o più oggetti con i dati di <table> (opzionale, in base alle modifiche).
fields	*array*	Array di nomi di campo.
-	*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, assente se non ci sono modifiche nelle macro del vault (opzionale).
proxy_group	*string*	Il nome del gruppo di proxy a cui appartiene il proxy.

name	value type	description
config_revision	*number*	Revisione della cache di configurazione - inviata con i dati di configurazione (opzionale).
del_hostids	*array*	Array degli hostid rimossi (opzionale).
-	*number*	Identificatore host.
del_macro_hostids	*array*	Array degli hostid da cui sono state rimosse tutte le macro (opzionale).
-	*number*	Identificatore 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, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "5
]
},
"items": {
"fields": ["itemid", "type", "snmp_oid", "hostid", "key_", "delay", "history", "status", "value_type", "tr
"data": [
[44161, 7, "", 10084, "agent.hostmetadata", "10s", "90d", 0, 1, "", "", "", "", 0, "", "", "", "", 0, null
[44162, 0, "", 10084, "agent.ping", "10s", "90d", 0, 3, "", "", "", "", 0, "", "", "", "", 0, 1, 0, "", nu
]
},
"item_rtdata": {
"fields": ["itemid", "lastlogsize", "mtime"],
"data": [
[44161, 0, 0],
[44162, 0, 0]
]
},
"item_preproc": {
```

```
"fields": ["item_preprocid", "itemid", "step", "type", "params", "error_handler", "error_handler_params"],
"data": []
},
"item_parameter": {
"fields": ["item_parameterid", "itemid", "name", "value"],
"data": []
},
"globalmacro": {
"fields": ["globalmacroid", "macro", "value", "type"],
"data": [
[2, "{$SNMP_COMMUNITY}", "public", 0]
]
},
"hosts_templates": {
"fields": ["hosttemplateid", "hostid", "templateid", "link_type"],
"data": []
},
"hostmacro": {
"fields": ["hostmacroid", "hostid", "macro", "value", "type", "automatic"],
"data": [
[5676, 10084, "{$M}", "AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix:Content", 2, 0]
]
},
"drules": {
"fields": ["druleid", "name", "iprange", "delay"],
"data": [
[2, "Local network", "127.0.0.1", "10s"]
]
},
"dchecks": {
"fields": ["dcheckid", "druleid", "type", "key_", "snmp_community", "ports", "snmpv3_securityname", "snmpv
"data": [
[2, 2, 9, "system.uname", "", "10052", "", 0, "", "", 0, 0, 0, "", 1, 0]
]
},
"regexps": {
"fields": ["regexpid", "name"],
"data": [
[1, "File systems for discovery"],
[2, "Network interfaces for discovery"],
[3, "Storage devices for SNMP discovery"],
[4, "Windows service names for discovery"],
[5, "Windows service startup states for discovery"]
]
},
"expressions": {
"fields": ["expressionid", "regexpid", "expression", "expression_type", "exp_delimiter", "case_sensitive"]
"data": [
[1, 1, "^(btrfs|ext2|ext3|ext4|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|apfs|refs|ntfs|fat32|zfs)$", 3, ",", 0
[3, 3, "^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$", 4, ",", 1],
[5, 4, "^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$", 4, ","
[6, 5, "^(automatic|automatic delayed)$", 3, ",", 1],
[7, 2, "^Software Loopback Interface", 4, ",", 1],
[8, 2, "^(In)?[Ll]oop[Bb]ack[0-9._]*$", 4, ",", 1],
[9, 2, "^NULL[0-9.]*$", 4, ",", 1],
[10, 2, "^[Ll]o[0-9.]*$", 4, ",", 1],
[11, 2, "^[Ss]ystem$", 4, ",", 1],
[12, 2, "^Nu[0-9.]*$", 4, ",", 1]
]
},
"settings": {
"fields": ["name", "type", "value_str", "value_int"],
```

```
"data": [
["autoreg_tls_accept", 2, "", 1],
["hk_history_global", 2, "", 0],
["snmptrap_logging", 2, "", 1],
["proxy_secrets_provider", 2, "", 0],
["hk_history", 1, "31d", 0],
["timeout_db_monitor", 1, "3s", 0],
["timeout_external_check", 1, "3s", 0],
["timeout_http_agent", 1, "3s", 0],
["timeout_simple_check", 1, "3s", 0],
["timeout_snmp_agent", 1, "3s", 0],
["timeout_ssh_agent", 1, "3s", 0],
["timeout_telnet_agent", 1, "3s", 0],
["timeout_zabbix_agent", 1, "3s", 0],
["timeout_browser", 1, "30s", 0]
]
},
"httptest": {
"fields": ["httptestid", "name", "delay", "agent", "authentication", "http_user", "http_password", "hostid
"data": []
},
"httptestitem": {
"fields": ["httptestitemid", "httptestid", "itemid", "type"],
"data": []
},
"httptest_field": {
"fields": ["httptest_fieldid", "httptestid", "type", "name", "value"],
"data": []
},
"httpstep": {
"fields": ["httpstepid", "httptestid", "name", "no", "url", "timeout", "posts", "required", "status_codes"
"data": []
},
"httpstepitem": {
"fields": ["httpstepitemid", "httpstepid", "itemid", "type"],
"data": []
},
"httpstep_field": {
"fields": ["httpstep_fieldid", "httpstepid", "type", "name", "value"],
"data": []
},
"config_autoreg_tls": {
"fields": ["autoreg_tlsid", "tls_psk_identity", "tls_psk"],
"data": [
[1, "", ""]
]
}
},
"macro.secrets": {
"AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix": {
"Content": "738"
}
},
"config_revision": 2
}
```


Richiesta dati

La richiesta `proxy data` viene inviata dal proxy per fornire dati sulla disponibilità delle interfacce del host, cronologia, individuazione e registrazione automatica. Questa richiesta viene inviata ogni `DataSenderFrequency` secondi (parametro di configurazione del proxy). Si noti che il proxy attivo continuerà comunque a interrogare Zabbix server ogni secondo per le attività dei 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*	*(opzionale)* Array di oggetti dati sulla disponibilità dell'interfaccia.
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*	*(opzionale)* Array di oggetti dati della cronologia.
itemid	*number*	Identificatore dell'item.
clock	*number*	Timestamp del valore dell'item (secondi).
ns	*number*	Timestamp del valore dell'item (nanosecondi).
value	*string*	*(opzionale)* Valore dell'item.
id	*number*	Identificatore del valore (contatore crescente, univoco all'interno di una sessione dati).
timestamp	*number*	*(opzionale)* Timestamp degli item di tipo log.
source	*string*	*(opzionale)* Valore source dell'item eventlog.
severity	*number*	*(opzionale)* Valore severity dell'item eventlog.
eventid	*number*	*(opzionale)* Valore eventid dell'item eventlog.
state	*string*	*(opzionale)* Stato dell'item: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(opzionale)* Ultima dimensione del log degli item di tipo log.
mtime	*number*	*(opzionale)* Ora di modifica degli item di tipo log.
discovery data	*array*	*(opzionale)* Array di oggetti dati di individuazione.
clock	*number*	Timestamp dei dati di individuazione.
druleid	*number*	Identificatore della regola di individuazione.
dcheckid	*number*	Identificatore del controllo di individuazione oppure null per i dati della regola di individuazione.
type	*number*	Tipo di controllo di individuazione: **-1** dati della regola di individuazione **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 del host.
dns	*string*	Nome DNS del host.
port	*number*	*(opzionale)* Numero di porta del servizio.
key_	*string*	*(opzionale)* Chiave item per il controllo di individuazione di tipo **9** *SVC_AGENT*

name	value type	description
value	*string*	*(opzionale)* Valore ricevuto dal servizio; per la maggior parte dei servizi può essere vuoto.
status	*number*	*(opzionale)* Stato del servizio: **0**, *DOBJECT_STATUS_UP* - Servizio UP **1**, *DOBJECT_STATUS_DOWN* - Servizio DOWN
autoregistration	*array*	*(opzionale)* Array di oggetti dati di registrazione automatica.
clock	*number*	Timestamp dei dati di registrazione automatica.
host	*string*	Nome del host.
ip	*string*	*(opzionale)* Indirizzo IP del host.
dns	*string*	*(opzionale)* Nome DNS risolto dall'indirizzo IP.
port	*string*	*(opzionale)* Porta del host.
host_metadata	*string*	*(opzionale)* Metadati del host inviati dall'agent (in base al parametro di configurazione dell'agent HostMetadata o HostMetadataItem).
tasks	*array*	*(opzionale)* 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*	*(opzionale)* Messaggio di errore.
parent_taskid	*number*	ID dell'attività padre.
more	*number*	*(opzionale)* 1 - ci sono altri dati della cronologia da inviare
clock	*number*	*(opzionale)* Timestamp del trasferimento dati (secondi).
ns	*number*	*(opzionale)* Timestamp del trasferimento dati (nanosecondi).
version	*string*	Versione del proxy (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Informazione sull'esito della richiesta ('success' o 'failed').
upload	*string*	Controllo del caricamento per i dati storici (cronologia, registrazione automatica, disponibilità del host, individuazione di rete). Valori possibili: **enabled** - funzionamento normale **disabled** - il server non accetta dati (possibilmente a causa del superamento del limite della cache interna)
tasks	*array*	*(opzionale)* 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 equivalente dal punto di vista funzionale 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*	*(opzionale)* Porta per i comandi Telnet e SSH.
authtype	*number*	*(opzionale)* Tipo di autenticazione per i comandi SSH.

name	value type	description
username	*string*	*(opzionale)* Nome utente per i comandi Telnet e SSH.
password	*string*	*(opzionale)* Password per i comandi Telnet e SSH.
publickey	*string*	*(opzionale)* Chiave pubblica per i comandi SSH.
privatekey	*string*	*(opzionale)* Chiave privata per i comandi SSH.
parent_taskid	*number*	ID dell'attività padre.
hostid	*number*	ID del host di destinazione.

Esempio:

proxy→server:


```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
```

```
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
}
```


server→proxy:


```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```


2 protocollo dell'agent/agent2 Zabbix

Riferirsi alla pagina Controlli attivi e passivi per più informazioni sui protocolli per agent Zabbix e agent 2 Zabbix.

4 Zabbix agent 2 plugin protocol

Zabbix agent 2 protocol is based on code, size and data model.

Codice

Tipo	Dimensione	Commenti
Byte	4	Tipo di payload, attualmente è supportato solo JSON.

Dimensione

Tipo	Dimensione	Commenti
Byte	4	Dimensione del payload corrente in byte.

Dati del payload

Tipo	Dimensione	Commenti
Byte	Definita dal campo *Dimensione*	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 utilizzato per collegare le richieste con le risposte. Unico all'interno di una direzione della richiesta (cioè da agent a plugin o da plugin ad agent). Per le risposte - 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 utilizzate 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}
```

Termina richiesta

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 dalla 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"}
```

Configurare la richiesta

Una richiesta per eseguire la funzione *Configure* dell'interfaccia *Configurator*.

direction	agent → plugin
response	n/a

Parametri specifici delle 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	sì

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 della risposta

Risposta della funzione *Validate* dell'interfaccia *Configurator*.

direzione	plugin → agent
risposta	n/a

Parametri specifici delle risposte *Validate*:

Nome	Tipo	Commenti
error	string (opzionale)	Messaggio di errore restituito se la funzione Validate non viene eseguita correttamente. Assente se l'esecuzione ha esito positivo.

Esempio:

```
{"id":8,"type":10}
```

oppure

```
{"id":8,"type":10,"error":"error message"}
```

5 Protocollo Zabbix sender

Panoramica

Il server Zabbix e il proxy Zabbix utilizzano un protocollo di comunicazione basato su JSON per la ricezione di dati da Zabbix sender. I dati possono essere ricevuti tramite un elemento trapper, o un elemento agente HTTP con il trapping abilitato.

I messaggi di richiesta e risposta devono iniziare con intestazione e lunghezza dati.

Richiesta 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 un 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"
}
```

6 Intestazione

Panoramica

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, se è un pacchetto grande oppure no.

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 inizia automaticamente a utilizzare il formato di pacchetto grande (flag 0x04), che aumenta il limite di dimensione del pacchetto a 16 GB.

Si noti che, sebbene il formato di pacchetto grande possa essere utilizzato per inviare qualsiasi dato, attualmente solo il sincronizzatore della configurazione di Zabbix proxy può gestire pacchetti di dimensioni superiori a 1 GB.

Struttura

L'intestazione è composta da quattro campi. Tutti i numeri nell'intestazione sono formattati in little-endian.

Campo	Dimensione	Dimensione (pacchetto grande)	Descrizione
<PROTOCOL>	4	4	"ZBXD" oppure 5A 42 58 44
<FLAGS>	1	1	Flag del protocollo: 0x01 - protocollo di comunicazione Zabbix 0x02 - compressione 0x04 - pacchetto grande
<DATALEN>	4	8	Lunghezza dei dati.
<RESERVED>	4	8	Quando viene utilizzata la compressione (flag 0x02) - la lunghezza dei dati non compressi Quando la compressione non viene utilizzata - 00 00 00 00

Esempi

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 1GB, pertanto il formato dei pacchetti grandi non viene utilizzato.

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"
```

7 Protocollo di esportazione JSON delimitato da newline

Questa sezione presenta i dettagli del protocollo di esportazione in formato JSON delimitato da newline, utilizzato in:

- esportazione dei dati in file
- streaming verso sistemi esterni

È possibile esportare quanto segue:

- eventi dei trigger
- valori degli item
- trend (solo esportazione in file)

Tutti i file hanno estensione .ndjson. Ogni riga del file di esportazione è un oggetto JSON.

Eventi trigger

Le seguenti informazioni vengono esportate per un evento di problema:

Campo		Tipo	Descrizione
clock		number	Numero di secondi trascorsi dall'Epoch fino al momento in cui il problema è stato rilevato (parte intera).
ns		number	Numero di nanosecondi da aggiungere a `clock` per ottenere l'ora precisa 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 - Media, 4 - Alta, 5 - Disastro).
hosts		array	Elenco degli host coinvolti nell'espressione del trigger; nell'array deve essere presente almeno un elemento.
-		object	
	host	string	Nome host.
	name	string	Nome host visibile.
groups		array	Elenco dei gruppi host di tutti gli host coinvolti nell'espressione del trigger; nell'array deve essere presente almeno un elemento.
-		string	Nome del gruppo 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:

Campo	Tipo	Descrizione
clock	number	Numero di secondi trascorsi dall'Epoch fino al momento in cui il problema è stato risolto (parte intera).
ns	number	Numero di nanosecondi da aggiungere a `clock` per ottenere l'ora precisa di risoluzione del problema.

Campo	Tipo	Descrizione
value	number	0 (sempre).
eventid	number	ID dell'evento di ripristino.
p_eventid	number	ID dell'evento di problema.

Esempi

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}
```

Valori degli item

Le seguenti informazioni vengono esportate per un valore raccolto di un item:

Campo	Tipo	Descrizione
host	object	Nome del host dell'item.
host	string	Nome del host.
name	string	Nome visibile del host.
groups	array	Elenco dei gruppi di host del host dell'item; nell'array deve essere presente almeno un elemento.
-	string	Nome del gruppo di 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 fino al momento in cui il valore è stato raccolto (parte intera).
ns	number	Numero di nanosecondi da aggiungere a `clock` per ottenere l'ora precisa di raccolta del valore.
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 raccolto dell'item.

Campo	Tipo	Descrizione
type	number	Tipo di valore raccolto: 0 - numerico in virgola mobile, 1 - carattere, 2 - log, 3 - numerico senza segno, 4 - testo, 5 - binario, 6 - JSON

Esempi

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 log:

```
{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Trend

Per un valore di trend calcolato vengono esportate le seguenti informazioni:

Campo	Tipo	Descrizione
host	object	Nome del host dell'item.
host	string	Nome del host.
name	string	Nome visibile del host.
groups	array	Elenco dei gruppi di host del host dell'item; nell'array deve essere presente almeno un elemento.
-	string	Nome del gruppo di 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 fino 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.
type	number	Tipo di valore: 0 - numerico in virgola mobile, 3 - numerico senza segno

Esempi

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":[{"
```

4 Item

Utilizzare la barra laterale per accedere ai contenuti di questa sezione.

1 parametri 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 in 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 di file a cui si è avuto accesso di recente
- **free** - memoria immediatamente disponibile a qualsiasi entità che ne faccia richiesta
- **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 uso interno
- **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 in 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+ (retroportato anche a 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 ad '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. Ad esempio, su FreeBSD:
* Le memorie active, inactive, wired e cached sono considerate usate, perché memorizzano 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

2 Controlli degli agenti passivi e attivi

Panoramica

Questa sezione fornisce dettagli sui 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. Zabbix server o proxy richiede alcuni dati (ad esempio, il carico della CPU) e Zabbix agent restituisce il risultato al server.

I controlli passivi 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.

L’agent poller tenterà di connettersi a tutti gli indirizzi restituiti dalla ricerca DNS. Questo garantisce che, se un indirizzo IP non è raggiungibile, il poller proverà l’indirizzo successivo disponibile, aumentando la probabilità di una connessione riuscita. Questo miglioramento si applica sia a Zabbix server che a proxy.

La concorrenza massima dei controlli asincroni è 1000 (definita da MaxConcurrentChecksPerPoller).

Il numero di agent poller asincroni è definito dal parametro StartAgentPollers.

Richiesta del server

Per la definizione dell’header e della lunghezza dei dati, fare riferimento ai dettagli del protocollo.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Campo		Tipo	Obbligatorio	Valore
request		*string*	yes	`"passive checks"`
data		*array of object*	yes	item di controllo passivo.
	key	*string*	yes	Chiave item con macro espanse.
	timeout	*number*	yes	Timeout di comunicazione.

Risposta dell’agent

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Campo	Tipo	Obbligatorio	Valore
version	*string*	yes	Il numero di versione dell’agent.
variant	*number*	yes	La variante dell’agent (*1* - Zabbix agent, *2* - Zabbix agent 2).

Campo	Tipo	Obbligatorio	Valore
data	*array of object*	yes	Contiene il risultato del controllo.
value	*string*	no	Il valore dell'item se il controllo è riuscito.
error	*string*	no	Il messaggio di errore se il controllo non è riuscito.

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, nel nostro caso '1'
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

Fallback 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 fallback al vecchio protocollo.

I controlli passivi vengono eseguiti utilizzando il protocollo JSON (7.0 e successive) dopo un riavvio o quando viene modificata la configurazione dell'interfaccia.
Se in risposta non viene ricevuto alcun JSON valido (l'agent ha inviato "ZBX_NOTSUPPORTED"), Zabbix memorizzerà 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 utilizzare 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 per l'elaborazione 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 vengono richiesti 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 codificati in modo fisso.

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'utilizzo delle risorse, viene eseguita una sincronizzazione incrementale della configurazione ogni 5 secondi (impostazione predefinita), durante la 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 o ai 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, si può prendere in considerazione l'uso dei soli 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 dei controlli attivi viene utilizzata per ottenere i controlli attivi che devono essere elaborati dall'agent. Questa richiesta viene inviata dall'agent all'avvio e successivamente agli 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"
}
```

Campo	Tipo	Obbligatorio	Valore
request	*string*	sì	`active checks`
host	*string*	sì	Nome del host.
host_metadata	*string*	no	Il parametro di configurazione HostMetadata o il valore della metrica HostMetadataItem.
interface	*string*	no	Il parametro di configurazione HostInterface o il valore della metrica HostInterfaceItem.
ip	*string*	no	Il primo IP del parametro di configurazione ListenIP, se impostato.
port	*number*	no	Il valore del parametro di configurazione ListenPort, se impostato e diverso dalla porta di ascolto predefinita dell'agent.
version	*string*	sì	Il numero di versione dell'agent.
variant	*number*	sì	La variante dell'agent (*1* - Zabbix agent, *2* - Zabbix agent 2).
config_revision	*number*	no	Identificatore di configurazione per la sincronizzazione incrementale della configurazione.
session	*string*	no	Identificatore di sessione per la sincronizzazione incrementale della configurazione.

Risposta del server

La risposta dei controlli attivi viene inviata dal server all'agent dopo l'elaborazione della richiesta dei controlli attivi.

```
{
  "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
    }
```

```
  ]
}
```

Campo	Tipo	Obbligatorio	Valore
response	*string*	sì	`success` | `failed`
info	*string*	no	Informazioni sull'errore in caso di errore.
data	*array of objects*	no	Item dei controlli attivi. Omissi se la configurazione del host non è cambiata.
key	*string*	no	Chiave dell'item con macro espanse.
itemid	*number*	no	Identificatore dell'item.
delay	*string*	no	Intervallo di aggiornamento dell'item. Gli intervalli flessibili/di pianificazione sono supportati sia da Zabbix agent che da Zabbix agent 2 a partire da Zabbix 7.0.
lastlogsize	*number*	no	lastlogsize dell'item.
mtime	*number*	no	mtime dell'item.
timeout	*string*	no	Timeout dell'item.
refresh_unsupported	*number*	no	Intervallo di aggiornamento degli item non supportati.
regexp	*array of objects*	no	Espressioni regolari globali.
name	*string*	no	Nome dell'espressione regolare globale.
expression	*string*	no	Espressione regolare globale.
expression_type	*number*	no	Tipo di espressione regolare globale.
exp_delimiter	*string*	no	Delimitatore dell'espressione regolare globale.
case_sensitive	*number*	no	Impostazione della distinzione tra maiuscole e minuscole per l'espressione regolare globale.
commands	*array of objects*	no	Comandi remoti da eseguire. Inclusi se l'esecuzione di comandi remoti è stata attivata da una operation di un'azione o dall'esecuzione manuale di uno script. Si noti che l'esecuzione di comandi remoti su un agent attivo è supportata a partire da Zabbix agent 7.0. Gli agent attivi meno recenti ignoreranno eventuali comandi remoti inclusi nella risposta del server dei controlli attivi.
command	*string*	no	Comando remoto.
id	*number*	no	Identificatore del comando remoto.
wait	*number*	no	Modalità di esecuzione del comando remoto ("0" (nowait) per i comandi dalle operations di un'azione; "1" (wait) per i comandi dall'esecuzione manuale di uno script).
config_revision	*number*	no	Identificatore di configurazione per la sincronizzazione incrementale della configurazione. Omissi se la configurazione del host non è cambiata. Incrementato se la configurazione del host cambia.

Il server deve rispondere con success.

Ad esempio:

1. L'agent apre una connessione TCP
2. L'agent richiede l'elenco dei controlli
3. Il server risponde con un elenco di item e comandi remoti da eseguire
4. L'agent analizza la risposta
5. La connessione TCP viene chiusa
6. L'agent avvia la raccolta periodica dei dati ed esegue i comandi remoti (supportato a partire 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 controllo attivo. Ciò è possibile perché chiunque può fingersi un agent attivo e richiedere i dati di configurazione degli item; l'autenticazione non avviene a meno che non si utilizzino le opzioni di encryption.

Invio dei dati raccolti

Invio dell'agent

La richiesta 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
}
```

Campo	Tipo	Obbligatorio	Valore
request	*string*	sì	`agent data`
data	*array of objects*	sì	Valori degli item.
id	*number*	sì	Identificatore del valore (contatore incrementale usato per verificare i valori duplicati in caso di problemi di rete).
itemid	*number*	sì	Identificatore dell'item.
value	*string*	no	Valore dell'item.
lastlogsize	*number*	no	lastlogsize dell'item.
mtime	*number*	no	mtime dell'item.
state	*number*	no	Stato dell'item.
source	*string*	no	Sorgente del log eventi del valore.
eventid	*number*	no	eventid del log eventi del valore.
severity	*number*	no	Severità del log eventi del valore.
timestamp	*number*	no	Timestamp del log eventi del valore.
clock	*number*	sì	Timestamp del valore (secondi dall'Epoch).
ns	*number*	sì	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 a partire da Zabbix agent 7.0. Gli agent attivi meno recenti 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 riuscita.
error	*string*	no	Messaggio di errore dell'esecuzione del comando remoto se l'esecuzione non è riuscita.
session	*string*	sì	Identificatore univoco della sessione generato ogni volta che l'agent viene avviato.
host	*string*	sì	Nome del host.
version	*string*	sì	Numero di versione dell'agent.

Campo	Tipo	Obbligatorio	Valore
variant	*number*	sì	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 sessione 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 ai dati dell'agent viene inviata dal server all'agent dopo l'elaborazione della richiesta dati dell'agent.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Campo	Tipo	Obbligatorio	Valore
response	*string*	sì	`success` | `failed`
info	*string*	sì	Risultati dell'elaborazione degli item.

Attention:
Se l'invio di alcuni valori non riesce sul server (per esempio, perché il host o l'item è stato disabilitato o eliminato), l'agent non ritenterà l'invio di tali valori.

Per esempio:

1. L'agent apre una connessione TCP
2. L'agent invia un elenco di valori
3. Il server elabora i dati e restituisce lo stato
4. La connessione TCP viene chiusa

Attention:
Il messaggio di errore verrà troncato a 2048 simboli sul lato server.

Messaggio heartbeat

L'agent invia

Il messaggio heartbeat viene inviato da un agent attivo a Zabbix server/proxy ogni HeartbeatFrequency secondi (configurato nel file di configurazione di Zabbix agent/ 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 del 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": "192.0.2.0:10055"
  }
}
```

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 64 KB, altrimenti verrà troncato a 64 KB durante la decodifica.

3 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 comuni dell'agent supportati su Windows

Item key	Gruppo utenti	
	Consigliato	Minimo tecnicamente accettabile (funzionalità limitate)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests

Item key	Gruppo utenti	
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Chiavi item specifiche di Windows

Chiave item	Gruppo utenti	
	Consigliato	Minimo tecnicamente accettabile (la funzionalità è limitata)
eventlog	Lettori registro eventi	Guests
net.if.list	Guests	Guests
perf_counter	Utenti Monitoraggio prestazioni	Utenti Monitoraggio prestazioni
proc_info	Amministratori	Guests
service.discovery	Guests	Guests

Chiave item	Gruppo utenti	
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Amministratori	Guests
vm.vmemory.size	Guests	Guests

4 Codifica dei valori restituiti

Zabbix server si aspetta che ogni valore testuale restituito sia in codifica 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 in codifica 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 i successivi verranno scartati quando il database memorizza questo valore. Non verrà scritto alcun messaggio di avviso in *zabbix_server.log*.
Rilevante almeno per MySQL versione 5.1.61

PostgreSQL

Se un valore contiene un carattere non ASCII in una codifica non UTF8, questo causerà il fallimento di una 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 PostgreSQL versione 9.1.3

5 Supporto per file di grandi dimensioni

Il supporto per file di grandi dimensioni, spesso abbreviato in LFS, è il termine usato per indicare la capacità di lavorare con file più grandi di 2 GB su 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 capacità del 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.

6 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]`.

7 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

Vedere nella tabella i valori supportati per il parametro 'memtype' su AIX.

Valore supportato	Descrizione	Origine nella struttura procentry64	Cerca di essere compatibile con
vsize [1]	Dimensione della memoria virtuale	pi_size	
pmem	Percentuale della memoria reale	pi_prm	ps -o pmem
rss	Dimensione del resident set	pi_trss + pi_drss	ps -o rssize
size	Dimensione del processo (codice + dati)	pi_dvm	"ps gvw" colonna SIZE
dsize	Dimensione dei dati	pi_dsize	
tsize	Dimensione del testo (codice)	pi_tsize	"ps gvw" colonna TSIZ
sdsize	Dimensione dei dati dalla libreria condivisa	pi_sdsize	
drss	Dimensione del resident set dei dati	pi_drss	
trss	Dimensione del resident set del testo	pi_trss	

Note per AIX:

1. Quando si scelgono i parametri per la chiave item proc.mem[] su AIX, cercare di specificare criteri di selezione del processo ristretti. In caso contrario, c'è 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 come specificare solo il parametro della riga di comando (espressione regolare da corrispondere) comporti l'auto-conteggio dello Zabbix agent, probabilmente non ciò che si desidera.

2. Non utilizzare "ps -ef" per esaminare i processi: mostra solo i processi non kernel. Usa "ps -Af" per vedere tutti i processi che saranno visibili a Zabbix agent.
3. Vediamo l'esempio di 'topasrec' per capire come Zabbix agent proc.mem[] seleziona i processi.

```
$ 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[] ha i seguenti argomenti:

`proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]`

Il 1° criterio è un nome di processo (argomento <name>). Nel nostro esempio, Zabbix agent lo vedrà come 'topasrec'. Per ottenere una corrispondenza, è necessario specificare 'topasrec' oppure lasciarlo vuoto. Il 2° criterio è un nome utente (argomento <user>). Per ottenere una corrispondenza, è necessario specificare 'root' oppure lasciarlo vuoto. Il 3° 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 ottenere una corrispondenza, è necessario specificare un'espressione regolare che corrisponda a questa stringa oppure lasciarlo vuoto.

Gli argomenti <mode> e <memtype> vengono applicati dopo l'uso dei tre criteri sopra menzionati.

FreeBSD

Vedere nella tabella i valori supportati per il parametro 'memtype' su FreeBSD.

Valore supportato	Descrizione	Sorgente nella struttura kinfo_proc	Cerca di essere compatibile con
vsize	Dimensione della memoria virtuale	kp_eproc.e_vm.vm_map_size or ki_size	ps -o vsz
pmem	Percentuale della memoria reale	calcolata da rss	ps -o pmem
rss	Dimensione del resident set	kp_eproc.e_vm.vm_rssize or 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 or ki_tsize	ps -o tsiz
dsize	Dimensione dei dati	kp_eproc.e_vm.vm_dsize or ki_dsize	ps -o dsiz
ssize	Dimensione dello stack	kp_eproc.e_vm.vm_ssize or ki_ssize	ps -o ssiz

Linux

Vedere nella tabella i valori supportati per il parametro 'memtype' su Linux.

Valore supportato	Descrizione	Sorgente nel file /proc/<pid>/status
vsize [1]	Dimensione della memoria virtuale	VmSize
pmem	Percentuale della memoria reale	(VmRSS/total_memory) * 100
rss	Dimensione del resident set	VmRSS
data	Dimensione del segmento dati	VmData
exe	Dimensione del segmento di codice	VmExe
hwm	Picco della dimensione del resident set	VmHWM
lck	Dimensione della memoria bloccata	VmLck
lib	Dimensione delle librerie condivise	VmLib
peak	Picco della dimensione della memoria virtuale	VmPeak
pin	Dimensione delle pagine bloccate in memoria	VmPin
pte	Dimensione delle voci della tabella delle pagine	VmPTE
size	Dimensione del codice del processo + segmenti dati + stack	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 l'auto-monitoraggio del processo di active check dello Zabbix agent con `proc.mem[...,...,...,...,data]` mostra un valore superiore di 4 kB rispetto a quello riportato dalla riga `VmData` nel file /proc/<pid>/status dell'agent. Al momento dell'auto-misurazione il segmento dati dell'agent aumenta di 4 kB e poi ritorna alla dimensione precedente.

Solaris

Vedere nella tabella i valori supportati per il parametro 'memtype' su Solaris.

Valore supportato	Descrizione	Origine nella struttura psinfo	Cerca di essere compatibile con
vsize [1]	Dimensione dell'immagine del processo	pr_size	ps -o vsz
pmem	Percentuale della 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.

8 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.

9 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 dettaglio in questa pagina per 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.

10 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

Su Linux, per ciascuna modalità vengono restituiti i seguenti parametri di processo:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: nome del processo
ppid: PID del padre	ppid: PID del 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 il processo è eseguito	pmem: percentuale della memoria reale
user: utente (reale) con cui il processo è eseguito	group: gruppo (reale) con cui il processo è eseguito	rss: dimensione del resident set
group: gruppo (reale) con cui il processo è eseguito	uid: ID utente	data: dimensione del segmento dati
uid: ID utente	gid: ID del gruppo con cui il processo è eseguito	exe: dimensione del segmento di codice

mode=process	mode=thread	mode=summary
gid: ID del gruppo con cui il processo è eseguito	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 totale di CPU trascorso in modalità utente (valore espresso in tick di clock — dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)	pin: dimensione delle pagine bloccate in memoria
rss: dimensione del resident set	cputime_system: tempo totale di CPU trascorso in modalità sistema (valore 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 context switch	stk: dimensione del segmento stack
hwm: picco della dimensione del resident set	page_faults: numero di page fault	swap: dimensione dello spazio di swap utilizzato
lck: dimensione della memoria bloccata		cputime_user: tempo totale di CPU trascorso in modalità utente (valore 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 totale di CPU trascorso in modalità sistema (valore espresso in tick di clock — dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)
peak: picco della dimensione della memoria virtuale		ctx_switches: numero di context switch
pin: dimensione delle pagine bloccate in memoria		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 proportional set size
stk: dimensione del segmento stack		
swap: dimensione dello spazio di swap utilizzato		
cputime_user: tempo totale di CPU trascorso in modalità utente (valore espresso in tick di clock — dividere per `getconf CLK_TCK` o `sysconf(_SC_CLK_TCK)` per ottenere i secondi)		
cputime_system: tempo totale di CPU trascorso in modalità sistema (valore 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, forma estesa)		
ctx_switches: numero di context switch		

mode=process	mode=thread	mode=summary
threads: numero di thread page_faults: numero di page fault pss: memoria 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

11 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 host viene considerata non raggiungibile dopo un controllo non riuscito (errore di rete, timeout) da parte di agent Zabbix, agent SNMP, agent IPMI o agent JMX. A partire da Zabbix 6.2.0, anche i controlli attivi dell'agent Zabbix 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 utilizzando uno degli item (incluse le regole LLD). Questi ricontrolli vengono eseguiti dagli unreachable poller (o dagli IPMI poller per i controlli IPMI). Per impostazione predefinita, l'intervallo tra controlli consecutivi di raggiungibilità è di 15 secondi.

Attention:
I controlli eseguiti da poller asincroni non vengono spostati agli unreachable poller.

Nel log del server Zabbix la non raggiungibilità è indicata da messaggi come:

```
L'item dell'agent Zabbix "system.cpu.load[percpu,avg1]" sull'host "New host" non è riuscito: primo errore
L'item dell'agent Zabbix "system.cpu.load[percpu,avg15]" sull'host "New host" non è riuscito: un altro err
```

I messaggi di log specificano l'item esatto che non è riuscito e il suo tipo (agent Zabbix).

Note:
Anche il parametro *Timeout* influisce su quanto presto un'interfaccia viene ricontrollata durante la non raggiungibilità. Se Timeout è impostato su 20 secondi e UnreachableDelay su 30 secondi, il controllo successivo avverrà 50 secondi dopo il 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.

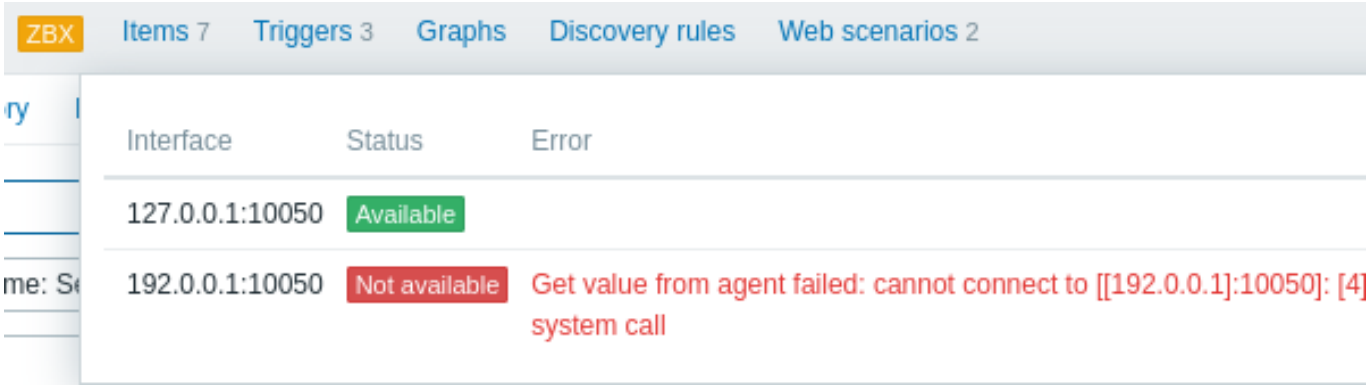
Interfaccia non disponibile

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
```

12 Monitoraggio remoto delle statistiche di Zabbix

Panoramica

È possibile rendere accessibili da remoto alcune metriche interne di Zabbix server e proxy da un'altra istanza di Zabbix o da uno strumento di terze parti. Questo può essere utile affinché i supporter/fornitori di 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, 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.

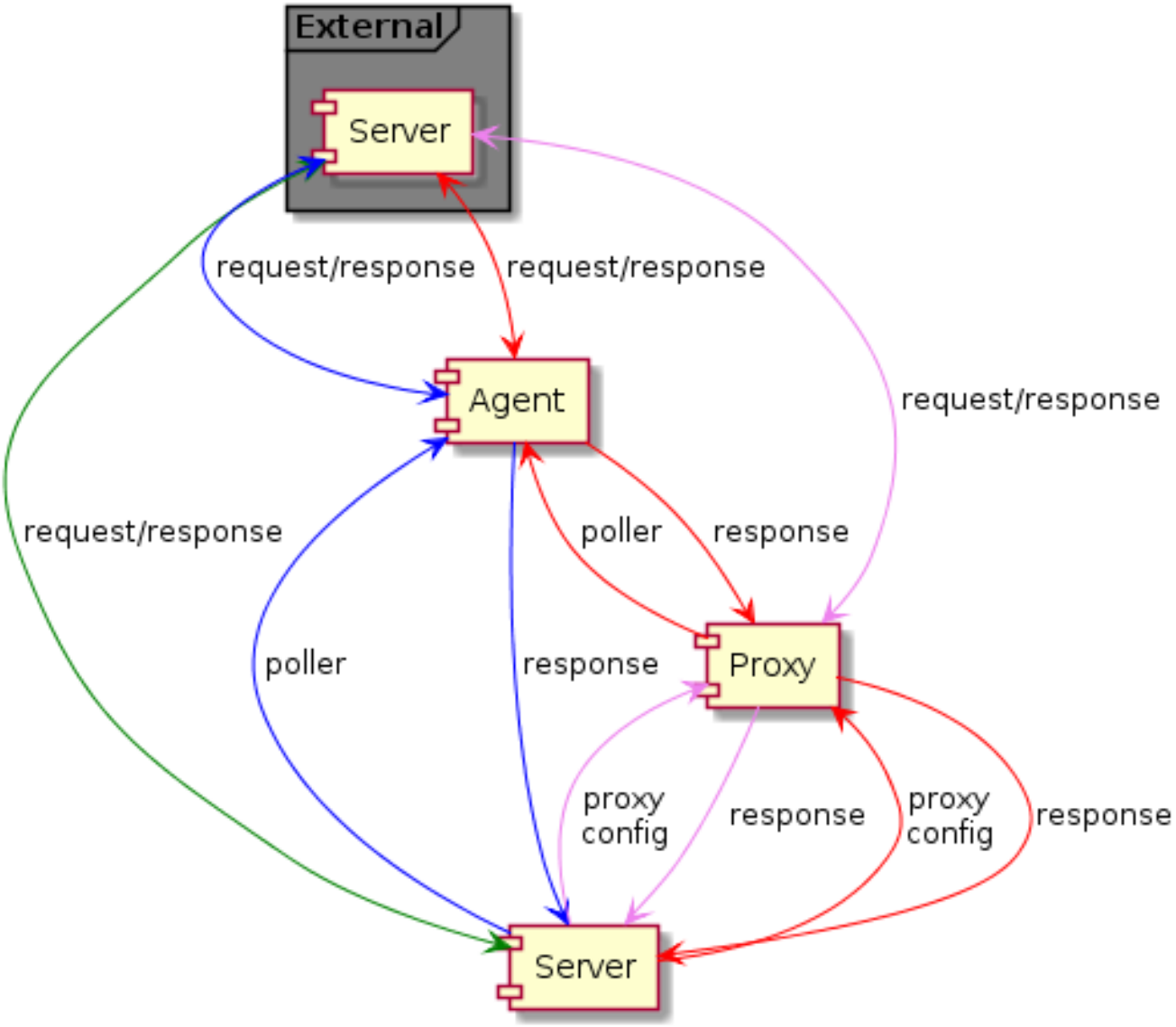
Item

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 di coda interni

Esistono anche altri due item che consentono specificamente di interrogare da remoto le statistiche della coda interna su un'altra istanza di Zabbix:

- item interno `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` - per interrogazioni dirette della coda interna verso un server/proxy Zabbix remoto
- item agent `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` - per interrogazioni della coda interna basate su agent verso un server/proxy Zabbix remoto

Vedi anche: Item interni, item Zabbix agent

13 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 utility client:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Durante la configurazione dei pacchetti, rispondere alle richieste, ad esempio:

```
Default Kerberos version 5 realm: EXAMPLE.COM
Kerberos servers for your realm: localhost (or your FQDN)
Administrative server for your Kerberos realm: localhost (or your FQDN)
```

2. Mappare un hostname descrittivo (opzionale, per test locali).

Modificare /etc/hosts e aggiungere una voce per il proprio DC e webserver se non si dispone di 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
```

Esempio di impostazioni:

```
[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 usare `.localdomain` o altri nomi non pubblici, aggiungere mappature esplicite dominio→realm in modo che la mappatura hostname→realm funzioni. Le incongruenze in questo punto causano errori `Server not found in Kerberos database`.

4. Inizializzare il database Kerberos (una sola volta, sul host KDC). Impostare una password master sicura quando richiesto:

```
sudo krb5_newrealm
```

5. Creare il principal `HTTP/host.fqdn@REALM` usando l'hostname esatto che i client utilizzeranno; preferire lettere minuscole (ad esempio `HTTP/web.example.com@EXAMPLE.COM`). Una mancata corrispondenza tra maiuscole/minuscole o nomi causa `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 sul web host (oppure lasciarlo in locale se si tratta della stessa macchina) e impostare permessi 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 Apache GSSAPI:

```
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 fallisce con `Invalid command 'GssapiCredStore'`, rimuovere la direttiva non supportata oppure aggiornare il modulo.

7. Configurare un VirtualHost (adattare `DocumentRoot` / il percorso alla propria UI di 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 il test (eseguire come l'utente che userà il ticket).

Ci si aspetta di vedere `krbtgt/EXAMPLE.COM@EXAMPLE.COM` nell'elenco dei ticket. 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 test SSO interattivi nel browser). I ticket emessi per un utente del sistema operativo diverso non saranno visibili a meno che `KRB5CCNAME` e i permessi non vengano adattati.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Testare lo scambio SPNEGO con curl (da un client con un TGT valido). Un `200 OK` (oppure un reindirizzamento all'applicazione) indica che SPNEGO è riuscito:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Facoltativamente, se la UI di Zabbix deve accettare accessi autenticati via HTTP, abilitare l'autenticazione HTTP nel frontend di Zabbix (ui/conf/zabbix.conf.php):

```
$ALLOW_HTTP_AUTH = true;
```

Nella web UI andare in *Users* > *Authentication* e passare alla scheda *HTTP settings*. Selezionare la casella *Enable HTTP authentication* e fare clic su *Ok* nella finestra pop-up. Selezionare "HTTP login form" nell'elenco a discesa *Default login form*. Decidere se *Case-sensitive login* è adatto alla propria policy di directory. Fare clic sul pulsante *Update* per completare.

12. Configurazione del browser (Firefox è usato come esempio): impostare `network.negotiate-auth.trusted-uris` sugli host che eseguono Negotiate (`dc01.example.com`) in modo che il browser invii automaticamente i token Kerberos.

In `about:config`:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Ora, visitando `http://dc01.example.com` si dovrebbe accedere direttamente a Zabbix senza il modulo di login.

13. Mantenere chiavi/ticket aggiornati. La durata predefinita di un ticket Kerberos è di circa 10 ore. Aggiungere un timer cron/systemd per evitare 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 di servizio sia presente nel keytab.
- `sudo tail -f /var/log/apache2/error.log` — monitorare eventuali errori GSSAPI (`gss_acquire_cred[_from]()` `failed to get server creds` indica keytab/permessi errati o principal mancante).
- `curl --negotiate` che restituisce 401/403 spesso indica principal errato, ticket assente, mancata corrispondenza dell'header host oppure un problema di permessi del filesystem; controllare i log e le mappature di 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.

Evitare di memorizzare password in chiaro a lunga durata sul host. Quando possibile, utilizzare keytab o principal di macchina aggiunti al dominio.

Quando si eseguono test o script che impostano `KRB5CCNAME` o copiano keytab, ricontrollare proprietà e permessi dopo l'operazione: il rifiuto delle credenziali da parte di un webserver è spesso dovuto a un problema di permessi dei file.

15 Creazione di nomi personalizzati dei contatori di prestazioni per VMware

Panoramica

Il percorso del contatore delle prestazioni VMware ha il formato `group/counter[rollup]` dove:

- `group` - il gruppo del contatore delle prestazioni, ad esempio *cpu*
- `counter` - il nome del contatore delle prestazioni, ad esempio *usagemhz*
- `rollup` - il tipo di rollup del contatore delle prestazioni, ad esempio *average*

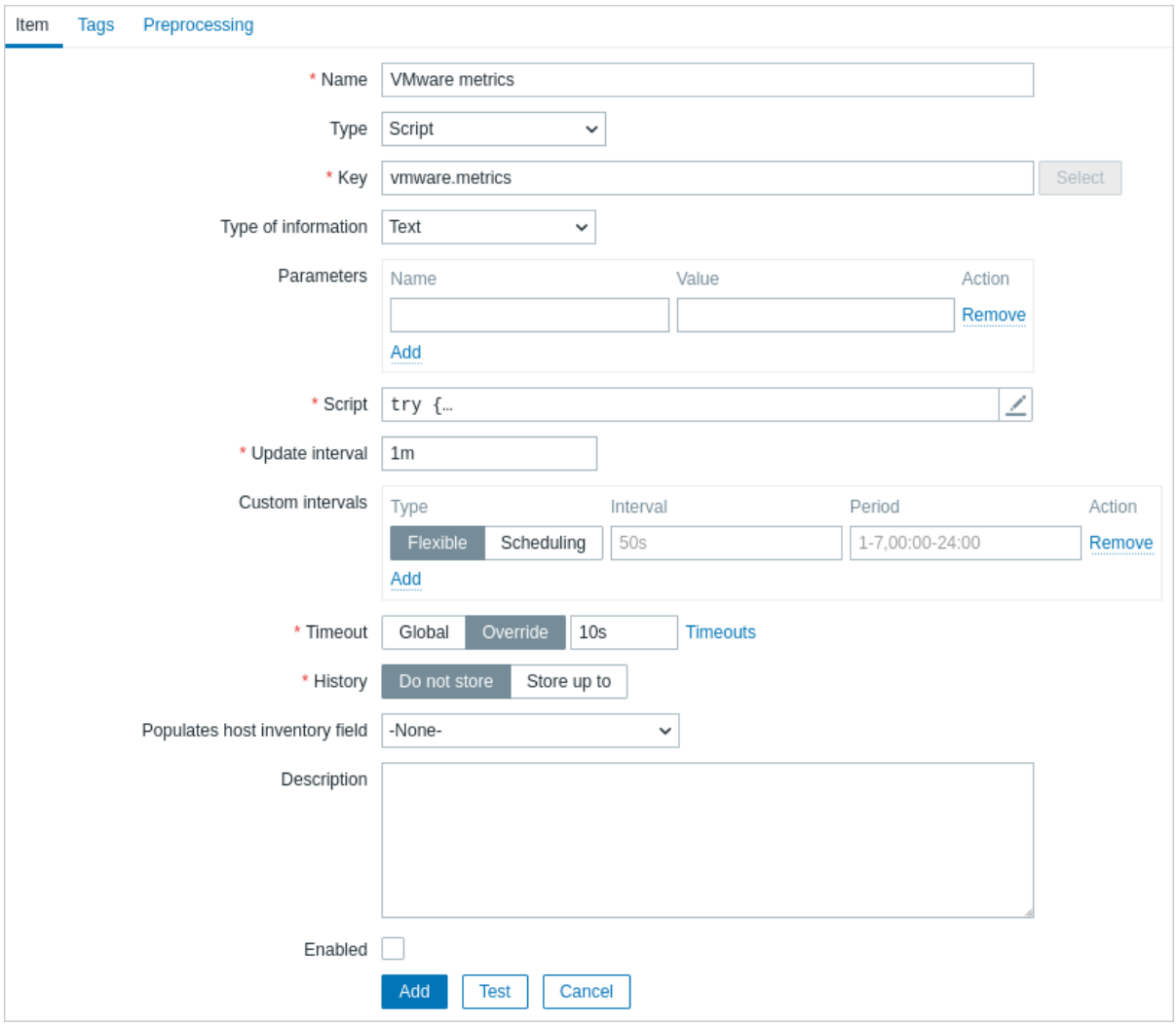
Quindi, l'esempio sopra riportato produrrebbe il seguente percorso del contatore: `cpu/usagemhz[average]`

Le descrizioni dei gruppi dei contatori delle prestazioni, i nomi dei contatori e i tipi di rollup sono disponibili nella documentazione VMware.

È possibile ottenere i nomi interni e creare nomi personalizzati dei contatori delle prestazioni utilizzando uno script item in Zabbix.

Configurazione

1. Creare un item Script disabilitato sul VMware host principale (dove è presente l'item **eventlog[]**) con i seguenti parametri:



- *Nome*: Metriche VMware
- *Tipo*: Script
- *Chiave*: vmware.metrics
- *Tipo di informazioni*: Testo
- *Script*: copiare e incollare lo script fornito di seguito
- *Timeout*: 10
- *Storico*: Non memorizzare
- *Abilitato*: 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, premere il pulsante *Test*, quindi premere *Get value*.



Copiare l'XML ricevuto in un qualsiasi formattatore XML e trovare 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>
```

Utilizzare XPath per estrarre il percorso del contatore dall'XML ricevuto. Per l'esempio sopra, l'XPath sarà:

campo	xPath	valore
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Il percorso risultante del contatore delle prestazioni in questo caso è: `cpu/usagemhz[average]`

16 Valori restituiti 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 chiavi:

- **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, impostare a 0)
- **arch** - architettura del pacchetto
- **buildtime** - un oggetto con 2 voci:
  - **timestamp** - timestamp UNIX del momento in cui il pacchetto è stato compilato (se non disponibile, impostare a 0)
  - **value** - data e ora in formato leggibile del momento in cui il pacchetto è stato compilato (se non disponibile, impostare a stringa vuota)
- **installtime** - un oggetto con 2 voci:
  - **timestamp** - timestamp UNIX del momento in cui il pacchetto è stato installato (se non disponibile, impostare a 0)
  - **value** - data e ora in formato leggibile del momento in cui il pacchetto è stato installato (se non disponibile, impostare 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": ""
        }
    }
]
```


17 Valori restituiti 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 server DNS, l'item `net.dns.get[192.0.2.0,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[192.0.2.0,example.com]` può restituire un errore:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: 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	
Il DNS non è disponibile.	-1	"Communication error"
Si verifica un errore durante l'analisi del JSON	-2	"Received unexpected response"

Con parametri aggiuntivi, l'item `net.dns.get[192.0.2.0,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": "192.0.2.0"
            },
            "rdlength": 4,
            "ttl": 19308,
            "type": "A"
        },
        {
            "class": "IN",
            "name": "example.com.",
            "rdata": {
                "algorithm": 13,
                "expiration": 1704715951,
                "inception": 1702910624,
                "key_tag": 21021,
                "labels": 2,
                "orig_ttl": 86400,
                "signature": "HVBOBcJJQy0S08J3f8kviPj8UkEUj7wmyiMyQqPSWgQIY9SCEJ5plq6KuxJmtAek1txZWXDo+6tp
                "signer_name": "example.com.",
                "type_covered": "A"
            },
            "rdlength": 95,
            "ttl": 19308,
            "type": "RRSIG"
        }
    ],
    "flags": [
        "RD",
        "RA",
        "AD",
        "CD"
    ],
    "query_time": "0.058221",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "ANY"
        }
    ],
    "response_code": "NOERROR",
    "zbx_error_code": 0
}
```


Vedi anche

Per ulteriori informazioni sui record DNS, vedere:

- Domain Names - Implementation and Specification
- Domain Name System (DNS) Parameters

18 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
```

19 Supporto per valori JSON di grandi dimensioni

Per garantire una gestione affidabile di valori di grandi dimensioni con il tipo di dati JSON (1MiB o più), rivedere e applicare, se necessario, le seguenti modifiche di configurazione:

1. Se si utilizza MySQL o MariaDB, aumentare la dimensione massima consentita dei pacchetti in `/etc/mysql/my.cnf` (su alcuni sistemi il valore predefinito può essere 1MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Regolare, se necessario, ulteriori variabili di sistema del server MySQL/MariaDB e variabili di sistema InnoDB. 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 del server Zabbix.

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 server/proxy Zabbix e agent (ad esempio, i timeout degli item o il valore del parametro `Timeout` nei file di configurazione di Zabbix server/proxy e agent). In caso contrario, i messaggi tra di essi potrebbero non essere completati e si potrebbero visualizzare errori come *"message length does not match expected length"*. Rivedere inoltre, se necessario, altri timeout, come il timeout dello script o il timeout del test dell'item.

Parametri di modbus.get 14

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

Parameter	Description	Defaults	Example
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
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

5 Funzioni supportate

Vedere Espressione del trigger.

6 Macro

È possibile utilizzare i macro supportati e i macro utente supportati per posizione.

1 Macro supportate per posizione

Panoramica

La tabella macro supportate per posizione contiene il riferimento di tutte le macro (predefinite, di espressione e definibili dall'utente) supportate nella posizione specifica.

Se sono supportate tutte le macro di un gruppo correlato, viene indicato solo il nome del gruppo. Fai clic su ciascun nome di macro o nome di gruppo per visualizzare tutti i dettagli delle macro.

L'elenco dettagli delle macro contiene un elenco dettagliato di tutte le macro **predefinite**, raggruppate per area applicativa. Nota che, per personalizzare i valori delle macro (ad esempio, abbreviare o estrarre sottostringhe specifiche), puoi utilizzare le funzioni macro.

Macro supportate per posizione

Posizione	Macro supportate	Commenti
Azione		

Posizione	Macro supportate	Commenti
Condizione del periodo di tempo, Durata predefinita del passo dell'operazione, Durata del passo	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate
Notifiche e co-mandi		
Basato su trigger	Macro Azione Macro Data/ora {ESC.HISTORY} Macro Evento, 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 definite dall'utente	Vedi anche: Uso delle macro nei messaggi
Aggiornamento problema	Macro Azione Macro Data/ora {ESC.HISTORY} Macro Evento, macro Evento causa/sintomo, macro Evento di ripristino Macro Aggiornamento evento: tutte (eccetto {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Macro di espressione: {?EXPRESSION} Macro Funzione 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 definite dall'utente	

Posizione	Macro supportate	Commenti
Ripristino problema	Macro Evento di ripristino {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Discovery	Macro Azione 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 Azione 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 Azione Macro Data/ora {ESC.HISTORY} Macro evento: tutte eccetto {EVENT.ACK.STATUS}, {EVENT.OPDATA} Macro Servizio Macro definite dall'utente	
Ripristino servizio	Macro Evento (eccetto {EVENT.ACK.STATUS}, {EVENT.OPDATA}), macro Evento di ripristino	
Aggiornamento servizio	Macro Azione Macro Data/ora {ESC.HISTORY} Macro Evento (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 Servizio Macro definite dall'utente	
Notifiche interne	Macro Azione Macro Data/ora {ESC.HISTORY} Macro Evento (eccetto {EVENT.NSEVERITY}, {EVENT.SEVERITY}), macro Evento di ripristino 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 low-level discovery	
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 definite dall'utente	

Posizione	Macro supportate	Commenti
Parametri script di avviso	Macro Avviso Macro definite dall'utente (solo globali)	
Connettore		
Campo URL, Username, Password, Bearer token, HTTP proxy, Certificato SSL, File chiave SSL, Password chiave SSL	Macro definite dall'utente (solo globali)	
Campo Timeout	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Nome grafico	Macro di espressione: {?EXPRESSION}	In questa posizione, in questa macro è supportata solo una singola funzione **avg**, **last**, **max**, **min** con secondi come parametro (possono essere usati i suffissi temporali). La macro {HOST.HOST<1-9>} può essere usata come host all'interno della macro (vedi anche macro indicizzate). Esempio di utilizzo: {?avg(/{HOST.HOST}/item.key,1h)} Utilizzo non supportato: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Host, pro-totipo host		
IP/DNS interfaccia	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Macro definite dall'utente [1]	[1] Per l'indirizzo IP, una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate
Porta interfaccia	Macro definite dall'utente	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate
Community SNMP Nome contesto SNMPv3, Nome sicurezza, Passphrase di autenticazione, Passphrase di privacy	Macro definite dall'utente	

Posizione	Macro supportate	Commenti
Username, password IPMI		
Item, prototipo item, regola LLD		
Nome	Macro definite dall'utente	Non supportate nei nomi delle regole LLD.
Descrizione	Macro definite dall'utente	
Parametri chiave	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite 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.
Campi Intervallo di aggiornamento, Intervalli personalizzati, Timeout (disponibile per i tipi di item supportati), Conserva fino a (per storico/trend)	Macro definite dall'utente	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Campi Elimina risorse perse, Disabilita risorse perse	Macro definite dall'utente	Questi campi sono supportati solo per le regole LLD. Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Parametri del passo di preprocessing (inclusi script personalizzati), *parametri personalizzati di gestione errori* (campi *Imposta valore a* e *Imposta errore a*)	Macro definite dall'utente	
Espressioni regolari del filtro	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	Questi campi sono supportati solo per le regole LLD.
Espressioni regolari del filtro di override	Macro definite dall'utente	Questi campi sono supportati solo per le regole LLD.

Posizione	Macro supportate	Commenti
Operazioni di override: *Intervallo di aggiornamento*, *Periodo di conservazione storico*, *Periodo di conservazione trend* (per prototipi item)	Macro definite dall'utente	Questi campi sono supportati solo per le regole LLD. Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
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 definite dall'utente	
Script item *nomi e valori dei parametri*		
Formula item **Calcolato/aggregato** (*costanti dell'espressione*, *parametri funzione*, *parametri chiave item*, (solo item aggregato) *condizioni del filtro* (nome gruppo host e nome tag))	Macro definite dall'utente macros	
Database monitor item *campi Username, Password, query SQL*	Macro definite dall'utente	
HTTP agent item *campi URL, Query, Corpo richiesta, Header (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 definite dall'utente	Gli URL che contengono una macro segreta non funzioneranno, poiché la macro al loro interno verrà risolta come "******"
HTTP agent item *campo HTTP Proxy*	Macro item: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Macro definite dall'utente	
HTTP agent item *campi Codici di stato richiesti, username/password autenticazione HTTP, password chiave SSL*	Macro definite dall'utente	
HTTP agent item *campo Host consentiti*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	
JMX agent item *campo Endpoint*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Macro definite dall'utente	
Trapper item *campo Host consentiti*		
SNMP agent item *campo OID SNMP*	Macro definite dall'utente	

Posizione	Macro supportate	Commenti
SSH e Telnet item *script*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	
SSH agent item *campi Username, File chiave pubblica, File chiave privata, Password*	Macro definite dall'utente	
Telnet agent item *campi Username, Password*	Macro definite dall'utente	
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 questa macro è supportata solo una singola funzione **avg**, **last**, **max**, **min** con secondi come parametro (possono essere usati i suffissi temporali). La macro {HOST.HOST<1-9>} può essere usata come host all'interno della macro. Esempio di utilizzo: {?avg(/{HOST.HOST}/item.key,1h)} Utilizzo 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}	
Network discovery		
Campo Intervallo di aggiornamento	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Campi Community SNMP, OID SNMP Campi Nome contesto SNMPv3, Nome sicurezza, Passphrase di autenticazione, Passphrase di privacy	Macro definite dall'utente (solo globali)	

Posizione	Macro supportate	Commenti
Proxy		
Indirizzo per active agents > *Porta* (se il proxy appartiene a un gruppo) Per la modalità proxy passivo: campi *Indirizzo* e *Porta* dell'interfaccia *Campi Timeout per tipi di item* in modalità override	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Gruppo proxy		
Campi Periodo di failover, Numero minimo di proxy	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Script		
Comandi, testo di conferma (Tipo: script, **azione host** 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 definite dall'utente	[1] Solo testo di conferma [2] Supportato per Zabbix server e Zabbix proxy
Comandi (Tipo: script, **azione evento** manuale)	Macro Data/ora Macro Evento, macro Evento causa/sintomo, macro Evento di ripristino (se il ripristino è avvenuto) 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 definite dall'utente	
Testo di conferma (Tipo: script, **azione evento** 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 definite dall'utente	

Posizione	Macro supportate	Commenti
URL, testo di conferma (Tipo: URL, **azione host/evento** manuale)		
Nome e valore tag	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 tag trigger) Macro definite dall'utente	Le macro di nome e valore 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 definite dall'utente	
Nome evento	Macro data/ora: {TIME}, {TIMESTAMP} Macro di espressione: {?EXPRESSION} Macro Funzione 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 definite dall'utente	
Espressione	{TRIGGER.VALUE} Macro definite dall'utente [1]	[1] Solo in costanti e parametri funzione; le macro segrete non sono supportate
Campi Tag per corrispondenza, Nome voce menu, URL voce menu	Macro definite dall'utente	Gli URL che contengono una macro segreta non funzioneranno, poiché la macro al loro interno 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 definite 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 User-Parameter Scenario web	Macro/riferimenti posizionali: $1...$9	
Campi Nome, Agent, HTTP proxy	Macro definite dall'utente	

Posizione	Macro supportate	Commenti
Campo Intervallo di aggiornamento	Macro definite dall'utente	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Campi Variabili, Header, Certificato SSL, File chiave SSL	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	
Campi Utente, Password, Password chiave SSL	Macro definite dall'utente	
Passo scenario web *campi Nome, Variabili (solo valori)*	Macro definite dall'utente	
Passo scenario web *campi Timeout, Codici di stato richiesti*	Macro definite dall'utente	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Passo scenario web *campi URL, Header (nomi e valori), Stringa richiesta*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	
Passo scenario web *campo Post*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Tipo di supporto webhook		
Valori dei parametri	Macro Avviso Tutte le macro supportate nelle notifiche di problema basate su trigger (vedi sopra)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Script JavaScript, nome parametro script *JavaScript*	Macro definite dall'utente (solo globali)	
Nome voce menu, URL voce 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 (eccetto {ITEM.STATE.ERROR}) Macro definite dall'utente (solo globali)	
Honeycomb widget *campo Etichetta primaria/secondaria*		
Valore item 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 definite dall'utente (solo globali)	

Posizione	Macro supportate	Commenti
URL widget (dinamico) *campo URL*	Macro host: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Macro definite dall'utente	
Altre posizioni		
Campo Orario di lavoro	Macro definite dall'utente (solo globali)	Una singola macro deve riempire l'intero campo; macro multiple/miste con testo non sono supportate.
Campo Timeout per tipi di item (impostazione globale)		
Supporto utente *campo Quando attivo*	Macro definite dall'utente (solo globali)	
Tipo di supporto email *campi Username, Password*	Macro definite dall'utente (solo globali)	Si consiglia l'uso di macro segrete.
Report pianificati *campi Oggetto, Messaggio*	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 fa riferimento il nome del tag.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri UTF multibyte non inglesi) deve essere racchiuso tra doppi apici. Le virgolette e le barre rovesciate all'interno di un nome di tag tra virgolette devono essere precedute da un carattere di 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 referenziato dal nome del tag.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri UTF multibyte non inglesi) deve essere racchiuso tra doppi apici. Le virgolette e le barre rovesciate all'interno di un nome di tag tra virgolette devono essere precedute da una barra rovesciata.
 Utilizzato 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 del host, nome dell'evento, gravità, età, tag del servizio e valori.
 Questa macro viene utilizzata 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 oggetti 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 dall'utente 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 aggiornamento).

{EVENT.UPDATE.ACTIONJSON}

Un array JSON contenente i dettagli dell'azione o delle azioni eseguite durante l'aggiornamento del problema.
 Valori possibili 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 soppresso indefinitamente);
- 0, 1, 2, 3, 4, 5 (per le proprietà di gravità `old` e `new`).

 Ad esempio: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"s`

{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 N-esima funzione basata su item nell'espressione del trigger al momento dell'evento.
 Vengono conteggiate solo le funzioni con `/host/key` come primo parametro.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {FUNCTION.VALUE<1-9>}, per fare riferimento alla prima, seconda, terza, ecc. funzione in un'espressione del trigger. Vedere macro indicizzate.

{FUNCTION.RECOVERY.VALUE}

Il valore della N-esima funzione basata su item nell'espressione di recovery al momento dell'evento.*
 Vengono conteggiate solo le funzioni con `/host/key` come primo parametro.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {FUNCTION.RECOVERY.VALUE<1-9>}, per fare riferimento alla prima, seconda, terza, ecc. funzione in un'espressione di trigger. Vedere macro indicizzate.

Macro host

{HOST.CONN}

L'indirizzo IP o il nome DNS del host, a seconda delle impostazioni del 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 anche un'interfaccia SNMP non è definita, verrà utilizzata l'interfaccia JMX principale. Se non è definita nemmeno un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se il host non dispone di alcuna interfaccia, la macro verrà risolta in 'UNKNOWN'.

 Questa macro può essere utilizzata con un indice numerico come {HOST.CONN<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{HOST.DESCRIPTION}

La descrizione dell'host.

 Questa macro può essere utilizzata con un indice numerico come {HOST.DESCRIPTION<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{HOST.DNS}

Il nome DNS del host.
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta nell'interfaccia agent principale. Se un'interfaccia agent non è definita, verrà utilizzata l'interfaccia SNMP principale. Se anche un'interfaccia SNMP non è definita, verrà utilizzata l'interfaccia JMX principale. Se non è definita nemmeno un'interfaccia JMX, verrà utilizzata l'interfaccia IPMI principale. Se il host non ha alcuna interfaccia, la macro viene risolta come 'UNKNOWN'.

 Questa macro può essere utilizzata con un indice numerico come {HOST.DNS<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{HOST.HOST}

Il nome tecnico del host.
 Le funzioni macro non sono supportate per questa macro se viene utilizzata come segnaposto nel primo parametro di una funzione di history, ad esempio `last(/{HOST.HOST}/{ITEM.KEY})`.

 Questa macro può essere utilizzata con un indice numerico come {HOST.HOST<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{HOST.ID}

L'ID host.

 Questa macro può essere utilizzata con un indice numerico come {HOST.ID<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{HOST.IP}

L'indirizzo IP del host.
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web, la macro verrà risolta nell'interfaccia agent principale. Se un'interfaccia agent non è definita, verrà utilizzata l'interfaccia SNMP principale. Se anche un'interfaccia SNMP non è definita, verrà utilizzata l'interfaccia JMX principale. Se nemmeno un'interfaccia JMX è definita, verrà utilizzata l'interfaccia IPMI principale. Se il host non ha alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere utilizzata con un indice numerico come {HOST.IP<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{HOST.METADATA}

I metadati dell'host.

 Utilizzati solo per l'autoregistrazione attiva dell'agent.

{HOST.NAME}

Il nome visibile dell'host.

 Questa macro può essere utilizzata con un indice numerico come {HOST.NAME<1-9>} per indicare il primo, il secondo, il terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{HOST.PORT}

La porta del host (agent).
 Negli script globali, nei campi IP/DNS dell'interfaccia e negli scenari web la macro verrà risolta nell'interfaccia agent principale. Se un'interfaccia agent non è definita, verrà utilizzata l'interfaccia SNMP principale. Se anche un'interfaccia SNMP non è definita, verrà utilizzata l'interfaccia JMX principale. Se nemmeno un'interfaccia JMX è definita, verrà utilizzata l'interfaccia IPMI principale. Se il host non ha alcuna interfaccia, la macro viene risolta in 'UNKNOWN'.

 Questa macro può essere utilizzata con un indice numerico come {HOST.PORT<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

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 utilizzata con un indice numerico, ad esempio {INVENTORY.ALIAS<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.ASSET.TAG}

Il campo *Asset tag* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.ASSET.TAG<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.CHASSIS}

Il campo *Chassis* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.CHASSIS<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.CONTACT}

Il campo *Contatto* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.CONTACT<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.CONTRACT.NUMBER}

Il campo *Numero contratto* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.CONTRACT.NUMBER<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.DEPLOYMENT.STATUS}

Il campo *Stato di distribuzione* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.DEPLOYMENT.STATUS<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedi macro indicizzate.

{INVENTORY.HARDWARE}

Il campo *Hardware* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HARDWARE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.HARDWARE.FULL}

Il campo *Hardware (dettagli completi)* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HARDWARE.FULL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.HOST.NETMASK}

Il campo *Maschera di sottorete dell'host* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HOST.NETMASK<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.HOST.NETWORKS}

Il campo *Reti del host* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HOST.NETWORKS<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.HOST.ROUTER}

Il campo *Host router* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HOST.ROUTER<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.HW.ARCH}

Il campo *Architettura hardware* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HW.ARCH<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.HW.DATE.DECOMM}

Il campo *Data di dismissione dell'hardware* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HW.DATE.DECOMM<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.HW.DATE.EXPIRY}

Il campo *Data di scadenza della manutenzione hardware* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HW.DATE.EXPIRY<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.HW.DATE.INSTALL}

Il campo *Data di installazione dell'hardware* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HW.DATE.INSTALL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.HW.DATE.PURCHASE}

Il campo *Data di acquisto dell'hardware* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.HW.DATE.PURCHASE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.INSTALLER.NAME}

Il campo *Nome dell'installatore* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.INSTALLER.NAME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.LOCATION}

Il campo *Posizione* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.LOCATION<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.LOCATION.LAT}

Il campo *Latitudine della posizione* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.LOCATION.LAT<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.LOCATION.LON}

Il campo *Longitudine della posizione* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.LOCATION.LON<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.MACADDRESS.A}

Il campo *MAC address A* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.MACADDRESS.A<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{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 fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.MODEL}

Il campo *Model* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.MODEL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.NAME}

Il campo *Nome* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.NAME<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.NOTES}

Il campo *Note* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.NOTES<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.OOB.IP}

Il campo *indirizzo IP OOB* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OOB.IP<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.OOB.NETMASK}

Il campo *maschera di sottorete OOB* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OOB.NETMASK<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.OOB.ROUTER}

Il campo *router OOB* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OOB.ROUTER<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.OS}

Il campo *OS* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OS<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.OS.FULL}

Il campo *OS (dettagli completi)* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OS.FULL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.OS.SHORT}

Il campo *OS (Short)* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.OS.SHORT<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.CELL}

Il campo *Cellulare principale del POC* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.CELL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.EMAIL}

Il campo *Email POC principale* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.NAME}

Il campo *Nome POC primario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.NAME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.NOTES}

Il campo *Note del POC primario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.NOTES<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.PHONE.A}

Il campo *Telefono A del POC primario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.PHONE.B}

Il campo *Telefono B del POC primario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.PRIMARY.SCREEN}

Il campo *Nome schermata POC primaria* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.CELL}

Il campo *Cellulare POC secondario* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.CELL<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.EMAIL}

Il campo *Email POC secondaria* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.NAME}

Il campo *Nome POC secondario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.NAME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.NOTES}

Il campo *Note del POC secondario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.NOTES<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.PHONE.A}

Il campo *Telefono A del POC secondario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.PHONE.B}

Il campo *Telefono B del POC secondario* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.POC.SECONDARY.SCREEN}

Il campo *Nome schermata POC secondaria* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SERIALNO.A}

Il campo *Numero di serie A* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SERIALNO.A<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SERIALNO.B}

Il campo *Numero di serie B* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SERIALNO.B<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ADDRESS.A}

Il campo *Indirizzo del sito A* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.A<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ADDRESS.B}

Il campo *Indirizzo del sito B* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.B<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ADDRESS.C}

Il campo *Indirizzo del sito C* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.ADDRESS.C<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SITE.CITY}

Il campo *Città del sito* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.CITY<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.SITE.COUNTRY}

Il campo *Paese del sito* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.COUNTRY<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SITE.NOTES}

Il campo *Note del sito* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.NOTES<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SITE.RACK}

Il campo *Posizione rack del sito* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.RACK<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SITE.STATE}

Il campo *Stato/provincia del sito* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.STATE<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SITE.ZIP}

Il campo *CAP/postale del sito* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SITE.ZIP<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE}

Il campo *Software* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.APP.A}

Il campo *Applicazione software A* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.A<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.APP.B}

Il campo *Applicazione software B* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.B<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.APP.C}

Il campo *Applicazione software C* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.C<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.APP.D}

Il campo *Applicazione software D* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.D<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.APP.E}

Il campo *Applicazione software E* nell'inventario del host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.APP.E<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.SOFTWARE.FULL}

Il campo *Software (dettagli completi)* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.SOFTWARE.FULL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.TAG}

Il campo *Tag* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.TAG<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.TYPE}

Il campo *Tipo* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.TYPE<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.TYPE.FULL}

Il campo *Tipo (dettagli completi)* nell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.TYPE.FULL<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.URL.A}

Il campo *URL A* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.URL.A<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.URL.B}

Il campo *URL B* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.URL.B<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

{INVENTORY.URL.C}

Il campo *URL C* nell'inventario dell'host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.URL.C<1-9>}, per puntare al primo, secondo, terzo, ecc. host in un'espressione di trigger. Vedere macro indicizzate.

{INVENTORY.VENDOR}

Il campo *Vendor* dell'inventario host.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {INVENTORY.VENDOR<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione trigger. Vedere macro indicizzate.

Macro degli item

{ITEM.DESCRIPTION}

La descrizione dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere utilizzata con un indice numerico come {ITEM.DESCRIPTION<1-9>} per indicare il primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.DESCRIPTION.ORIG}

La descrizione originale (con le macro non risolte) dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere utilizzata con un indice numerico come {ITEM.DESCRIPTION.ORIG<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.ID}

L'ID numerico dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere utilizzata con un indice numerico come {ITEM.ID<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.KEY}

La chiave dell'N-esimo item nell'espressione del trigger che ha causato una notifica.
 Le funzioni macro non sono supportate per questa macro se viene utilizzata come segnaposto nel primo parametro di una funzione di storico, ad esempio `last(/{HOST.HOST}/{ITEM.KEY})`.

 Questa macro può essere utilizzata con un indice numerico come {ITEM.KEY<1-9>} per fare riferimento al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.KEY.ORIG}

La chiave originale (con le macro non risolte) dell'N-esimo item nell'espressione del trigger che ha causato una notifica.

 Questa macro può essere utilizzata con un indice numerico come {ITEM.KEY.ORIG<1-9>} per puntare al primo, secondo, terzo, ecc. host in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.LASTVALUE}

L'ultimo valore dell'N-esimo item nell'espressione del trigger che ha causato una notifica.
 Nel frontend verrà risolto come *UNKNOWN* se l'ultimo valore della cronologia è stato raccolto più tempo fa rispetto al *Periodo massimo di visualizzazione della cronologia* (impostato nella sezione di menu Amministrazione → Generale).
 Quando viene utilizzata nel nome del problema, la macro non verrà risolta con l'ultimo valore dell'item durante la visualizzazione degli eventi di problema; manterrà invece il valore dell'item del momento in cui si è verificato il problema.
Quando viene utilizzata nelle notifiche, in alcuni casi la macro potrebbe non essere risolta con l'ultimo valore dell'item nel 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 quello in cui è stata creata la notifica. Per evitare questo comportamento, è possibile usare la macro {ITEM.VALUE}, che viene risolta con il valore presente nel 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 testo/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 valore dell'item.
Per ottenere il valore completo, è possibile usare le funzioni macro, poiché nessun valore viene troncato dal server. Ad esempio: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LASTVALUE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{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'utente (ad esempio 1m 45s).

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LASTVALUE.AGE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{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 (in Monitoring -> Problems list), 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 utilizzata con un indice numerico, ad esempio {ITEM.LASTVALUE.DATE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.LASTVALUE.TIME}

L'ora in cui è stato raccolto l'ultimo valore dell'item.
 Visualizzata nel formato `HHMMSS`.
 Il fuso orario del server viene utilizzato nei nomi dei trigger (in Monitoring -> Problems list), 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 utilizzata con un indice numerico, ad esempio {ITEM.LASTVALUE.TIME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.LASTVALUE.TIMESTAMP}

Il timestamp UNIX relativo al momento in cui è stato raccolto l'ultimo valore dell'item.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LASTVALUE.TIMESTAMP<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.LOG.AGE}

L'età dell'evento di log.
Con precisione fino al secondo.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.AGE<1-9>}, per indicare il primo, il secondo, il terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.LOG.DATE}

La data in cui la voce di log è stata scritta nel log.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.DATE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.LOG.EVENTID}

L'ID dell'evento nel log eventi.
 Solo per il monitoraggio del log eventi di Windows.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.EVENTID<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.LOG.NSEVERITY}

La gravità numerica dell'evento nel log eventi.
 Solo per il monitoraggio del log eventi di Windows.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.NSEVERITY<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere macro indicizzate.

{ITEM.LOG.SEVERITY}

La gravità testuale dell'evento nel log eventi.
 Solo per il monitoraggio del log eventi di Windows.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.SEVERITY<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.LOG.SOURCE}

L'origine dell'evento nel log eventi.
 Solo per il monitoraggio del log eventi di Windows.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.SOURCE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedi macro indicizzate.

{ITEM.LOG.TIME}

L'ora in cui la voce di log è stata scritta nel log.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.LOG.TIME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.LOG.TIMESTAMP}

Il timestamp UNIX che indica quando 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 fare riferimento al primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere macro indicizzate.

{ITEM.NAME}

Il nome dell'item con tutte le macro risolte.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.NAME<1-9>}, per indicare il primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere macro indicizzate.

{ITEM.NAME.ORIG}

Il nome originale (con le macro non risolte) dell'item.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.NAME.ORIG<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.STATE}

L'ultimo stato dell'N-esimo item nell'espressione del trigger che ha causato una notifica.
 Valori possibili: *Non supportato*, *Normale*.

 Questa macro può essere utilizzata 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 macro indicizzate.

{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}

Risolto in uno dei seguenti modi:

- Il valore storico (al momento dell'evento) dell'N-esimo item nell'espressione del trigger, se usato nel contesto del cambiamento di stato del trigger, ad esempio durante la visualizzazione degli eventi o l'invio di notifiche;

- L'ultimo valore dell'N-esimo item nell'espressione del trigger, se usato al di fuori del contesto del cambiamento di stato del trigger, ad esempio durante la visualizzazione di un elenco di trigger in una finestra popup di selezione. In questo caso funziona allo stesso modo di {ITEM.LASTVALUE}.

Nel primo caso verrà risolto in *UNKNOWN* se il valore storico è già stato eliminato o non è mai stato memorizzato.
Nel secondo caso, e solo nel frontend, verrà risolto in *UNKNOWN* se l'ultimo valore storico è stato raccolto più tempo fa del periodo *Max history display period* (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 valore dell'item.
Per risolvere al valore completo, è possibile usare le funzioni macro, 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 fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{ITEM.VALUE.AGE}

Il tempo trascorso tra la 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 utilizzata con un indice numerico, ad esempio {ITEM.VALUE.AGE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{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 (in Monitoring -> Problems list), 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 utilizzata con un indice numerico, ad esempio {ITEM.VALUE.DATE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.VALUE.TIME}

L'ora in cui è stato raccolto il valore dell'item.
 Visualizzata 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 utilizzata con un indice numerico, ad esempio {ITEM.VALUE.TIME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.VALUE.TIMESTAMP}

Il timestamp UNIX relativo al momento in cui il valore dell'item è stato raccolto.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.VALUE.TIMESTAMP<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{ITEM.VALUETYPE}

Il tipo di valore dell'N-esimo item nell'espressione del trigger che ha causato una notifica.
 Valori possibili: *0* - Numerico in virgola mobile, *1* - Carattere, *2* - Log, *3* - Numerico senza segno, *4* - Testo.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {ITEM.VALUETYPE<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

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}

L'ultimo stato della regola di low-level discovery.
 Valori possibili: *Non supportato*, *Normale*.

 Questa macro può essere utilizzata con un indice numerico, ad esempio {LLDRULE.STATE<1-9>}, per puntare al primo, secondo, terzo, ecc. item in un'espressione del trigger. Vedere macro indicizzate.

{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 valori:

- Il proxy dell'N-esimo item nell'espressione del trigger (nelle notifiche basate su trigger). Qui è possibile usare macro indicizzate;
- Il proxy che ha eseguito il discovery (nelle notifiche di discovery). Usare qui {PROXY.DESCRIPTION}, senza indicizzazione;
- Il proxy a cui si è registrato un agent attivo (nelle notifiche di autoregistrazione). Usare qui {PROXY.DESCRIPTION}, senza indicizzazione.

Questa macro può essere usata con un indice numerico, ad esempio {PROXY.DESCRIPTION<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione trigger. Vedere macro indicizzate.

{PROXY.NAME}

Il nome del proxy.
 Si risolve in uno dei seguenti valori:

- Proxy dell'N-esimo item nell'espressione del trigger (nelle notifiche basate su trigger). Qui è possibile usare macro indicizzate;
- Proxy che ha eseguito il discovery (nelle notifiche di discovery). Usare qui {PROXY.NAME}, senza indicizzazione;
- Proxy a cui si è registrato un agent attivo (nelle notifiche di autoregistrazione). Usare qui {PROXY.NAME}, senza indicizzazione.

Questa macro può essere usata con un indice numerico, ad esempio {PROXY.NAME<1-9>}, per fare riferimento al primo, secondo, terzo, ecc. item in un'espressione di trigger. Vedere 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 evento del servizio separati da virgole.
 I tag evento del servizio possono essere definiti nella sezione dei tag della configurazione del servizio. Viene espanso in una stringa vuota se non esistono tag.

{SERVICE.TAGSJSON}

Un array JSON contenente oggetti tag dell'evento del servizio.
 I tag dell'evento del servizio possono essere definiti nella sezione tag della configurazione del servizio. Viene espanso in un array vuoto se non esistono tag.

{SERVICE.TAGS.<tag name>}

Il valore del tag dell'evento del servizio referenziato dal nome del tag.
 I tag degli eventi del servizio possono essere definiti nella sezione dei tag della configurazione del servizio.
 Un nome di tag contenente caratteri non alfanumerici (inclusi caratteri UTF multibyte 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 precedute da un carattere di 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 di 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 come scritte nell'espressione. Può essere utilizzata per il debug delle espressioni di 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 sugli item vengono valutate e sostituite dai risultati al momento della generazione dell'evento, mentre tutte le altre funzioni vengono visualizzate come scritte nell'espressione. Può essere utilizzata 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'N-esimo item, funzione o host nel campo *Expression* di un trigger:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** vengono risolte negli indirizzi IP del primo, secondo e terzo host nell'espressione del trigger (se presenti).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** vengono risolte nei valori del primo, secondo e terzo item nell'espressione del trigger al momento dell'evento (se presenti).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** vengono risolte 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 a *Recovery expression*. Ad esempio, in un evento di ripristino, **{ITEM.VALUE2}** verrà risolta nel valore del secondo item dell'espressione del problema al momento del ripristino.

Anche la macro **{HOST.HOST<1-9>}** è supportata all'interno della macro di espressione `{?func(/host/key,param)}` nei **nomi dei grafici**. Ad esempio, `{?func(/{HOST.HOST2}/key,param)}` in un nome di grafico verrà risolta nell'host del secondo item nel grafico.

Warning:
Le macro indicizzate non verranno risolte in nessun altro contesto oltre ai casi qui menzionati. Per altri contesti, utilizzare macro senza indice (**{HOST.HOST}, {HOST.IP}**, ecc.).

2 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	

Posizione		Macro multiple/miste con testo[1]
	Community SNMP	yes
SNMP v3		
	Nome del contesto	yes
	Nome di sicurezza	yes
	Passphrase di autenticazione	yes
	Passphrase di privacy	yes
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	Macro multiple/miste con testo[1]
Nome item	sì
Parametri della chiave item	sì
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

Posizione	Macro multiple/miste con testo[1]
Item calculato/aggregato	
Formula (costanti di espressione e parametri di funzione; parametri della chiave item; (*solo item aggregato*) condizioni di filtro (nome del gruppo host e nome del tag))	sì
Monitor 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
Proxy HTTP	sì
Nome utente per autenticazione HTTP	sì
Password per 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
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 trapper	
Host consentiti	sì
Tag[2]	
Nomi dei tag	sì
Valori dei tag	sì

Posizione	Macro multiple/miste con testo[1]
Passi di pre-pro-cess-ing	
Parametri (inclusi script personalizzati)	sì
Parametri personalizzati di gestione degli errori (campi *Imposta valore a* e *Imposta errore a*)	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 della chiave pubblica	sì
File della 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
Override	
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:

Posizione	Macro multiple/miste con testo[1]
Indirizzo per agent attivi > Porta (se il proxy appartiene a un gruppo)	no
Indirizzo e porta dell'interfaccia (per proxy passivo)	no
Timeout per i tipi di item	no

Gruppi di proxy

In una configurazione di proxy group, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Macro multiple/miste 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

Posizione	Macro multiple/miste con testo[1]
Dati operativi	sì
Espressione (solo nelle costanti e nei parametri di funzione; le macro segrete non sono supportate)	sì
Tag per la corrispondenza	sì
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

Posizione		Macro multiple/miste con testo[1]
	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

Altre posizioni

Oltre alle posizioni elencate qui, le macro utente possono essere utilizzate nei seguenti campi:

Posizione	Macro multiple/miste 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 del tipo di dati *Text* nel widget dashboard *Top hosts*	yes
Parametro *Description* nei widget dashboard *Item value* e *Gauge*	yes
Parametro *Text* dell'etichetta primaria/secondaria nel widget dashboard *Honeycomb*	yes
Parametro *URL*[3] nel widget dashboard *URL*	yes
Utenti → *Utenti* → *Supporti*	
Quando attivo	yes
Amministrazione → *Generale* → *GUI*	
Orario di lavoro	no

Posizione	Macro multiple/miste con testo[1]
Amministrazione → Generale → Timeout	
Timeout per i tipi di item	no
Amministrazione → Generale → Connettori	
URL	yes
Nome utente	yes
Password	yes
Token Bearer	yes
Timeout	no
Proxy HTTP	yes
File del certificato SSL	yes
File della chiave SSL	yes
Password della chiave SSL	yes
Avvisi → Tipi di supporto → Modelli di messaggio	
Oggetto	yes
Messaggio	yes
Avvisi → Tipi di supporto → Script	
Parametri dello script	yes
Avvisi → Tipi di supporto → Tipo di supporto	

Posizione	Macro multiple/miste con testo[1]
Campi *Username* e *Password* per il tipo di supporto *Email* (quando *Authentication* è impostato su "Username and password"; si consiglia l'uso di macro segrete)	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 utilizzate nei nomi e nei valori dei tag vengono risolte solo durante il processo di generazione degli eventi.

[3] Gli URL che contengono una macro segreta non funzioneranno, poiché la macro al loro interno verrà risolta come "******".

7 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 di tempo

Il frontend di Zabbix supporta i seguenti suffissi di tempo nella configurazione delle entità:

- **s** - secondi *(quando utilizzato, funziona come il valore grezzo)*
- **m** - minuti
- **h** - ore
- **d** - giorni
- **w** - settimane
- **M** - mesi *(funzioni di trend solo)*
- **y** - anni *(funzioni di trend solo)*

I suffissi di tempo sono supportati solo con numeri interi. Ad esempio, 1h è supportato, ma 1,5h o 1.5h non sono supportati; utilizzare invece 90m.

Suffissi delle dimensioni della memoria

Zabbix supporta i seguenti suffissi per le dimensioni della memoria:

- **K** - kilobyte
- **M** - megabyte
- **G** - gigabyte
- **T** - terabyte

Suffissi dei valori degli item

I suffissi possono essere utilizzati anche per visualizzare i valori numerici degli item in un formato leggibile.

Per abilitare questa funzionalità, utilizzare uno dei seguenti suffissi nel campo *Units* durante la configurazione di un item:

- **B** - byte
- **Bps** - byte al secondo
- **s** - secondi, visualizzati utilizzando fino a tre delle più grandi unità di tempo non nulle
- **uptime** - tempo trascorso nel formato hh:mm:ss oppure N giorni, hh:mm:ss
- **unixtime** - timestamp Unix, formattato come yyyy.mm.dd hh:mm:ss

Si applicano inoltre le seguenti regole aggiuntive al modo in cui questi suffissi vengono interpretati e visualizzati:

- Per B e Bps, Zabbix utilizza la conversione in base 2 (1K = 1024B), seguendo lo standard JEDEC.
- Per le altre unità (come Hz, W, ecc.), Zabbix utilizza 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 delle più grandi unità di tempo non nulle (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 utilizzate le *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, utilizzare il prefisso `!` (ad esempio, !B oppure !s).

I seguenti esempi 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 hardcoded delle unità composta da `ms`, `rpm`, `RPM`, `%`. Questa stoplist è stata deprecata, pertanto il modo corretto per impedire la conversione di tali unità è `!ms`, `!rpm`, `!RPM`, `!%`.

8 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`

9 Esecuzione dei comandi

Zabbix utilizza funzionalità comuni per i controlli esterni, i parametri utente, gli item system.run, gli script di avviso 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 secondo le seguenti condizioni:

- Solo per script di avviso 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 dell'errore standard e dell'output standard per le esecuzioni non riuscite viene raccolto ed è disponibile nel frontend (dove viene visualizzato il risultato dell'esecuzione).
- È possibile creare una voce di log aggiuntiva per i comandi remoti eseguiti su Zabbix agent/proxy abilitando il parametro LogRemoteCommands su agent/proxy.

Possibili messaggi nel frontend e voci di log per comandi/script non riusciti:

- Contenuto dell'errore standard e dell'output standard per le esecuzioni non riuscite (se presente).
- "Il processo è terminato con il codice: N." (per output vuoto e codice di uscita diverso da 0).
- "Il processo è stato terminato dal segnale: N." (per processi terminati da un segnale, solo su Linux).
- "Il processo è terminato in modo imprevisto." (per processi terminati per motivi sconosciuti).

Vedi anche

- Controlli esterni
- Parametri utente
- item system.run
- Script di avviso personalizzati
- Comandi remoti
- Script globali

10 Compatibilità della versione

Agent supportati

Per essere compatibile con Zabbix 8.0, Zabbix agent non deve essere precedente alla versione 1.4 e non deve essere successivo alla 8.0.

Potrebbe essere necessario rivedere la configurazione degli agent più vecchi, 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
Corrente	Sì	Sì	Sì
Obsoleta	Sì	No	Comandi remoti (ad es. script shell); Controlli immediati del valore degli item (cioè *Esegui ora*); Nota: i test di preprocessing con un valore reale non sono supportati.
Non supportata	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.

12 Libreria a collegamento dinamico di Zabbix sender per Windows

Panoramica

In un ambiente Windows, le applicazioni possono inviare dati al server/proxy Zabbix utilizzando la libreria a collegamento dinamico di Zabbix sender (zabbix_sender.dll) invece di dover avviare un processo esterno (zabbix_sender.exe).

zabbix_sender.h e zabbix_sender.lib sono necessari per compilare le applicazioni utente con zabbix_sender.dll.

Come ottenerla

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`. Estrarre i file dove necessario.

2. Compilare zabbix_sender.dll dal codice sorgente (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 header zabbix_sender.h ed eseguire il collegamento con la libreria zabbix_sender.lib.

Vedere anche

- esempio di una semplice utility Zabbix sender implementata con la libreria a collegamento dinamico di Zabbix sender per illustrare l'utilizzo della libreria;
- file zabbix_sender.h per le funzioni di interfaccia della libreria a collegamento dinamico di Zabbix sender. Questo file contiene documentazione che spiega lo scopo di ciascuna funzione di interfaccia, i relativi argomenti e il valore di ritorno.

13 Aggiornamento del monitoraggio dei servizi

Overview In Zabbix 6.0, service monitoring functionality has been reworked significantly (see What's new in Zabbix 6.0.0 for the list of changes).

This page describes how services and SLAs defined in earlier Zabbix versions are changed during an upgrade to Zabbix 6.0 or newer.

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 "Servizio figlio" era stato precedentemente collegato a "Servizio padre 1" tramite una dipendenza hard e inoltre a "Servizio padre 2" tramite una dipendenza soft, dopo un aggiornamento il "Servizio figlio" avrà due servizi padre: "Servizio padre 1" e "Servizio padre 2".

La mappatura basata su trigger tra problemi e servizi 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/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 il problem tag `ServiceLink` con lo stesso valore.

Regole di calcolo dello stato

L'algoritmo di calcolo dello stato verrà aggiornato utilizzando le seguenti regole:

- Non calcolare → Imposta lo stato su OK
- Problema, se almeno un figlio ha un problema → Il più critico tra i servizi figli
- Problema, se tutti i figli hanno problemi → Il più critico se tutti i figli hanno problemi

SLA In precedenza, gli obiettivi SLA dovevano essere definiti separatamente per ciascun servizio. A partire da Zabbix 6.0, lo SLA è diventato un'entità separata, che contiene informazioni sulla pianificazione del servizio, sull'obiettivo di livello di servizio previsto (SLO) e sui periodi di inattività da escludere dal calcolo. Una volta configurato, uno SLA può essere assegnato a più servizi tramite i tag di servizio.

Durante un aggiornamento:

- Gli SLA identici definiti per ciascun servizio verranno raggruppati e verrà creato uno SLA per ogni gruppo.
- A ciascun servizio interessato verrà assegnato un tag speciale `SLA:<ID>` e lo stesso tag verrà specificato nel parametro *Tag di servizio* dello 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.

14 Altri problemi

Accesso e systemd

Raccomandiamo di creare un utente *zabbix* come utente di sistema, cioè senza possibilità di effettuare l'accesso. Alcuni utenti ignorano questa raccomandazione e usano lo stesso account per accedere (ad es. tramite SSH) all'host che esegue Zabbix. Questo

potrebbe causare l'arresto anomalo del demone Zabbix al momento della disconnessione. In questo caso nel log del server Zabbix verrà visualizzato qualcosa di simile:

```
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 di Zabbix agent:

```
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 ci si disconnette dal sistema, i semafori creati in precedenza da Zabbix vengono rimossi, causando l'arresto anomalo.

Una citazione dalla documentazione di systemd:

```
RemoveIPC=

Controlla se gli oggetti IPC System V e POSIX appartenenti all'utente debbano essere rimossi quando
l'utente termina completamente la sessione. Accetta un argomento booleano. Se abilitato, l'utente non può
dopo la terminazione dell'ultima sessione dell'utente. Questo include semafori System V, memoria condivisa
e code di messaggi, nonché memoria condivisa POSIX e code di messaggi. Si noti che gli oggetti IPC dell'ut
root e di altri utenti di sistema sono esclusi dall'effetto di questa impostazione. Il valore predefinito
```

Esistono 2 soluzioni a questo problema:

1. (consigliata) Smettere di usare l'account *zabbix* per qualsiasi altra cosa diversa dai processi di Zabbix e creare un account dedicato per altre attività.
2. (sconsigliata) Impostare `RemoveIPC=no` in `/etc/systemd/logind.conf` e riavviare il sistema. Si noti che `RemoveIPC` è un parametro valido per l'intero sistema; modificarlo influenzerà l'intero sistema.

Utilizzo del frontend di Zabbix dietro proxy

Se il frontend di Zabbix viene eseguito dietro un server proxy, il percorso del cookie nel file di configurazione del proxy deve essere riscritto in modo che corrisponda al percorso del reverse proxy. Vedere gli esempi riportati di seguito. Se il percorso del cookie non viene riscritto, gli utenti potrebbero riscontrare problemi di autorizzazione durante il tentativo di accesso al frontend di Zabbix.

Esempio di configurazione per nginx

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Configurazione di esempio per Apache

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

16 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.

21 Guide rapide di riferimento

Panoramica

Questa sezione della documentazione contiene procedure rapide per configurare Zabbix in base ad alcuni obiettivi di monitoraggio comunemente richiesti.

È pensata per i nuovi utenti di Zabbix e può essere utilizzata come guida attraverso altre sezioni della documentazione che contengono le informazioni necessarie per portare a termine l'attività.

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 utilizzando controlli attivi
- Monitorare i siti web con Browser item
- Monitorare i certificati dei siti web con Zabbix agent 2 (passivo)
- Monitorare uno switch o un router di rete con Zabbix
- Monitorare il registro eventi di Windows utilizzando controlli attivi

1 Monitora Linux con l'agente Zabbix

Introduzione Questa pagina illustra i passaggi necessari per iniziare 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 rivolta 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 cerchi 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.

Controlla la versione del tuo Zabbix server:

```
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 Zabbix server 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 restituisce il risultato al server. I controlli attivi richiedono un'elaborazione più complessa. L'agent deve prima recuperare dal server o dai server un elenco di item da elaborare in modo indipendente e quindi inviare in blocco i dati di ritorno. Per maggiori informazioni, vedere Controlli agent passivi e attivi.

I template di monitoraggio forniti da Zabbix offrono solitamente 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.

L'ulteriore configurazione 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 l'agent è installato.

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

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

```
Server=192.0.2.0:10051
```

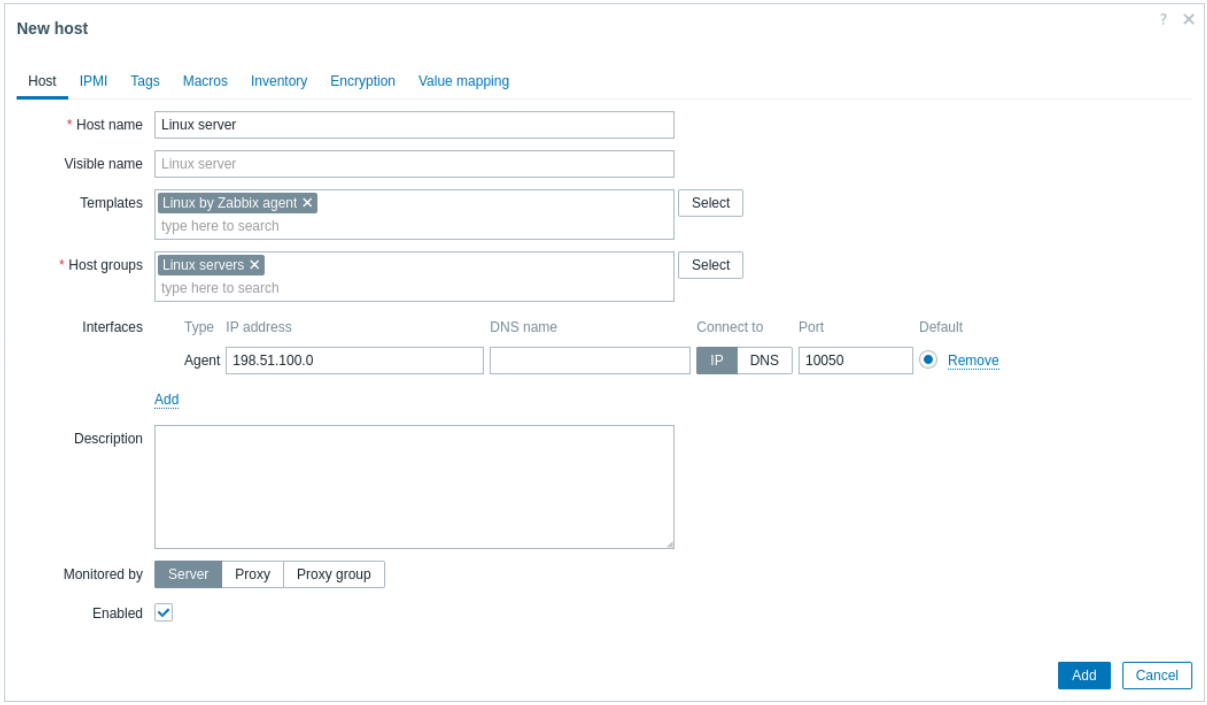
Zabbix agent utilizzerà questo indirizzo per accettare connessioni in ingresso solo dai server o proxy Zabbix specificati, al fine di recuperare i dati.

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 di host (ad esempio, "Linux servers").
   - Nel parametro *Interfaces*, aggiungi un'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 l'agent è installato.

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

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

```
ServerActive=192.0.2.0:10051
```

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

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

```
Hostname=Linux server
```

Il valore di Hostname deve corrispondere perché, per i controlli attivi, Zabbix agent lo utilizza per recuperare dal server la configurazione corretta del 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 quel host. Se questi valori differiscono, l'agent non riceverà la configurazione appropriata, con conseguente assenza di metriche o problemi di monitoraggio.

4. Riavviare Zabbix agent.

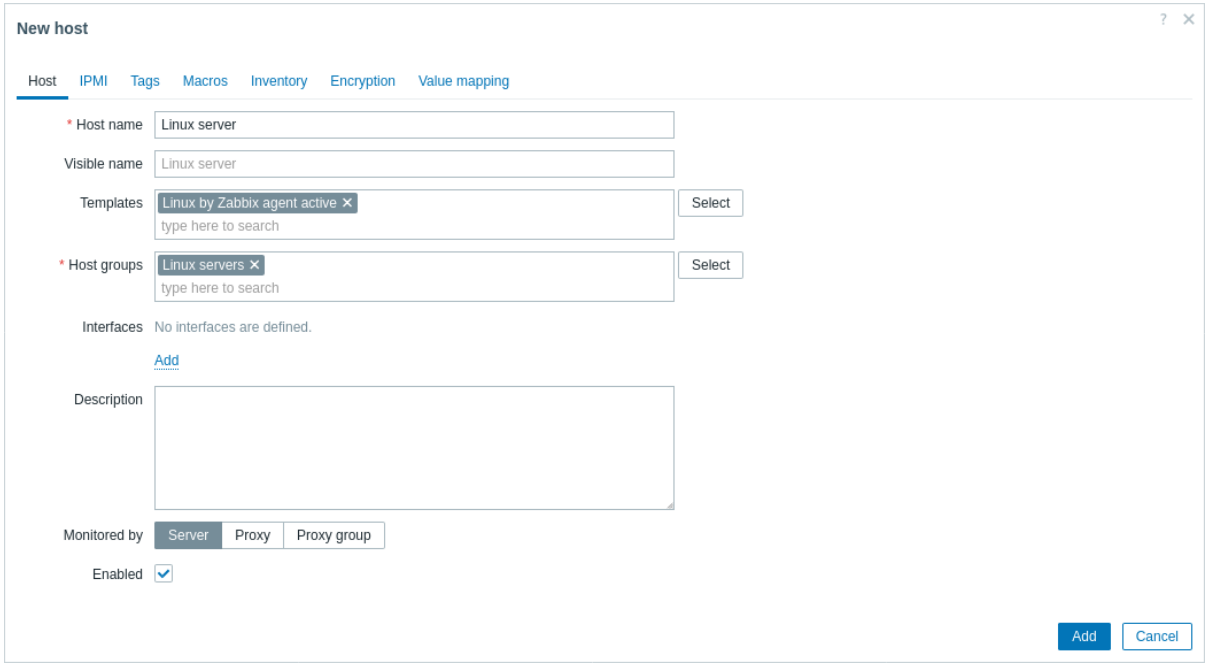
```
systemctl restart zabbix-agent
```

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") 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 di 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 restare in ascolto di una connessione dal server.



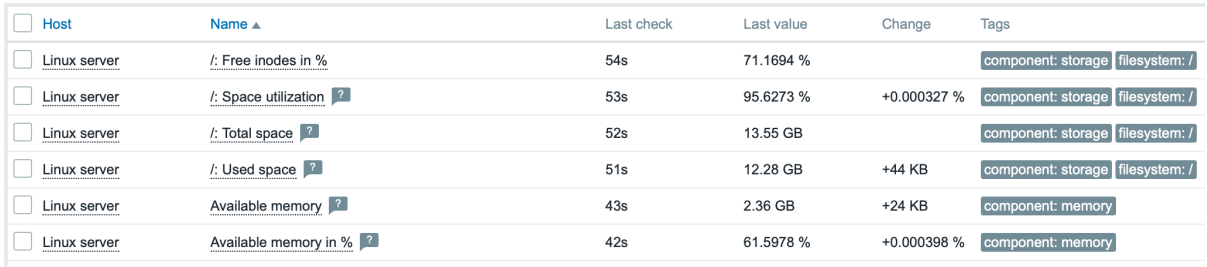
3. Fai clic su *Add* per aggiungere l'host. Questo host rappresenterà la macchina Linux monitorata.

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.

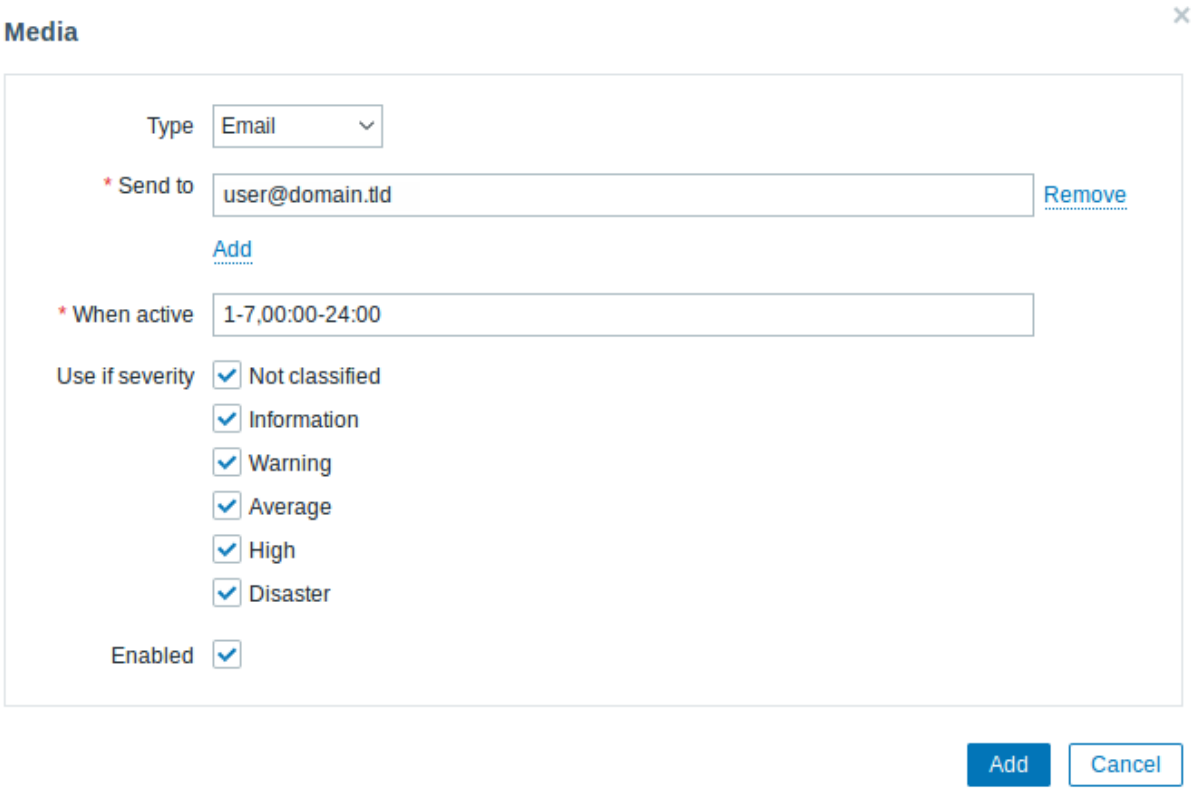


Questa azione aprirà un elenco di tutte le metriche più recenti raccolte dall'host del server Linux.



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

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



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

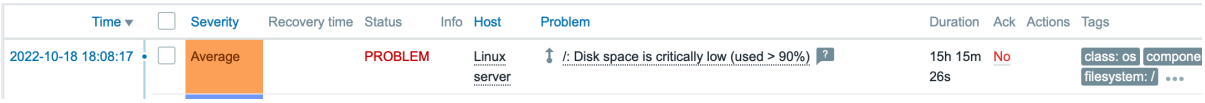
La prossima volta che Zabbix rileverà un problema, dovresti ricevere un avviso via email.

Verifica la tua configurazione Su Linux, puoi simulare un carico elevato della CPU e, di conseguenza, ricevere un avviso di problema eseguendo:

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

Potrebbe essere necessario eseguire diversi processi md5sum affinché il carico della CPU superi la soglia.

Quando Zabbix rileva il problema, questo apparirà nella sezione Monitoraggio->Problemi.



Se gli avvisi 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 utilizzando l'agent Zabbix su Windows.
- Escalation dei problemi - come creare scenari di avviso in più fasi (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).
- Installazione dai pacchetti - come installare i componenti di Zabbix utilizzando i pacchetti RPM e DEB ufficiali per varie distribuzioni Linux, garantendo l'accesso alle funzionalità più recenti e alle correzioni di bug.

2 Monitorare Windows con Zabbix agent

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

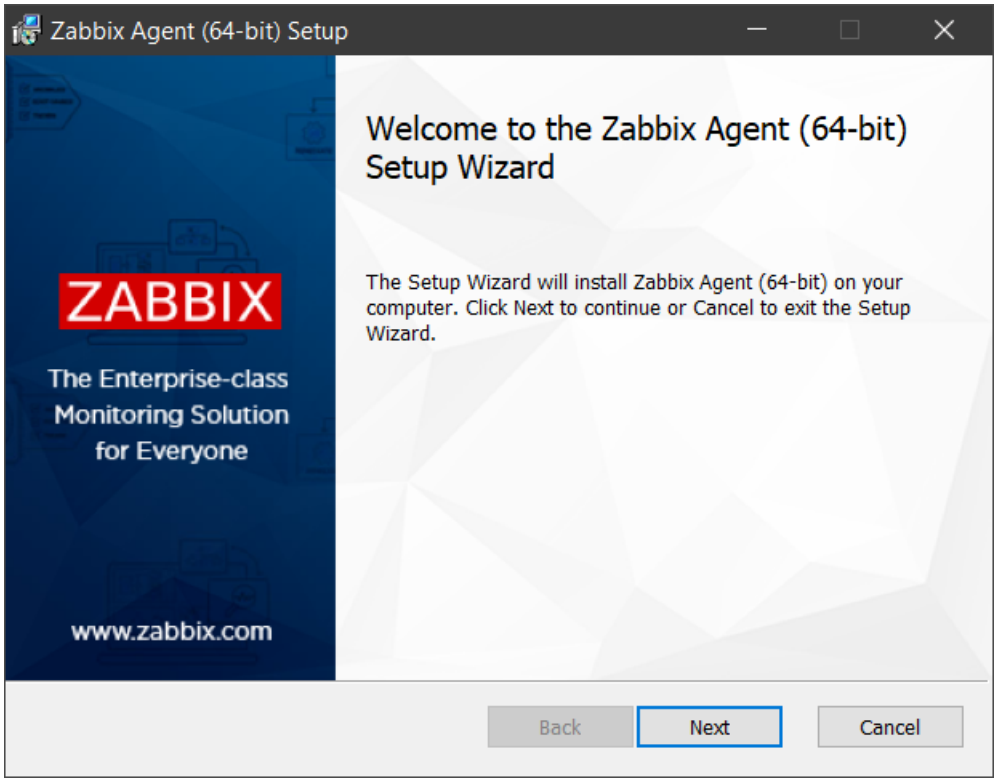
A chi è rivolta questa guida

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 Windows.
Se stai cercando opzioni di personalizzazione più approfondite o hai bisogno di una
configurazione più avanzata, consulta la sezione Configuration del manuale di Zabbix.

Prerequisiti

Prima di procedere con questa guida all'installazione, devi scaricare e installare Zabbix server e Zabbix frontend secondo le istruzioni per il tuo 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 o dai server un elenco di item da elaborare in modo indipendente e quindi inviare in blocco i dati raccolti. Per maggiori informazioni, vedere Controlli agent passivi e attivi.

I template di monitoraggio forniti da Zabbix offrono solitamente 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.

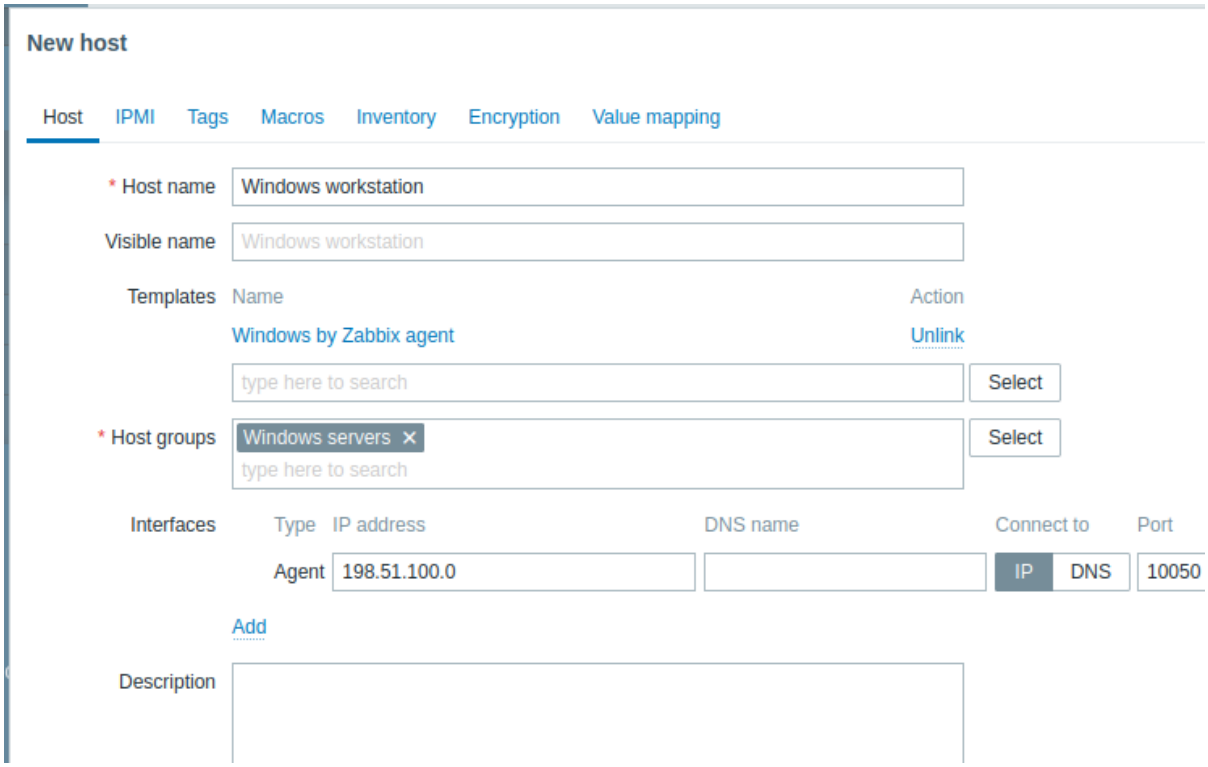
L'ulteriore configurazione di Zabbix dipende dal fatto che si selezioni un template per i controlli Zabbix agent attivi o passivi.

Controlli passivi Zabbix frontend

1. Log into Zabbix frontend.
2. Create a host in Zabbix web interface.

This host will represent your Windows machine.

3. In the *Interfaces* parameter, add *Agent* interface and specify the IP address or DNS name of the Windows machine where the agent is installed.
4. In the *Templates* parameter, type or select *Windows by Zabbix agent*.



Zabbix agent

For passive checks Zabbix agent needs to know the IP address or DNS name of Zabbix server. If you have provided correct information during the agent installation, the configuration file is already updated. Otherwise, you need to manually specify it. Go to the `C:\Program files\Zabbix Agent` folder, open the file *zabbix_agentd.conf* and add the IP/DNS of your Zabbix server to the *Server* parameter.

Example:

```
Server=192.0.2.22
```

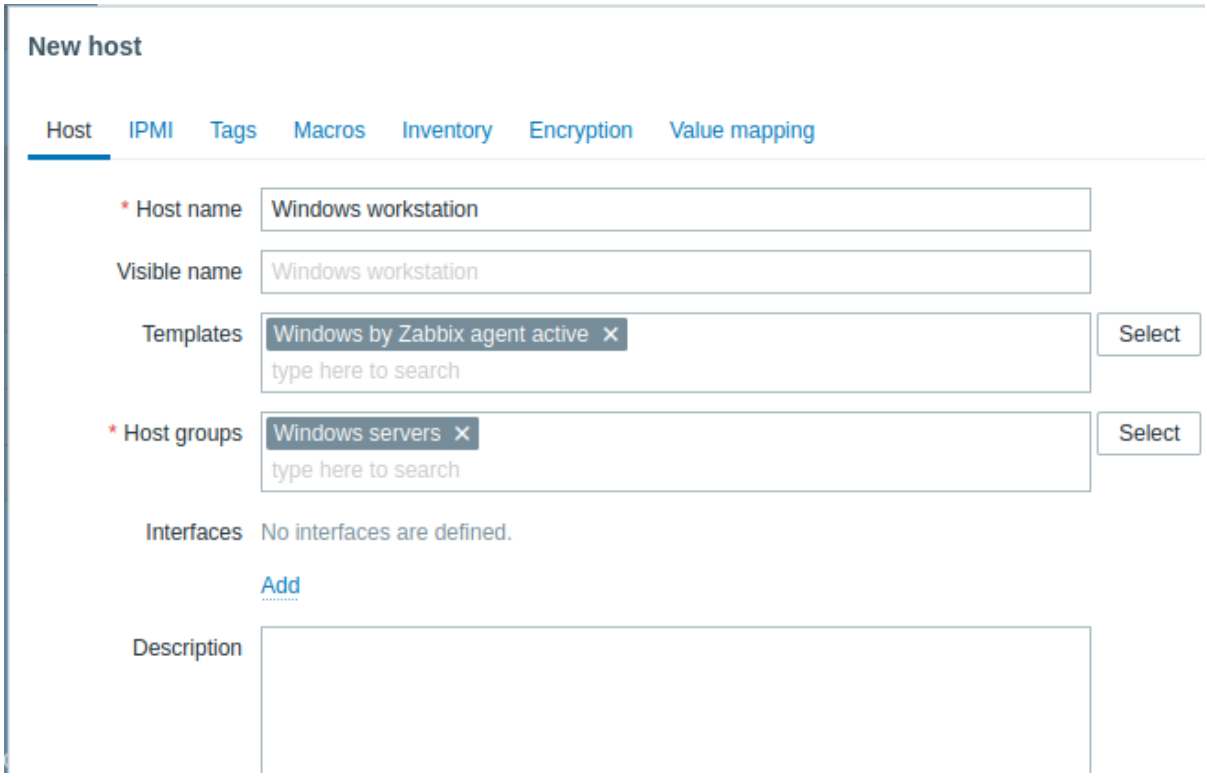
Controlli attivi Frontend di Zabbix

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

In the `C:\Program files\Zabbix Agent` folder open the file *zabbix_agentd.conf* and add:

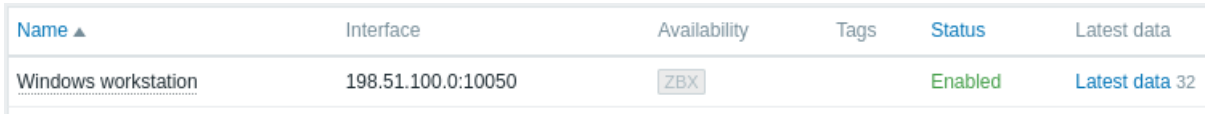
- The name of the host you created in Zabbix web interface to the *Hostname* parameter.
- The IP address or DNS name of your Zabbix server machine to the *ServerActive* parameter (might be prefilled if you have provided it during Zabbix agent setup).

Example:

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

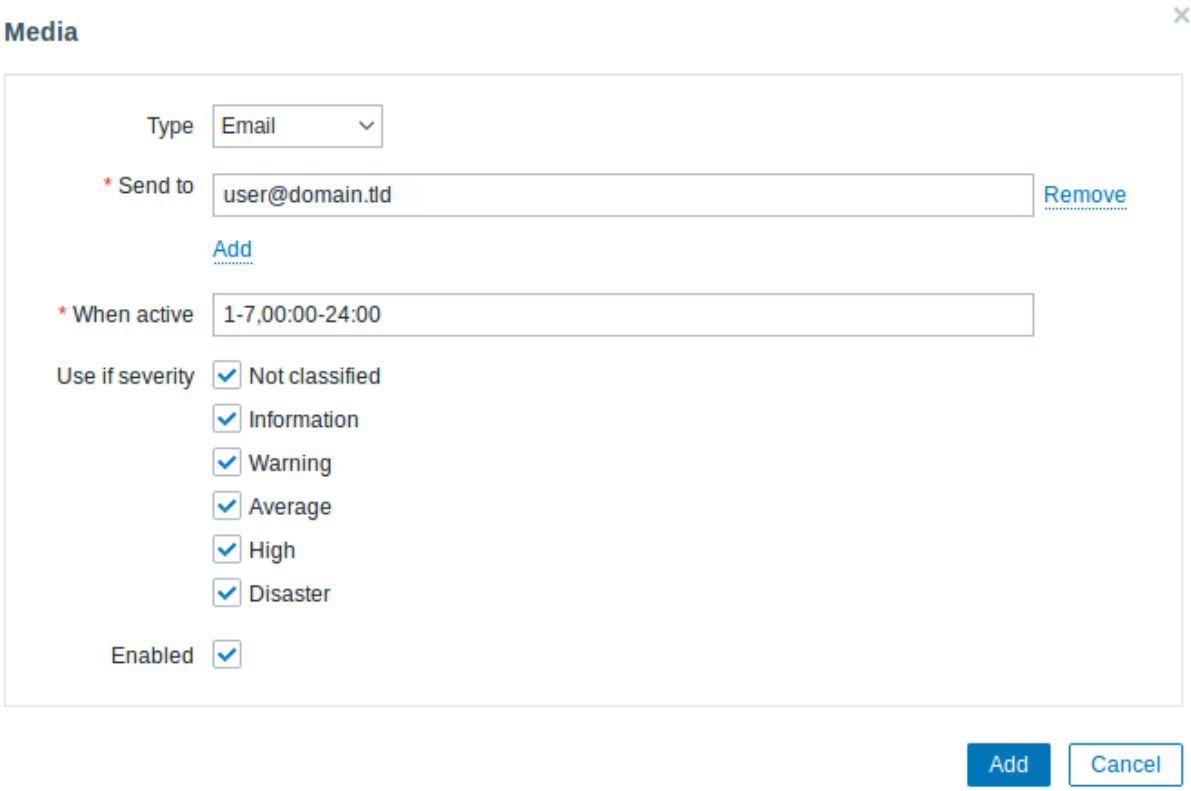
Visualizzare le metriche raccolte Congratulazioni! A questo punto, Zabbix sta già monitorando la tua macchina Windows.

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 notificare un problema nella tua infrastruttura utilizzando vari metodi. Questa guida fornisce i passaggi di configurazione per l'invio di avvisi email.

1. Vai in *Impostazioni utente -> Profilo*, passa alla scheda *Supporti* e aggiungi il tuo indirizzo 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:**
> In Windows, puoi utilizzare l'utility CpuStres per simulare un carico elevato della CPU e, di conseguenza, ricevere un avviso di 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 utilizzando l'agent Zabbix in Windows.
- Escalation dei problemi - come creare scenari di avviso in più fasi (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).

3 Monitora 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 rivolta 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 Apache. 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 all'installazione, è necessario scaricare e installare Zabbix server e Zabbix frontend secondo le istruzioni per il proprio sistema operativo.

Preparare Apache 1. Verificare quale versione di Apache si sta utilizzando:

Su un sistema basato su RHEL, eseguire:

```
httpd -v
```

Su Debian/Ubuntu, eseguire:

```
apache2 -v
```

2. Assicurarsi che il modulo Status sia abilitato nella propria istanza Apache.

Su un sistema basato su RHEL, eseguire:

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

Su Debian/Ubuntu, eseguire:

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

Se `status_module` non compare nell'elenco, abilitare il modulo eseguendo:

Su un sistema basato su RHEL, eseguire:

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

Su Debian/Ubuntu, eseguire:

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

3. Modificare il file di configurazione di Apache per consentire l'accesso ai report di stato dall'indirizzo 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
```

Aggiungere al file le seguenti righe (**sostituire 198.51.100.255** con l'indirizzo IP del proprio 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. Riavviare Apache

Su un sistema basato su RHEL, eseguire:

```
sudo systemctl restart httpd
```

Su Debian/Ubuntu, eseguire:

```
sudo systemctl restart apache2
```

5. Per verificare che tutto sia configurato correttamente, eseguire (**sostituire 198.51.100.255** con l'indirizzo IP del proprio server Zabbix):

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

La risposta dovrebbe contenere le statistiche del server web Apache.

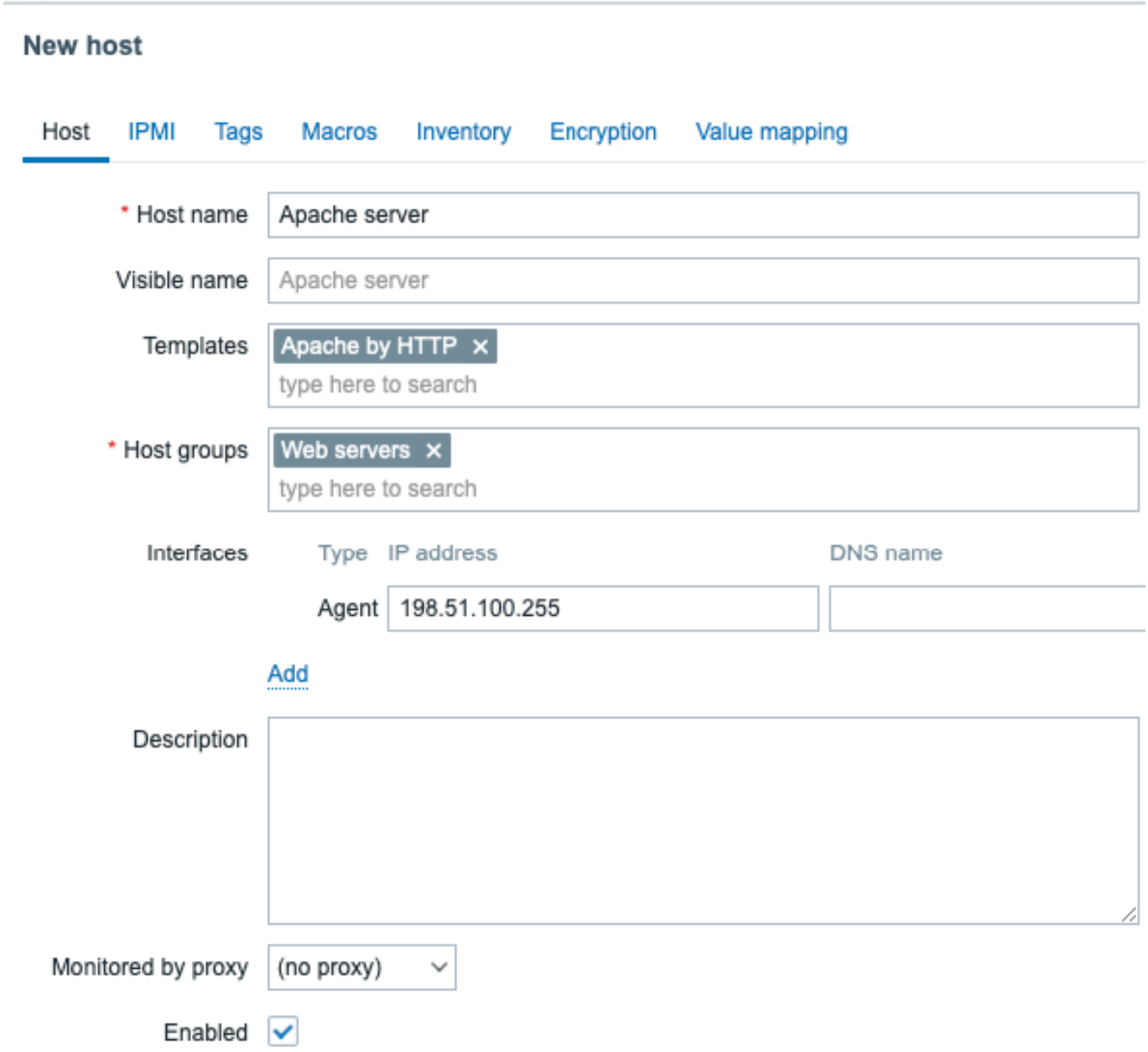
Configurare Zabbix per il monitoraggio 1. Accedi al frontend di Zabbix.

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

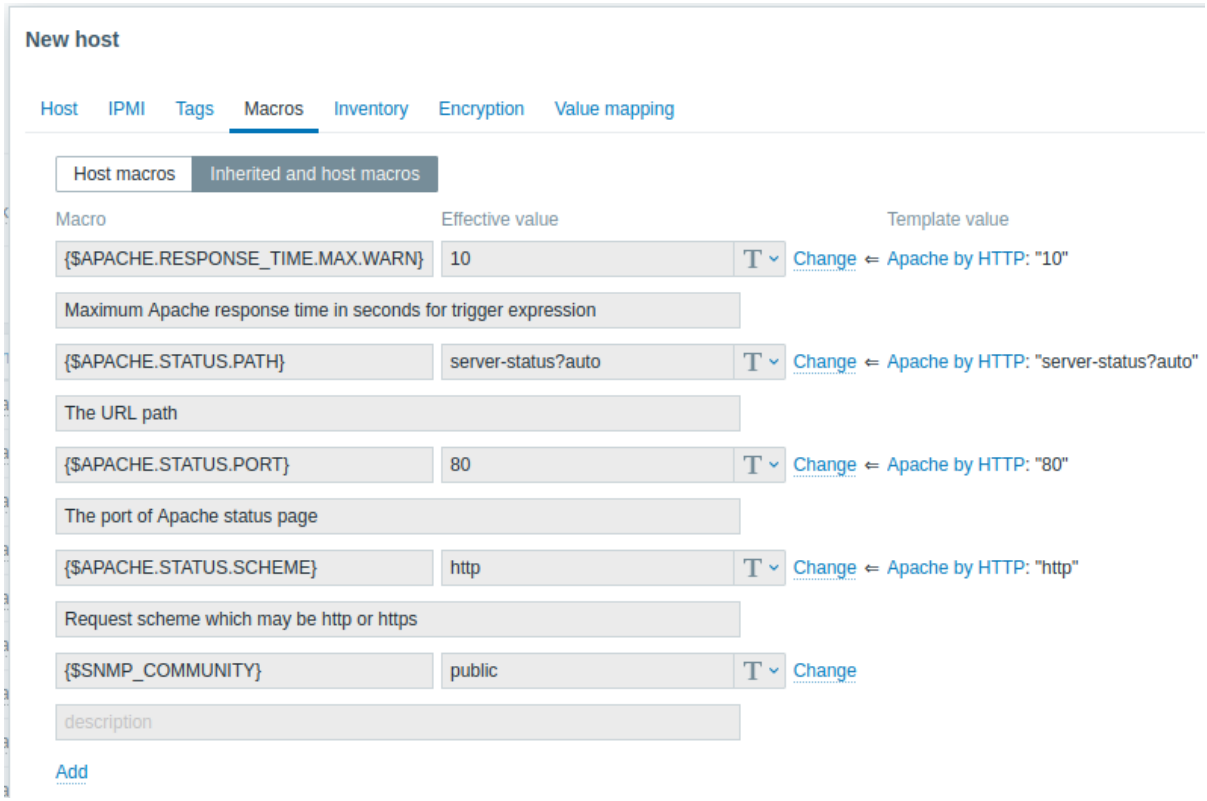
Questo host rappresenterà il tuo server Apache.

3. Nel parametro *Interfaces*, aggiungi un'interfaccia *Agent* e specifica l'indirizzo IP della tua istanza Apache. **Non è necessario installare Zabbix agent sulla macchina**, l'interfaccia verrà utilizzata solo per risolvere la macro {HOST.CONN}. Questa macro viene utilizzata nei 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 server web Apache.

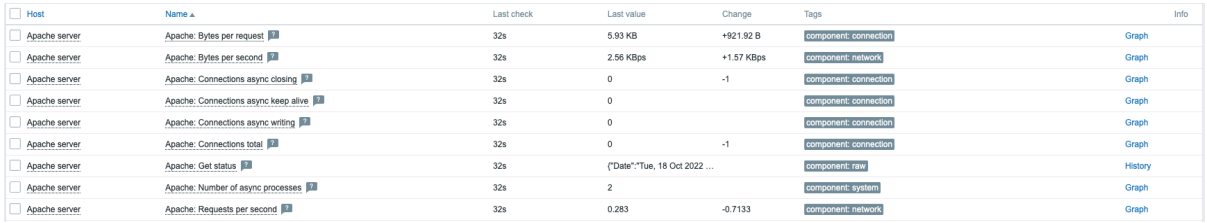
Per visualizzare le metriche raccolte, apri la sezione del menu *Monitoring->Hosts* e fai clic su *Dashboards* accanto all'host.



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

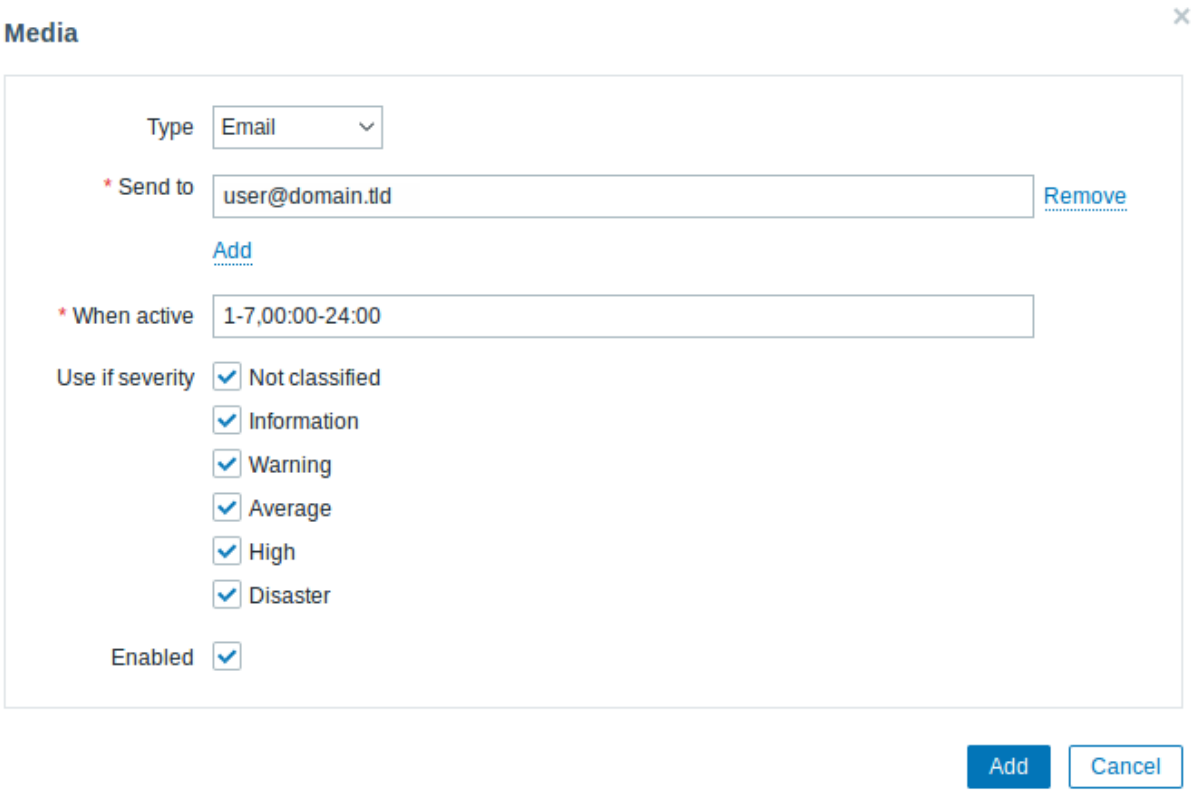


In alternativa, da *Monitoring->Hosts*, puoi fare clic su *Latest data* per visualizzare in un elenco tutte le metriche raccolte più di recente.



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

1. Vai a *User settings -> Profile*, 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.

Verifica la configurazione Per simulare un problema reale e ricevere un avviso di problema di test:

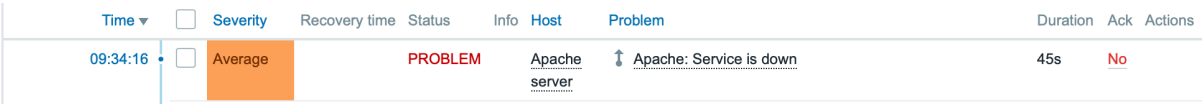
1. Apri la configurazione del host *Apache server* in Zabbix.

2. Passa alla scheda Macro e seleziona *Macro ereditate e del host*.

3. Premi *Modifica* accanto alla macro {$APACHE.STATUS.PORT} e imposta una porta diversa.

4. Premi Aggiorna per salvare la configurazione del host.

5. Entro pochi minuti, Zabbix rileverà il problema *Il servizio Apache non è disponibile*, perché ora non riesce a connettersi all'istanza. Comparirà nella sezione Monitoraggio->Problemi.



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

6. Riporta il valore della macro a quello originale per risolvere il problema e continuare il monitoraggio di Apache.

Vedi anche:

- Protezione avanzata del web server - impostazioni consigliate per una maggiore sicurezza del web server.
- Creazione di un item - come iniziare a monitorare metriche aggiuntive.
- Item HTTP - come monitorare metriche personalizzate utilizzando HTTP agent.
- Escalation dei problemi - come creare scenari di avviso in più fasi (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).

4 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 rappresenta l'approccio **consigliato** grazie alla sua configurazione semplice e uniforme in vari scenari. Tuttavia, questa pagina fornisce anche istruzioni per gli altri approcci, quindi sentiti libero di scegliere quello più adatto alle tue esigenze.

A chi è rivolta 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 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 2 in base alle istruzioni per il proprio sistema operativo.

A seconda dell'ambiente in uso, 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 il semplice monitoraggio (privilegi per eseguire il DROP dei database, eliminare 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>* dell'utente "zbx_monitor" con una password a tua scelta), e GRANT i privilegi necessari all'utente:

```
mysql -u root -p
# Inserisci la password:

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

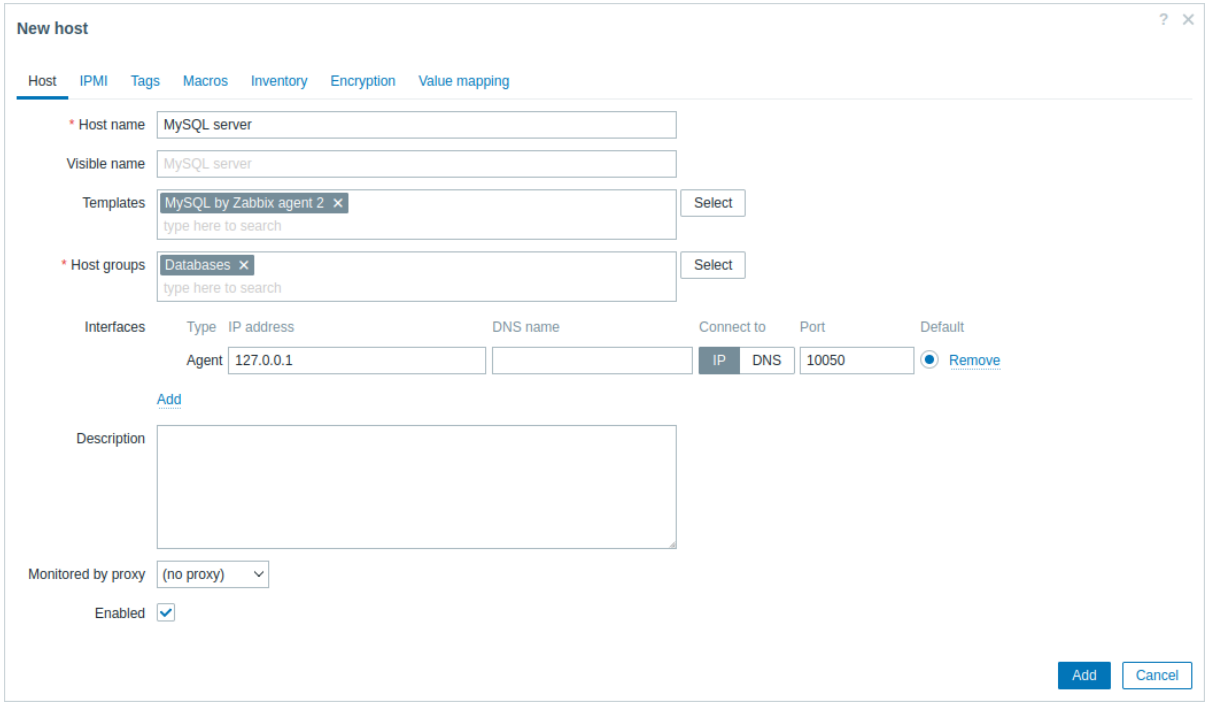
Una volta creato l'utente, puoi passare al passaggio successivo.

Configurare il frontend di Zabbix

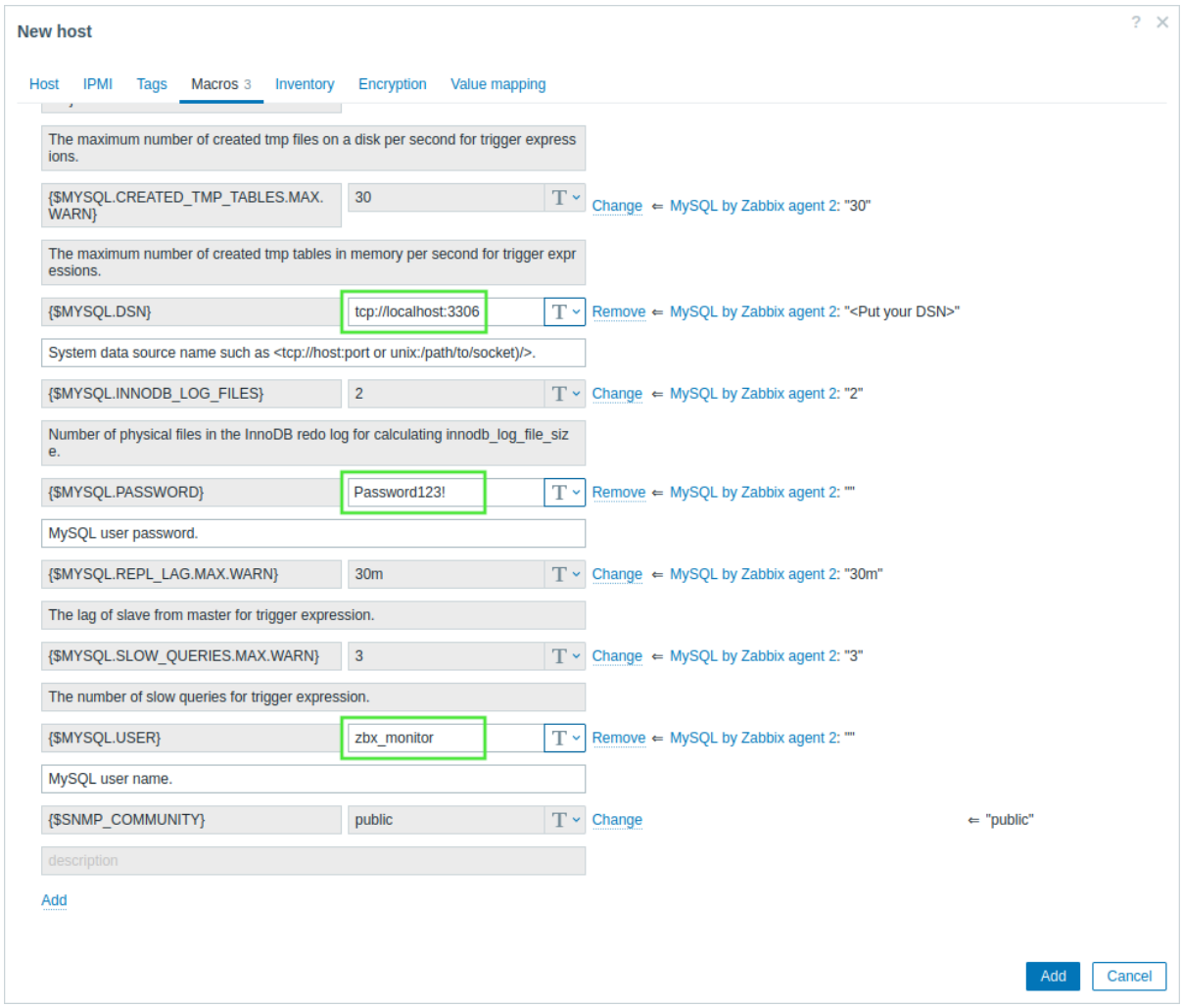
1. Accedere al frontend di Zabbix.

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

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



- 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:
  - {$MYSQL.DSN} - impostare l'origine dati del server MySQL (la stringa di connessione di una sessione denominata dal file di configurazione del plugin MySQL di Zabbix agent 2). Questa guida utilizza l'origine dati predefinita "tcp://localhost:3306" per monitorare un server MySQL installato sulla stessa macchina del server Zabbix e di Zabbix agent 2.
  - {$MYSQL.PASSWORD} - impostare la password dell'utente MySQL creato in precedenza "zbx_monitor".
  - {$MYSQL.USER} - impostare il nome dell'utente MySQL creato in precedenza "zbx_monitor".

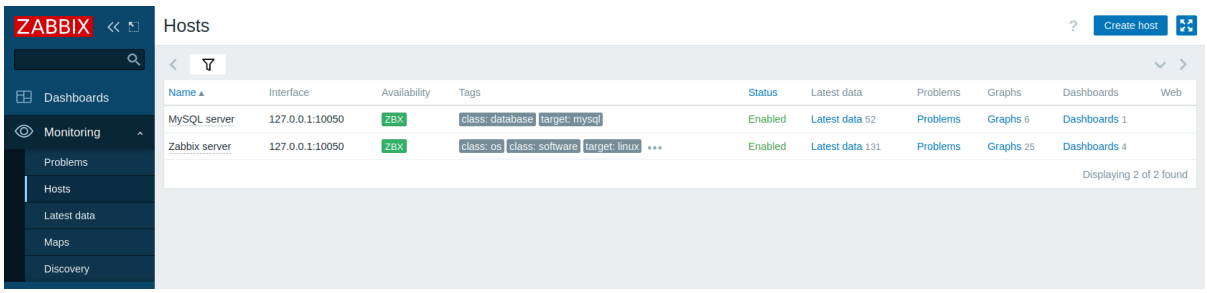


3. Fare clic su *Add* per aggiungere l'host. Questo host rappresenterà il server MySQL.

Visualizzare le metriche raccolte

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

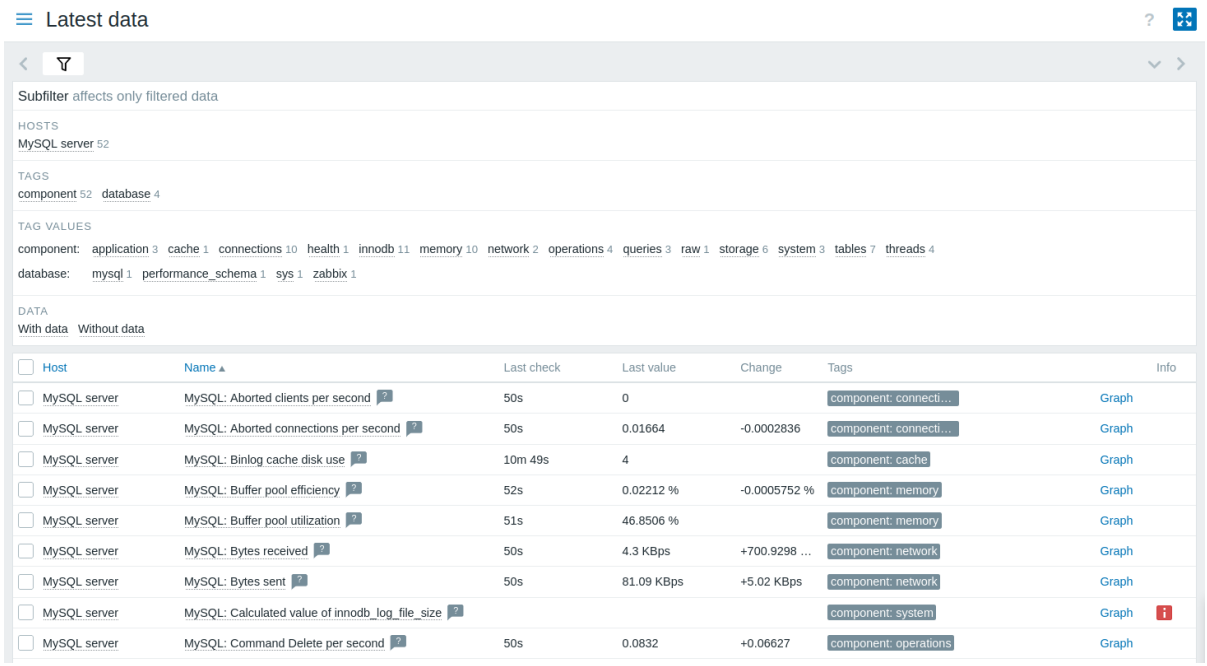
Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoring → Hosts* e fai clic su *Dashboards* accanto all'host.



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



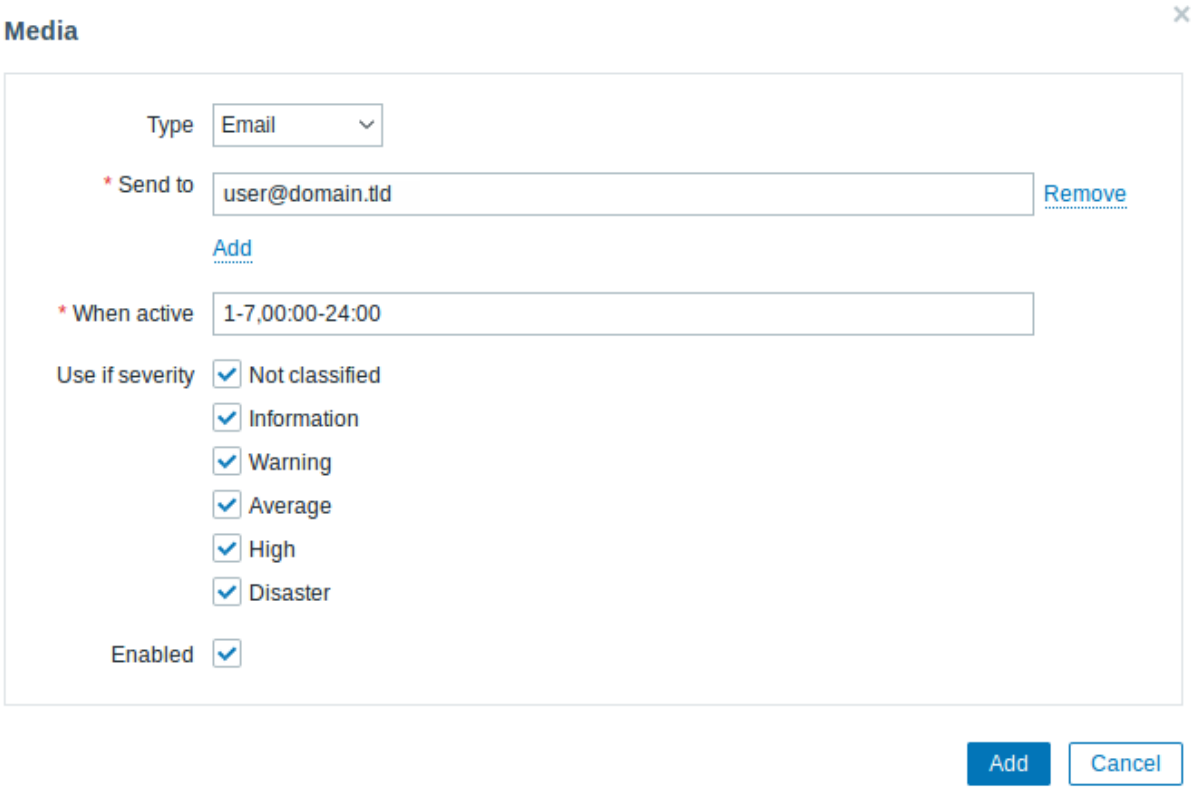
In alternativa, dalla sezione di menu *Monitoring → Hosts*, puoi fare clic su *Latest data* per visualizzare in un elenco tutte le metriche raccolte più recenti. Tieni presente che è previsto che l'item *MySQL: Calculated value of innodb_log_file_size* non abbia dati, poiché il valore verrà calcolato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



Configurare gli avvisi di problema

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

1. Vai a *Impostazioni utente → Profilo*, passa alla scheda *Supporti* 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.

Testa la tua configurazione

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

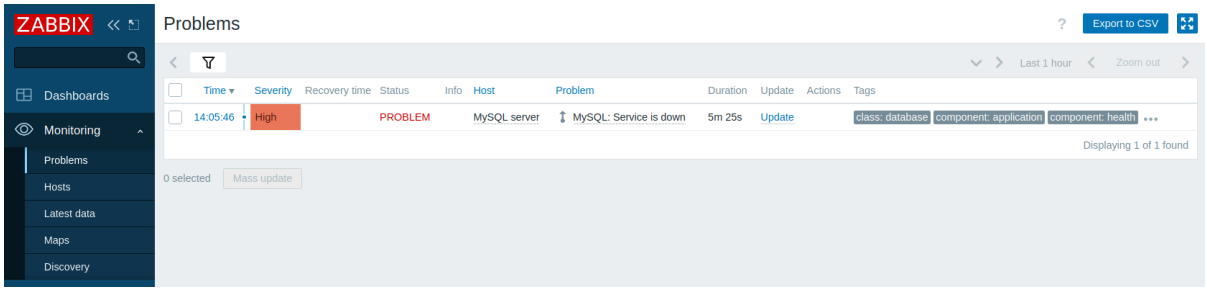
1. Apri la configurazione del host del tuo server MySQL in Zabbix.

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

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

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

5. Tra 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 il monitoraggio del server MySQL.

Altri approcci per monitorare MySQL

Invece di monitorare un server MySQL con Zabbix agent 2, è possibile utilizzare anche Zabbix agent o lo standard Open Database Connectivity (ODBC). Sebbene sia consigliato l'uso di Zabbix agent 2, potrebbero esserci alcune configurazioni che non lo supportano 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 utilizzando le sue 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 dei passaggi di configurazione siano simili al monitoraggio di un server MySQL con Zabbix agent 2, esistono alcune differenze significative: è necessario configurare Zabbix agent o ODBC per poter monitorare un server MySQL. Le istruzioni seguenti illustrano 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 secondo 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 *Creare un utente MySQL*.

Dopo aver creato l'utente MySQL, è necessario configurare Zabbix agent affinché sia in grado di stabilire una connessione con il server MySQL e monitorarlo. Ciò include la configurazione di più parametri utente per eseguire controlli personalizzati dell'agent, nonché la fornitura a Zabbix agent delle credenziali necessarie per connettersi al server MySQL come utente "zbx_monitor" creato in precedenza.

Configurare Zabbix agent

1. Passare alla directory delle configurazioni aggiuntive di Zabbix agent.

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

> **Attention:**
> La directory delle configurazioni aggiuntive di Zabbix agent dovrebbe trovarsi nella stessa directory del file di configurazione di Zabbix agent (*zabbix_agentd.conf*). A seconda del sistema operativo e dell'installazione di Zabbix, questa directory potrebbe trovarsi in un percorso diverso da quello specificato in questa guida. Per i percorsi predefiniti, controllare il parametro `Include` nel file di configurazione di Zabbix agent.

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

2. Creare un file *template_db_mysql.conf* nella directory delle configurazioni aggiuntive di Zabbix agent.

```
vi template_db_mysql.conf
```

3. Copiare il contenuto del file *template_db_mysql.conf* (presente nel repository di Zabbix) nel file *template_db_mysql.conf* creato e salvare.

4. Riavviare Zabbix agent per aggiornare la configurazione.

```
systemctl restart zabbix-agent
```

Una volta configurati i parametri utente di Zabbix agent, è possibile passare alla configurazione delle credenziali che consentiranno a Zabbix agent di accedere al server MySQL.

5. Passare alla directory home di Zabbix agent (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 di Zabbix agent.

```
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>'
```

Configurare il frontend di Zabbix e testare la configurazione

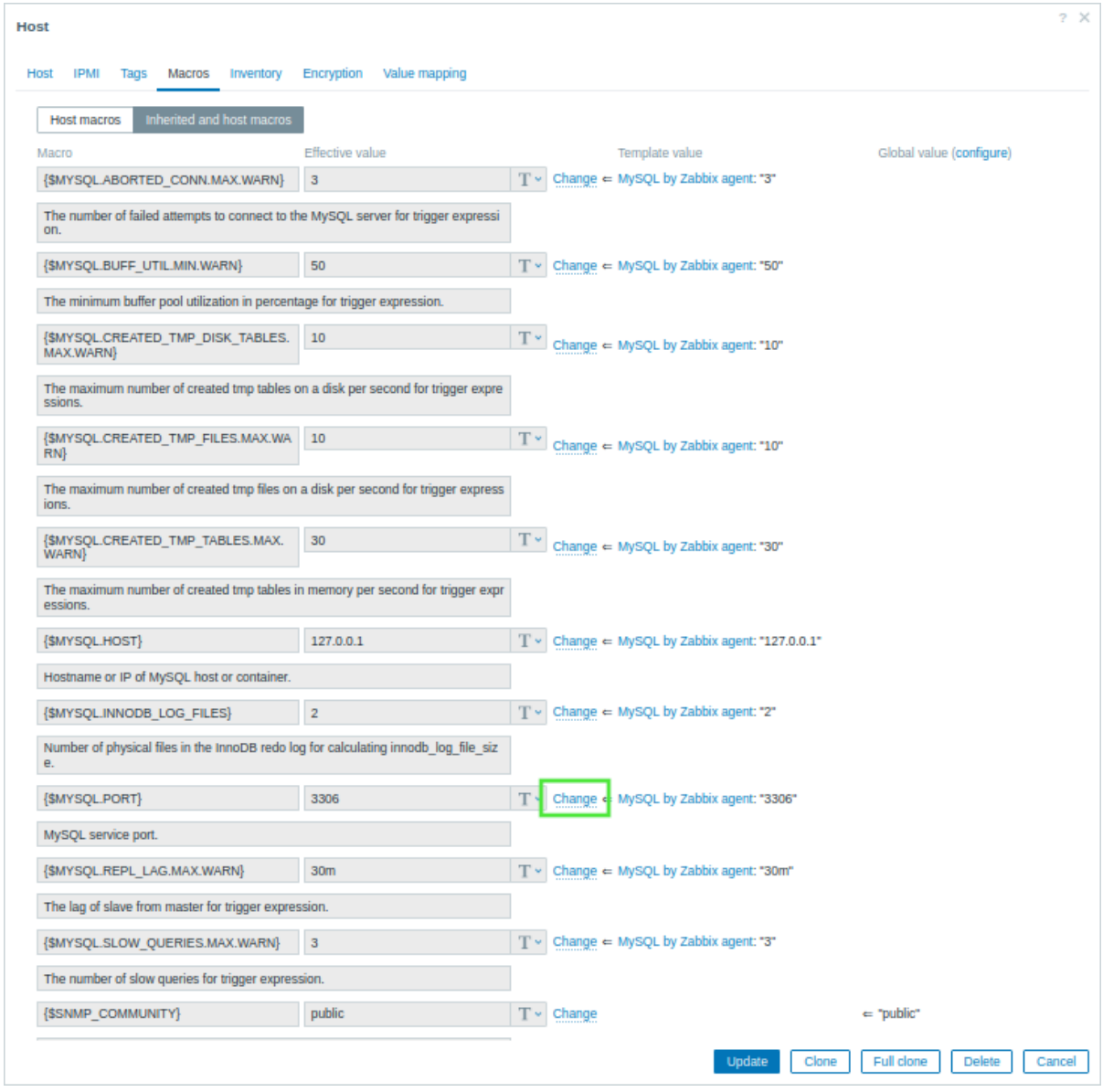
Per configurare il frontend di Zabbix, seguire le istruzioni nella sezione *Configurare il frontend di Zabbix* con le seguenti modifiche:

- Nel campo *Templates*, digitare o selezionare il template "MySQL by Zabbix agent" che verrà collegato all'host.
- La configurazione dei *Macros* non è richiesta.

Una volta configurato il frontend di Zabbix, è possibile visualizzare le metriche raccolte e impostare gli avvisi dei problemi.

Per testare la configurazione, seguire le istruzioni nella sezione *Testare la configurazione* con le seguenti modifiche:

- Nella sezione *Inherited and host macros* della configurazione dell'host del server MySQL, fare clic su *Change* accanto al valore della macro {$MYSQL.PORT} e impostare 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 *Creare un utente MySQL*.

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

Configurare ODBC

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

```
apt install unixodbc
```

2. Installare il driver di database MariaDB unixODBC. Anche se si dispone di un database MySQL, il driver MariaDB unixODBC viene utilizzato 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 utilizza "127.0.0.1" (localhost) come indirizzo del server MySQL per monitorare un server MySQL installato sulla stessa macchina del driver ODBC. Notare il nome dell'origine dati (DSN) "test", che sarà richiesto quando si configura il frontend 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, seguire le istruzioni nella sezione *Configurare il frontend di Zabbix* con le seguenti modifiche:

- Nel campo *Templates*, digitare o selezionare il template "MySQL by ODBC" che verrà collegato all'host.
- La configurazione di *Interfaces* non è richiesta.
- 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, è possibile visualizzare le metriche raccolte, impostare 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 a più fasi (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).

- 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*.

5 Monitorare VMware con Zabbix

Introduzione

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

A chi è rivolta 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 approfondite o hai bisogno di una configurazione più avanzata, consulta la sezione Monitoraggio delle macchine virtuali o la sezione Configurazione del manuale di Zabbix.

Prerequisiti

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

A seconda dell'ambiente in uso, alcuni passaggi di questa guida potrebbero differire leggermente. Questa guida si basa su un ambiente con Ubuntu in esecuzione.

Si presuppone 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 più (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
```

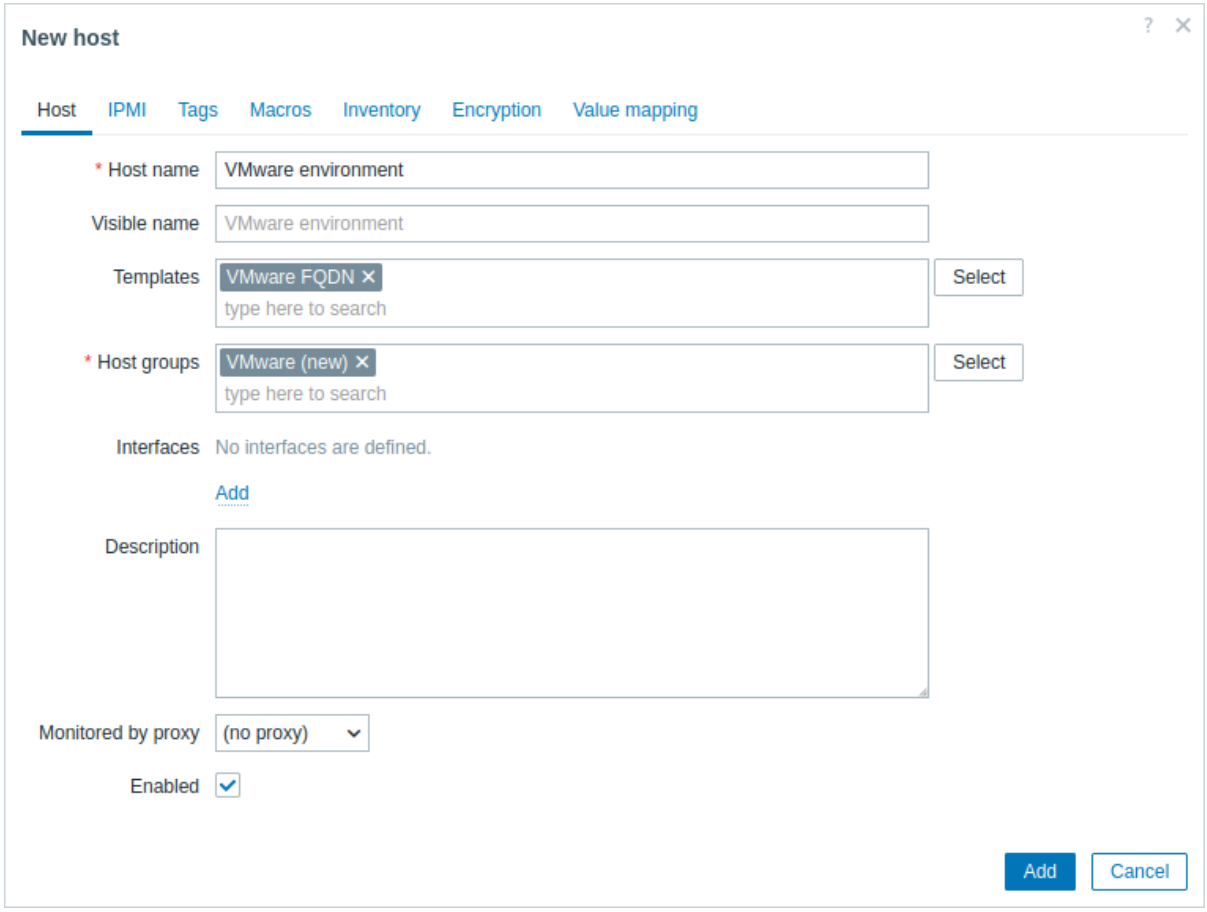
Una volta avviati i processi *vmware collector*, passare al passaggio successivo.

Configurare il frontend di Zabbix

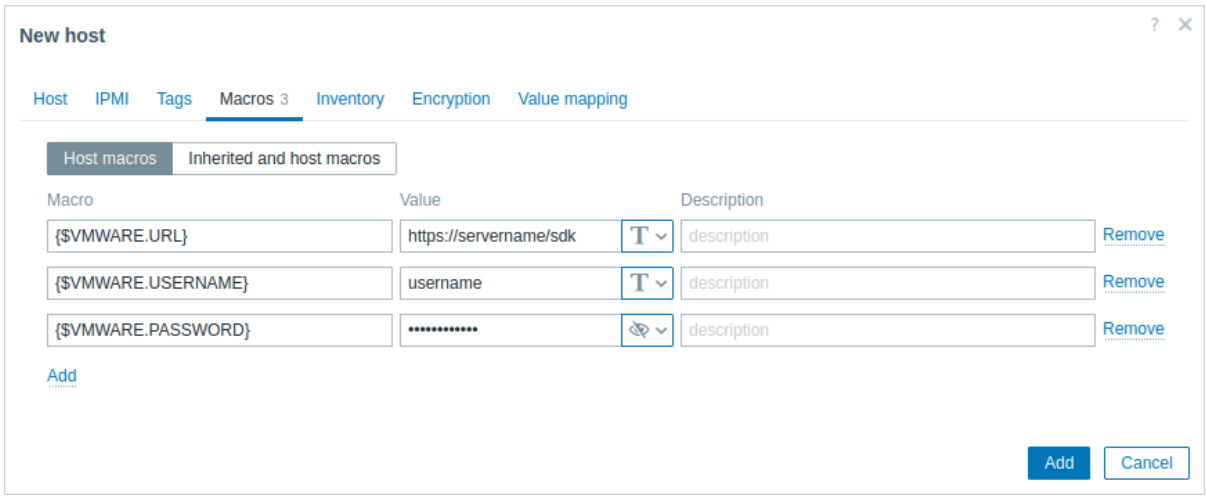
1. Accedere al frontend di Zabbix.

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

- Nel campo *Host name*, inserire un nome host (ad esempio, "VMware environment").
- Nel campo *Templates*, digitare o selezionare il template "VMware FQDN" (oppure "VMware"). Per ulteriori informazioni su questi template, vedere Monitoraggio delle macchine virtuali.
- Nel campo *Host groups*, digitare o selezionare un gruppo host (ad esempio, un nuovo gruppo host "VMware").



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



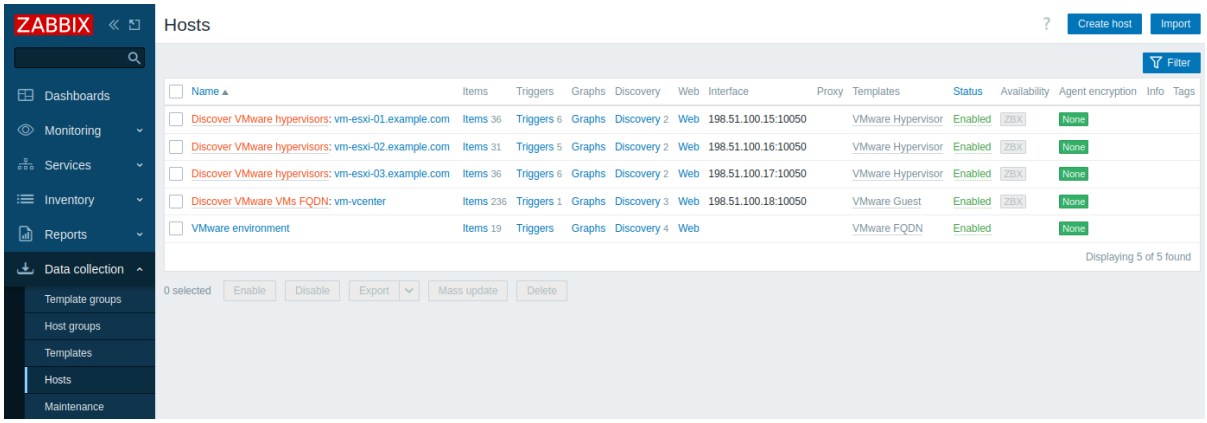
3. Fare clic sul pulsante *Add* per creare l'host. Questo host rappresenterà l'ambiente VMware.

Visualizzare le metriche raccolte

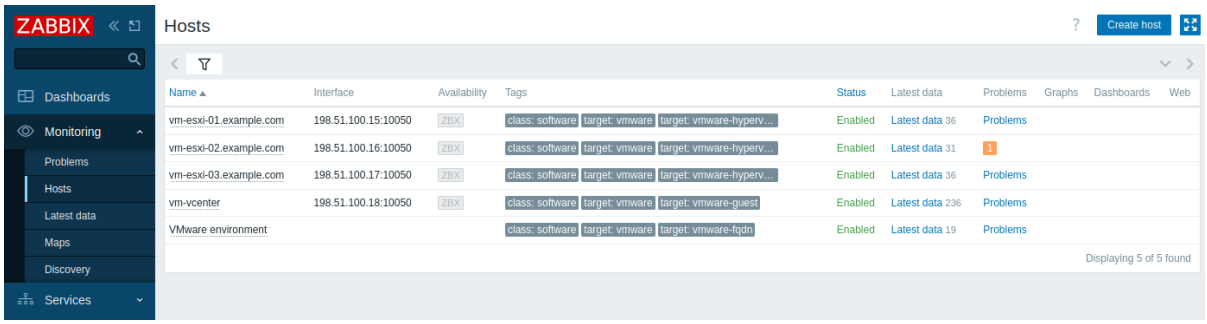
Congratulazioni! A questo punto, Zabbix sta già monitorando il tuo ambiente VMware.

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.

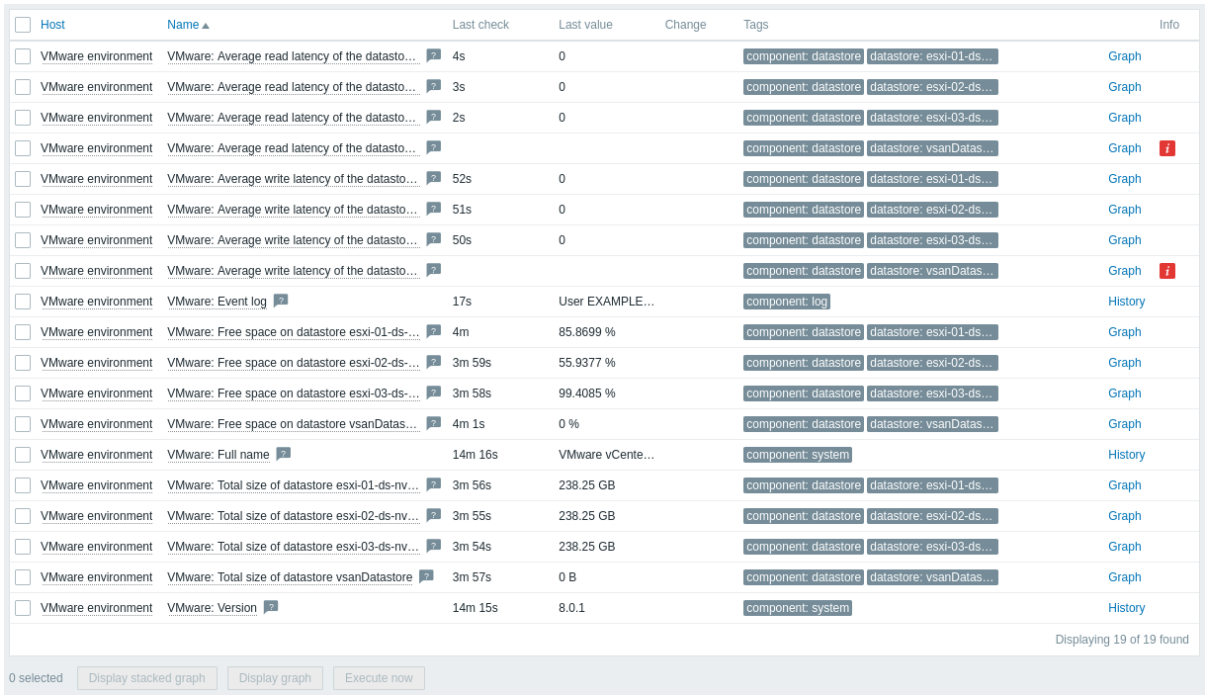
Per visualizzare gli host creati, vai alla sezione di menu *Data collection → Hosts*.



Per visualizzare le metriche raccolte, vai alla sezione di menu *Monitoring → Hosts* e fai clic su *Latest data* accanto all'host "VMware environment" creato oppure a uno degli host creati per le entità rilevate.



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

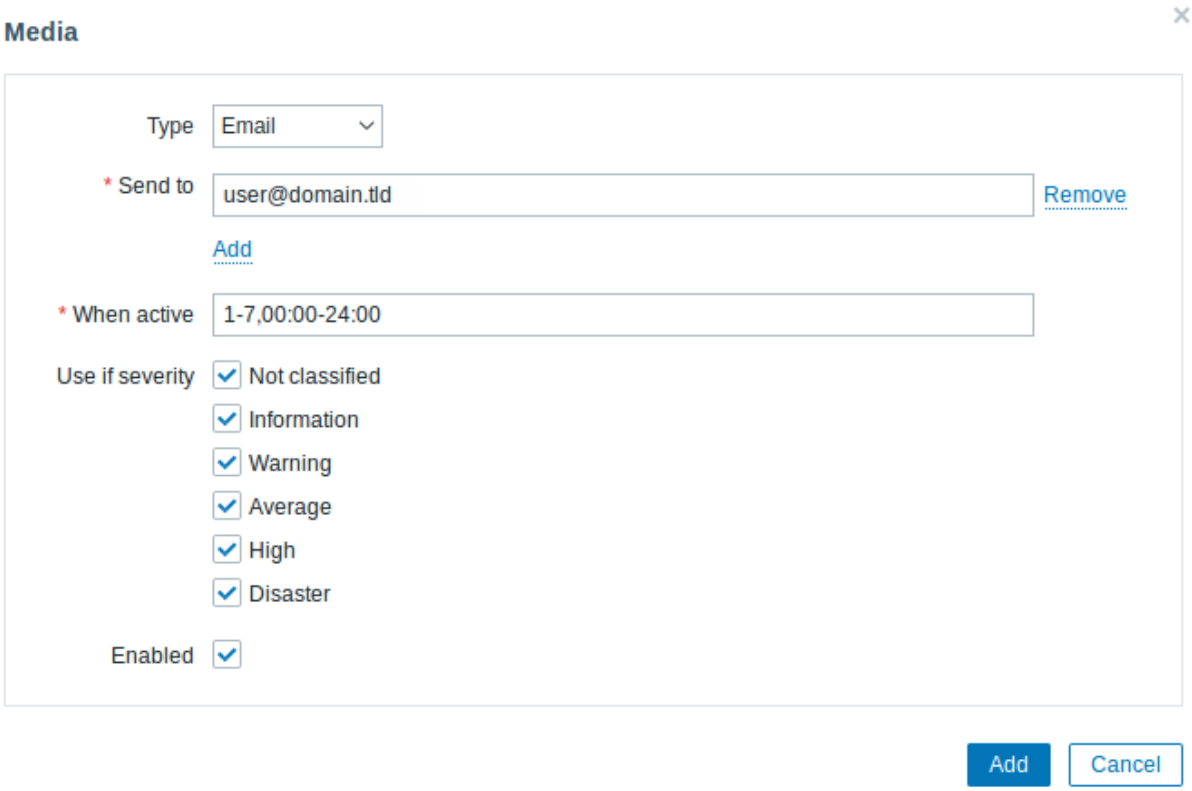


Tieni presente che alcuni item non hanno dati e si trovano nello stato *Not supported*. Questo accade perché Zabbix non riesce a trovare contatori delle prestazioni validi nel datastore specifico, poiché non è abilitato nell'ambiente VMware monitorato.

Configurare gli avvisi di problema

Zabbix può notificare un problema nella tua infrastruttura utilizzando vari metodi. Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi email.

1. Vai a *Impostazioni utente → Profilo*, passa alla scheda *Supporti* 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 in più fasi (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).
- Monitoraggio delle macchine virtuali - informazioni aggiuntive sul monitoraggio VMware (processo di raccolta dei dati, opzioni di configurazione del server, indicazioni per la risoluzione dei problemi, ecc.).
- Chiavi item di monitoraggio VMware - un elenco completo delle metriche VMware che possono essere monitorate utilizzando Zabbix.
- Template *VMware* - informazioni aggiuntive sul template *VMware*.
- Template *VMware FQDN* - informazioni aggiuntive sul template *VMware FQDN*.

6 Monitorare il traffico di rete con Zabbix

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

A chi è rivolta 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 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 in base alle 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 Zabbix agent può raccogliere metriche in modalità attiva o passiva (simultaneamente). Per ulteriori informazioni, vedere Controlli agent passivi e attivi. In questa guida verrà descritto il monitoraggio tramite controlli passivi.

Configurazione di Zabbix agent

1. Aprire il file di configurazione dell'agent sulla macchina in cui è installato (per impostazione predefinita, il percorso è */usr/local/etc/zabbix_agentd.conf*):

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

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

```
Server=192.0.2.22
```

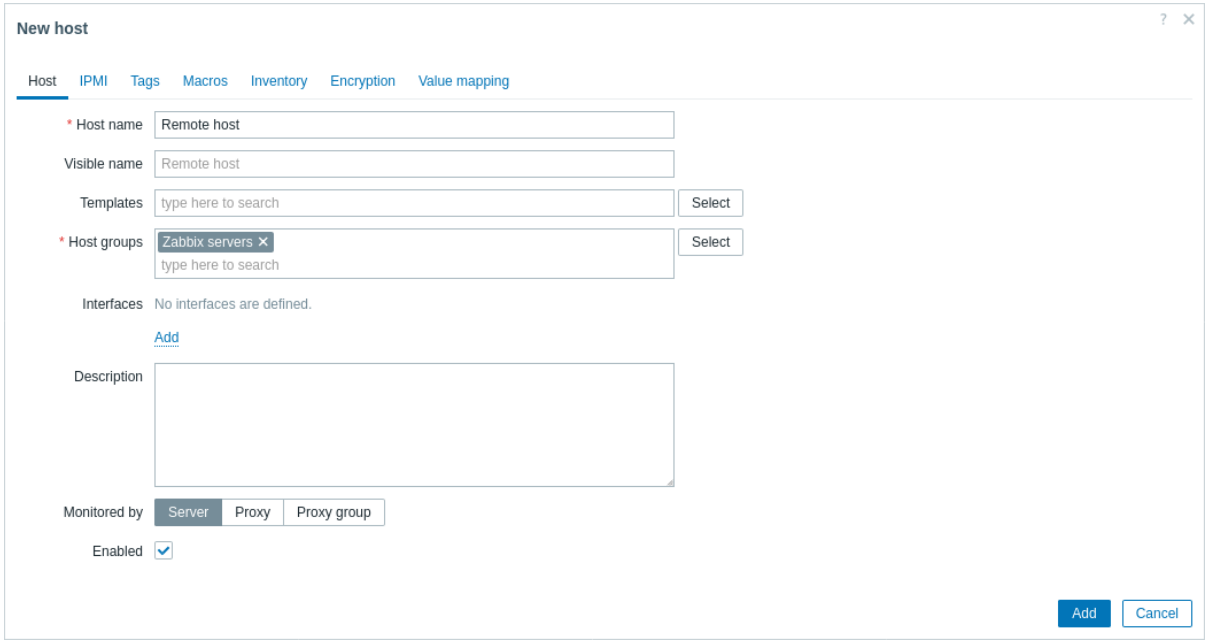
3. Riavviare Zabbix agent:

```
systemctl restart zabbix-agent
```

Frontend Zabbix

1. Accedere al frontend Zabbix.

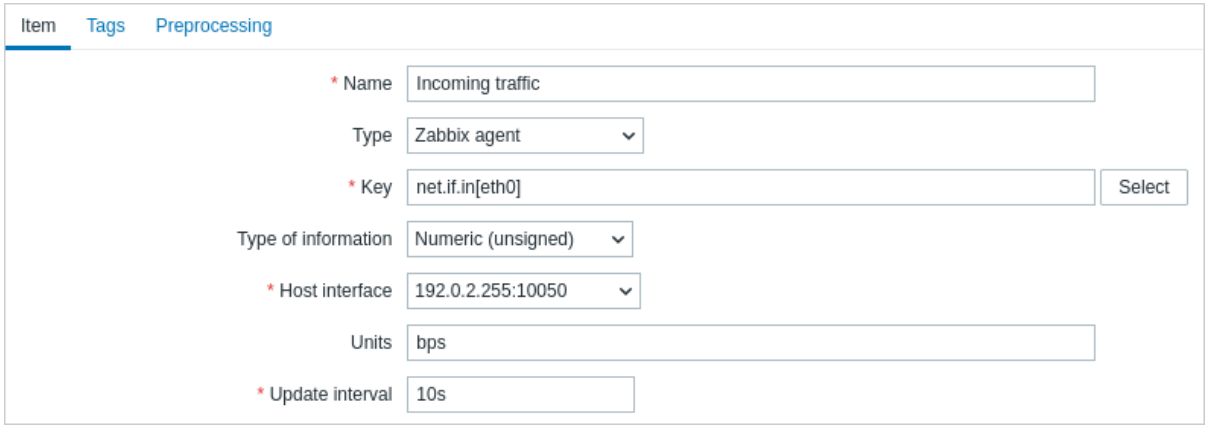
2. Creare un host nell'interfaccia web di Zabbix, specificando l'indirizzo IP o il nome DNS del computer su cui è installato l'agent.



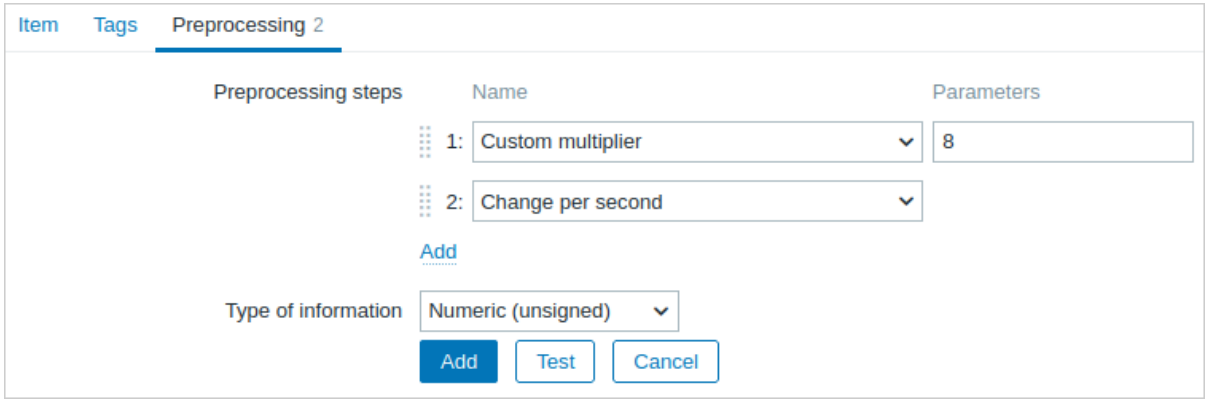
Creare item Segui le istruzioni su come creare un item per aggiungere gli item per il monitoraggio del traffico, vale a dire:

- Traffico in ingresso
- Traffico in uscita
- Traffico totale

Una configurazione semplice per l'item di monitoraggio del traffico in ingresso apparirebbe come segue:

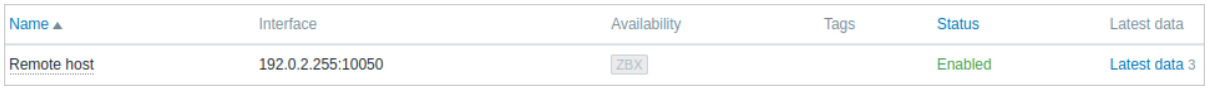


Per rendere i dati raccolti adatti all'uso pratico, puoi impostare alcuni passaggi di preprocessing durante la creazione degli item. Nel caso specifico, questi possono essere la moltiplicazione per 8 (per convertire i byte in bit) e la visualizzazione come variazione al secondo.

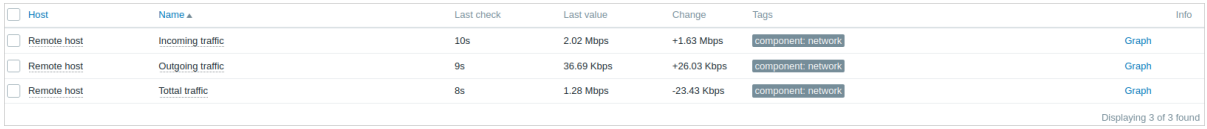


Visualizzare i dati raccolti Congratulazioni! A questo punto, Zabbix sta già monitorando il traffico di rete.

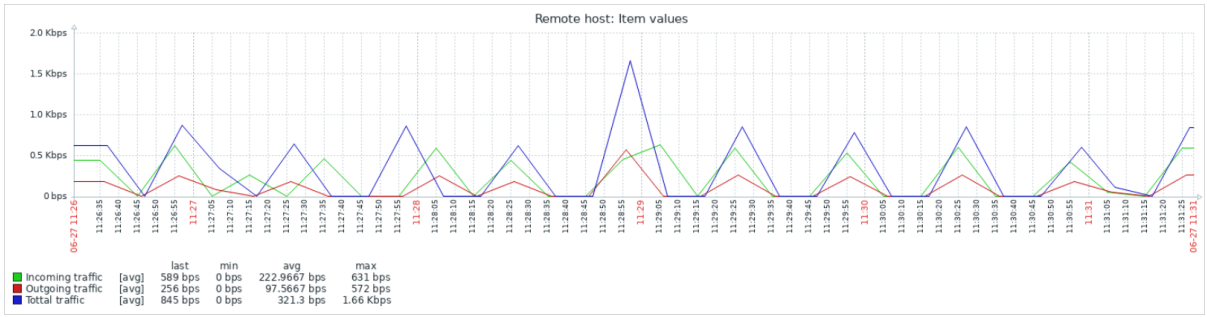
Per visualizzare le metriche raccolte, apri la sezione di menu *Monitoring → Hosts* e fai clic su *Latest data* nella riga del host.



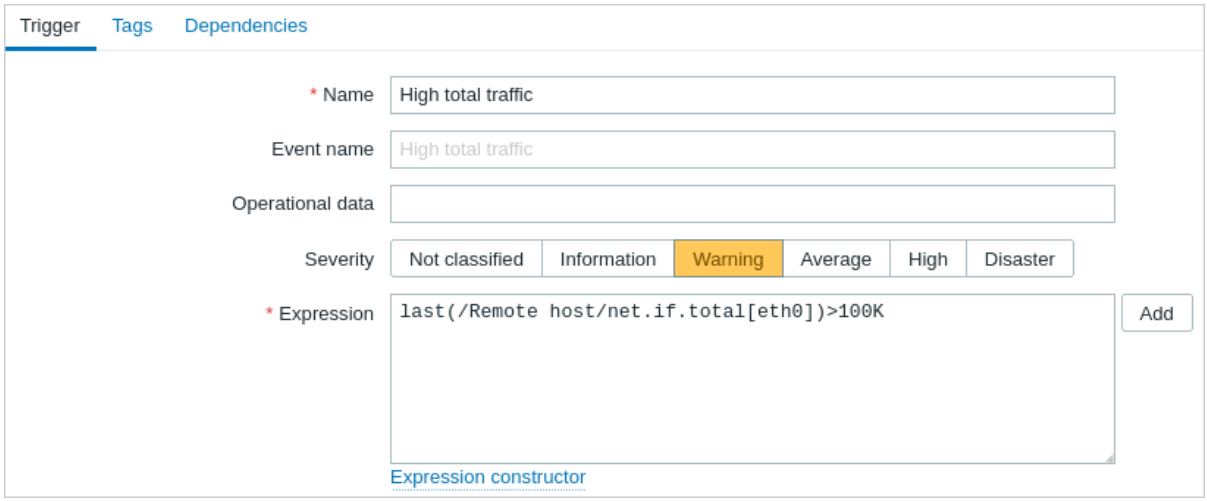
Vedrai i dati sul traffico.



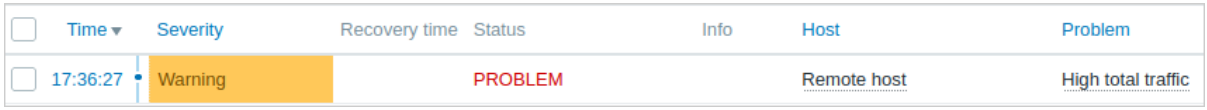
Visualizzare i grafici I dati raccolti possono essere visualizzati come grafici. Per visualizzarli, nella sezione *Ultimi dati* fare clic su *Grafico* nella riga dell'item oppure selezionare gli item richiesti e fare clic su *Visualizza grafico* in basso.



Configurare i trigger È possibile impostare dei trigger per rilevare traffico di rete anomalo. Consultare le istruzioni su come configurare un trigger e aggiungere il trigger che segnala che il traffico totale è troppo elevato, ad esempio:



Ora fare in modo che il traffico superi la soglia impostata nell'espressione del trigger e andare in *Monitoring → Problems* per verificare che il problema sia elencato lì.



Configurare gli avvisi di problema Esistono diversi modi per ricevere notifiche relative a un problema. L'email è il più diffuso; segui le istruzioni per configurare una notifica di problema via email. Puoi anche scegliere altri tipi di supporto da utilizzare per la consegna delle notifiche.

Vedi anche:

- Escalation dei problemi - come creare scenari di avviso in più fasi (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).
- Riconoscimento del problema - come indicare che il problema è noto, aggiungere commenti sulla sua risoluzione, sopprimere o chiudere il problema.
- Monitorare Linux con Zabbix agent - come avviare il monitoraggio di base degli item più importanti collegando un template preconfigurato.

7 Monitorare il traffico di rete utilizzando controlli attivi

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

A chi è rivolta 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 più 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 in base alle 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 Zabbix agent può raccogliere metriche in modalità attiva o passiva (simultaneamente). Per ulteriori informazioni, vedere Controlli agent passivi e attivi. In questa guida verrà descritto il monitoraggio tramite **controlli attivi**.

Configurazione di Zabbix agent

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

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 l'eventuale porta) del proprio Zabbix server al parametro ServerActive. Ad esempio:

```
ServerActive=192.0.2.0:10051
```

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

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

```
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 del 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 quel host. Se questi valori differiscono, l'agent non riceverà la configurazione appropriata, con conseguente assenza di metriche o problemi di monitoraggio.

4. Riavviare Zabbix agent.

Se si utilizza Zabbix agent:

```
systemctl restart zabbix-agent
```

Se si utilizza Zabbix agent 2:

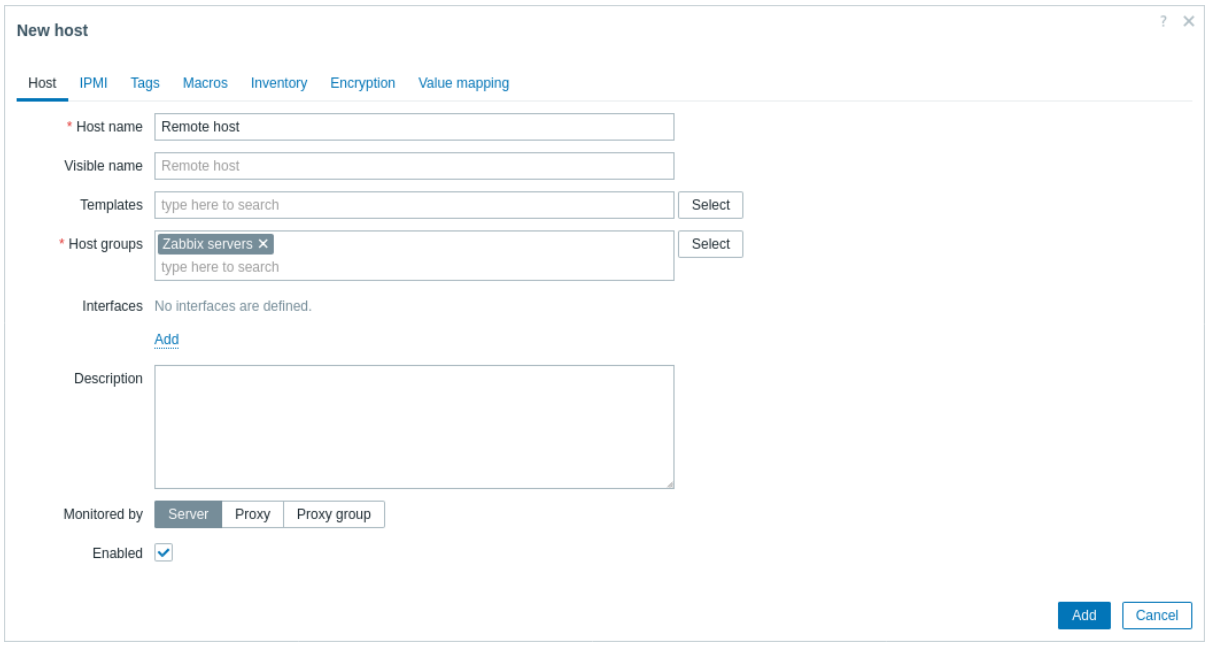
```
systemctl restart zabbix-agent2
```

Zabbix frontend

1. Accedi a Zabbix frontend.

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

- Nel campo *Host name*, inserisci 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*, digita o seleziona un gruppo di host (ad esempio, "Zabbix servers").
- Lascia *Interfaces* non definito; un'interfaccia non è necessaria per i controlli attivi perché è l'agent ad avviare la connessione al server invece di restare in ascolto di una connessione dal server.

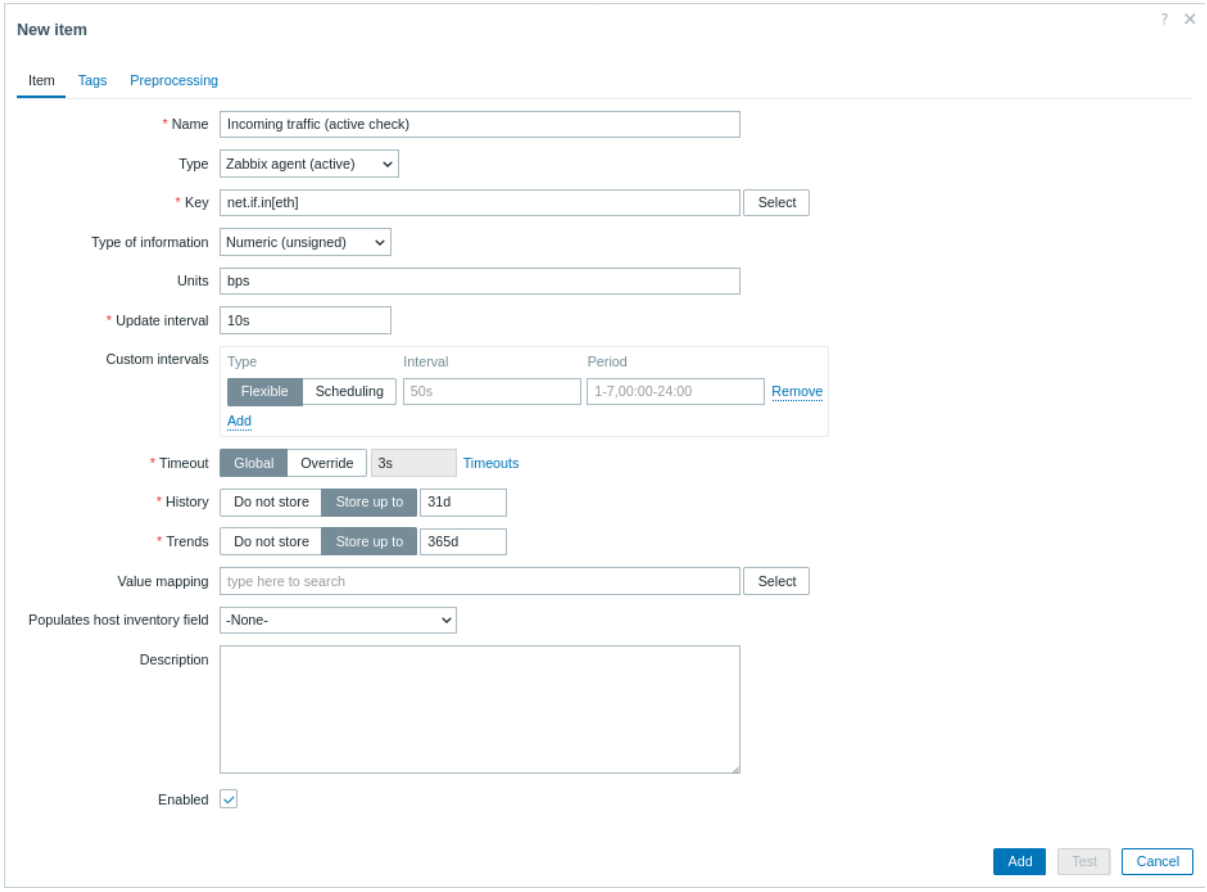


3. Fai clic su *Add* per aggiungere l'host. Questo host rappresenterà la macchina Linux monitorata.

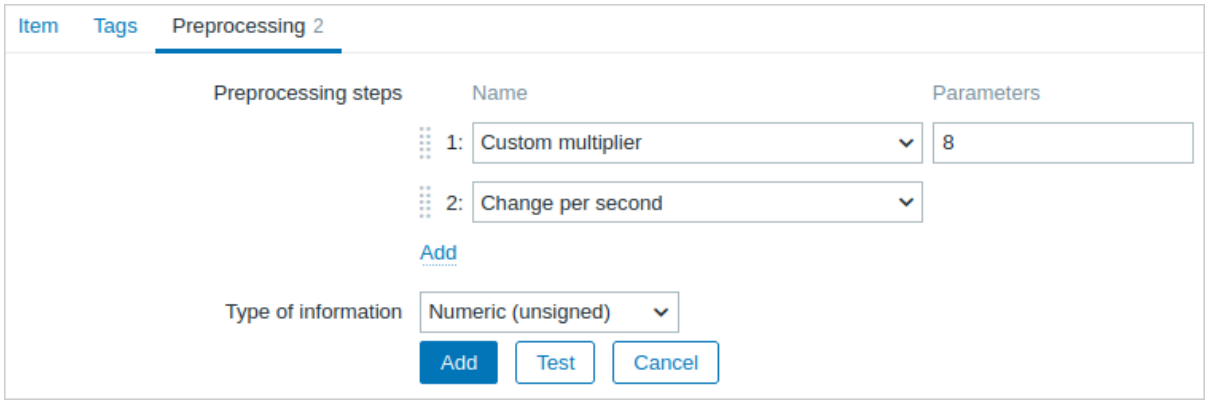
Creare item Segui le istruzioni su come creare un item per aggiungere gli item per il monitoraggio del traffico, vale a dire:

- Traffico in ingresso
- Traffico in uscita
- Traffico totale

Una configurazione semplice per l'item di monitoraggio del traffico in ingresso con un controllo attivo apparirebbe come segue:

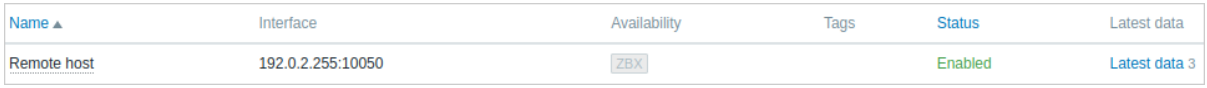


Per rendere i dati raccolti adatti all'uso pratico, puoi impostare alcuni passaggi di preprocessing durante la creazione degli item. Nel caso specifico, questi possono essere la moltiplicazione per 8 (per convertire i byte in bit) e la visualizzazione come variazione al secondo.

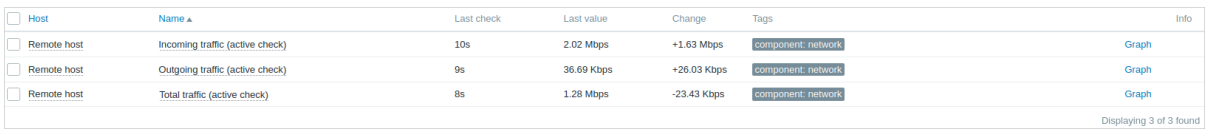


Visualizzare i dati raccolti Congratulazioni! A questo punto, Zabbix sta già monitorando il traffico di rete.

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



Verranno visualizzati i dati sul traffico.



Vedi anche:

- Visualizzare i grafici - come visualizzare i dati raccolti sotto forma di grafici.
- Configurare i trigger - come impostare i trigger per rilevare traffico di rete anomalo.
- Impostare gli avvisi di problema - come configurare 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, quindi, se un problema non viene risolto entro 45 minuti, inviare un messaggio al responsabile del data center).
- Riconoscimento del problema - come indicare che il problema è noto, aggiungere commenti sulla sua risoluzione, sopprimere o chiudere il problema.
- Monitorare 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 di Zabbix utilizzando i pacchetti RPM e DEB ufficiali per varie distribuzioni Linux, garantendo l'accesso alle funzionalità più recenti e alle correzioni di bug.

8 Monitorare i siti web con gli item Browser

Introduzione

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

A chi è rivolta 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 i Browser item. Se stai cercando opzioni di personalizzazione approfondite o hai bisogno di una configurazione più avanzata, consulta la pagina Browser item o la sezione Configurazione 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 del browser richiedono un framework di automazione (Selenium Server oppure un semplice WebDriver, ad esempio ChromeDriver) come endpoint di web testing che controlla e interagisce con un browser, eseguendo comandi di test come il clic sui pulsanti o l'inserimento di 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** - esegue un nuovo container Docker denominato "browser";
- **-p 4444:4444** - mappa la porta 4444 sulla macchina host alla porta 4444 del container (questa è la porta utilizzata da Selenium Server per accettare i comandi);
- **-p 7900:7900** - mappa la porta 7900 sulla macchina host alla porta 7900 del container (questa è la porta utilizzata dal server Virtual Network Computing (VNC), che consente 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 affinché Chrome funzioni correttamente, poiché può richiedere una quantità significativa di memoria condivisa per evitare arresti anomali);
- **-d** - esegue il container in modalità detached, il che significa che verrà eseguito in background;
- **selenium/standalone-chrome:latest** - specifica l'immagine Docker da utilizzare; in questo caso, l'ultima versione di Selenium Server with Chrome.

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

2. Assicurarsi 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 Zabbix server

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

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

1. Aprire il file di configurazione di Zabbix server.

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

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

```
##### Option: WebDriverURL
###URL HTTP[S] dell'interfaccia WebDriver. Ad esempio, http://localhost:4444 utilizzato con il server star
###
### Mandatory: no
### Default:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Riavviare Zabbix server.

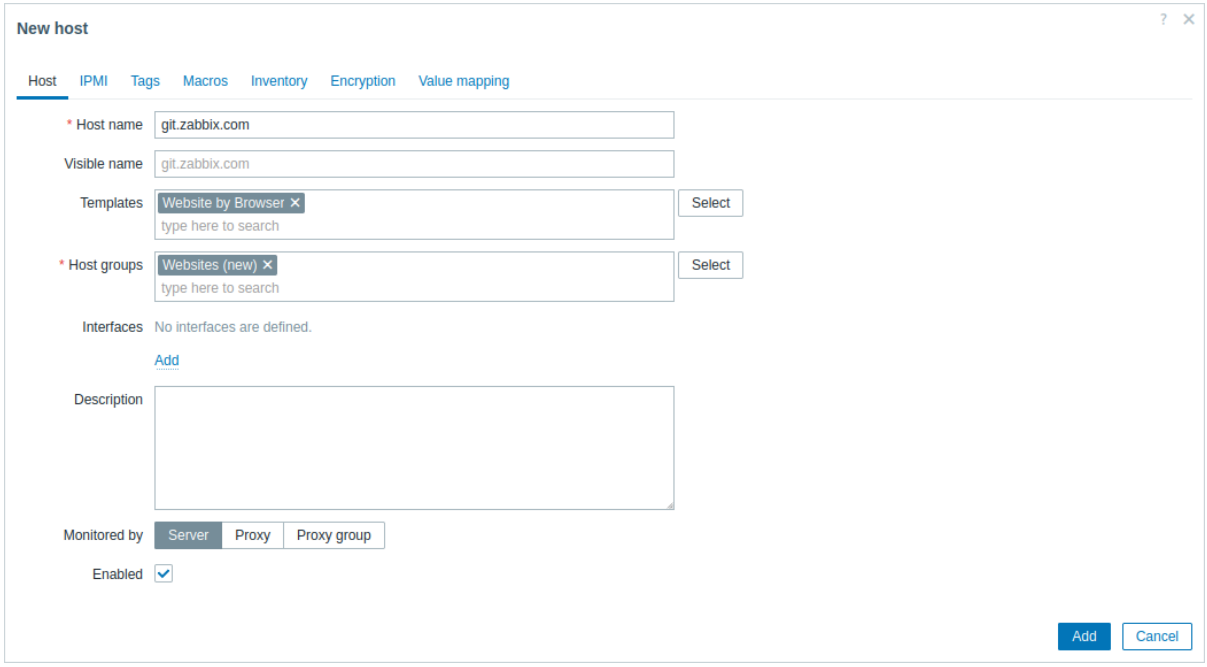
```
systemctl restart zabbix-server
```

Configurare il frontend di Zabbix

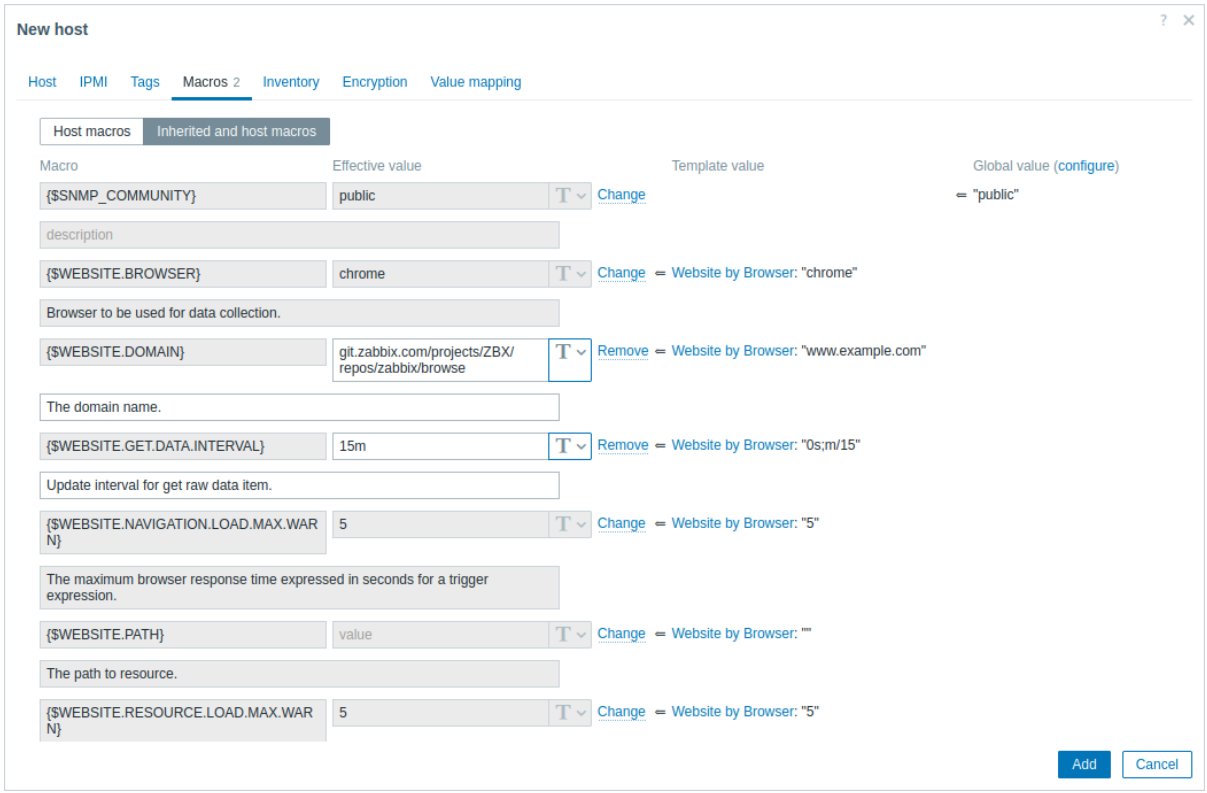
1. Accedi al frontend di Zabbix.

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, vedi Website by Browser.
- Nel campo *Host groups*, digita o seleziona un gruppo di host (ad esempio, un nuovo gruppo di 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)

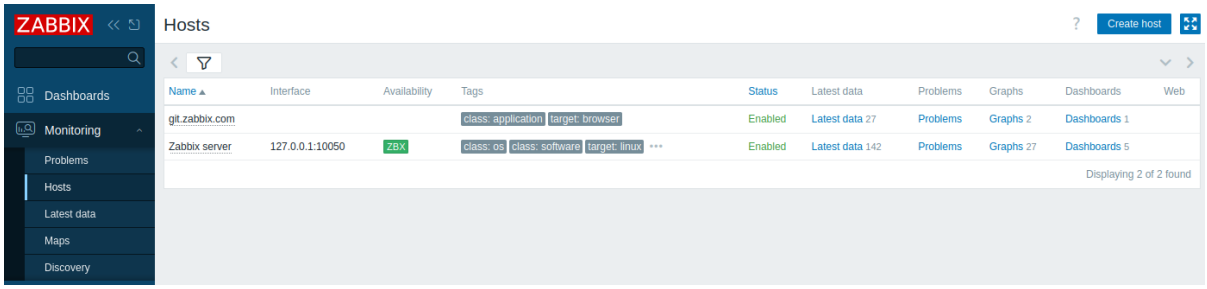


3. Fai clic sul pulsante *Add* per creare l'host. Questo host rappresenterà il sito web che desideri monitorare.

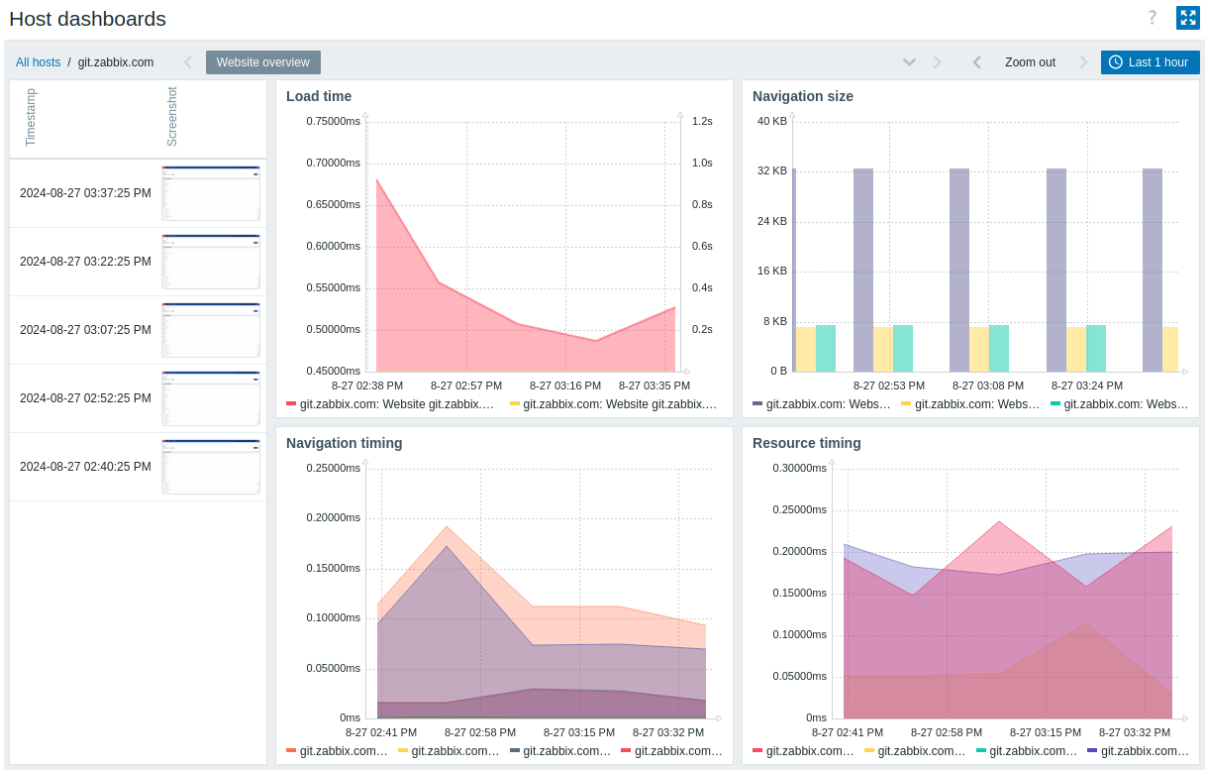
Visualizzare le metriche raccolte

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.



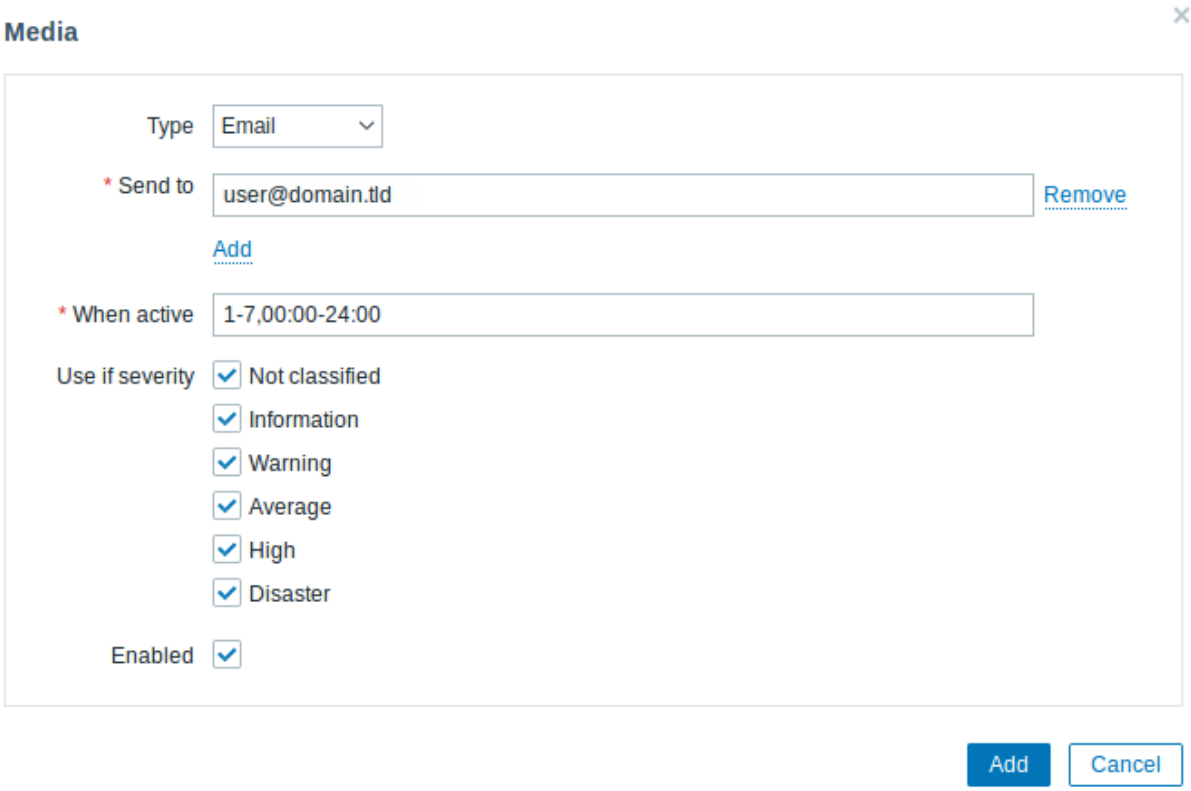
Questa azione ti porterà alla dashboard dell'host (configurata a livello di template) con le metriche più importanti raccolte dal sito web.



Configurare gli avvisi di problema

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

1. Vai a *Impostazioni utente → Profilo*, passa alla scheda *Supporti* 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.

Testare la configurazione

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

1. Apri la configurazione del 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} configurata 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 del host.

5. Tra 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 avvisi sono configurati, riceverai anche la notifica del problema.

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

Vedi anche

- Creazione di un item - come iniziare a monitorare metriche aggiuntive.
- Escalation dei problemi - come creare scenari di avviso in più fasi (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).
- Item Browser - come configurare gli item Browser.
- Template Website by Browser - informazioni aggiuntive sul template *Website by Browser*.

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

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, garantendo agli amministratori di poter identificare rapidamente potenziali problemi come certificati scaduti o non validi.

A chi è rivolta 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 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 2 secondo le istruzioni per il proprio sistema operativo. Questo tutorial presuppone che sia Zabbix server sia agent siano installati sulla stessa macchina; pertanto, nella configurazione viene utilizzato `127.0.0.1`.

Configurare Zabbix agent 2

1. Aprire il file di configurazione di Zabbix agent (percorso predefinito: `/etc/zabbix/zabbix_agent2.conf`):

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

2. Impostare il parametro Server su 127.0.0.1, poiché agent e 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 l'installazione e la configurazione di 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, quindi non è richiesta alcuna installazione o configurazione separata.

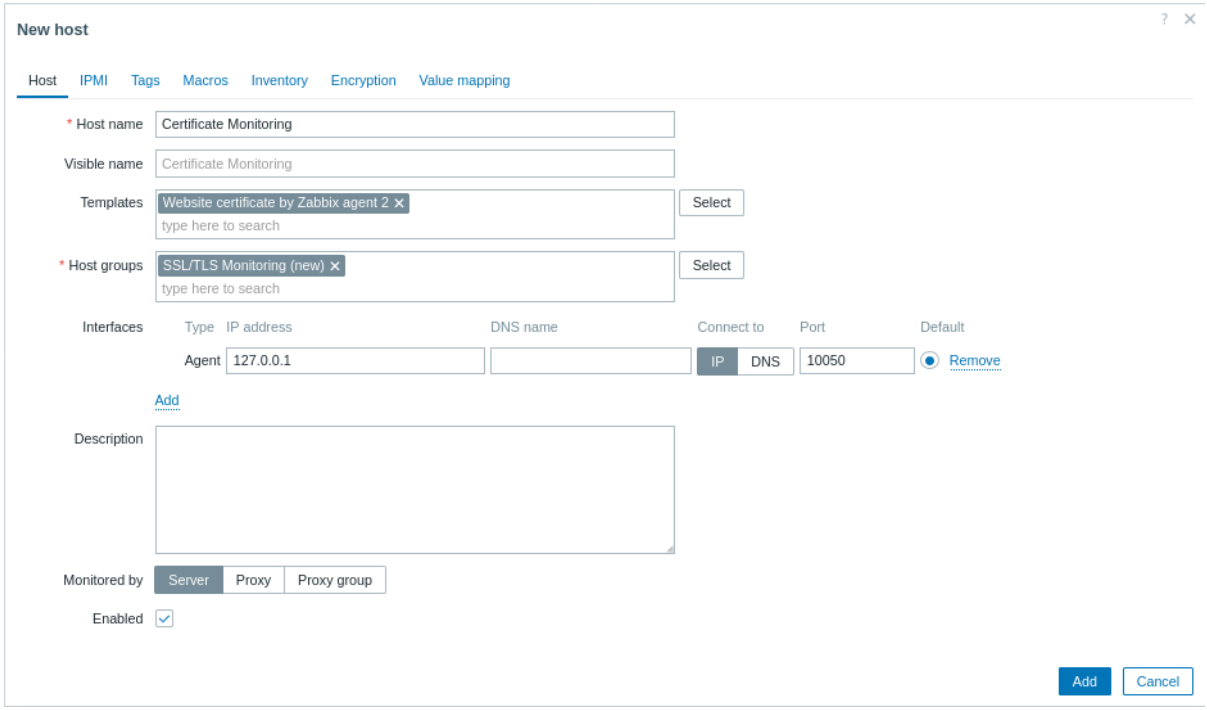
Configurare il frontend di Zabbix

1. Accedi al frontend di Zabbix.

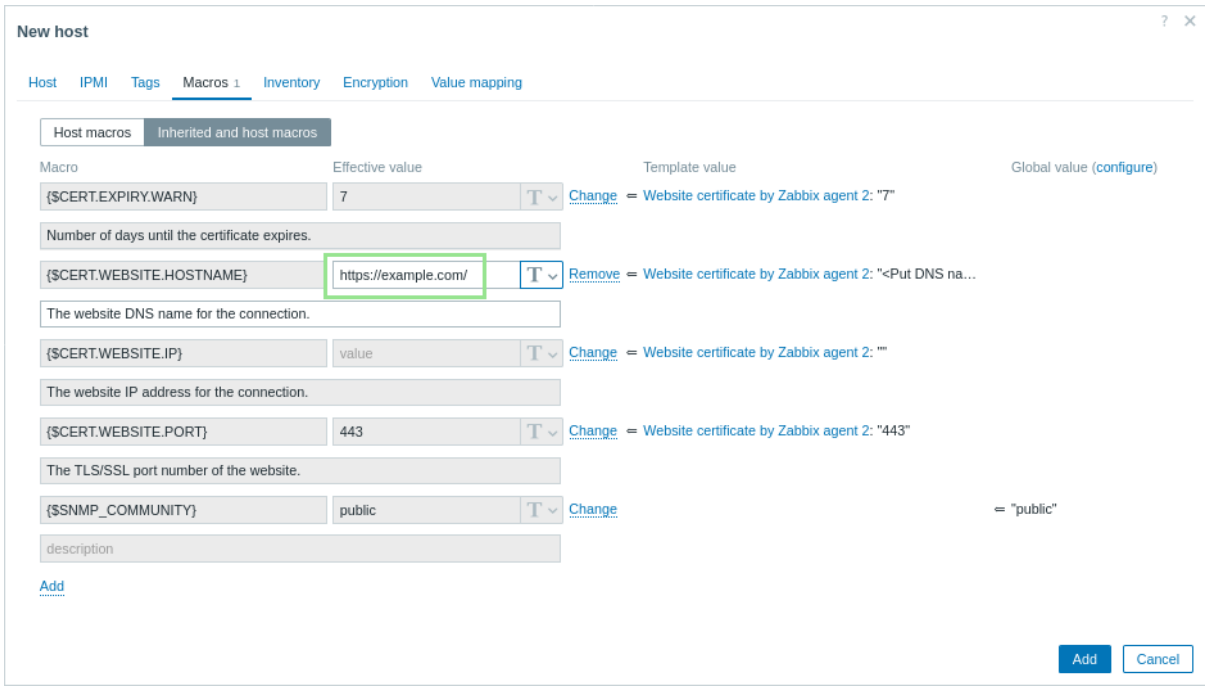
2. Vai a *Monitoring* > *Hosts*.

3. Fai clic su un host esistente in cui vuoi monitorare i certificati dei siti web, oppure crea un host se necessario:

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



- 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:
  - `{$CERT.WEBSITE.HOSTNAME}` - inserisci come valore il nome DNS del sito web desiderato.



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

Per monitorare più siti web, puoi specificare un elenco di nomi host separati da virgole nella macro `{$CERT.WEBSITE.HOSTNAME}`. Facoltativamente, puoi anche fornire un elenco di porte separato da virgole nella macro `{$CERT.WEBSITE.PORT}`, in cui ogni porta corrisponde al rispettivo nome host nello stesso 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

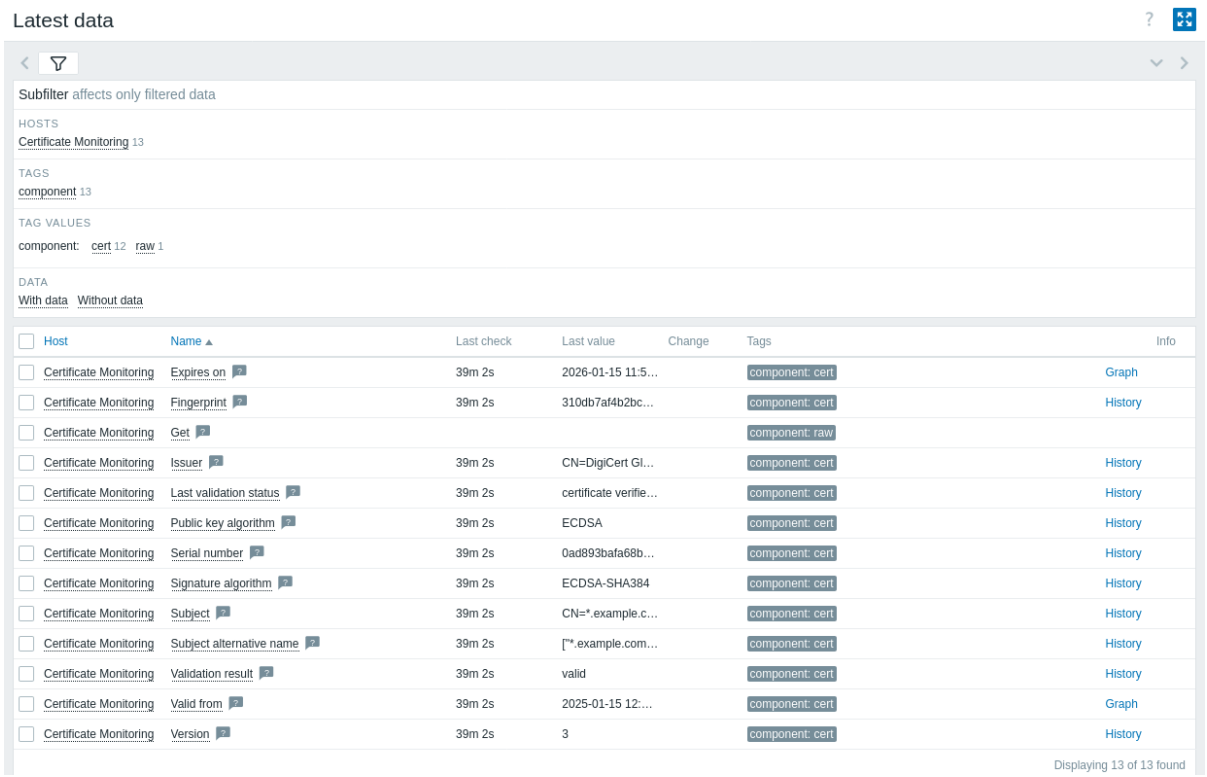
Per ogni sito web specificato nella macro `{$CERT.WEBSITE.HOSTNAME}`, Zabbix creerà un insieme corrispondente di item e trigger. Ciò consente il monitoraggio e l'invio di avvisi individuali per il certificato SSL di ciascun sito web.

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.

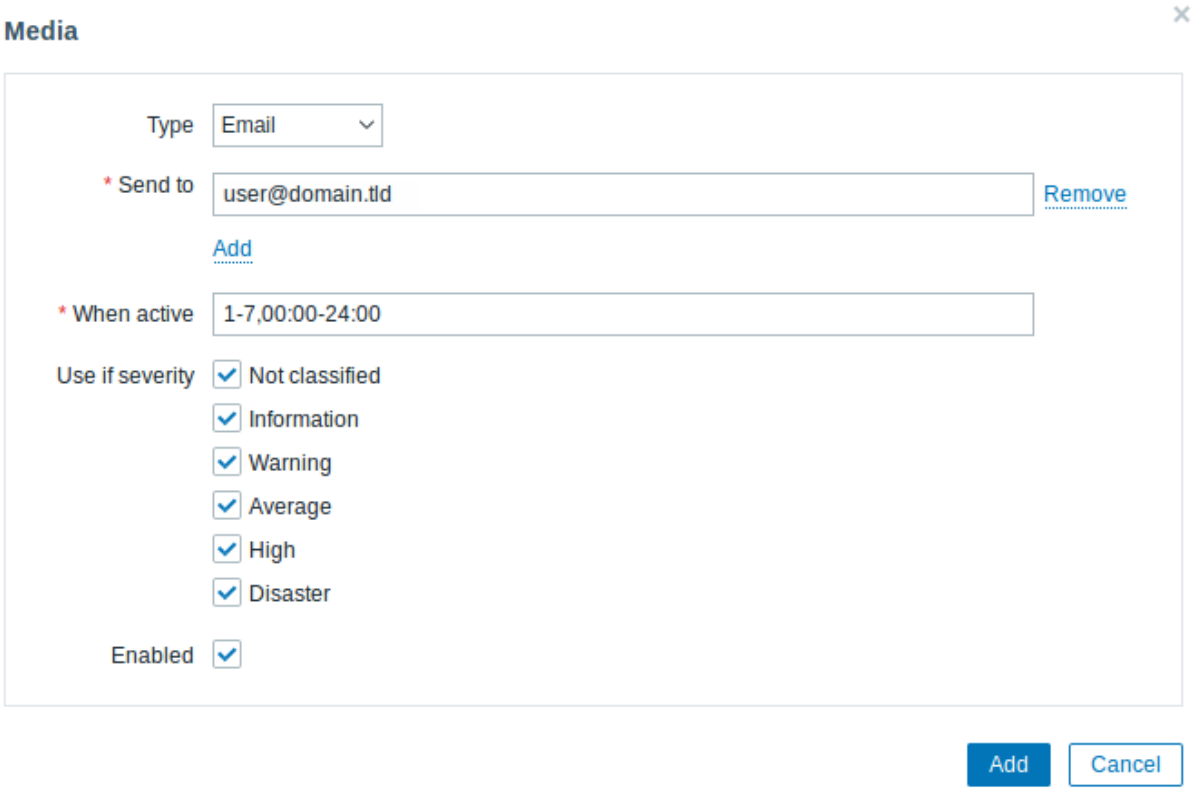
Per visualizzare le metriche raccolte, andare alla sezione di menu *Monitoring > Hosts* e fare clic su *Latest data* accanto all'host per visualizzare in un elenco tutte le metriche raccolte più recenti, come la data di scadenza, l'emittente e il soggetto.



Configurare gli avvisi di problema

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

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Supporti* 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.

Testa la tua configurazione

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

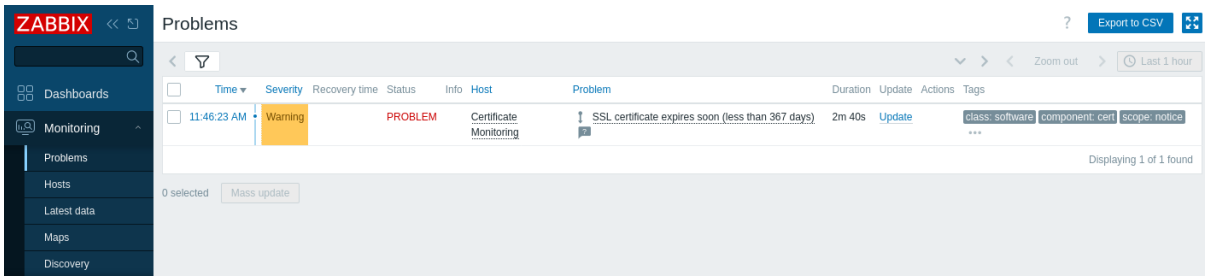
1. Apri la configurazione del tuo host "Certificate Monitoring" in Zabbix.

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

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

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

5. Tra qualche istante, Zabbix rileverà il problema "SSL certificate expires soon", con il numero di giorni rimanenti alla scadenza. 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 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 gli item Browser.

10 Monitorare uno switch o un router di rete con Zabbix

Introduzione Questa guida illustra i passaggi necessari per avviare il monitoraggio di base del tuo switch o router di rete utilizzando Zabbix. Come esempio viene usato un router Cisco, ma la procedura si applica a qualsiasi dispositivo di rete con SNMP abilitato.

A chi è rivolta 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: esegui l'installazione secondo le istruzioni per il tuo sistema operativo (vedi Installation from packages e Web interface installation).
- Zabbix agent installato se devi monitorare metriche di rete locali.
- Dispositivo con SNMP abilitato: 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. 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 per il download dei MIB:

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

Se devi aggiungere MIB specifici del fornitore (ad esempio, di Cisco o Juniper), inseriscili nella directory MIB appropriata:

- Per i sistemi basati su Linux, i percorsi 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 d'ambiente MIBDIRS o il file snmp.conf includano 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 uno `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 con Ubuntu in esecuzione 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 in modo da consentire le query SNMP. L'esempio seguente è relativo a 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 indicati di seguito.

Esempio SNMPv2 1. Abilitare SNMP e impostare la stringa di community.

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

```
configure terminal
```

Quindi, abilita SNMP specificando una stringa di community in sola lettura. Per esempio:

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

Sostituisci `<your_community_string>` con la tua stringa di community 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, si consiglia di limitare l'accesso SNMP solo ai dispositivi necessari. Per ulteriori indicazioni sulla configurazione delle liste di controllo degli accessi (ACL), consulta la documentazione ufficiale di Cisco.

2. Salvare la configurazione.

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

```
write memory
```

Esempio SNMPv3 SNMPv3 offre una sicurezza avanzata con autenticazione e crittografia. La sua configurazione è più sicura rispetto a 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 per il modello, è possibile fare riferimento ai tutorial esterni sulla configurazione SNMP di Cisco. Questa guida, tuttavia, 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.

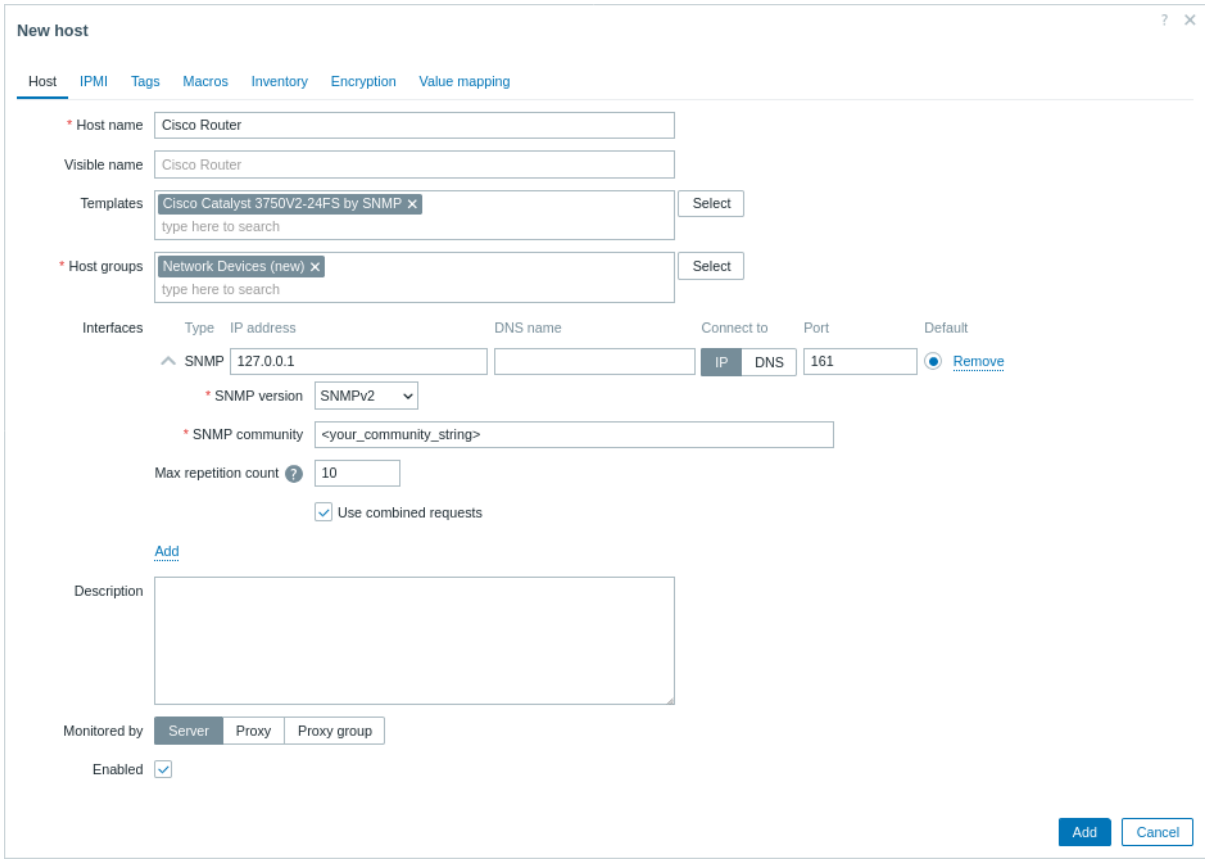
Vai a *Data collection* > *Hosts* e fai clic su *Create host*.

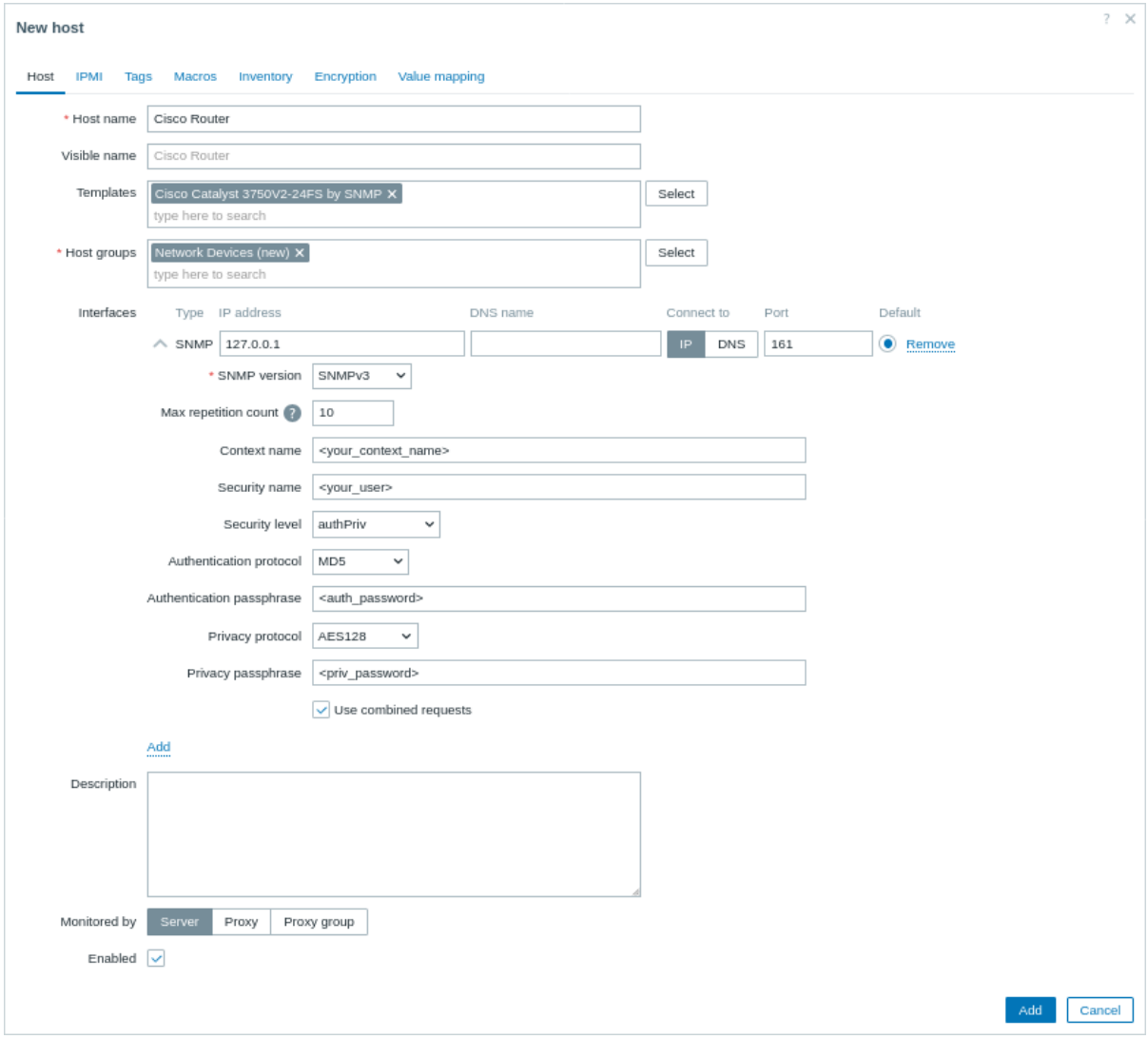
- *Host name*: inserisci un nome per il tuo dispositivo (ad esempio, ”Cisco Router”).
- Host groups: seleziona un gruppo esistente oppure crea un nuovo gruppo, ad esempio ”Network Devices”.
- Interfaces:
  - Fai clic su *Add* sotto Interfaces.
  - Scegli *SNMP* come tipo di interfaccia.
  - Inserisci 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 stringa di community 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 tuo dispositivo (come Cisco IOS SNMP o Cisco Catalyst 3750<modello dispositivo> SNMP).

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

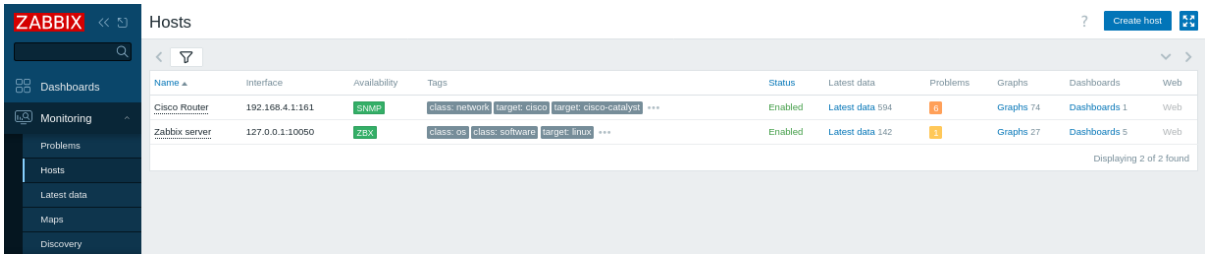




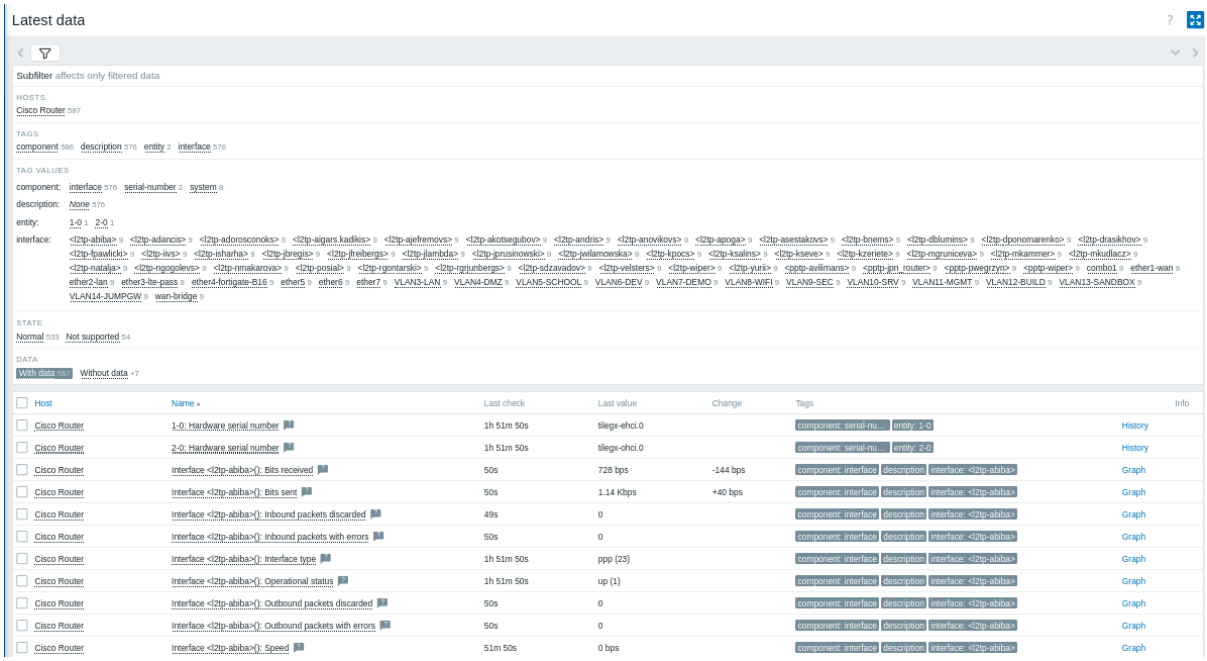
Visualizzare le metriche raccolte Congratulazioni! Zabbix è ora configurato per monitorare il tuo dispositivo di rete.

Dati più recenti:

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



- Seleziona il tuo host “Cisco Router” (o gli host rilevati automaticamente) per visualizzare metriche quali uptime dell’hardware e della rete, perdita ICMP, ping e 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 passaggio successivo, puoi:

- Aggiungere item SNMP personalizzati per monitorare metriche aggiuntive.
- Configurare avvisi di problema 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 utilizzando un template, poiché i template contengono già insiemi predefiniti di item.

1. Identificare l'OID SNMP:

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

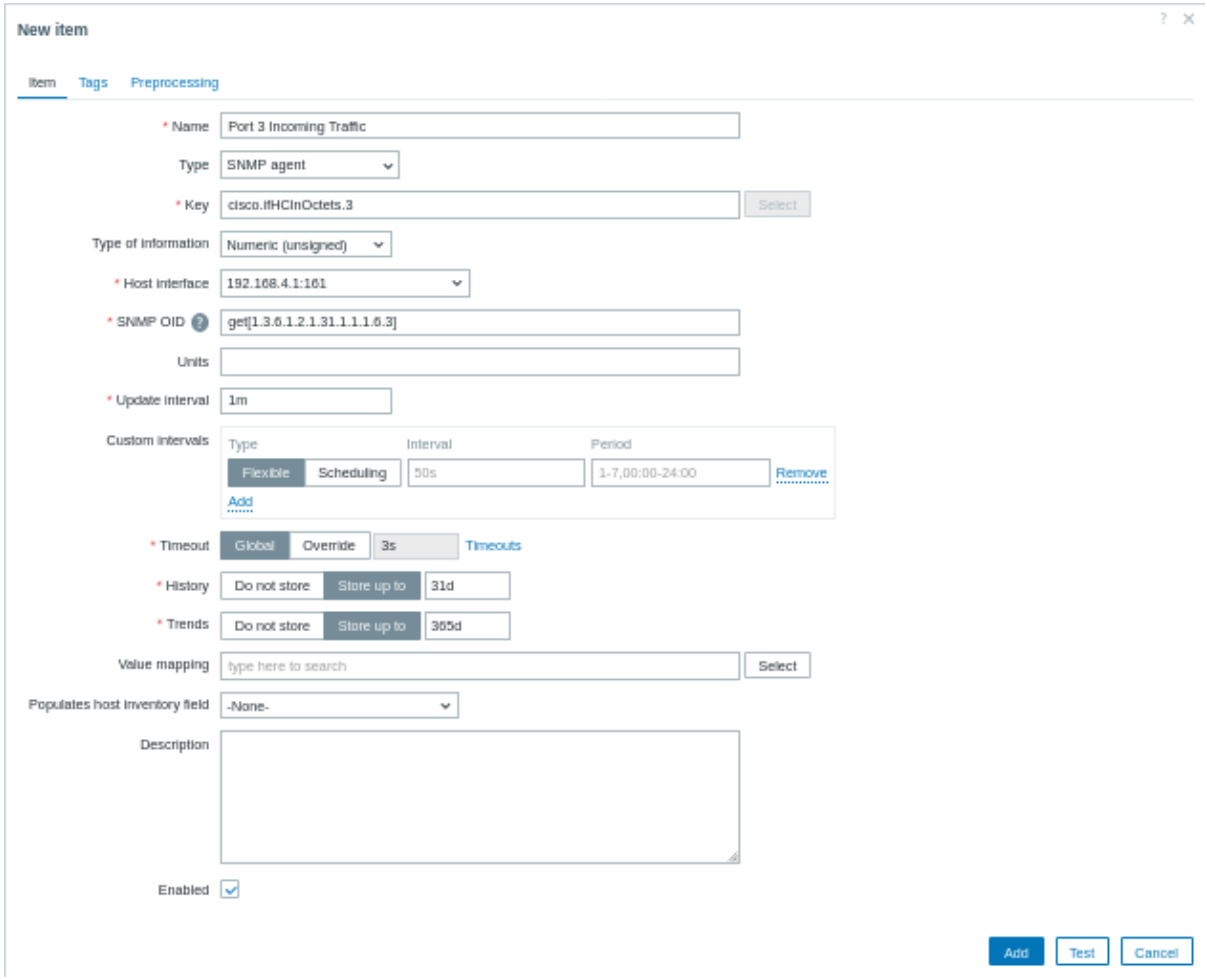
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Trova l'OID della 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:

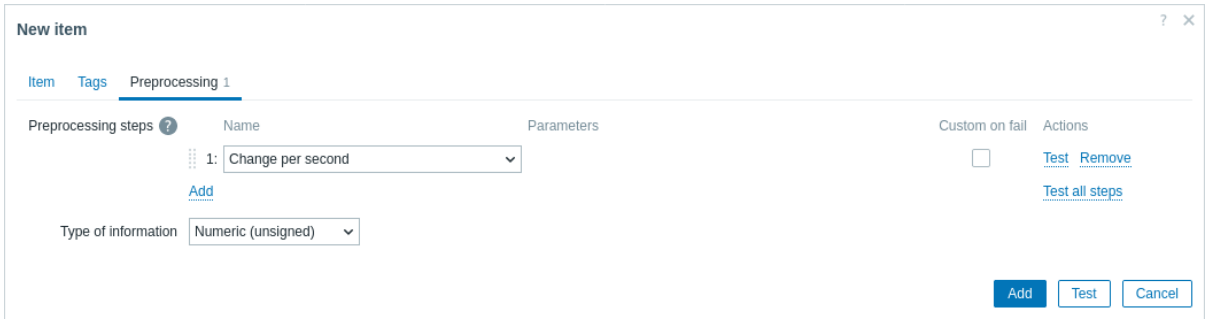
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Creare un item SNMP:

- Vai a *Data collection* > *Hosts* e fai clic sulla scheda *Items* del 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, per 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 la velocità.



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

Traduzione degli OID tra nomi numerici e MIB Quando si lavora con SNMP, potrebbe essere necessario convertire tra OID numerici e i corrispondenti nomi MIB. Questa traduzione aiuta a identificare e risolvere i problemi delle metriche più facilmente.

- Traduzione di un nome MIB in un OID numerico: utilizzare il comando `snmptranslate` con l'opzione `-On`. Ad esempio, per tradurre il nome MIB `IF-MIB::ifHCInOctets.3` nel relativo OID numerico, eseguire:

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

Questo comando potrebbe restituire:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Traduzione di un OID numerico nel relativo nome MIB: utilizzare il comando `snmptranslate` con l'opzione `-IR` (oppure `-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 relativo nome MIB, eseguire:

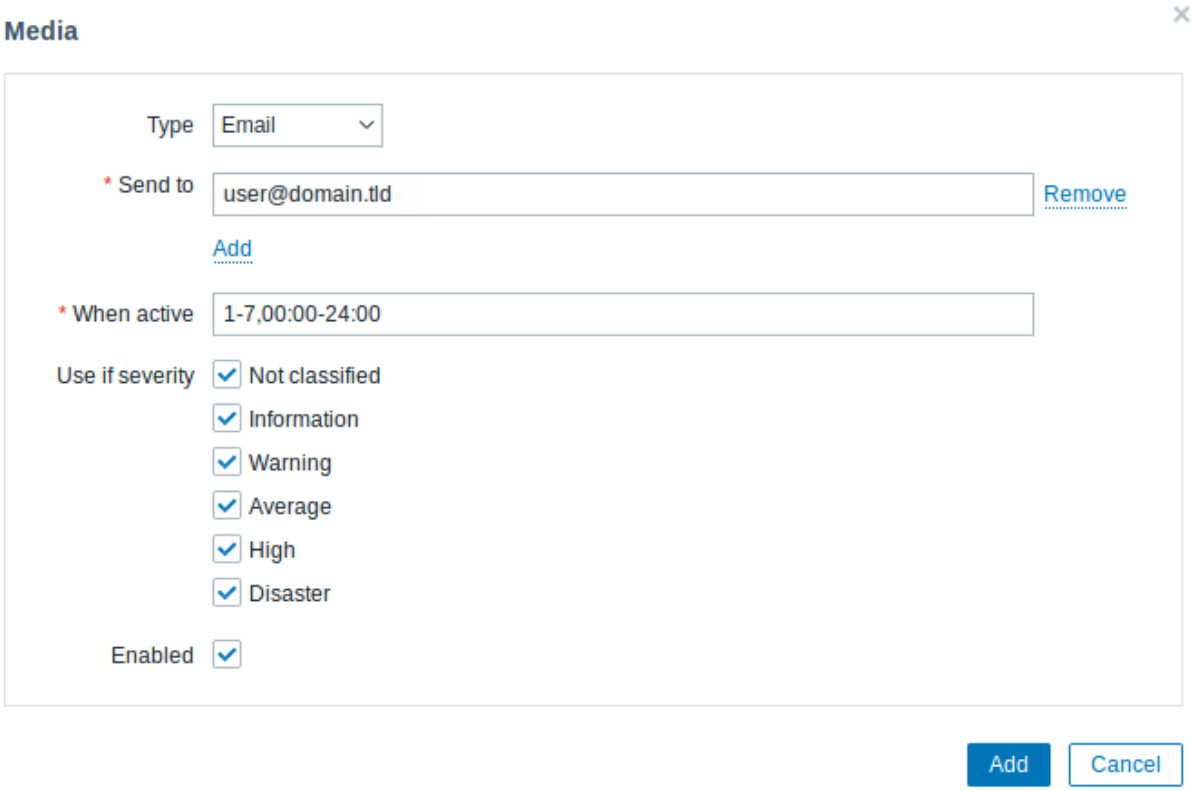
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Questo comando potrebbe restituire:

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

Configurare gli avvisi di problema Questa guida fornisce i passaggi di configurazione di base per l'invio di avvisi email.

1. Vai a *Impostazioni utente > Profilo*, passa alla scheda *Supporti* 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.

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

1. Apri la configurazione del tuo host "Cisco Router" in Zabbix.

2. Vai alla scheda *Macro* e seleziona *Macro ereditate e dell'host*.

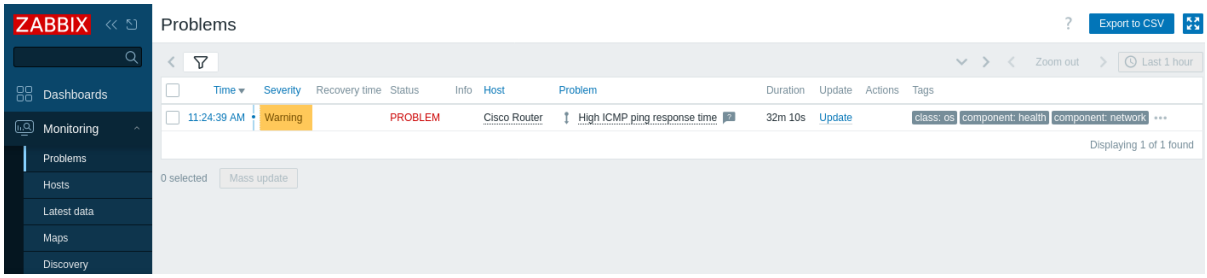
3. Individua la macro `{$ICMP_RESPONSE_TIME_WARN}` (o una macro simile per la soglia del tempo di risposta).

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 *Aggiorna* per applicare le modifiche.

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

7. Vai a *Monitoraggio > Problemi* per verificare che venga visualizzato un avviso (ad esempio, "Tempo di risposta del ping ICMP elevato").



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

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

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

Risoluzione dei problemi del monitoraggio SNMP Se noti che l'icona SNMP nel frontend di Zabbix appare ROSSA oppure non viene raccolto alcun dato, 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 tuo dispositivo. Ad esempio, controlla le impostazioni SNMP nella configurazione del host in Zabbix e confrontale con la configurazione del tuo dispositivo. Su un dispositivo Cisco, puoi controllare le impostazioni SNMP eseguendo:

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

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

4. Verifica che SNMP sia correttamente abilitato 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 (in genere 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. Controlla i file di log del server Zabbix per eventuali errori relativi a SNMP, in modo da individuare il problema:

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

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

Vedi anche:

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

11 Monitorare il log eventi di Windows utilizzando controlli attivi

Introduzione

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

A chi è rivolta questa guida

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

Prerequisiti

Prima di procedere con questa guida, è necessario scaricare e installare Zabbix server e Zabbix frontend secondo le istruzioni per il proprio sistema operativo. È inoltre necessario che Zabbix agent sia scaricato e installato sulla macchina Windows che si desidera monitorare.

Configurare Zabbix agent per il monitoraggio del registro eventi di Windows

1. Aprire `zabbix_agentd.conf` (percorso predefinito `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) sul proprio host Windows e verificare che il parametro ServerActive sia impostato sull'indirizzo IP del proprio Zabbix server e che il parametro Hostname corrisponda al nome host che verrà definito nel Zabbix frontend. Questo consente all'agent di richiedere controlli attivi per il proprio host e dal Zabbix server specificato. Ad esempio:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

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

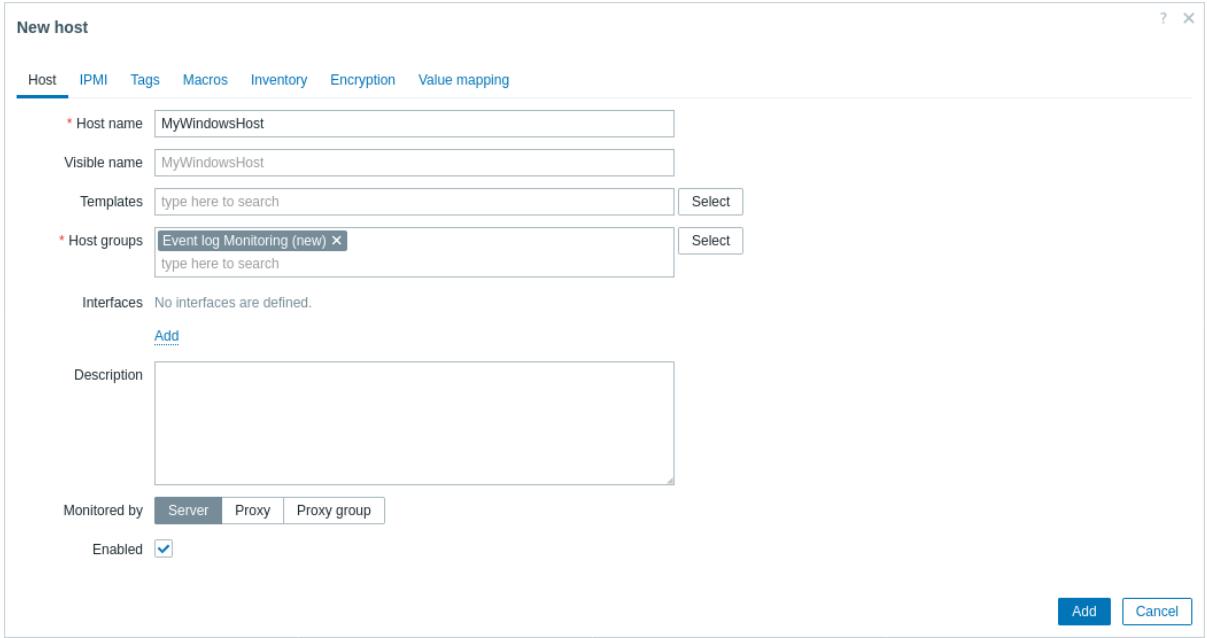
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Configurare il frontend di Zabbix

1. Andare su *Raccolta dati* > *Host* e creare un host:

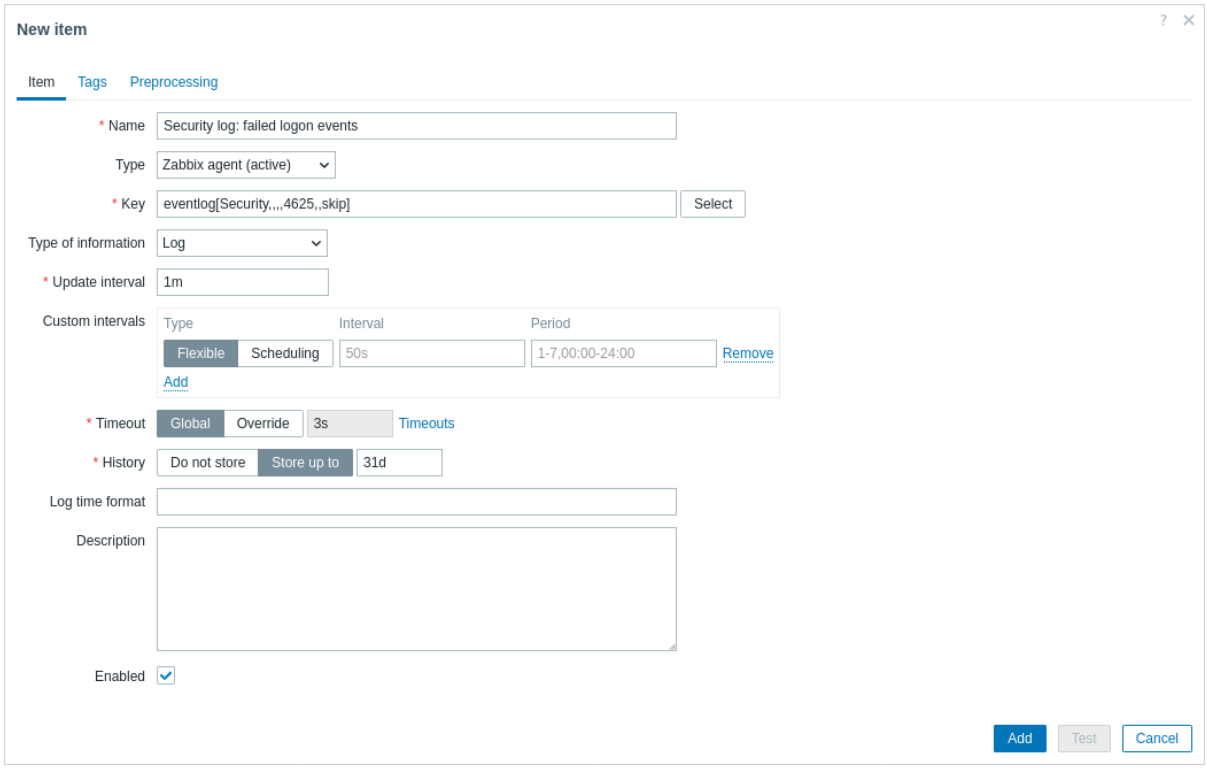
- Nel campo *Nome host*, inserire un nome host (ad esempio, "MyWindowsHost").
- Nel campo *Gruppi host*, digitare o selezionare un gruppo host (ad esempio, "Event log Monitoring").
- Premere *Aggiungi* per salvare l'host configurato.

Note:
Nel campo *Template* è possibile aggiungere il template "Windows by Zabbix agent active" per facilitare la risoluzione dei problemi, osservando se altri item attivi sullo stesso host vengono aggiornati.



2. Creare un nuovo item con i seguenti parametri:

- Nel campo *Nome*, inserire un nome descrittivo per l'item (ad esempio, "Security log: failed logon events").
- Nel menu a discesa *Tipo*, selezionare "Zabbix agent (active)" (obbligatorio per il monitoraggio dell'Event log).
- Nel campo *Chiave*, utilizzare la chiave item eventlog. Ad esempio, per monitorare i tentativi di accesso non riusciti (ID evento: 4625) nel log Security e ignorare le voci più vecchie dell'ultimo controllo dell'item (utilizzando il parametro `skip`), inserire la seguente chiave item: `eventlog[Security,,,,4625,,skip]`
- Nel menu a discesa *Tipo di informazione*, selezionare "Log".



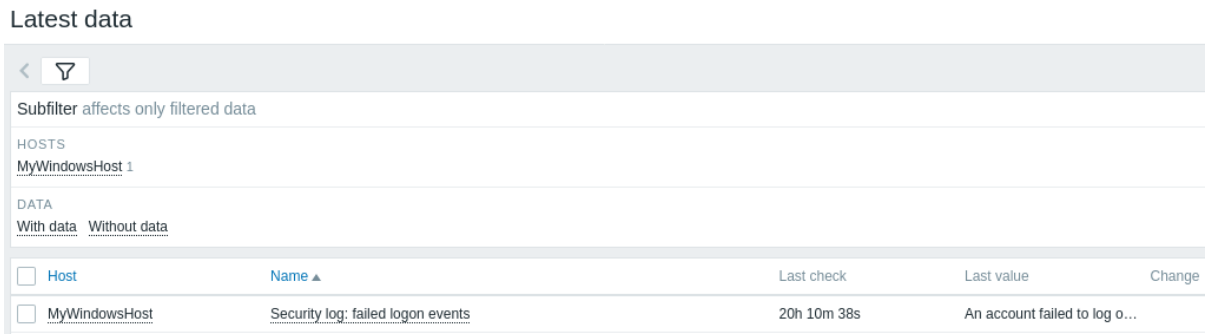
3. Fare clic su *Aggiungi* per salvare l'item.

Testare e visualizzare 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 con 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 non sono presenti, passa alla sezione Risoluzione dei problemi della guida.

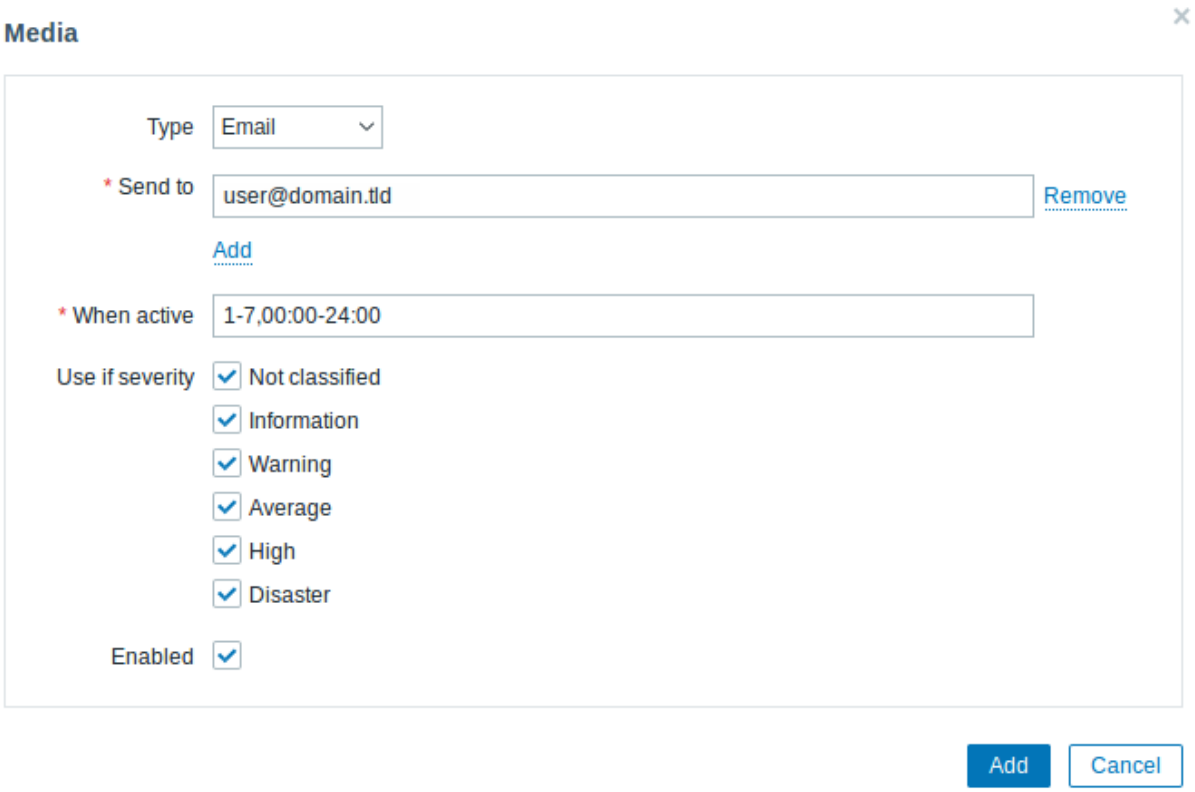
Configurare gli avvisi di problema

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

1. Vai a Data collection > Hosts per definire un trigger che si attivi quando il tuo item del log eventi registra il pattern di tuo interesse. 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 a *User settings > Profile*, 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 con la raccolta o la visualizzazione dei log eventi di Windows, utilizza 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 del log esatto (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 di 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 macchine Windows tramite Zabbix agent.
- Chiavi item specifiche per Windows - informazioni dettagliate sulle chiavi item specifiche per Windows supportate dagli agent Zabbix, 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.

Cos'è Zabbix

Zabbix è stato creato da Alexei Vladishev e attualmente è sviluppato attivamente e supportato 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 e lo stato di salute 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 via email per praticamente qualsiasi evento. Questo 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 web. Un frontend web garantisce che lo stato della rete e lo stato di salute dei server possano essere valutati da qualsiasi posizione. Se configurato correttamente, Zabbix può svolgere un ruolo importante nel monitoraggio dell'infrastruttura IT. Questo vale sia per le piccole organizzazioni con pochi server sia per le grandi aziende con una moltitudine di server.

Zabbix è gratuito. Zabbix è scritto e distribuito con licenza AGPL-3.0. Ciò significa che il suo codice sorgente è distribuito liberamente ed è disponibile al pubblico.

È disponibile il supporto commerciale, fornito da Zabbix Company e dai suoi partner in tutto il mondo.

Per saperne di più sulle funzionalità di Zabbix.

Utenti di Zabbix Molte organizzazioni di diverse dimensioni in tutto il mondo si affidano a Zabbix come piattaforma di monitoraggio principale.

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 riportano informazioni sulla disponibilità e l'integrità, nonché le statistiche. Il server è il repository centrale in cui vengono memorizzati tutti i dati di configurazione, statistici e operativi.

Archiviazione del database

Tutte le informazioni di configurazione, così come i dati raccolti da Zabbix, sono memorizzati in un database.

Interfaccia web

Per un facile accesso a Zabbix da qualsiasi luogo e da qualsiasi piattaforma, viene fornita un'interfaccia basata sul web. L'interfaccia fa parte del server Zabbix e di solito (ma non necessariamente) viene eseguita sulla stessa macchina fisica su cui è in esecuzione il server.

Proxy

Lo Zabbix proxy può raccogliere dati sulle prestazioni e sulla disponibilità per conto di Zabbix server. Un proxy è una parte opzionale di un'installazione di Zabbix; tuttavia, può essere molto utile per distribuire il carico di un singolo Zabbix server.

Agent

Gli agent Zabbix vengono distribuiti sui target di monitoraggio per monitorare attivamente le risorse locali e le applicazioni e riportare i dati raccolti al server Zabbix. 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 dare uno sguardo al flusso complessivo dei dati all'interno di Zabbix. Per creare un item che raccolga dati, è necessario prima creare un host. Spostandosi all'altro estremo dello spettro di Zabbix, per creare un trigger è necessario prima avere un item. 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 attiva se la CPU è troppo alta, seguito da un'azione che invia un'email. Anche se può sembrare un numero elevato di passaggi, con l'uso dei template in realtà non lo è. Tuttavia, grazie a questa progettazione, 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, facendo riferimento ai valori del database backend

Avvisi altamente configurabili

- l'invio delle notifiche può essere personalizzato in base alla pianificazione dell'escalation, al destinatario e al tipo di supporto
- le notifiche possono essere rese significative e utili utilizzando le variabili macro
- le azioni automatiche includono comandi remoti

Creazione di grafici in tempo reale

- gli item monitorati vengono immediatamente rappresentati graficamente utilizzando la funzionalità di creazione di grafici integrata

Funzionalità di monitoraggio web

- Zabbix può seguire un percorso di clic del mouse simulati su un sito web e verificarne la funzionalità e il tempo di risposta

Ampie opzioni di visualizzazione

- possibilità di creare grafici personalizzati che possono combinare più item in un'unica vista
- mappe di rete
- slideshow in una panoramica in stile dashboard
- report
- vista di alto livello (business) delle risorse monitorate

Archiviazione dei dati storici

- dati archiviati in un database
- storico 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 da altri template

Rilevamento della rete

- rilevamento automatico dei dispositivi di rete
- autoregistrazione dell'agent
- rilevamento di file system, interfacce di rete e OID SNMP

Interfaccia web veloce

- un frontend web basato su PHP
- accessibile da qualsiasi luogo
- è possibile navigare facilmente con un clic
- log di audit

Zabbix API

- Zabbix API fornisce un'interfaccia programmabile a Zabbix per operazioni di massa, integrazione con software di terze parti e altri scopi.

Sistema di autorizzazioni

- autenticazione utente sicura
- alcuni utenti possono essere limitati a determinate viste

agent completo e facilmente estendibile

- distribuito sugli obiettivi di monitoraggio
- può essere distribuito sia su Linux che su Windows

Demoni binari

- scritti in C, per prestazioni elevate e un ridotto utilizzo della memoria
- facilmente portabili

Pronto per ambienti complessi

- il monitoraggio remoto è semplificato grazie all'uso di uno Zabbix proxy

Glossario In questa sezione puoi apprendere il significato di alcuni termini comunemente usati in Zabbix.

host

- qualsiasi dispositivo fisico o virtuale, applicazione, servizio o qualsiasi altra raccolta logicamente correlata di parametri monitorati.

gruppo di host

- un raggruppamento logico di host. I gruppi di host vengono utilizzati quando si assegnano diritti di accesso agli host per diversi gruppi di utenti.

item

- uno specifico dato che si desidera ricevere da un host, una metrica di dati.

preelaborazione del valore

- una trasformazione del valore della metrica ricevuta prima di salvarlo 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 sono al di sopra della soglia, i trigger passano da 'Ok' a uno stato di 'Problem'. Quando i dati ricevuti sono al di sotto della soglia, i trigger rimangono in/tornano a uno stato di '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.

Il compito dei template è accelerare la distribuzione delle attività di monitoraggio su un host; inoltre rendono più semplice applicare modifiche di massa alle attività di monitoraggio. I template sono collegati direttamente ai singoli host.

gruppo di template

- un raggruppamento logico di template. I gruppi di template vengono utilizzati quando si assegnano diritti di accesso ai template per diversi gruppi di utenti.

evento

- una singola occorrenza di qualcosa che merita attenzione, come un trigger che cambia stato o il verificarsi di una discovery/autoregistrazione dell'agent.

tag evento

- un marcatore predefinito per l'evento. Può essere utilizzato nella correlazione degli eventi, nella granularità dei permessi, ecc.

correlazione degli eventi

- 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 diverso metodo di raccolta dati.

problema

- un trigger che si trova nello stato "Problem".

aggiornamento del problema

- opzioni di gestione dei problemi fornite da Zabbix, come aggiungere un commento, confermare, cambiare gravità o chiudere manualmente.

azione

- un mezzo predefinito per reagire a un evento.

Un'azione consiste in operazioni (ad es. invio di una notifica) e condizioni (*quando* l'operazione viene eseguita)

escalation

- uno scenario personalizzato per eseguire operazioni all'interno di un'azione; una sequenza di invio di notifiche/esecuzione di comandi remoti.

macro

- un segnaposto variabile che viene risolto in un valore specifico a seconda del contesto.

media

- un mezzo per recapitare notifiche; canale di consegna.

notifica

- un messaggio relativo a un evento inviato a un utente tramite il canale media scelto.

comando remoto

- un comando predefinito che viene eseguito automaticamente su un host monitorato al verificarsi di una determinata condizione.

scenario web

- una o più richieste HTTP per verificare la disponibilità di un sito web.

frontend

- l'interfaccia web fornita con Zabbix.

dashboard

- sezione personalizzabile dell'interfaccia web che visualizza riepiloghi e rappresentazioni di informazioni importanti in unità visive chiamate widget.

widget

- unità visiva che mostra informazioni di un determinato tipo e origine (un riepilogo, una mappa, un grafico, l'orologio, ecc.), utilizzata nel 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 i proxy e gli agent di Zabbix, calcola i trigger, invia notifiche; un archivio centrale dei dati.

Zabbix proxy

- un processo che può raccogliere dati per conto di Zabbix server, sottraendo parte del carico di elaborazione al server.

Zabbix agent

- un processo distribuito sui target di monitoraggio per monitorare attivamente le risorse locali e le applicazioni.

Zabbix agent 2

- una nuova generazione di Zabbix agent per monitorare attivamente risorse locali e applicazioni, consentendo l'uso di plugin personalizzati per il monitoraggio.

> **Attention:**
> Poiché Zabbix agent 2 condivide molte funzionalità con Zabbix agent, il termine "Zabbix agent" nella documentazione indica entrambi - Zabbix agent e Zabbix agent 2, se il comportamento funzionale è lo stesso. Zabbix agent 2 viene nominato specificamente solo dove la sua funzionalità differisce.

crittografia

- supporto delle comunicazioni crittografate tra i componenti Zabbix (server, proxy, agent, utility zabbix_sender e zabbix_get) tramite il protocollo Transport Layer Security (TLS).

autoregistrazione dell'agent

- processo automatizzato mediante il quale un Zabbix agent stesso viene registrato come host e il monitoraggio viene avviato.

network discovery

- discovery automatizzata dei dispositivi di rete.

low-level discovery

- discovery automatizzata di entità di basso livello su un particolare dispositivo (ad es. file system, interfacce di rete, ecc.).

regola di low-level discovery

- insieme di definizioni per la discovery automatizzata di entità di basso livello su un dispositivo.

prototipo di item

- una metrica con determinati parametri come variabili, pronta per la low-level discovery. Dopo la low-level discovery le variabili vengono automaticamente sostituite con i parametri reali rilevati e la metrica inizia automaticamente a raccogliere dati.

prototipo di trigger

- un trigger con determinati parametri come variabili, pronto per la low-level discovery. Dopo la low-level discovery le variabili vengono automaticamente sostituite con i parametri reali rilevati e il trigger inizia automaticamente a valutare i dati.

I *prototipi* di alcune altre entità Zabbix sono utilizzati anche nella low-level discovery - prototipi di grafico, prototipi di host, prototipi di gruppo di host.

Zabbix Cloud

Zabbix Cloud

Scopri le opzioni di configurazione disponibili e i controlli di accesso per organizzazioni e nodi in Zabbix Cloud.

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.

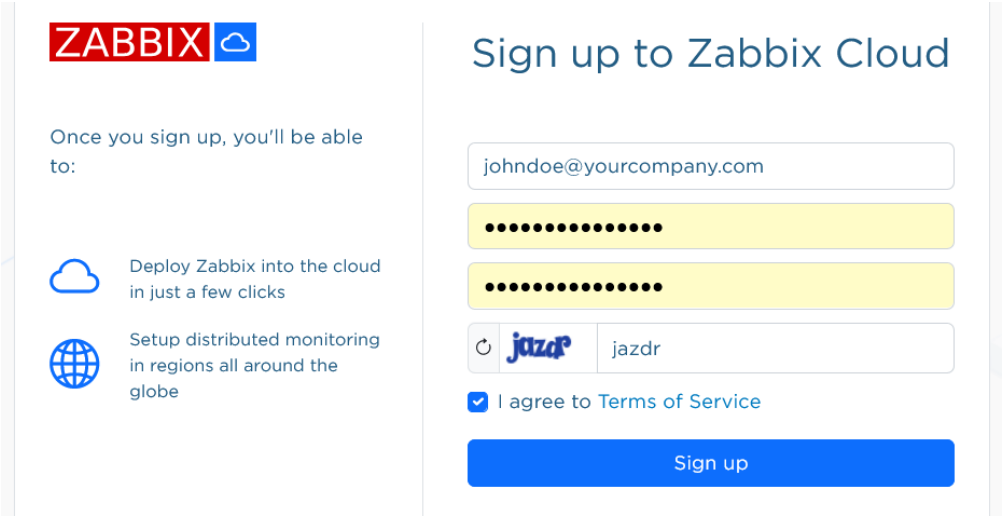
Distribuire Zabbix nel cloud

Per avviare Zabbix nel cloud, segui questa procedura:

- Accedi o registrati su cloud.zabbix.com
- Crea un nodo cloud
- Inizia a utilizzare Zabbix nel cloud

Accedi o registrati Puoi accedere su cloud.zabbix.com con un account Google, GitHub o Microsoft esistente.

In alternativa, vai alla pagina di registrazione per creare un account.



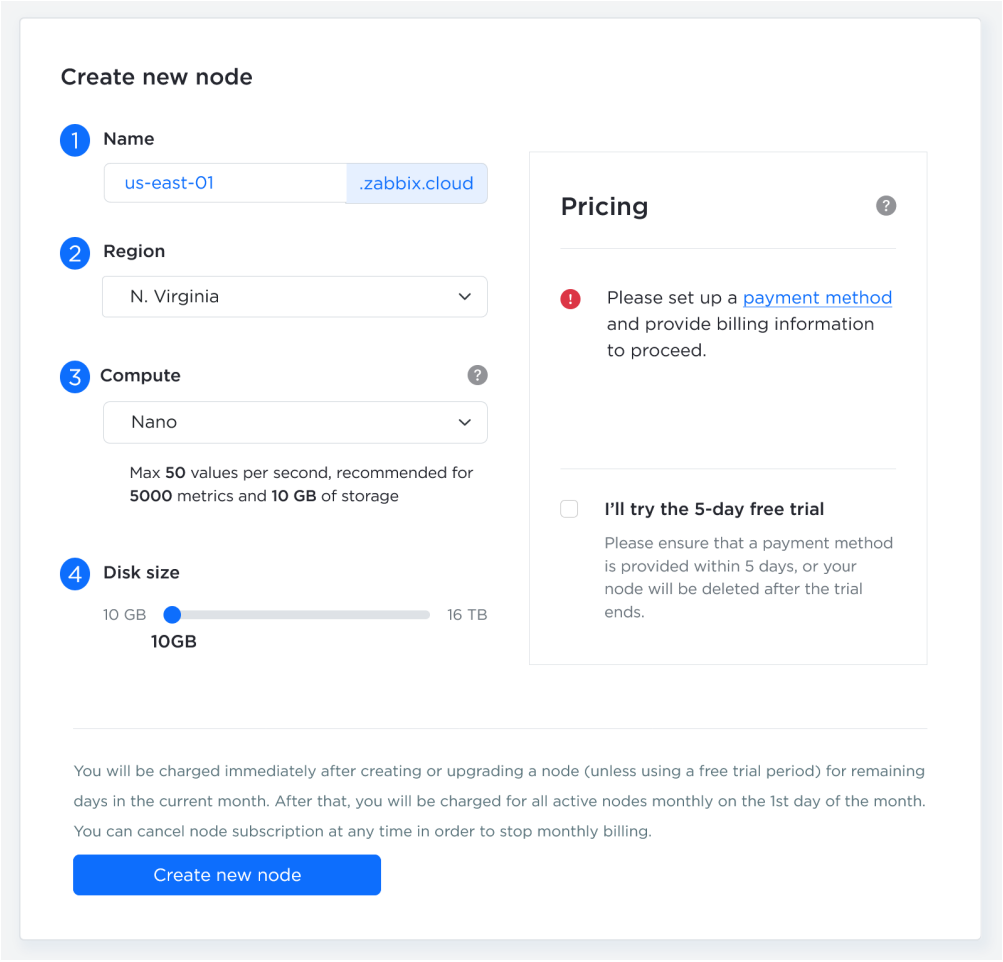
Dopo la registrazione, riceverai nell'email una conferma del tuo nuovo account.

Vai su cloud.zabbix.com per accedere. Al momento dell'accesso ti verrà richiesto un codice monouso. Questo codice ti sarà inviato via email.

Crea un nodo cloud Un nodo cloud è un'istanza di Zabbix basata nel cloud. Include il server Zabbix, il database e il web frontend.

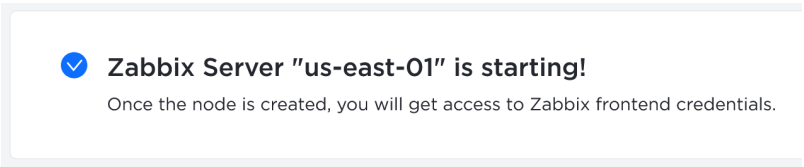
Dopo aver effettuato l'accesso, vedrai che per il tuo account esiste già per impostazione predefinita un'organizzazione *Private*. Potrai rinominare l'organizzazione predefinita in seguito.

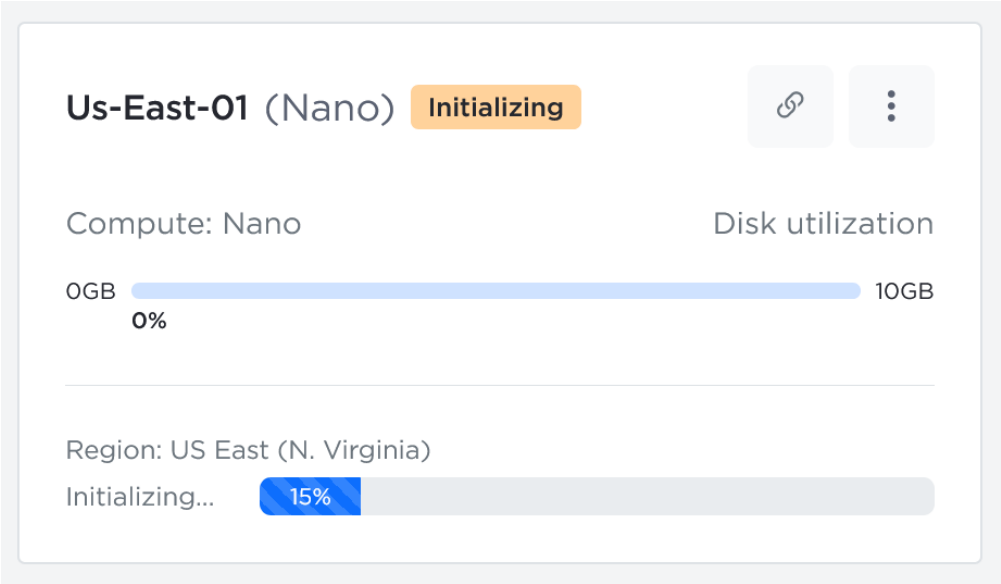
Fai clic su *Create New Node*. Si aprirà un modulo di configurazione del nodo.

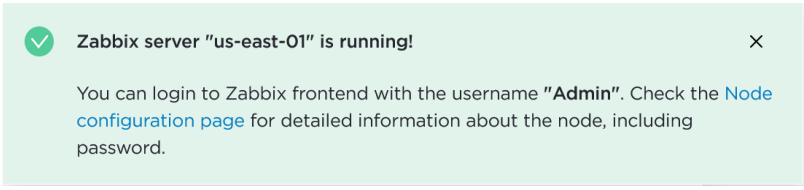


Parameter	Description
Name	Seleziona un nome univoco per il nodo. Il nome del nodo genera un URL univoco per il server/frontend cloud, nel formato `<your-name>.zabbix.cloud`.
Region	Seleziona la posizione del data center più vicina per ridurre al minimo la latenza di rete.
Compute	Seleziona la potenza di calcolo per il nodo. Per determinare quale livello di risorse ti serve, consulta i dettagli dei prezzi. Se avrai bisogno di maggiore potenza, potrai sempre effettuare un upgrade in seguito. Prima di creare un nodo non di prova sarà necessario un metodo di fatturazione. Aggiungi le informazioni di fatturazione tramite il menu *Billing*. È disponibile una **prova gratuita** di 5 giorni per testare un nodo di livello *Nano*. Durante il periodo di prova non ti verrà addebitato nulla, ma una volta terminato il periodo, il nodo verrà rimosso se non sarà convertito in un nodo a pagamento.
Disk size	Il cursore è impostato per default su un minimo consigliato in base al livello. Aumentalo se archivi grandi volumi di storico o trend. Tieni presente che i dischi più grandi comportano costi mensili più elevati e che non puoi ridurre autonomamente la dimensione di un disco: per farlo dovrai contattare il supporto.

Dopo aver completato il modulo, fai clic su *Create New Node*. Durante l'inizializzazione del nodo, viene visualizzato un messaggio corrispondente e il nodo appare già nella sezione Organizations.





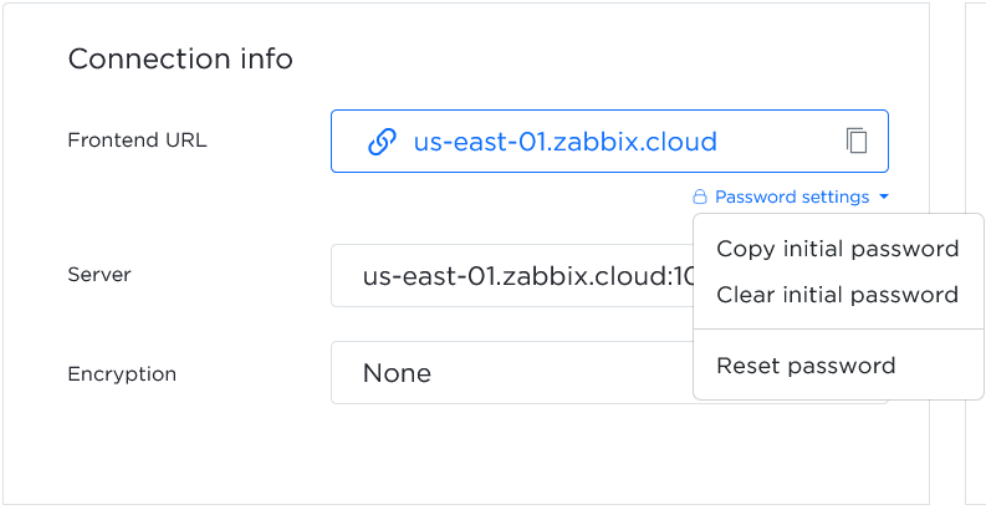


Inizia a usare Zabbix

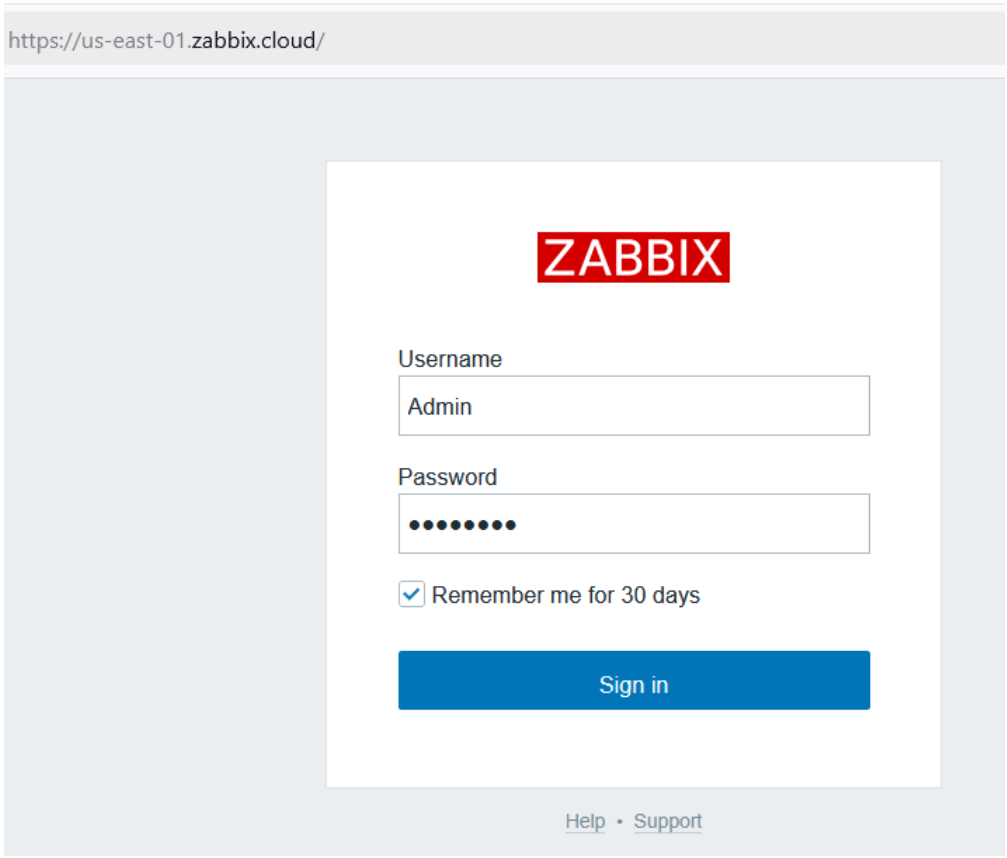
Quando lo stato del nodo cambia da *Initializing* a *Running*, puoi accedere al frontend di Zabbix.

Per accedere alla password iniziale del frontend, fai clic sul nodo creato.

Verrai reindirizzato alle impostazioni del nodo (panoramica, filtri di accesso, crittografia, backup, ecc.). In **Connection info**, fai clic su *Password settings* e seleziona *Copy initial password* dal menu a discesa.



Fai clic sull'URL del frontend per aprire il frontend di Zabbix (`us-east-01.zabbix.cloud`).



Accedi utilizzando le seguenti credenziali:

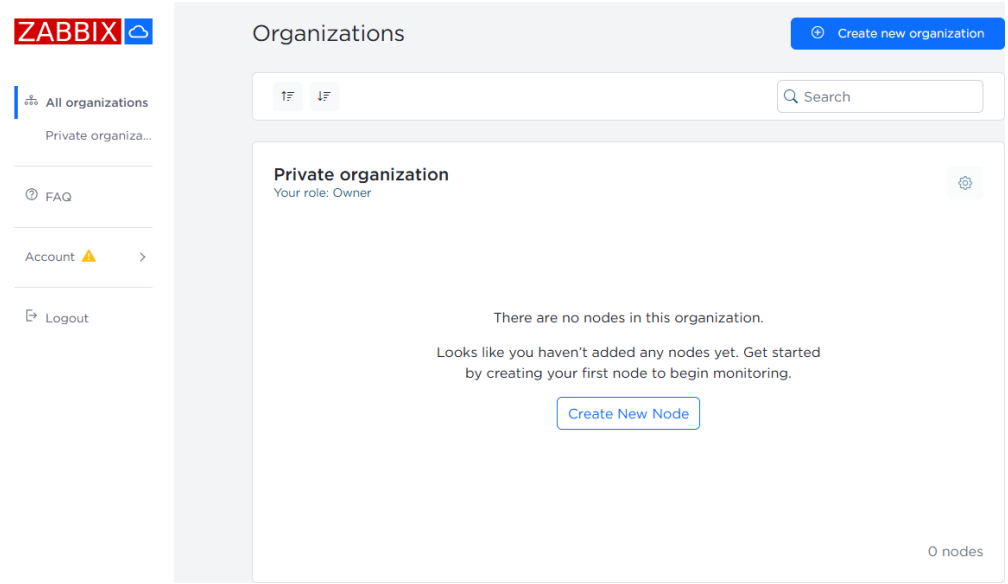
- *Username* - `Admin`
- *Password* - inserisci la password iniziale (ad esempio, HzTG9t7Y)

Per iniziare il monitoraggio, devi definire i filtri di accesso: l'indirizzo IP o gli indirizzi IP autorizzati a connettersi al server cloud di Zabbix. Deve esserne definito almeno uno.

Gestione delle organizzazioni Un'organizzazione è un contenitore per uno o più nodi. Le organizzazioni aiutano inoltre a strutturare le risorse e a 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, per il proprio account viene creata un'organizzazione privata.



In qualità di proprietario di questa organizzazione, è possibile gestire utenti, amministratori, nodi e il nome dell'organizzazione.

Fare clic sul pulsante delle impostazioni  per accedere alle opzioni di gestione.

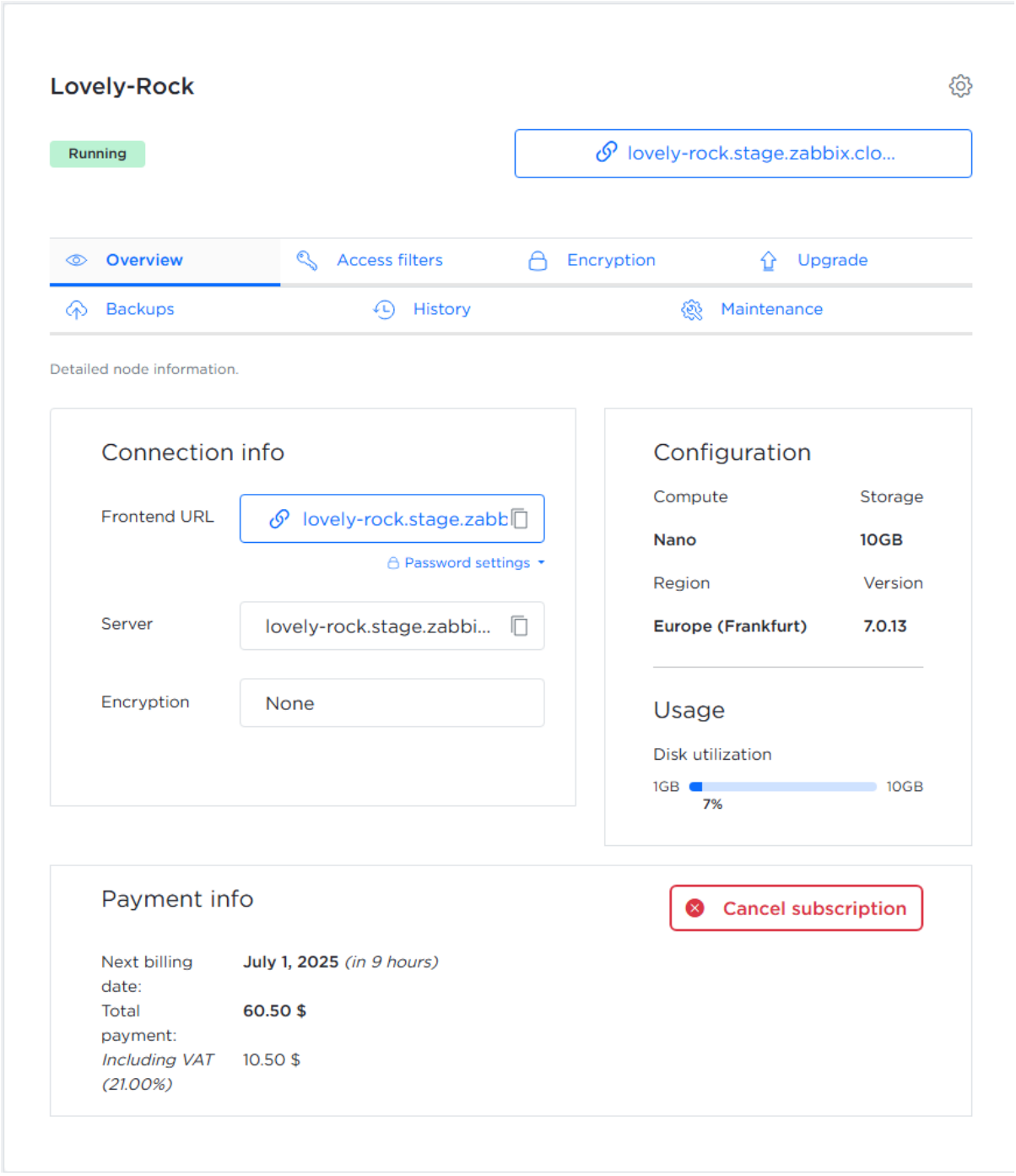
Configurazione del nodo

Panoramica

Per accedere alla pagina di configurazione del nodo, fare clic sul nodo in *Tutte le organizzazioni*, dal proprio menu dell'organizzazione oppure utilizzando il link *Pagina di configurazione del nodo* mostrato dopo che un nodo è stato avviato.

Nella vista di configurazione del nodo, ogni scheda consente di gestire un aspetto specifico del nodo:

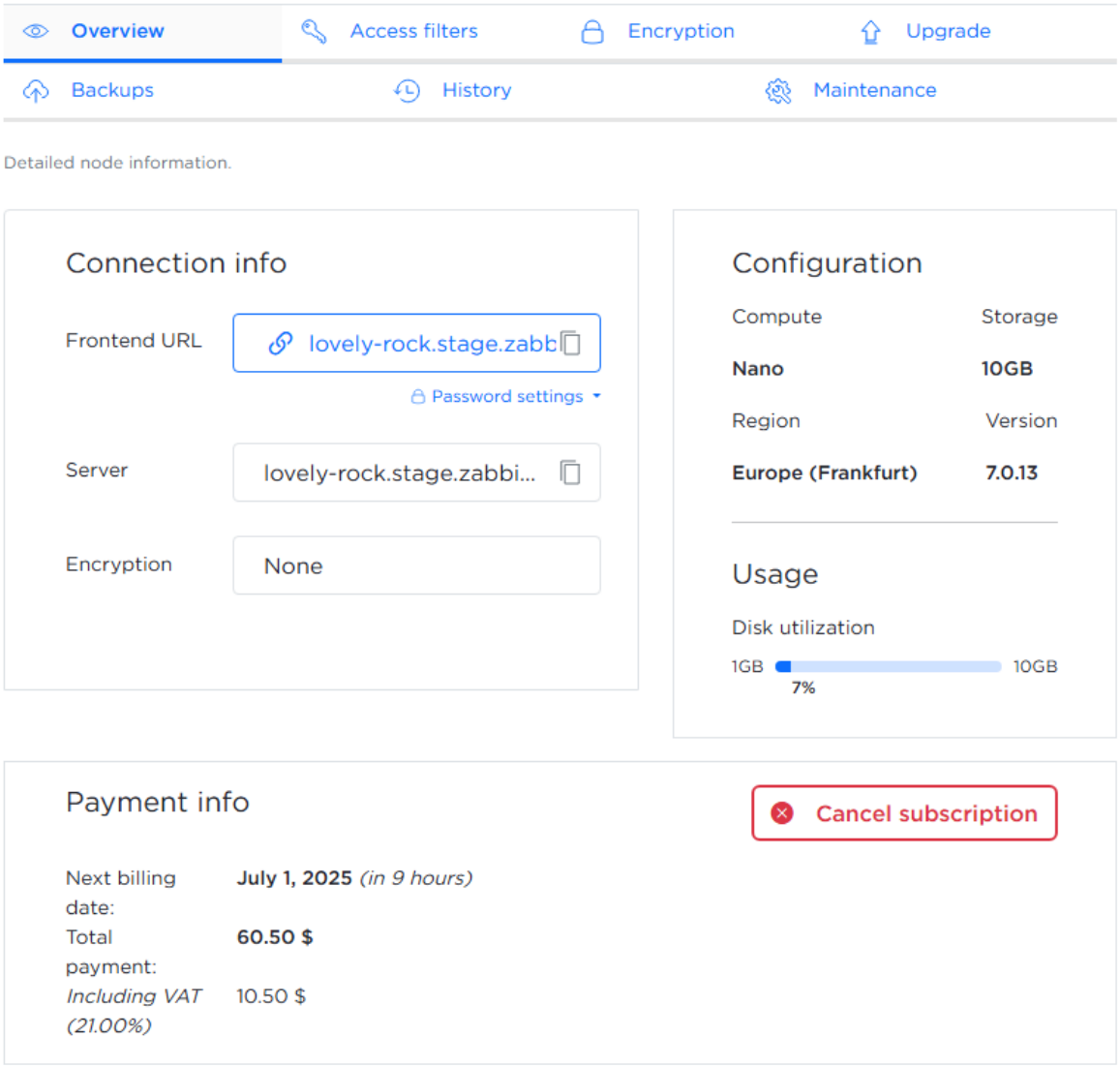
- Panoramica - visualizzare i dettagli del nodo, aprire il relativo frontend e gestire gli utenti del nodo.
- Filtri di accesso - controllare l'accesso per indirizzo IP o sottorete.
- Crittografia - configurare TLS caricando certificati, chiavi ed eventuali CRL.
- Backup - creare, ripristinare ed eliminare snapshot del nodo; visualizzare il prossimo backup pianificato.
- Cronologia - impostare i periodi di conservazione per log di audit, cronologia e trend, con unità predefinite e un limite di aggiornamento di 3 volte al giorno.
- Manutenzione - definire finestre di manutenzione settimanali in base a fuso orario, giorno e ora.
- Aggiornamento - aumentare le risorse di calcolo/archiviazione o gestire lo stato della prova/del pagamento.



Note:
Alcune schede potrebbero non essere disponibili a seconda del tipo di utente che visualizza la pagina e dello stato del nodo (prova/non prova).

Scheda Panoramica

La scheda *Panoramica* fornisce informazioni dettagliate sul nodo.



È suddivisa in tre sezioni:

Informazioni di connessione:

- URL del frontend cliccabile con opzione di copia negli appunti
- Impostazioni della password
- Metodo di crittografia corrente

Configurazione e utilizzo:

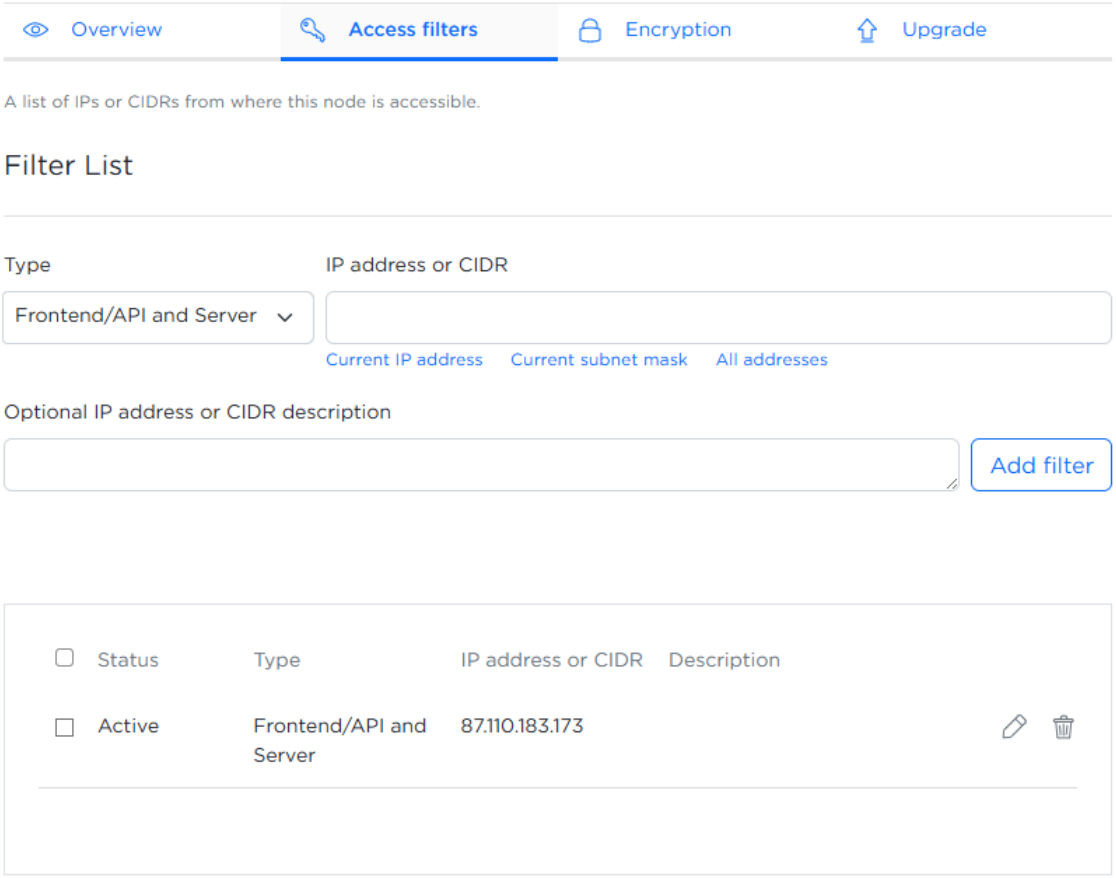
- Livello di calcolo, storage massimo, regione
- Versione dei componenti di Zabbix
- Utilizzo del disco

Informazioni sul nodo/Informazioni di pagamento:

- Data di scadenza (per i nodi di prova e prepagati)
- Dopo che il nodo ha completato l'inizializzazione, viene visualizzato un pulsante specifico in base al tipo di nodo: - *Passa a pagamento* - disponibile per i nodi in prova gratuita - *Annulla abbonamento* - disponibile per i nodi a pagamento con un abbonamento attivo - *Riattiva abbonamento a pagamento* - mostrato se un abbonamento è stato annullato - Per i nodi prepagati non viene visualizzato alcun pulsante di azione, poiché hanno una lunga validità e non possono essere annullati

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* - utilizzare il menu a discesa per scegliere quale componente di Zabbix limitare: solo frontend/API, solo server oppure entrambi.
- *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 ciascun IP o subnet.

Dopo aver compilato i campi, fare clic su *Aggiungi filtro*.
I filtri aggiunti vengono visualizzati nella tabella sottostante, con stato, tipo, indirizzo IP o subnet e 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 da una rete diversi, è necessario aggiungere qui tale indirizzo, altrimenti l'accesso verrà bloccato.

È possibile:

- - modificare un filtro.
- - eliminare un filtro oppure più filtri utilizzando le caselle di controllo.

Scheda Crittografia

La scheda *Crittografia* consente di configurare la crittografia TLS tra i componenti di Zabbix caricando i file di certificato richiesti.

Per abilitare la comunicazione sicura tramite certificati, selezionare *CERTIFICATE* e caricare dal proprio computer i seguenti file:

- *Certificati Root CA* - utilizzati per verificare l'autenticità dei certificati delle controparti.
  Fare clic su *Scegli file* e selezionare il file contenente i certificati Root CA attendibili (in genere un file `.crt` o `.pem`, codificato PEM).

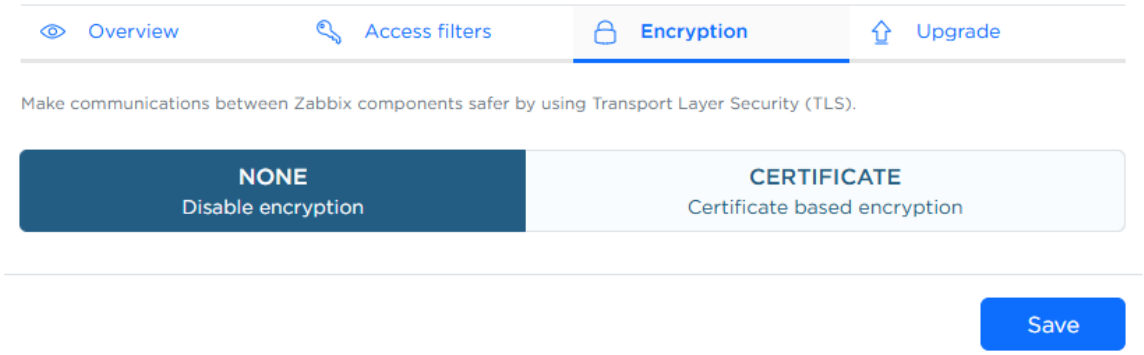
- *Catena di certificati* - rappresenta la catena di attendibilità del proprio certificato.
  Fare clic su *Scegli file* e selezionare il file codificato PEM (ad esempio `.crt` o `.pem`) che include il proprio certificato ed

eventuali certificati CA intermedi.

- *Chiave privata* - la chiave privata corrispondente al proprio certificato.
  Fare clic su *Scegli file* e selezionare il file `.key` codificato PEM.

- *Certificati revocati* (facoltativo) - se si gestisce un elenco di revoca dei certificati (CRL), fare clic su *Scegli file* e selezionare il file CRL per specificare i certificati che non devono più essere considerati attendibili.

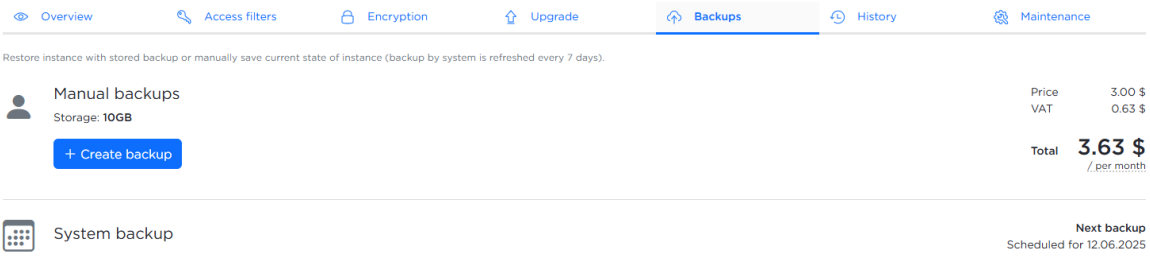
Dopo aver selezionato i file, fare clic su *Salva*. I file verranno caricati, convalidati e applicati per configurare la crittografia TLS tra i componenti di Zabbix.



Scheda Backup

La scheda *Backup* consente di visualizzare, creare, ripristinare ed eliminare i backup del tuo nodo Zabbix.

Puoi ripristinare la tua istanza da qualsiasi backup salvato oppure creare un nuovo backup manuale in qualsiasi momento. La scheda mostra anche la data del prossimo backup pianificato.



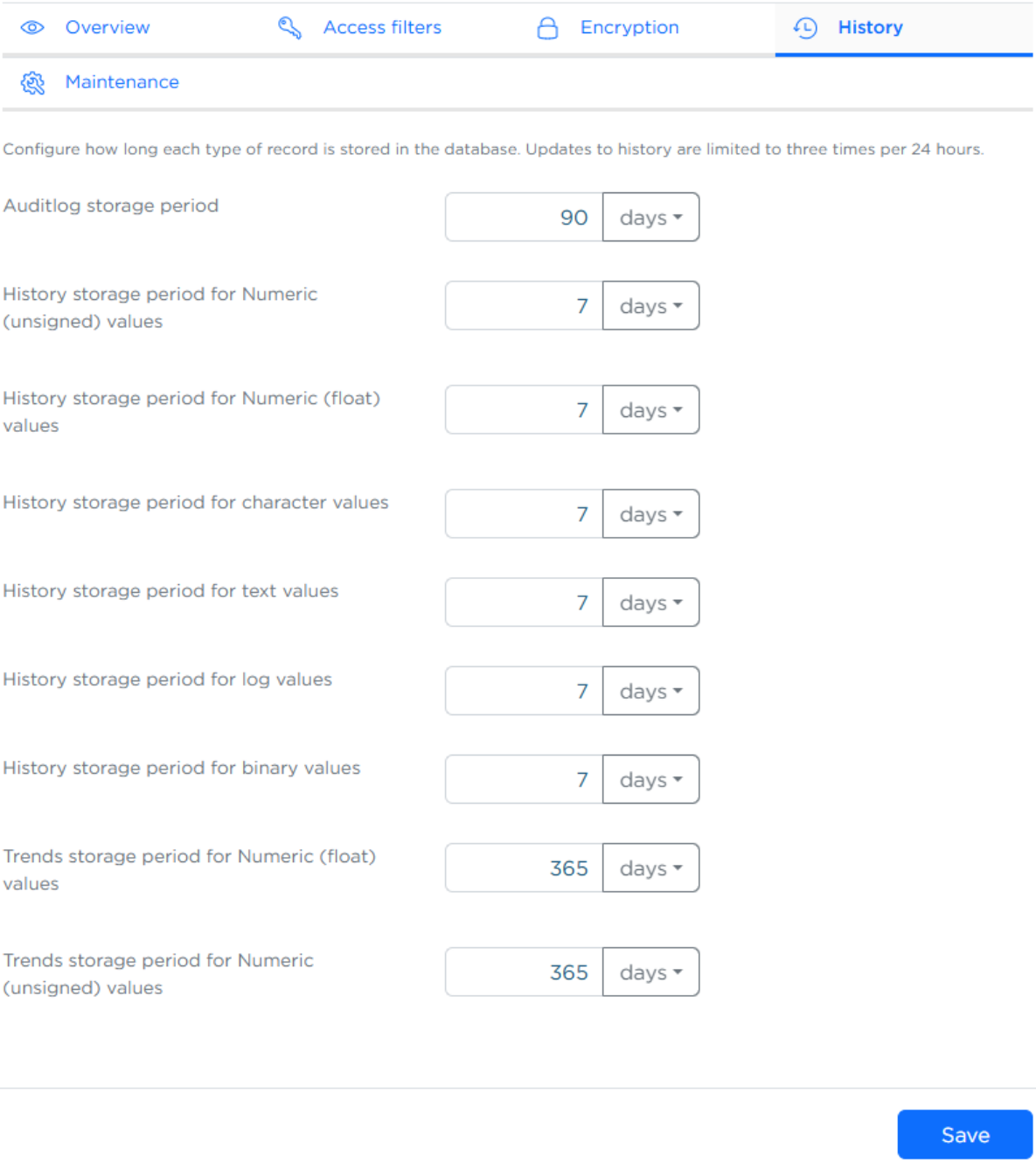
Per creare un backup:

1. Fai clic sul pulsante *Create backup*.
2. Inserisci una descrizione facoltativa nella finestra popup.
3. Fai clic su *Create* per confermare.

Warning:
I backup aggiuntivi vengono addebitati mensilmente in base allo spazio di archiviazione utilizzato.

Scheda Cronologia

La scheda *Cronologia* consente di configurare per quanto tempo vengono archiviati i diversi tipi di dati storici.



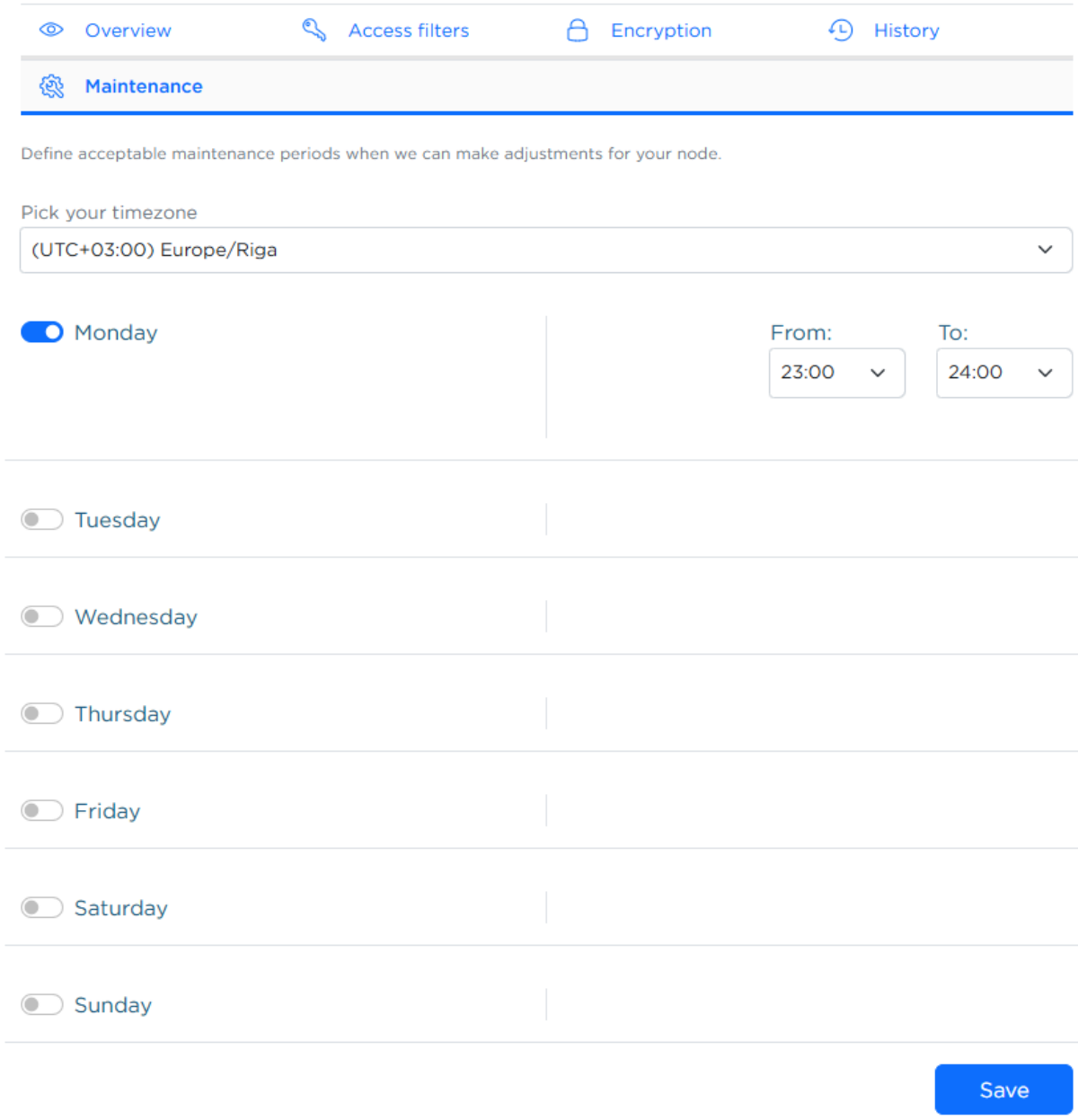
Puoi impostare il periodo di archiviazione per ciascun tipo di dato (ad esempio, numerico, testo, log) e scegliere l'unità di misura (giorni, mesi, anni). Le modifiche possono essere effettuate fino a tre volte ogni 24 ore.

Dopo aver aggiornato i valori, fai 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 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 rapide e una gestione della conservazione più prevedibile.

Scheda Manutenzione

La scheda *Manutenzione* consente di definire finestre di manutenzione accettabili durante le quali è possibile eseguire aggiornamenti o modifiche sul nodo.

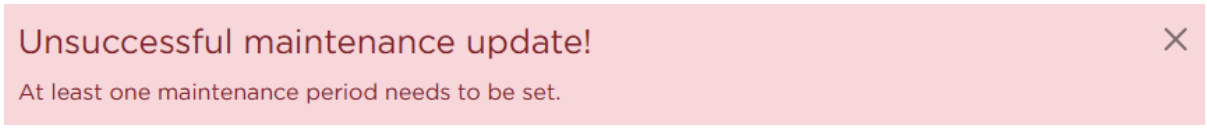


È necessario:

- Selezionare il proprio fuso orario dal menu a discesa.
- Scegliere almeno un giorno e una fascia oraria per la manutenzione.

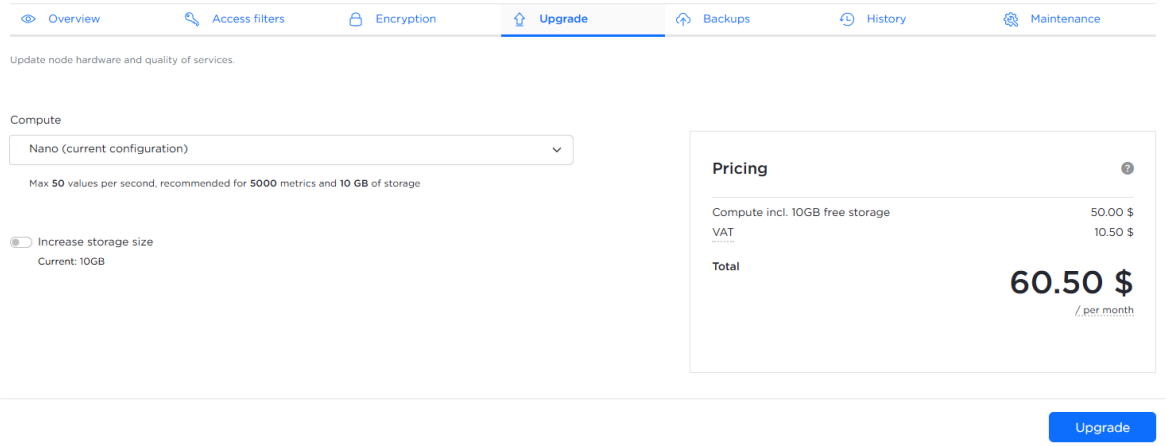
Fare clic su *Salva* per applicare le modifiche.

Se non viene selezionato alcun giorno o alcuna fascia oraria, verrà visualizzato un errore.



Scheda Upgrade

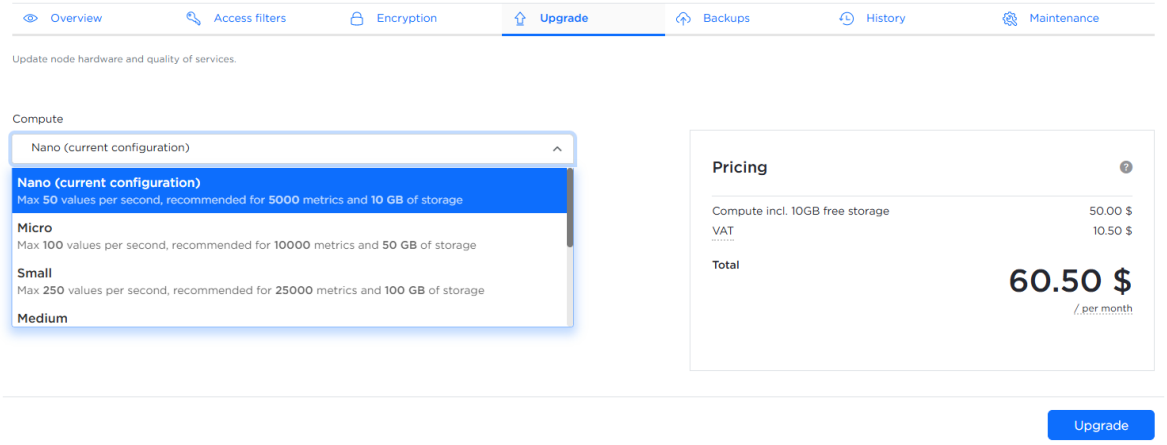
La scheda *Upgrade* consente di aumentare le risorse del nodo o gestirne l'abbonamento.



È possibile aumentare il livello di calcolo o la dimensione del disco: si noti che la dimensione del disco può essere aumentata solo una volta ogni 24 ore e non può essere ridotta 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 calcolo:

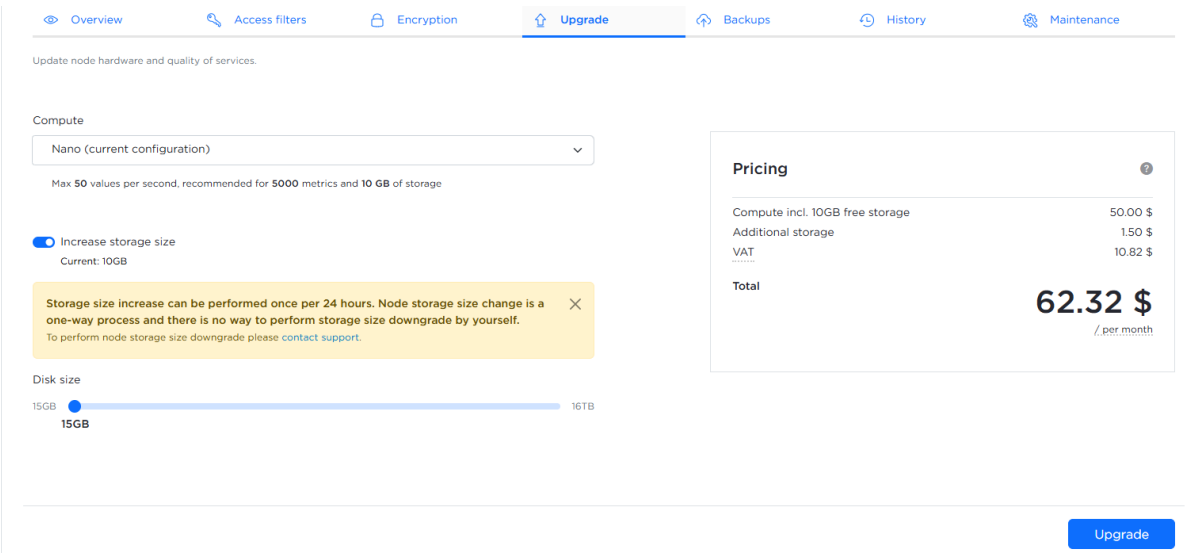
1. Fare clic sul menu a discesa *Compute*.



2. Selezionare il livello di configurazione desiderato.
3. Fare clic sul pulsante *Upgrade*.

Per aumentare la dimensione del disco:

1. Fare clic sul pulsante di opzione *Increase storage size*.
2. Regolare il cursore *Disk size* sul valore desiderato.



3. Fare clic sul pulsante *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.

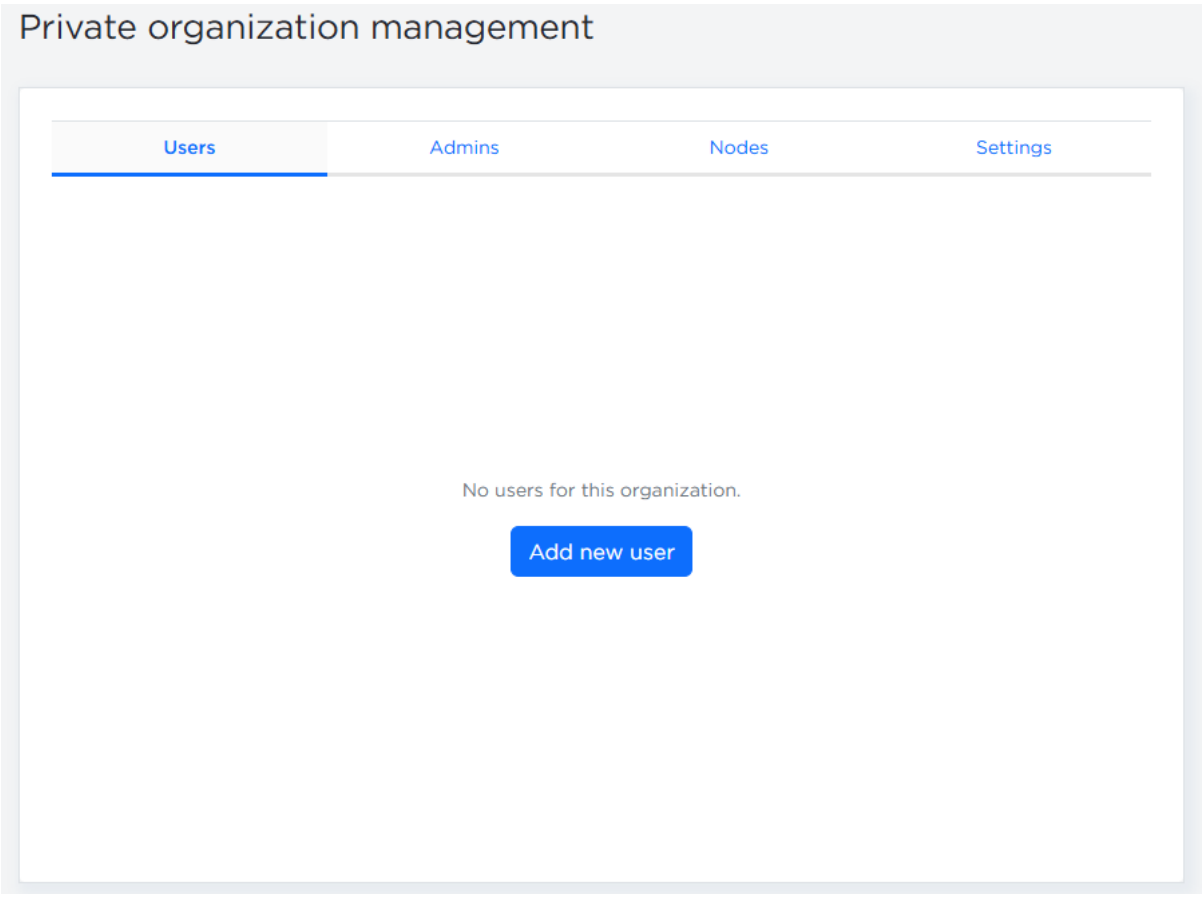
Aggiunta di utenti alle organizzazioni

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*.

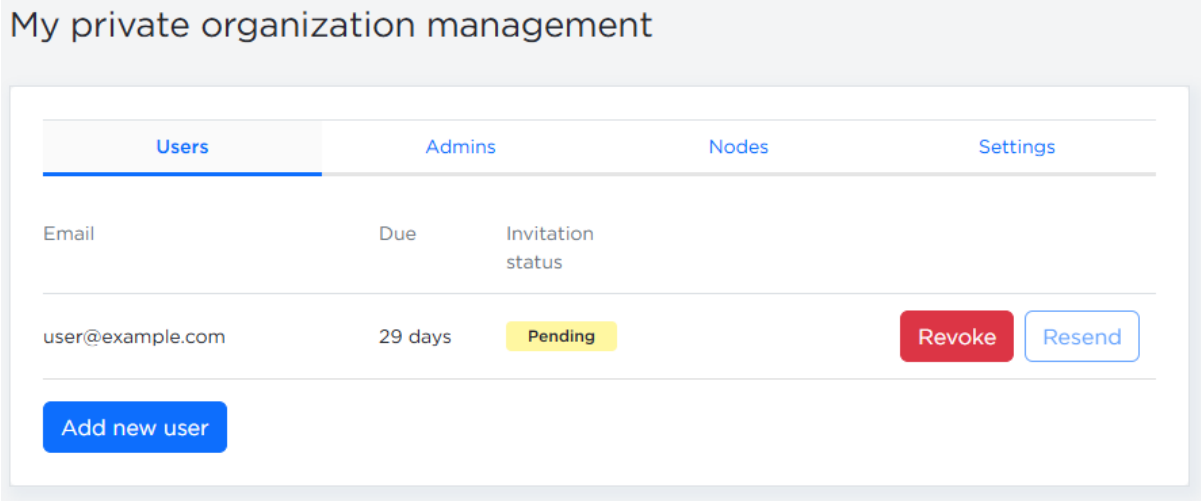
Tieni presente che selezionando il ruolo *Utente* l'utente viene aggiunto all'organizzazione, ma non gli viene concesso l'accesso ad alcun nodo.

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.

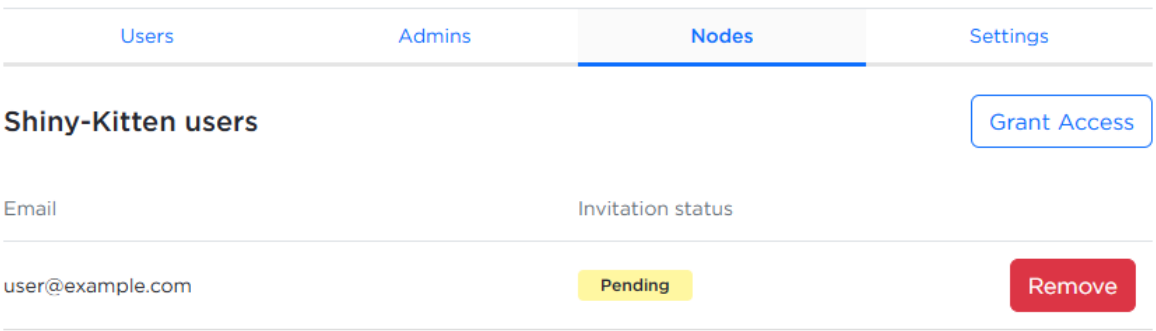
L'utente invitato verrà aggiunto con stato *In sospeso*. Puoi revocare l'invito o inviarlo nuovamente (il reinvio è disponibile una volta ogni 10 minuti: il pulsante sarà disabilitato se questo limite viene raggiunto).



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

Tabella comparativa Sebbene gran parte delle funzionalità si sovrapponga tra le installazioni Zabbix on-premise e i nodi cloud, esistono anche delle differenze. Di seguito è riportata una tabella comparativa riassuntiva. Le sezioni successive descrivono in dettaglio alcuni item selezionati.

Funzionalità / Aspetto	Zabbix Cloud	Zabbix on-premises
Ciclo di rilascio	Allineato alle release LTS	Scegli tu 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 di sicurezza	Patch automatiche di OS/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; nessun DELETE di lunga durata	Partizionamento e pulizia manuali (possibili transazioni DELETE lunghe)
Ottimizzazione delle prestazioni	Preconfigurata dal team Zabbix Cloud (nessun changelog)	Visibilità e controllo completi sui parametri di ottimizzazione
Controllo del firewall	Gestito dal team Zabbix Cloud; le porte GUI/trapper non sono esposte pubblicamente	Controllo completo delle regole del firewall e dell'esposizione delle porte
Isolamento e protezione del DB	Elevato isolamento, nessun accesso diretto a DB/SSH	Dipende dalla tua rete e dalla sicurezza del host
Alta disponibilità (HA)	Eseguito su una piattaforma cloud ad alta disponibilità, i proxy Zabbix possono essere utilizzati per una maggiore resilienza della raccolta dati	HA gestita dall'utente
Trap SNMP	Solo tramite proxy dedicato (nessun ricevitore trap HA/bilanciato)	Direttamente o tramite proxy, con possibilità di HA/bilanciamento del carico
Polling SNMP	Difficile senza proxy (ogni dispositivo richiede NAT/porta personalizzata)	Polling SNMP nativo; proxy opzionale
Impostazioni di conservazione	History/Trends/Audit solo tramite UI (nessuna API o file di configurazione)	Configurabile nel file di configurazione del server Zabbix o tramite API
Script personalizzati	`AlertScriptsPath`, `ExternalScripts`, frontend e moduli della community non supportati	Completamente supportati (percorso degli script, moduli, integrazioni)
Monitoraggio ODBC	Solo PostgreSQL (driver `{postgresql}`); plugin MariaDB presente come segnaposto	ODBC per PostgreSQL, MySQL, Oracle, ecc. - completamente configurabile
Limitazione delle chiamate ODBC	Non possibile (`StartODBCPollers=1` soltanto; query sincrone pesanti bloccano altre attività)	Numero di poller e pianificazione completamente ottimizzabili
Caricamento del certificato SAML	Non supportato	Supportato tramite UI o API
Report pianificati	È necessario creare un proprio supporto Email (supporto script non supportato)	Supporta sia il supporto script sia Email pronti all'uso
Interfaccia del host per i controlli attivi	Indirizzo IP emesso dalla piattaforma (creato automaticamente)	Interfacce gestite da te; IP sotto il tuo 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.

Interfaccia per controlli attivi

I controlli attivi nel cloud creano automaticamente un'interfaccia host con un IP non correlato alla rete on-premise. Per impostazione predefinita, questo IP può essere esterno. Per mantenere la coerenza, è possibile 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 supporto predefinito *Cloud Email* utilizza un trasporto tramite script e non può essere usato per i report pianificati. Per inviare report tramite email:

1. Creare un nuovo tipo di supporto Email.
2. Assegnare il supporto email al proprio utente in *Impostazioni utente* > *Supporti*.
3. Selezionare quel tipo di supporto durante la configurazione dei report pianificati.

Centro sviluppatori

Centro sviluppatori

Inizia a sviluppare estensioni personalizzate per Zabbix con guide su moduli, widget e plugin.

Sviluppare su Zabbix

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 integrare sistemi esterni.

Plugin

Panoramica I plugin caricabili personalizzati estendono le funzionalità di Zabbix Agent 2. Sono compilati separatamente, ma utilizzano un pacchetto condiviso con Zabbix Agent 2.

Ogni plugin è un pacchetto *Go* che definisce la struttura e implementa una o più interfacce del plugin (*Exporter*, *Configurator*, *Runner*).

Vai a:

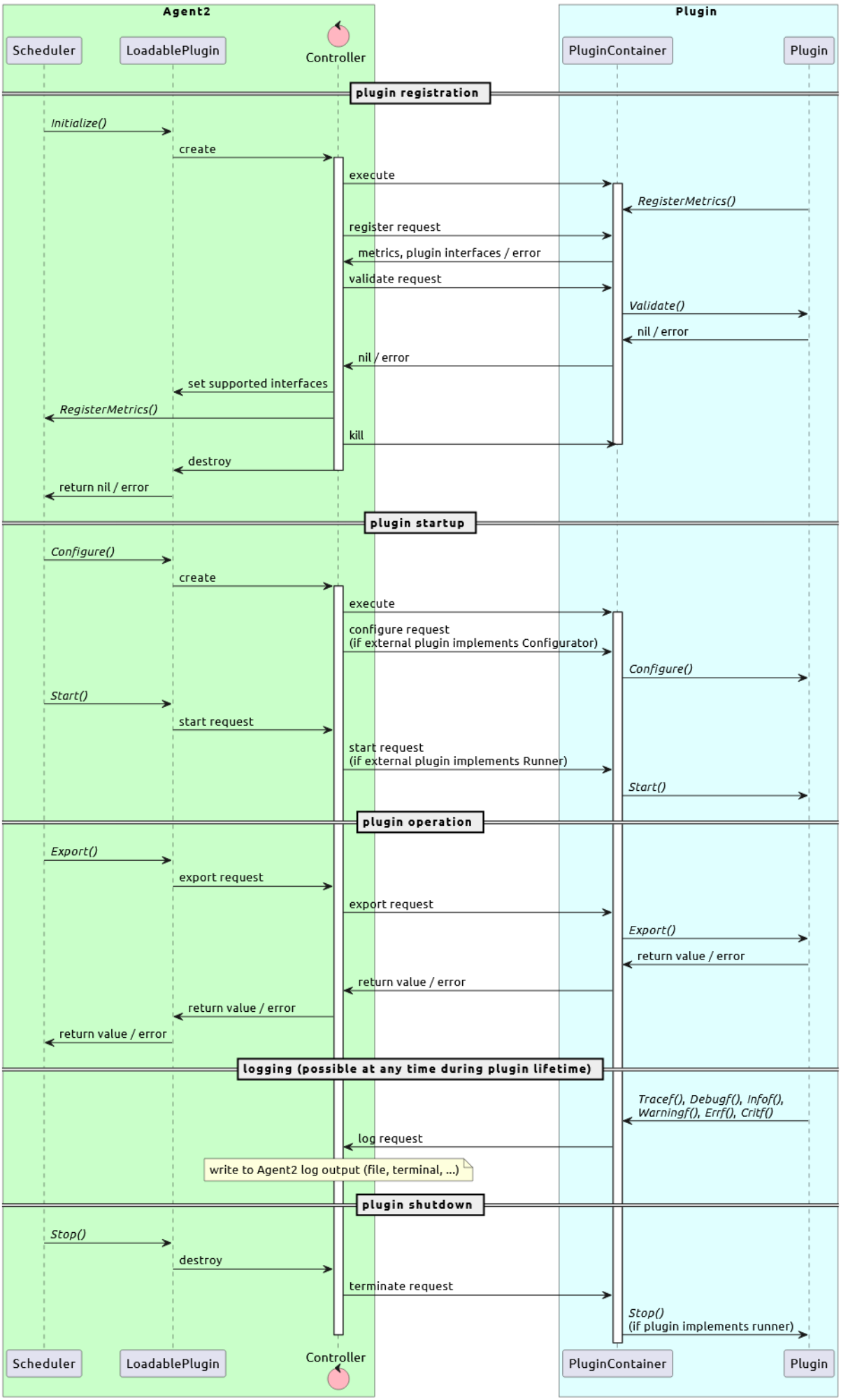
- Scrivi il tuo primo plugin
- Interfacce del plugin

Vedi anche:

- Plugin di esempio per Zabbix Agent 2

Diagramma di connessione Zabbix Agent 2 si connette bidirezionalmente ai plugin utilizzando socket UNIX su Linux e Named Pipes su Windows.

Il seguente diagramma di connessione 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

What you'll create This is ste-by-step tutorial for creating a simple loadable plugin named **MyIP**. This plugin will implement a single item key (`myip`) returning the external IP address of the Zabbix agent host.

Step 1: Setup

1. A plugin is a standard Go module. Start by initializing the `go.mod` file in the plugin directory to track plugin dependencies:

```
cd path/to/plugins/myip # Switch to your plugin directory
go mod init myip
```

2. Install the mandatory dependency Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Replace `$LATEST_COMMIT_HASH` with the latest `HEAD` commit hash from the `golang.zabbix.com/sdk` repository in the appropriate release branch. For example:

```
go get golang.zabbix.com/sdk@af85407
```

Note that `golang.zabbix.com/sdk` versioning is currently not supported, but this may change in the future.

Additional dependencies can be installed as needed using `go get`.

3. Create an empty `main.go` file for the plugin source code:

```
touch main.go
```

Now the initial setup is complete, and the plugin is ready for development.

Step 2: Plugin structure The `golang.zabbix.com/sdk` module, installed in the previous step, provides the necessary framework for plugin development and ensures all plugins have a consistent structure.

1. Set up basic execution flow.

Start by defining the main execution flow of the plugin. Add the following code to `main.go`:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

This establishes the basic execution flow for the plugin. The run function will later contain the core logic of the plugin.

2. Explore the plugin interfaces.

A Zabbix agent 2 plugin shall be represented by a struct that implements interfaces from the `golang.zabbix.com/sdk/plugin` package:

- *Accessor* - defines essential methods all plugins must implement, such as setting the plugin name and handling item key timeouts.
- One or more of the following functional plugin interfaces:
  - *Exporter* - performs a poll and returns a value (or values), nothing, or an error; often used alongside the *Collector* interface.
  - *Collector* - manages the periodic collection of data.
  - *Runner* - defines plugin startup and shutdown procedures.
  - *Watcher* - allows to implement independent metric polling, bypassing the agent's internal scheduler; useful for trap-based or event-driven monitoring.
  - *Configurator* - defines how the plugin reads and applies its configuration settings.

You can either implement these interfaces yourself or use the default ones provided by the Zabbix Go SDK, modifying them as needed. This tutorial uses the default implementations.

3. Create the plugin struct.

Now, import the *plugin* package and create a `myIP` struct that embeds the `plugin.Base` struct:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

The myIP struct currently satisfies the Accessor interface. A method for implementing one of the functional plugin interfaces, the `Exporter`, will be added later in the tutorial.

Step 3: Define item keys Your plugin needs the item keys to collect data and provide it to Zabbix server or proxy.

1. Import *errs* package for error handling:

```
import "golang.zabbix.com/sdk/errs"
```

2. Register item keys using the `plugin.RegisterMetrics()` function within the `run()` function:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

To register several item keys, repeat the parameters *metric name* and *description* for each metric. For example:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

Step 4: Set up the handler The handler facilitates communication between the agent and the plugin.

1. Import the *container* package:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Indise the `run()` function add code to create and set up a handler:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                          // Plugin name
        "myip",                          // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }
```

```
    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request is received from the agent.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}
```

Step 5: Implement data collection Data collection is done via the Exporter interface, which describes the `Export` method:

```
func Export(
  key string,             // The item key to collect.
  params []string,        // Arguments passed to the item key (`myip[arg1, arg2]`).
  context ContextProvider // Metadata about the item key data collection.
) (any, error)
```

1. Import the required packages for HTTP requests and response reading:

```
import (
    "io"
    "net/http"
)
```

2. Implement the `Export` method for the `myIP` struct:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Step 6: Build and configure the plugin

1. To build the plugin, run:

```
go mod tidy
go build
```

This should create an executable `myip` in the current directory.

2. Configure Zabbix agent 2 to use the plugin:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Replace `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` with the path to the `myip` created in step 5.

The plugin name in the configuration parameter name (*MyIP* in this tutorial) must match the plugin name defined in the *plugin.RegisterMetrics()* function.

3. To test the plugin and its `myip` item, run:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

The output should contain an external IP address of your host and look similar to this:

```
myip                                            [s|192.0.2.0]
```

With that, you have created a simple loadable plugin for Zabbix agent 2. Congrats!

```go
package main

import (
    "io"
    "net/http"

    "golang.zabbix.com/sdk/errs"
    "golang.zabbix.com/sdk/plugin"
    "golang.zabbix.com/sdk/plugin/container"
)

var _ plugin.Exporter = (*myIP)(nil)

type myIP struct {
    plugin.Base
}

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Create a new handler.
```

```
    h, err := container.NewHandler("MyIP") // Plugin name
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Setup logging to forward logs from the plugin to the agent.
    // Available via p.Logger.Infof, p.Logger.Debugf, etc.
    p.Logger = h

    // Start plugin execution.
    // Blocks until a termination request from the agent is received.
    err = h.Execute()
    if err != nil {
        return errs.Wrap(err, "failed to execute plugin handler")
    }

    return nil
}

func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // The plugin can use different data collection logic based on the `key` parameter.
    // This implementation only verifies that the provided `key` is supported.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // The log will get forwarded to the agent 2 log.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Collect the data and return it.

    resp, err := http.Get("https://api.ipify.org")
    if err != nil {
        return nil, errs.Wrap(err, "failed to get IP address")
    }

    defer resp.Body.Close()

    body, err := io.ReadAll(resp.Body)
    if err != nil {
        return nil, errs.Wrap(err, "failed to read response body")
    }

    return string(body), nil
}
```

Complete source code

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

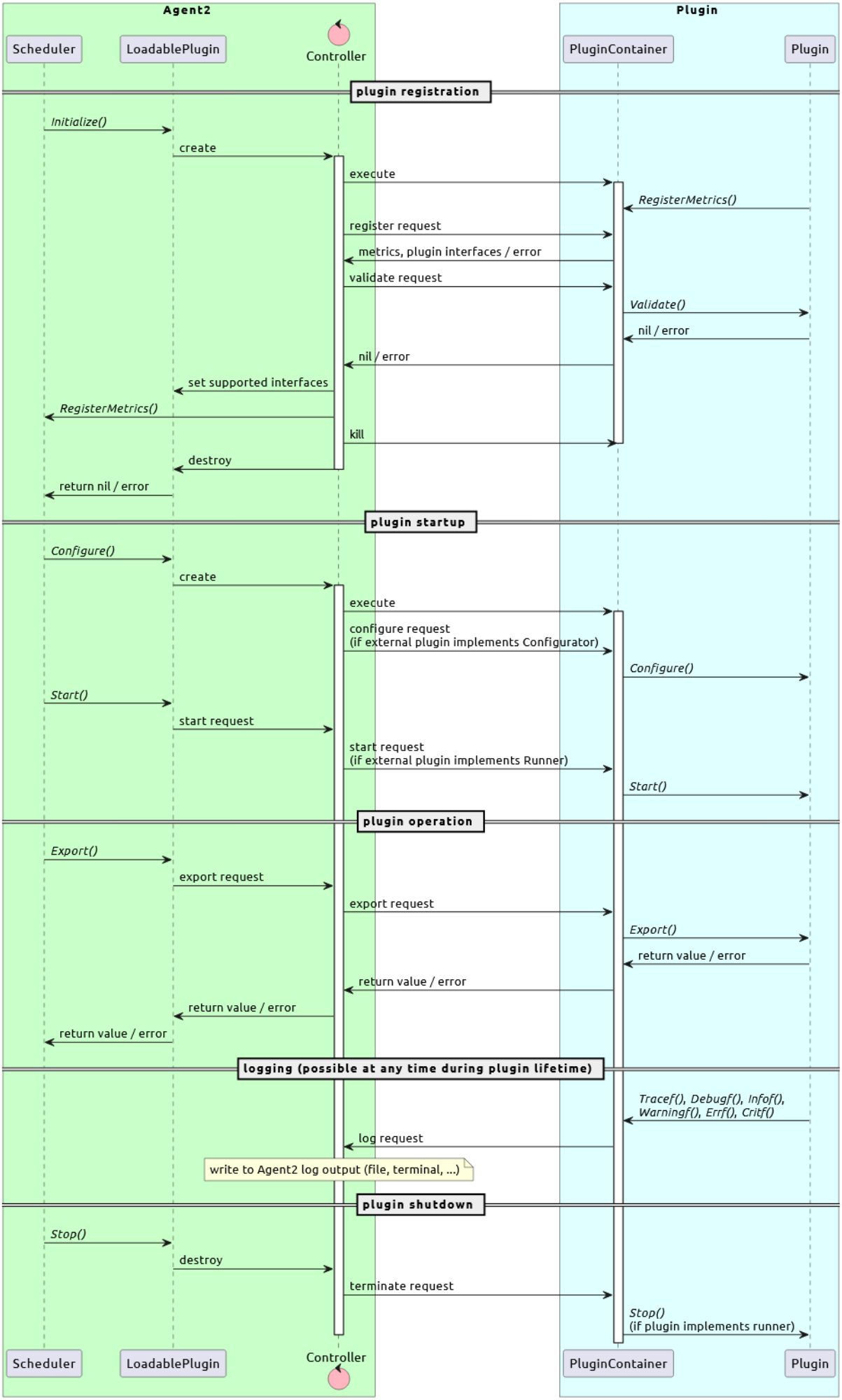
nessun metodo può essere chiamato se un plugin sta già eseguendo un'attività. Inoltre, esiste un limite massimo di 100 chiamate simultanee *Export()* per plugin, che può essere ridotto in base ai requisiti di ciascun plugin.

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.

Connection diagram Zabbix agent 2 connects bidirectionally to the plugins using UNIX sockets on Linux and Named Pipes on Windows.

The connection diagram below illustrates the communication process between Zabbix agent 2 and a loadable plugin and the metrics collection process.


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

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 di `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='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

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 il `Getter` utilizzerà durante la connessione a Zabbix agent:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Utilizzo del timeout

Puoi impostare un `timeout` di risposta per `Getter` per controllare per quanto tempo il tuo script deve attendere una risposta da Zabbix agent prima di rinunciare:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

Utilizzo della crittografia

Getter 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
    # ...

agent = Getter(
    host='192.0.2.0',
    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 al tuo script di raccogliere valori senza attendere che ciascuno arrivi. Questo può rendere il tuo script più efficiente quando deve raccogliere molti valori o quando alcuni valori richiedono molto tempo per essere raccolti.

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 AsyncGetter invece di Getter.
- Scrivi il tuo codice all'interno di una funzione async.
- Usa await quando chiami il metodo get().

Ad esempio, per raccogliere un singolo valore utilizzando la modalità asincrona:

```
### 1. Import asyncio for asynchronous mode, and AsyncGetter from zabbix_utils:
import asyncio
from zabbix_utils import AsyncGetter

### 2. Define the main async function where all data requests will be executed:
async def main():
    agent = AsyncGetter(host='192.0.2.0', port=10050)

    # 3. Fetch the system.uname value from Zabbix agent (must await):
    response = await agent.get('system.uname')

    # 4. Print the value returned by Zabbix agent:
    print(response.value)

### 5. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

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

Puoi fare in modo che zabbix_utils legga i parametri `Server` o `ServerActive` da un file di configurazione locale di Zabbix agent o 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 server, Sender invia i dati al primo server disponibile in ciascun cluster. Se ServerActive non è impostato, viene utilizzato l'indirizzo di 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 modulo Un esempio di processo di scrittura di un modulo consiste nei seguenti passaggi (ove disponibili, fare clic sul nome del file o della cartella per visualizzare ulteriori dettagli sul passaggio):

1. Creare una nuova directory per il modulo in **zabbix/ui/modules/**.
2. Aggiungere il file manifest.json con i metadati del modulo.
3. Creare la cartella *views* e definire una o più viste del modulo.
4. Creare la cartella *actions* e definire una o più azioni del modulo.
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. Crea la cartella *assets* per i file JavaScript (inseriscili in *assets/js*), gli stili CSS (inseriscili in *assets/css*) o qualsiasi altro file aggiuntivo.
7. Assicurati di specificare le viste, le azioni e i file asset richiesti nel file manifest.json.
8. Registra il modulo nel frontend di Zabbix e inizia a utilizzarlo.

> **Note:**
> Puoi registrare e abilitare un modulo non appena crei il file manifest.json. Una volta abilitato il modulo, puoi visualizzare in anteprima tutte le modifiche apportate ai file del modulo aggiornando il frontend di Zabbix.

manifest.json

Tutti i moduli devono contenere il file manifest.json. Il file deve essere posto nella cartella principale del modulo (ad esempio, *zabbix/ui/modules/module_name/manifest.json*).

Come minimo, manifest.json dovrebbe specificare questi campi:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

Parametri supportati in **manifest.json** (clicca 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 mostrato nella sezione Administration.	
namespace	Namespace PHP per le classi del modulo.	
version	Versione del Modulo.	
type	Tipo del modulo. Per i widget deve essere impostato a *widget*	Yes per widgets, altimenti No
widget	Configurazione del Widget . Utilizzato solo per i Widget.	
actions	Actions da registrare con il modulo.	
assets	Stili CSS e file JavaScript da includere.	No
author	Autore del modulo.	
config	Valori di default per le opzioni custom del modulo.	
description	Descrizione del modulo.	
url	Un link 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 link alla descrizione del modulo. Per i widget, questo link si aprirà cliccando sull'icona della guida nella finestra *Aggiungi widget* o *Modifica widget*. Se l'**url** non viene specificato, cliccando 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 doAction(): 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(_('Nome')))
```

Questo aggiungerà una nuova colonna con nome *Nome* e applicherà alla riga superiore della tabella lo stesso stile usato 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"]
}
```

CSS styles

I file CSS possono contenere un attibuto custom `theme` così da definire differenti temi per uno specifico tema di frontend.

Temi disponibili e relativo valore degli attributi:

- **Blue** - [theme='blue-theme']
- **Dark** - [theme='dark-theme']
- **High-contrast light** - [theme='hc-light']
- **High-contrast dark** - [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.

Aggiungere un modulo Apri la pagina *Amministrazione→Generale→Moduli* e premi *Scansiona directory*.



Individua il tuo modulo nell'elenco e attivalo.

Per attivare un modulo, premi il collegamento ipertestuale *Disabilitato*: lo stato del modulo cambierà in *Abilitato*.

Premi sul nome del modulo per visualizzare ulteriori informazioni sul modulo, come autore, versione o breve descrizione (se definita nel file 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*, determinandone l'aspetto e il comportamento quando vengono renderizzati nella vista di configurazione.

La classe *CWidgetFormView* supporta i seguenti metodi:

- *addField()* - riceve come parametro un'istanza della classe *CWidgetFieldView*; ogni classe *CWidgetField* ha una rispettiva classe *CWidgetFieldView* 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 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 del modulo (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList*, ecc.). Le classi che estendono *CWidgetField* sono responsabili della ricezione, del

salvataggio e della validazione dei valori di configurazione del widget.

Sono disponibili le seguenti classi *CWidgetField*.

Classe CWidgetField	Tipo di campo del database	Descrizione
CWidgetFieldCheckBox	int32	Casella di controllo singola.
CWidgetFieldCheckBoxList	array of int32	Più caselle di controllo in un singolo campo di configurazione.
CWidgetFieldColor	string	Campo di selezione del colore.
CWidgetFieldDatePicker	string	Campo di selezione della data.
CWidgetFieldHostPatternSelect	string	Campo di selezione multipla che consente di selezionare uno o più host. Supporta la definizione di modelli di nome host (verranno selezionati tutti gli host corrispondenti).
CWidgetFieldIntegerBox	int32	Campo per inserire un numero intero. Può essere usato per configurare valori minimi e massimi.
CWidgetFieldLatLng	string	Casella di testo che consente di inserire latitudine, longitudine e livello di zoom della mappa separati da virgole.
CWidgetFieldMultiSelectAction	ID	Campo di selezione multipla per selezionare azioni (dall'elenco delle azioni definite in *Avvisi* → *Azioni*).
CWidgetFieldMultiSelectGraph	ID	Campo di selezione multipla per selezionare grafici personalizzati.
CWidgetFieldMultiSelectGraphPrototype	ID	Campo di selezione multipla per selezionare prototipi di grafico personalizzati.
CWidgetFieldMultiSelectGroup	ID	Campo di selezione multipla per selezionare gruppi di host.
CWidgetFieldMultiSelectHost	ID	Campo di selezione multipla per selezionare host.
CWidgetFieldMultiSelectItem	ID	Campo di selezione multipla per selezionare item.
CWidgetFieldMultiSelectItemPattern	ID	Campo di selezione multipla per selezionare modelli di item.
CWidgetFieldMultiSelectItemPrototype	ID	Campo di selezione multipla per selezionare prototipi di item.
CWidgetFieldMultiSelectMap	ID	Campo di selezione multipla per selezionare mappe.
CWidgetFieldMultiSelectMediaType	ID	Campo di selezione multipla per selezionare tipi di supporto.
CWidgetFieldMultiSelectOverrideHost	ID	Campo di selezione multipla per selezionare una sorgente dati (dashboard o altro widget) contenente un host per il quale il widget può visualizzare dati.
CWidgetFieldMultiSelectService	ID	Campo di selezione multipla per selezionare servizi.
CWidgetFieldMultiSelectSla	ID	Campo di selezione multipla per selezionare SLA.
CWidgetFieldMultiSelectUser	ID	Campo di selezione multipla per selezionare utenti.
CWidgetFieldNumericBox	string	Campo per inserire un numero in virgola mobile.
CWidgetFieldRadioButtonList	int32	Gruppo di pulsanti di opzione composto da uno o più pulsanti di opzione.
CWidgetFieldRangeControl	int32	Cursore per selezionare un valore di tipo intero.
CWidgetFieldReference	string	Crea un identificatore univoco per questo widget nella dashboard. Viene utilizzato per fare riferimento a questo widget da altri widget.
CWidgetFieldSelect	int32	Casella di selezione a discesa.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* preimpostato con le severity dei trigger.
CWidgetFieldTags	array of (string, int32, string)	Consente di configurare una o più righe di filtro dei 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.
CWidgetFieldTimePeriod	array of string	Campo di selezione dell'intervallo di tempo.
CWidgetFieldTimeZone	string	Menu a discesa con i fusi orari.
CWidgetFieldThresholds	array of (string, string)	Consente di configurare coppie colore-numero.
CWidgetFieldUrl	string	Casella di testo che consente di inserire URL.

Le seguenti classi *CWidgetField* sono state create per widget specifici. Queste classi hanno casi d'uso molto specifici, ma possono anche essere riutilizzate se necessario.

Classe CWidgetField	Tipo di campo del database	Descrizione
CWidgetFieldColumnsList	array of (multiple mixed)	Per il widget *Top hosts*. Crea una tabella con colonne personalizzate dei tipi consentiti.

Classe CWidgetField	Tipo di campo del database	Descrizione
CWidgetFieldNavTree	string	Per il widget *Map navigation tree*. Sostituisce la vista del widget in modalità di modifica con l'albero di selezione della mappa.

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.

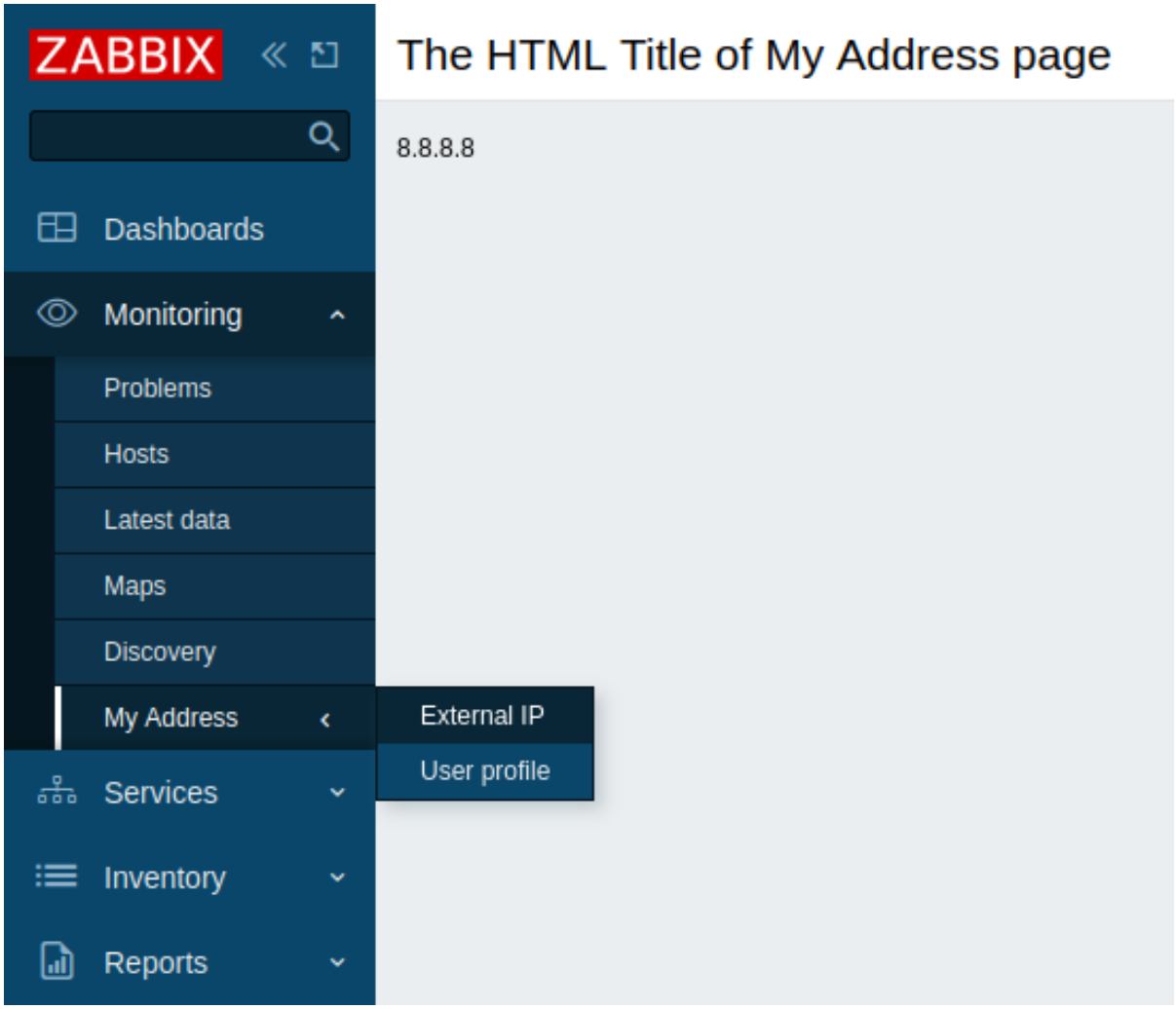
Crea un modulo (tutorial)

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, prima creerai un modulo frontend che aggiunge una nuova sezione di menu *Il mio indirizzo* 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 - su una nuova pagina nella sezione di menu *Il mio indirizzo* appena creata. Ecco come apparirà il modulo finito:



Parte I - Nuova sezione del menu

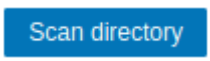
Aggiungere un modulo vuoto al frontend di Zabbix

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).

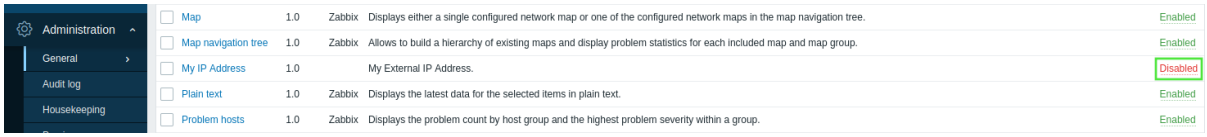
ui/modules/MyAddress/manifest.json

```
{
    "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 cambiare 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 ancora necessario definire la funzionalità del modulo. Una volta aggiunto il contenuto alla directory del modulo, le modifiche saranno immediatamente visibili nel frontend di Zabbix aggiornando la 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, è possibile usare l'azione predefinita *userprofile.edit*, che aprirà la pagina *Profilo utente*. Nella parte III di questa esercitazione, imparerai a creare un'azione personalizzata.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

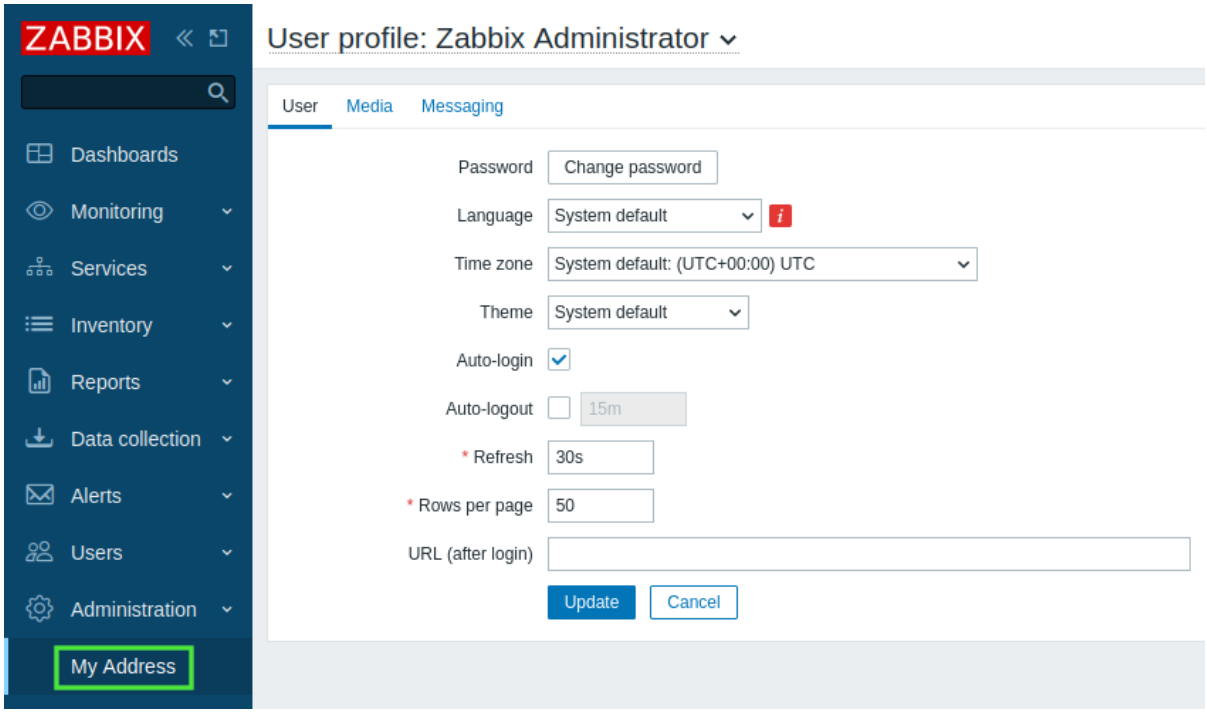
use Zabbix\Core\CModule,
    APP,
    CMenuItem;

class Module extends CModule {

    public function init(): void {
        APP::Component()->get('menu.main')
            ->add((new CMenuItem(_('My Address')))
            ->setAction('userprofile.edit'));
    }
}
```

Note:
È possibile sostituire `'userprofile.edit'` con altre azioni, ad esempio `'charts.view'` (apre i grafici personalizzati), `'problems.view'` (apre *Monitoraggio* → *Problemi*) oppure `'report.status'` (apre il report *Informazioni di sistema*).

3. Aggiornare il frontend di Zabbix. Ora è presente una nuova sezione *My Address* nella parte inferiore del menu principale di Zabbix. Fare clic su *My Address* per aprire la pagina *Profilo utente*.



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 nidificato. Di conseguenza, gli utenti potranno accedere a due pagine di sottomenu dalla sezione di menu *Monitoring* → *My Address*.

1. Apri e modifica il file *Module.php*.

ui/modules/MyAddress/Module.php

```
<?php

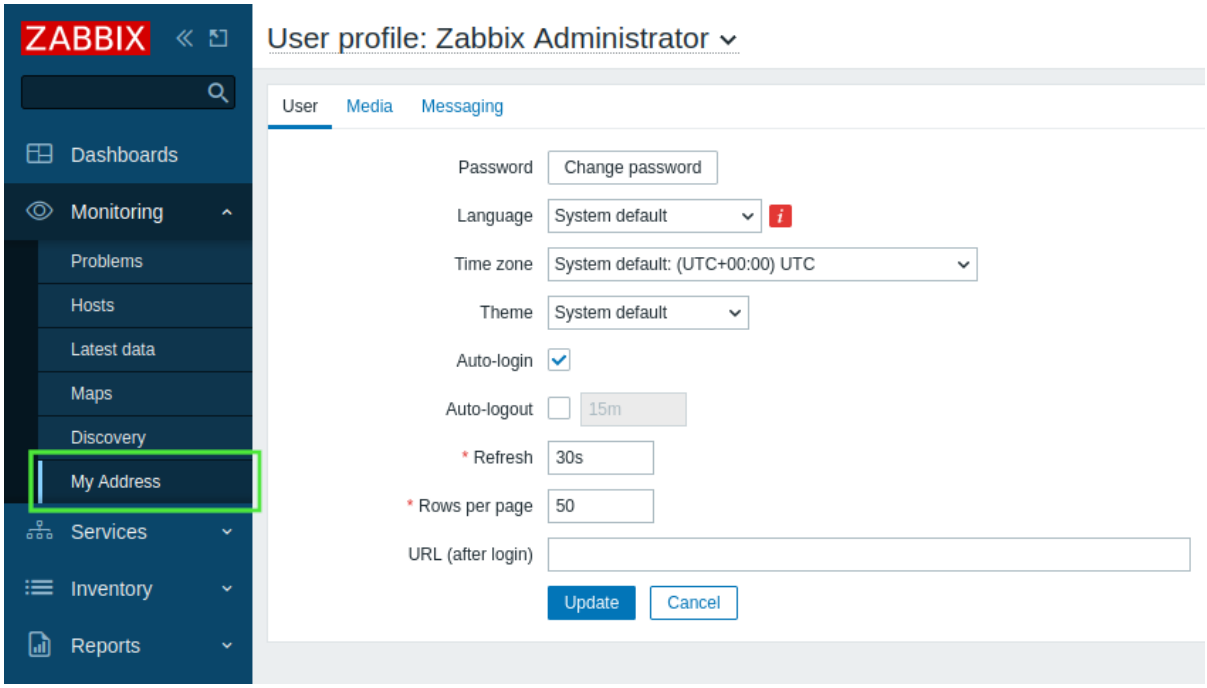
namespace Modules\MyAddress;

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* ora si trova sotto la sezione *Discovery*.



3. Per aggiungere pagine nidificate alla sezione di menu *My Address*, apri e modifica di nuovo 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 nidificato è necessario utilizzare la classe *CMenu* oltre alle classi usate nei passaggi precedenti.

ui/modules/MyAddress/Module.php

```
<?php

namespace Modules\MyAddress;

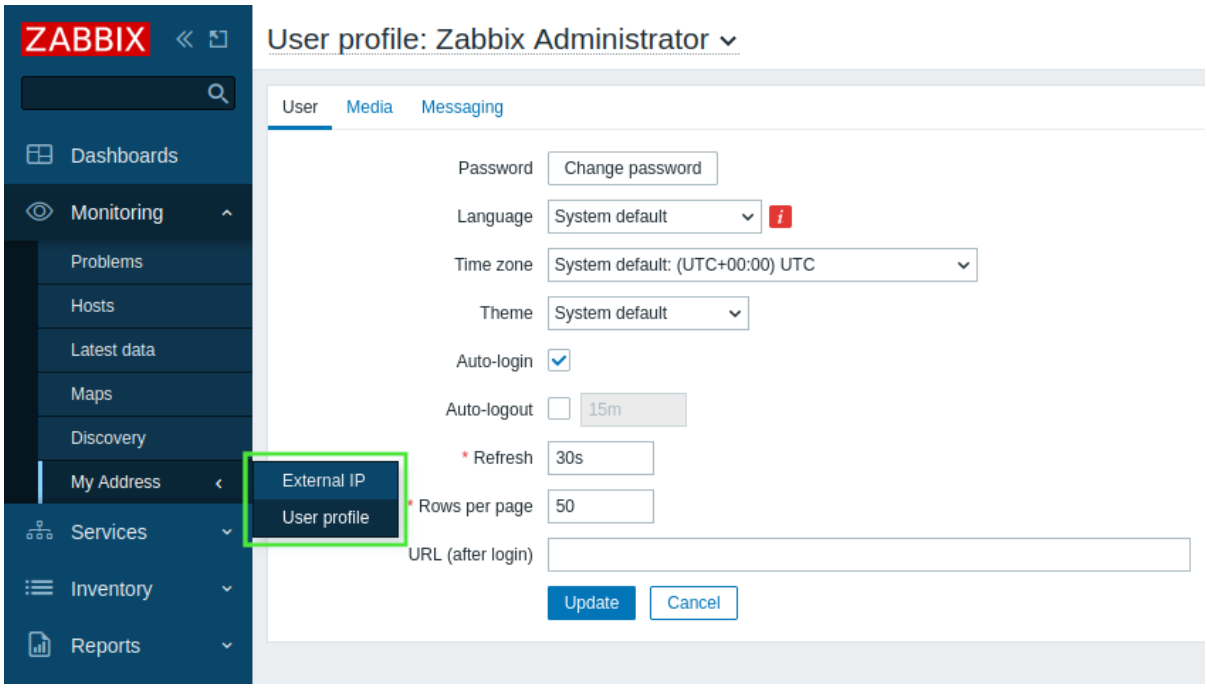
use Zabbix\Core\CModule,
```

```
    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 è 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 vista.

1. Creare una directory *actions* nella directory *MyAddress*.
2. Creare 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 richiama la funzione *doAction()* quando l'azione viene richiesta. 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 vista.

ui/modules/MyAddress/actions/MyAddress.php

```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. Creare una nuova directory *views* nella directory *MyAddress*.
4. Creare un file *my.address.php* nella directory *views* e definire la vista del modulo.

Si noti che la variabile $data è disponibile nella vista senza doverla definire esplicitamente. Il framework passa automaticamente l'array associativo alla vista.

ui/modules/MyAddress/views/my.address.php

```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*. Aprire *manifest.json* e aggiungere un nuovo oggetto `actions` che contenga:

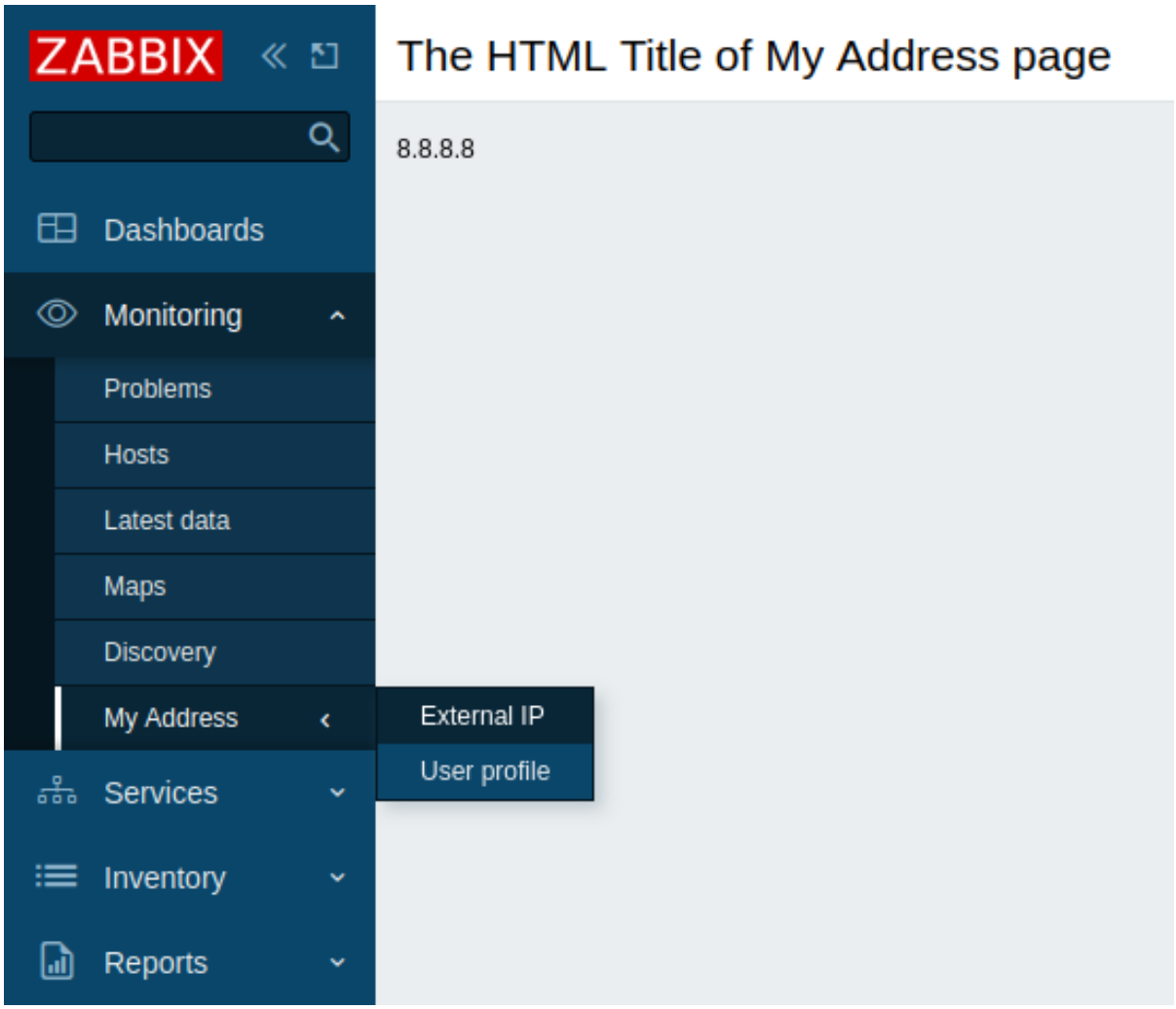
- la chiave dell'azione con il nome dell'azione scritto in minuscolo (a-z) e con le parole separate da punti (ad esempio, `my.address`);
- il nome della classe dell'azione (`MyAddress`) come valore della chiave `class` dell'oggetto `my.address`;
- il nome della vista dell'azione (`my.address`) come valore della chiave `view` dell'oggetto `my.address`.

ui/modules/MyAddress/manifest.json

```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. Aggiornare Zabbix frontend. Fare clic su *My address* → *External IP* per visualizzare l'indirizzo IP del proprio 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 realizzerai Durante questo tutorial, prima creerai un widget base "Hello, world!" e poi lo convertirai in un widget più avanzato che visualizza il valore di un elemento come un grafico a indicatori. Ecco come apparirà il widget finito:

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 trattati come 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).

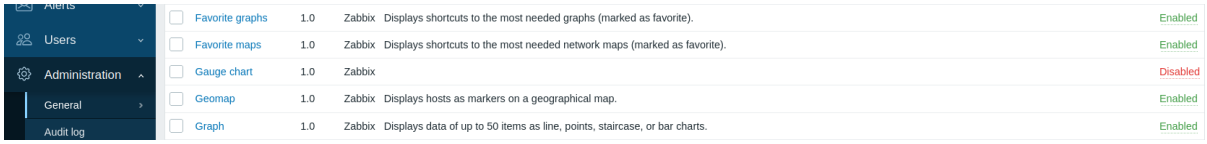
ui/modules/lesson_gauge_chart/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

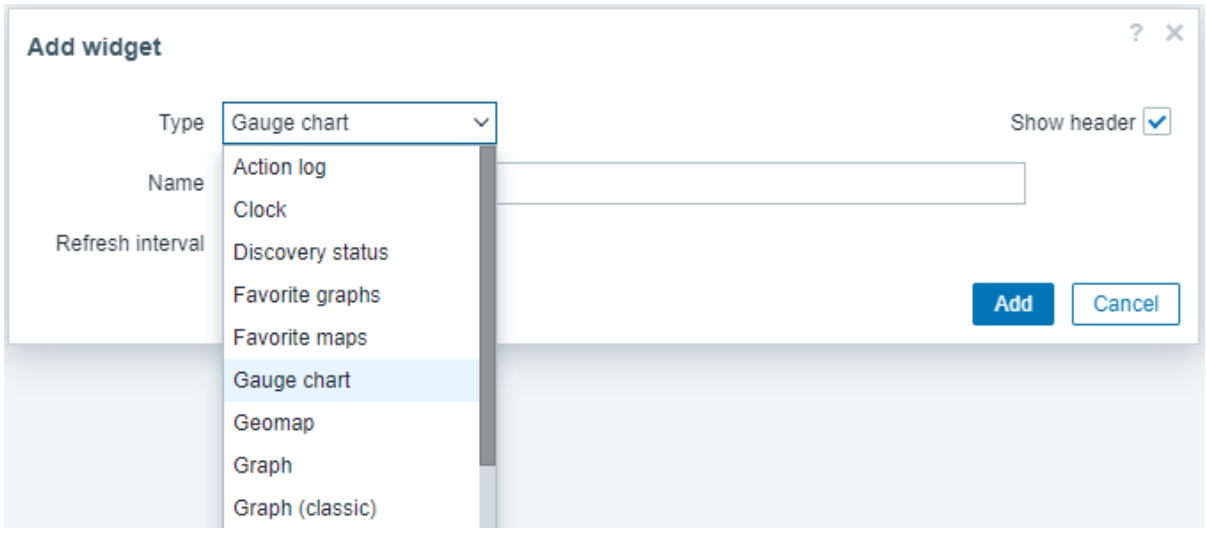
3. Nel frontend di Zabbix, andare alla sezione *Administration* → *General* → *Modules* e fare clic sul pulsante *Scan directory*.



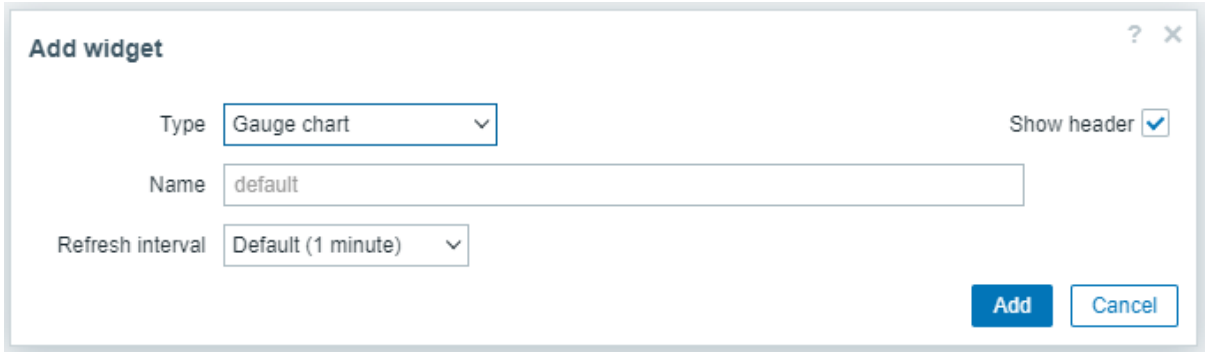
4. Individuare il nuovo modulo *Gauge chart* nell'elenco e fare clic sul collegamento ipertestuale "Disabled" per cambiare lo stato del modulo da "Disabled" a "Enabled" (se il modulo non è elencato, vedere la sezione troubleshooting).



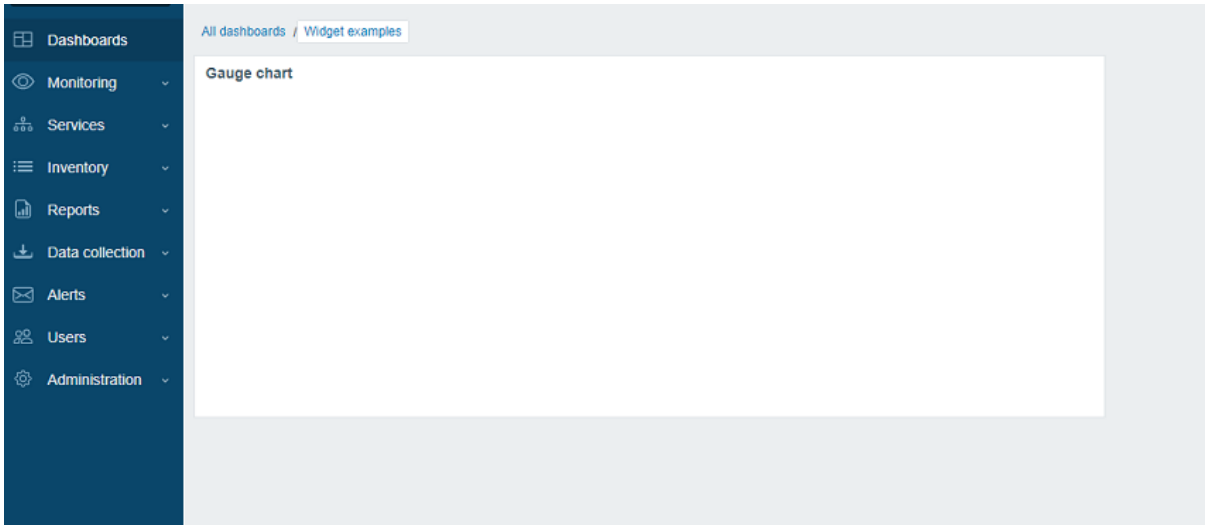
5. Aprire una dashboard, passare alla modalità di modifica e aggiungere un nuovo widget. Nel campo *Type*, selezionare "Gauge chart".



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. Sulla dashboard dovrebbe apparire un widget vuoto. Fare clic su *Save changes* nell'angolo superiore destro 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*.

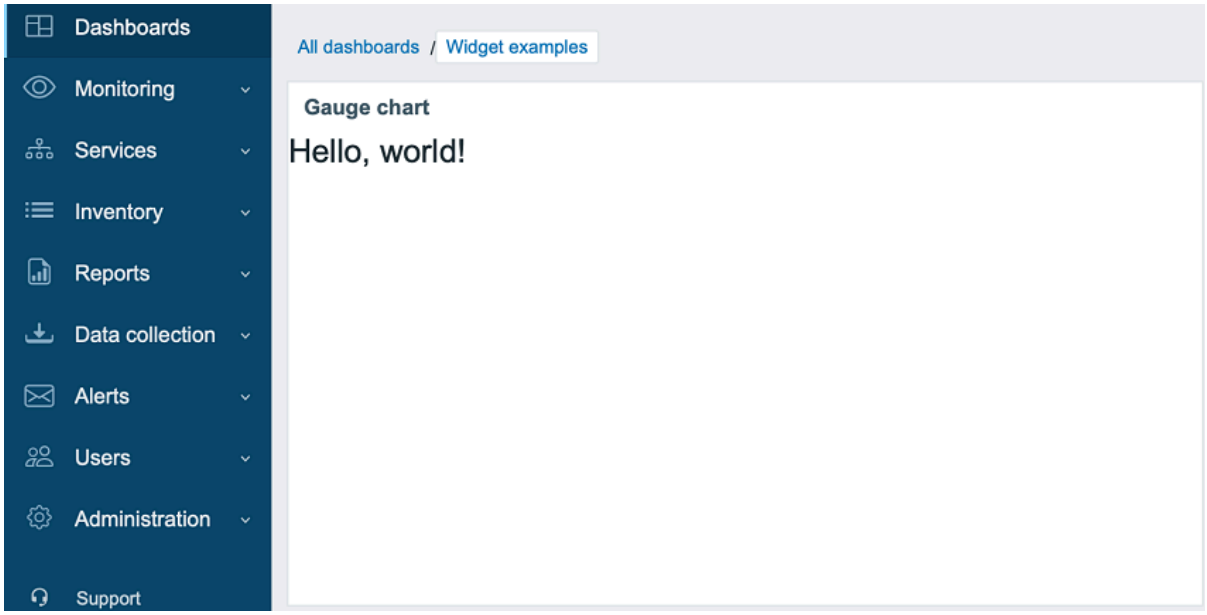
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?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 visualizza ”Hello, world!”.



Parte II - Grafico degli indicatori Aggiungere impostazioni a una vista di configurazione e usarle in una vista 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 modulo (*Zabbix\Widgets\CWidgetForm*) e da una vista del modulo del widget (*widget.edit.php*). Per aggiungere campi (*Zabbix\Widgets\CWidgetField*), è necessario creare una classe *WidgetForm*, che estenderà *Zabbix\Widgets\CWidgetForm*.

Il modulo contiene un insieme di campi (*Zabbix\Widgets\CWidgetField*) di vari tipi, utilizzati per convalidare i valori immessi dall'utente. Il campo del modulo (*Zabbix\Widgets\CWidgetField*) per ciascun tipo di elemento di input converte il valore in un unico formato per memorizzarlo 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 modulo di configurazione del widget. Si tratta di un normale campo di testo, in cui un 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 della 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*.

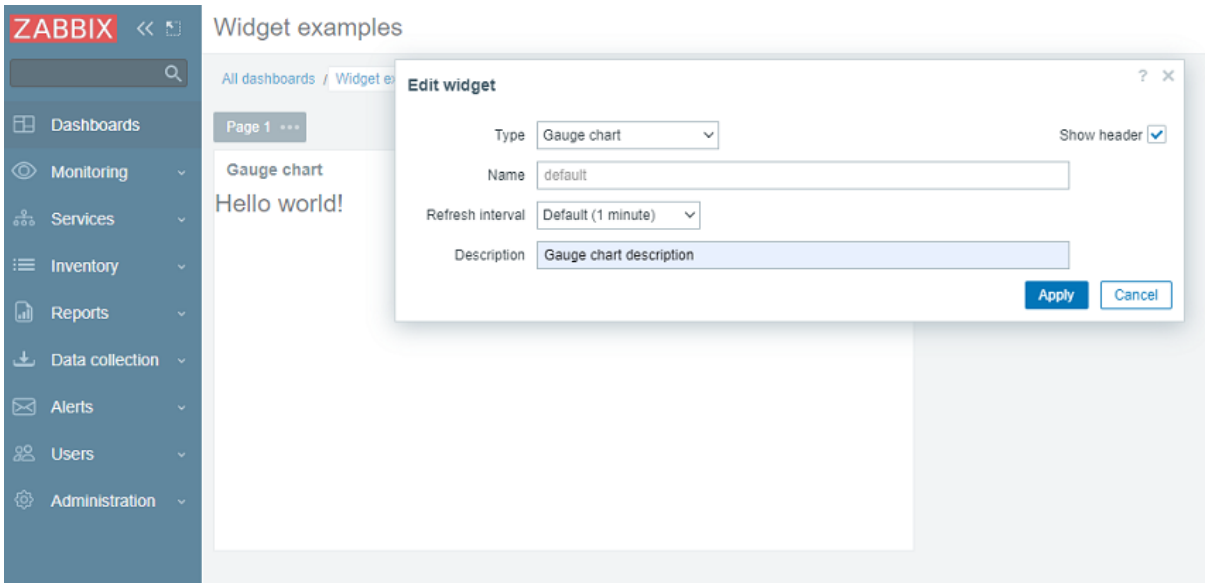
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?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 dell'ingranaggio nel widget per aprire il modulo di configurazione del widget.
6. Il modulo 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 modulo di configurazione del widget. Quindi fai clic su *Save changes* nell'angolo superiore destro per salvare la dashboard. Nota che la nuova descrizione non è visibile da nessuna parte e il widget continua a visualizzare "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 farlo, è necessario creare una classe action.

8. Crea una nuova directory *actions* nella directory *lesson_gauge_chart*.
9. Crea un file *WidgetView.php* nella directory *actions*. La classe action *WidgetView* estenderà la classe *CControllerDashboard-WidgetView*.

I valori dei campi di configurazione del widget sono memorizzati nella proprietà **$fields_values** della classe action.

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 action 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']*.

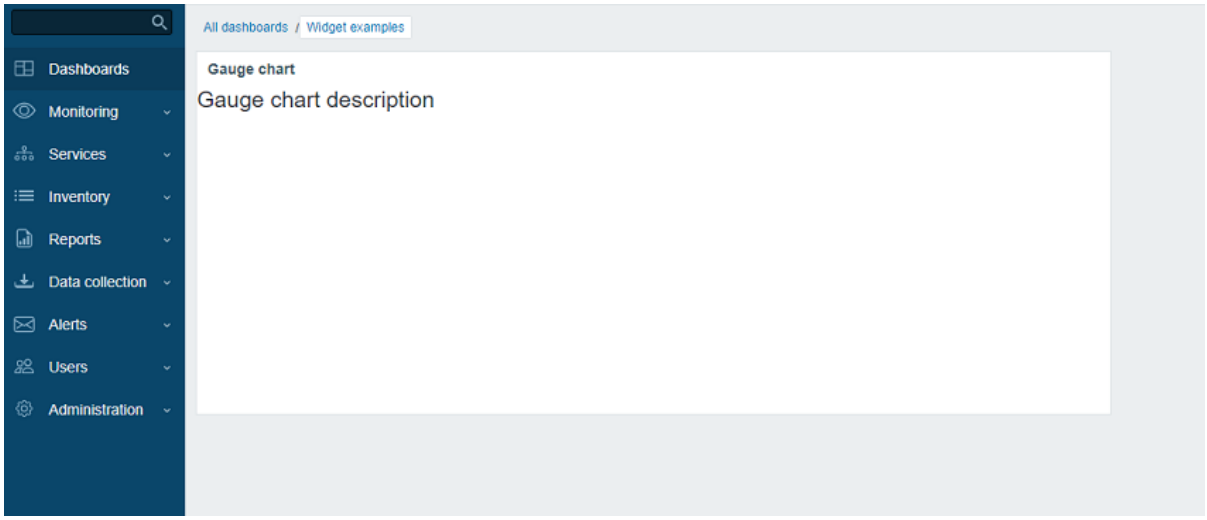
ui/modules/lesson_gauge_chart/views/widget.view.php

```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 un valore di item tramite API

Il widget deve 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. Successivamente, il controller del widget sarà in grado di recuperare i dati dell'item e il relativo 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
};

/**
 * Modulo del widget Gauge chart.
 */
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.

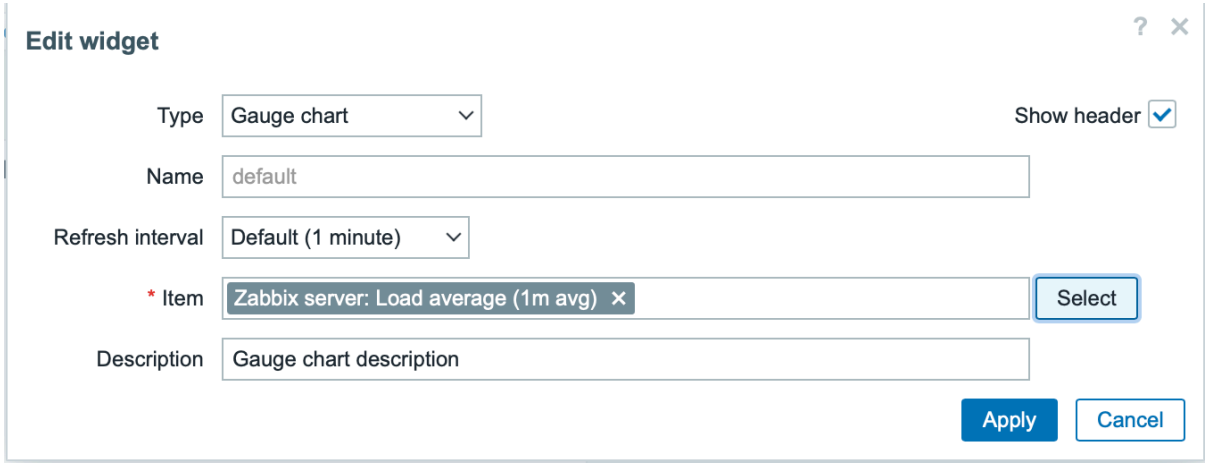
ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 * Vista del modulo del widget Gauge chart.
 *
 * @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 dell'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. Quindi 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 in poi, 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.

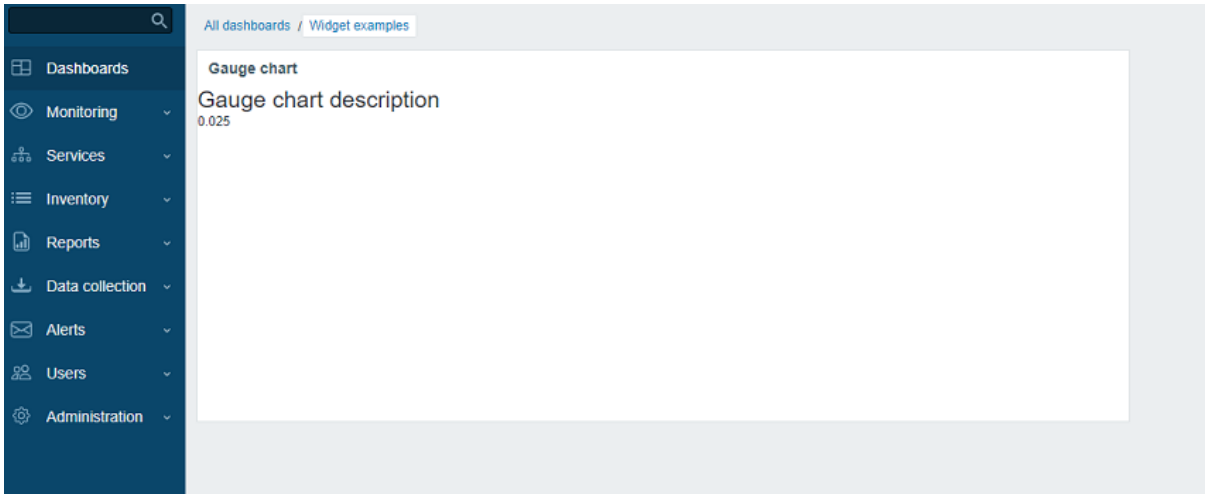
ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Vista del widget Gauge chart.
 *
 * @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 visualizzerà 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 *Descrizione* 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). JavaScript, fornito di seguito, visualizza la stringa "No data" in caso di dati mancanti. La stringa "No data" è presente nei file di traduzione della UI di Zabbix.

Se sono presenti costanti del widget, si consiglia di specificarle anch'esse 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 tavolozza 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
```

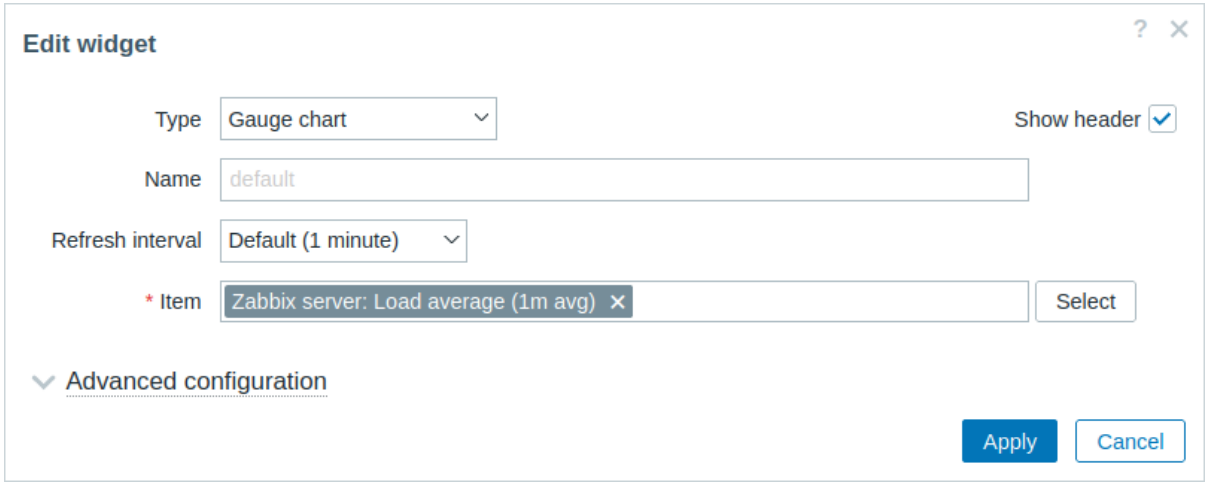
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($data['fields']['value_units']);
$lefty_static_units = (new CWidgetFieldTextBoxView($data['fields']['value_static_units']))
    ->setPlaceholder(_('value'))
    ->setWidth(ZBX_TEXTAREA_TINY_WIDTH);

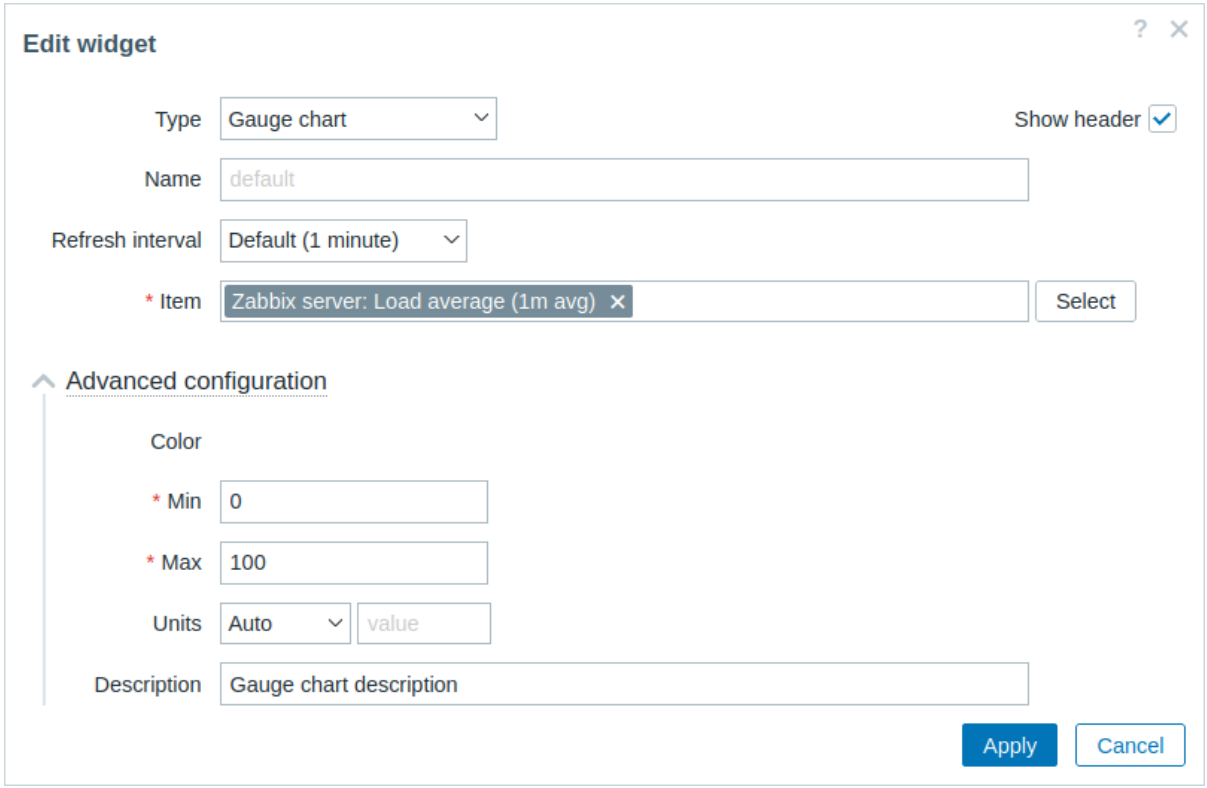
(new CWidgetFormView($data))
    ->addField(
        (new CWidgetFieldMultiSelectItemView($data['fields']['itemid']))
            ->setPopupParameter('numeric', true)
    )
    ->addFieldset(
        (new CWidgetFormFieldsetCollapsibleView(_('Advanced configuration')))
            ->addField(
                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 dell'ingranaggio nel widget per aprire il modulo di configurazione del widget. Il modulo di configurazione del widget ora contiene una nuova sezione espandibile/comprimibile *Configurazione avanzata*.



5. Espandi la sezione *Configurazione avanzata* per visualizzare ulteriori campi 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 sezione successiva - *Aggiungere JavaScript al widget*.



Aggiungere JavaScript al widget

In questa sezione imparerai come aggiungere un grafico gauge, 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 colore 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

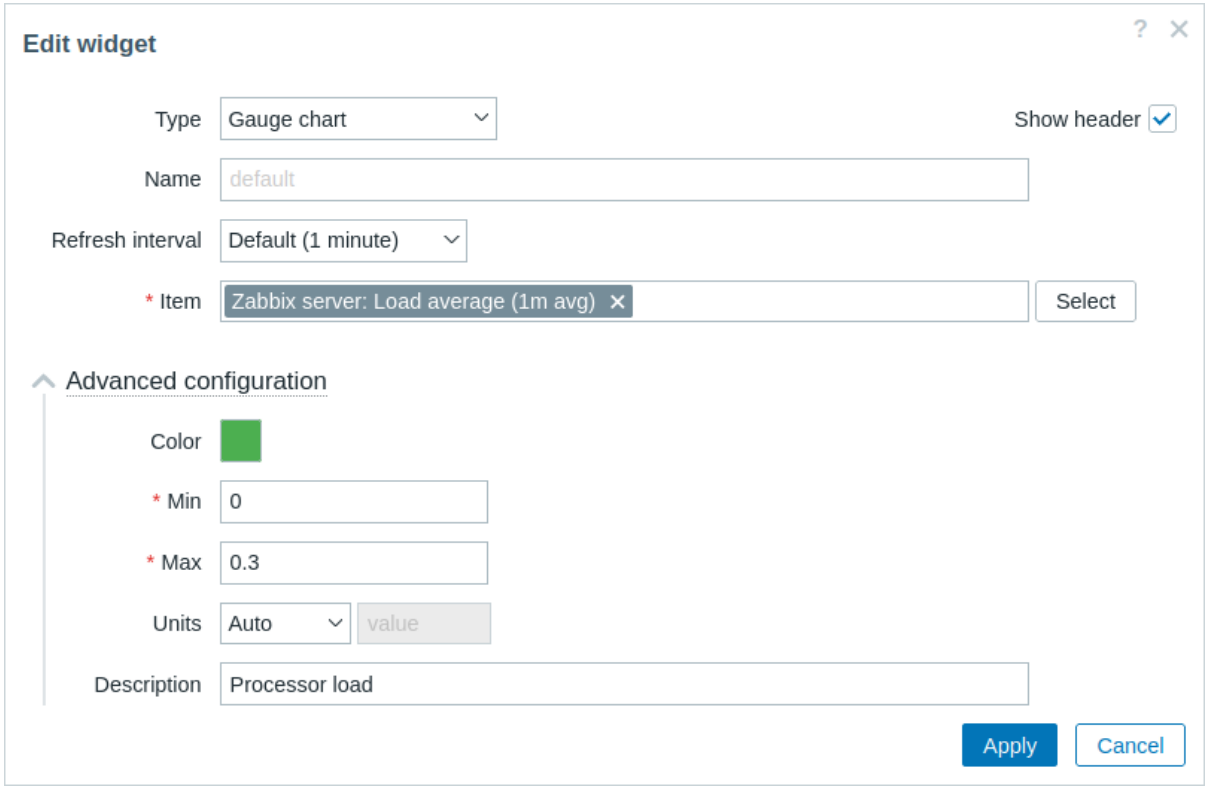
/**
 * Gauge chart widget form view.
 *
 * @var CView $this
 * @var array $data
 */

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 dell'ingranaggio nel widget per aprire il modulo di configurazione del widget. Ora espandi la sezione *Advanced configuration* per vedere il selettore di colore inizializzato. Compila i campi con i valori e seleziona un colore per il grafico gauge.



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 gauge, 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 archiviare JavaScript, CSS e potenzialmente qualsiasi altra risorsa, 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 di base di tutti i widget della dashboard, *CWidget*.

La dashboard si basa su una corretta implementazione di un widget e comunica al widget tutte le informazioni rilevanti chiamando i rispettivi metodi JavaScript. La dashboard si aspetta inoltre che il widget generi eventi quando si verifica qualche interazione. Per questo motivo, 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, pertanto verrà implementata una logica personalizzata 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 gauge) se il processo di aggiornamento del widget è andato a buon fine e senza errori (vedi il metodo *processUpdateResponse(response)* e i relativi metodi *_resizeChart()* e *_updatedChart()*)
- ridimensionamento del widget (vedi il metodo *onResize()* e il relativo metodo *_resizeChart()*)

Per gli altri aspetti del widget del grafico gauge 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 di 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 accattivante.

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 applicare lo stile agli elementi del widget, usa il selettore *div.dashboard-widget-{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;
}
```

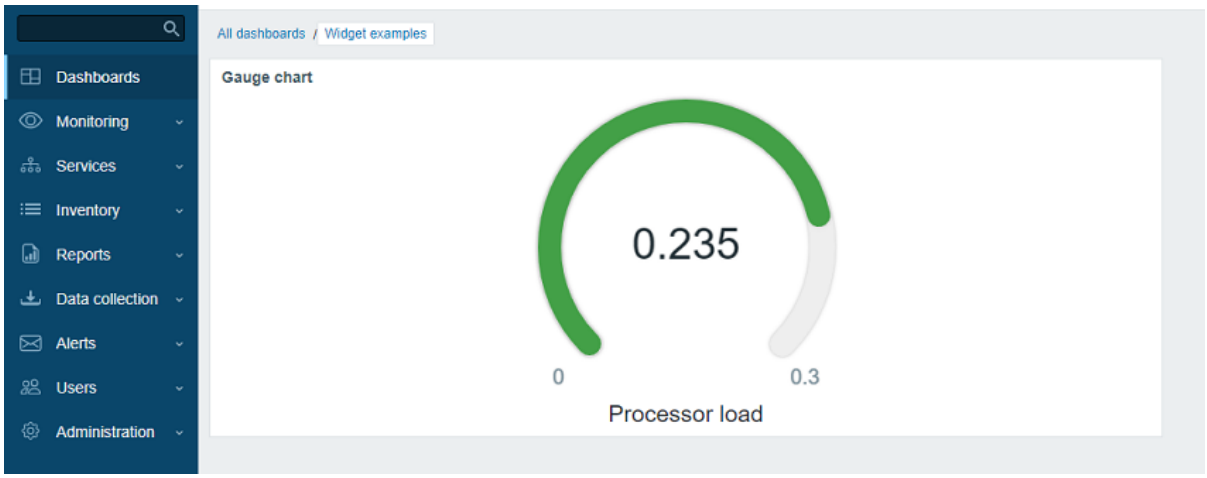
3. Apri *manifest.json* e aggiungi il nome del file CSS (*widget.css*) all'array nella sezione *assets/css*. Questo consentirà agli stili CSS definiti in *widget.css* di essere caricati con la pagina della dashboard.

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 della 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

Questa pagina elenca tutte le modifiche (se presenti) allo sviluppo di estensioni Zabbix personalizzate.

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

Sezione: Comandi di manutenzione (8)
Aggiornato: 2019-01-29
Indice Torna al contenuto principale

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 il monitoraggio dei parametri di vari servizi.

OPZIONI

-c, **--config** *file-di-configurazione* Utilizza il *file-di-configurazione* alternativo invece di quello predefinito.

-R, **--runtime-control** *opzione-runtime* Esegue le funzioni amministrative in base all'*opzione-runtime*.

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 conosciuti ed esce. Per ogni item vengono utilizzati i 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 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 da Zabbix server o zabbix_get durante l'interrogazione di un demone agent in esecuzione, poiché i permessi o l'ambiente potrebbero essere diversi. I tipi di valori restituiti sono:

d
Numero con una parte decimale.

m
Non supportato. Ciò potrebbe 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. Anche problemi di permessi o parametri utente errati potrebbero causare lo stato non supportato.

s
Testo. Lunghezza massima non limitata.

t
Testo. Uguale a **s**.

u
Intero senza segno.

-t, **--test** *item-key*
Testa 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 conosciuti.

-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).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

FILE

VEDI ANCHE

AUTORE

Questo documento è stato creato il: 14:07:57 GMT, 22 novembre 2021

zabbix_agentd

Sezione: Comandi di manutenzione (8)
Aggiornato: 2019-01-29
Indice Torna al contenuto principale

NOME

zabbix_agentd - Demone Zabbix agent

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** *file-di-configurazione* Utilizza il *file-di-configurazione* alternativo invece di quello predefinito.

-f, **--foreground** Esegui Zabbix agent in primo piano.

-R, **--runtime-control** *opzione-runtime* Esegue le funzioni amministrative in base all'*opzione-runtime*.

Opzioni di controllo in fase di esecuzione userparameter_reload
Ricarica i parametri utente dal file di configurazione

log_level_increase[=*target*]
Aumenta il livello di log; se *target* non è specificato, influisce su tutti i processi

log_level_decrease[=*target*]
Diminuisce il livello di log; se *target* non è specificato, influisce su tutti i processi

Destinazioni del controllo del livello di log *process-type*
Tutti i processi del tipo specificato (active checks, 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 la destinazione come "process-type,N"

-p, **--print**
Stampa gli item noti ed esce.
Per ogni item vengono usati i 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 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 durante l'interrogazione di un demone agent in esecuzione, poiché i permessi o l'ambiente potrebbero essere diversi.
I tipi di valori restituiti sono:

d
Numero con una 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 dei log o un item che richiede più valori raccolti.
Anche problemi di permessi o parametri utente errati possono causare lo stato di non supportato.

s
Testo.
Lunghezza massima non limitata.

t
Testo.
Uguale a **s**.

u
Intero senza segno.

-t, **--test** *item-key*
Testa 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 di Zabbix agent (se non modificata in fase di compilazione).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Indice

NOME
SINOSSI
DESCRIZIONE
OPZIONI
FILE
VEDI ANCHE
AUTORE

Questo documento è stato creato il: 20:50:13 GMT, 22 novembre 2021

zabbix_get

Sezione: Comandi utente (1)
Aggiornato: 2022-01-06
Index Torna ai contenuti principali

NOME

zabbix_get - Zabbix get utility

SINOSSI

zabbix_get -s *nom-d'equip-o-IP* [**-p** *port*] [**-I** *adreça-IP*] [**-t** *timeout*] **-k** *clau-d'element*
zabbix_get -s *nom-d'equip-o-IP* [**-p** *port*] [**-I** *IP-address*] [**-t** *temps-d'espera*] **--tls-connect cert --tls-ca-file** *arxiu-CA* [**--tls-crl-file** *arxiu-CRL*] [**--tls-agent-cert-issuer** *cert-issuer*] [**--tls-agent-cert-subject** *cert-subject*] **--tls-cert-file** *arxiu-cert* **--tls-key-file** *arxiu-key* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *clau-d'element*
zabbix_get -s *nom-d'equip-o-IP* [**-p** *port*] [**-I** *adreça-IP*] [**-t** *temps-d'espera*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *arxiu-PSK* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *item-key*
zabbix_get -h
zabbix_get -V

DESCRIZIONE

zabbix_get è un'utilità 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 (predefinito: 30)

-k, **--key** *item-key*
Specifica la chiave dell'item di cui recuperare il valore.

-P, **--protocol** *value*
Protocollo utilizzato per comunicare con l'agent. Valori:

auto si connette utilizzando il protocollo JSON, in caso di errore ripiega e ritenta con il protocollo plaintext (predefinito)

json si connette utilizzando il protocollo JSON

plaintext si connette utilizzando il protocollo plaintext in cui viene inviata solo la chiave dell'item (release 6.4.x e precedenti)

--tls-connect *value*
Come connettersi all'agent. Valori:

non crittografato
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 della 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-agent-cert-issuer *cert-issuer*
Autorità emittente del certificato agent consentita.

--tls-agent-cert-subject *cert-subject*
Soggetto del certificato 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 di identità PSK.

--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 di selezione predefiniti della suite di cifratura. Questa opzione non è disponibile se la versione di OpenSSL è inferiore alla 1.1.1.

--tls-cipher *cipher-string*
Stringa di priorità GnuTLS (per TLS 1.2 e versioni successive) oppure stringa di cifratura OpenSSL (solo per TLS 1.2). Sovrascrive i criteri di selezione predefiniti della suite di cifratura.

-h, **--help**
Visualizza questa guida ed esce.

-V, **--version**
Mostra le informazioni sulla versione ed esce.

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 operations,O=Example Corp,DC=example,DC=com" --tls-cert-file /home/zabbix/zabbix_get.crt --tls-key-file /home/zabbix/zabbix_get.key
zabbix_get -s 127.0.0.1 -p 10050 -k "system.cpu.load[all,avg1]" --tls-connect psk --tls-psk-identity "PSK ID Zabbix agentd" --tls-psk-file /home/zabbix/zabbix_agentd.psk

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

ESEMPI

VEDERE ANCHE

AUTORE

Questo documento è stato creato il: 08:42:29 GMT, 11 giugno 2021

zabbix_js

Sezione: Comandi utente (1)
Aggiornato: 2022-01-06
Index Torna ai contenuti principali

NOME

zabbix_js - Utilità Zabbix JS

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 come nome del file viene specificato '-', lo script verrà letto da stdin.

-p, **--param** *input-param*
Specifica il parametro di input.

-i, **--input** *input-file*
Specifica il nome del file del parametro di input. Se come nome del file viene specificato '-', l'input verrà letto da stdin.

-w, **--webdriver** *url*
Specifica l'URL del webdriver.

-l, **--loglevel** *log-level*
Specifica il livello di log.

-t, **--timeout** *timeout*
Specifica il timeout in secondi. Intervallo valido: 1-600 secondi (predefinito: 10)

-h, **--help**
Visualizza questo aiuto ed esce.

-V, **--version**
Mostra le informazioni sulla versione ed esce.

ESEMPI

zabbix_js -s script-file.js -p esempio

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

ESEMPI

VEDI ANCHE

Questo documento è stato creato il: 21:23:35 GMT, March 18, 2020

zabbix_proxy

Sezione: Comandi di manutenzione (8)
Aggiornato: 2020-09-04
Indice Torna al contenuto principale

NOME

zabbix_proxy - Demone proxy 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 demone che raccoglie i dati di monitoraggio dai dispositivi e li invia al server Zabbix.

OPZIONI

-c, **--config** *file-di-configurazione* Utilizza il *file-di-configurazione* invece di quello predefinito.

-f, **--foreground** Esegue il proxy Zabbix in primo piano.

-R, **--runtime-control** *opzione-runtime* Esegue le funzioni amministrative in base all'*opzione-runtime*.

Opzioni di controllo in fase di esecuzione 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.
Verrà utilizzato il file di configurazione predefinito (a meno che non sia specificata l'opzione **-c**) per individuare il file PID e il segnale verrà inviato al processo elencato nel file PID.

snmp_cache_reload
Ricarica la cache SNMP.
Si noti che Zabbix memorizza nella cache le associazioni SNMPv3 EngineID→IP e può riutilizzare automaticamente gli EngineID per ridurre il sovraccarico del polling.

housekeeper_execute
Esegue l'housekeeper.
Ignorato se l'housekeeper è attualmente in esecuzione.

diaginfo[=*section*]
Registra nel log le 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.

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 (availability manager, browser poller, configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, icmp pinger, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector)

process-type,N
Tipo di processo e numero (ad esempio, poller,3)

pid
Identificatore del processo, fino a 65535. Per valori più grandi specificare la destinazione come ”process-type,N”

Destinazioni del controllo della profilazione

process-type
Tutti i processi del tipo specificato (configuration syncer, data sender, discovery manager, history syncer, housekeeper, http poller, preprocessing manager, icmp pinger, ipmi manager, ipmi poller, java poller, poller, agent poller, http agent poller, snmp poller, self-monitoring, snmp trapper, task manager, trapper, unreachable poller, vmware collector, history poller, availability manager, odbc poller)

process-type,N
Tipo di processo e numero (ad esempio, history syncer,1)

pid
Identificatore del processo, fino a 65535. Per valori più grandi specificare la destinazione come ”process-type,N”

scope
Ambito della profilazione (rwlock, mutex, processing) utilizzabile con il tipo di processo (ad esempio, history syncer,1,processing)

-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_proxy.conf
Posizione predefinita del file di configurazione di Zabbix proxy (se non modificata in fase di compilazione).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

FILE

VEDI ANCHE

AUTORE

Questo documento è stato creato il: 16:12:22 GMT, 04 settembre 2020

zabbix_sender

Sezione: Comandi utente (1)
Aggiornato: 2021-06-01
Indice Torna al contenuto principale

NOME

zabbix_sender - Utilità Zabbix sender

SINOSSI

zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect cert --tls-ca-file** *CA-file* [**--tls-crl-file** *CRL-file*] [**--tls-server-cert-issuer** *cert-issuer*] [**--tls-server-cert-subject** *cert-subject*] **--tls-cert-file** *cert-file* **--tls-key-file** *key-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] **-s** *host* **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-z** *server* [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] **-k** *key* **-o** *value*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *server*] [**-p** *port*] [**-I** *IP-address*] [**-t** *timeout*] [**-s** *host*] **--tls-connect psk --tls-psk-identity** *PSK-identity* **--tls-psk-file** *PSK-file* [**--tls-cipher13** *cipher-string*] [**--tls-cipher** *cipher-string*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *input-file*
zabbix_sender -h
zabbix_sender -V

DESCRIZIONE

zabbix_sender è un'utilità della riga di comando per l'invio di dati di monitoraggio al server o proxy Zabbix. Sul server Zabbix un elemento di tipo **Zabbix trapper** deve essere creato con la chiave corrispondente. Nota che i valori in arrivo verranno accettati solo dagli host specificati nel campo **Host consentiti** per questo elemento.

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**.
Il nome host definito tramite il parametro **HostnameItem** non verrà rilevato; in questo caso il nome host deve essere specificato tramite riga di comando (vedere l'opzione -s).
Tutti gli indirizzi definiti nel parametro di configurazione **ServerActive** dell'agent vengono utilizzati per l'invio dei dati. Se l'invio di dati in batch a un indirizzo non riesce, i batch successivi non vengono inviati a questo indirizzo.

-z, **--zabbix-server** *server*
Nome host o indirizzo IP del server Zabbix. Se un host è monitorato da un proxy, deve essere usato invece il nome host o l'indirizzo IP del proxy. Se usato insieme a **--config**, sovrascrive le voci del parametro **ServerActive** specificate nel file di configurazione di agentd.

-p, **--port** *port*
Specifica il numero di porta del trapper del server Zabbix in esecuzione sul server. Il valore predefinito è 10051. Se usato insieme a **--config**, sovrascrive le voci di porta del parametro **ServerActive** specificate nel file di configurazione di agentd.

-I, **--source-address** *IP-address*
Specifica l'indirizzo IP sorgente. Se usato insieme a **--config**, sovrascrive 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 di Zabbix). L'indirizzo IP e il nome DNS dell'host non funzionano. Se usato insieme a **--config**, sovrascrive 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 da un file di input. Specificare **-** come **<input-file>** per leggere i valori dallo standard input. Ogni riga del file contiene voci delimitate da spazi: **<hostname> <key> <value>**. Ogni valore deve essere specificato sulla propria riga. Ogni riga deve contenere 3 voci delimitate da spazi: **<hostname> <key> <value>**, dove "hostname" è il nome dell'host monitorato come registrato nel frontend di 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**.

Un esempio di riga di un file di input:

"Linux DB3" db.connections 43

Il tipo di valore deve essere impostato correttamente nella configurazione dell'item nel frontend di Zabbix. Zabbix sender invierà fino a 250 valori in una connessione. Il limite di dimensione per l'invio di valori da un file di input dipende dalla dimensione descritta nel protocollo di comunicazione di Zabbix. Il contenuto del file di input deve essere in codifica 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 carattere di quoting è il doppio apice.
- Le voci con spazi devono essere racchiuse tra virgolette.
- I caratteri doppio apice e backslash all'interno di una voce tra virgolette devono essere preceduti da un backslash.
- L'escaping non è supportato nelle voci non tra virgolette.
- Le sequenze di escape di avanzamento riga (\n) sono supportate nelle stringhe tra virgolette.
- Le sequenze di escape di avanzamento riga 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 voci delimitate da spazi: **<hostname> <key> <timestamp> <value>**. Il timestamp deve essere specificato nel formato timestamp Unix. Se l'item di destinazione ha trigger che vi fanno riferimento, tutti i timestamp devono essere in ordine crescente, altrimenti il calcolo degli eventi non sarà corretto.

Un esempio di riga del file di input:

"Linux DB3" db.connections 1429533600 43

Per maggiori dettagli vedere l'opzione **--input-file**.

Se un valore con timestamp viene inviato per un host che si trova in una manutenzione di tipo “no data”, questo valore verrà scartato; tuttavia, è possibile inviare un valore con timestamp per un periodo di manutenzione scaduto e 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 voci delimitate da spazi: **<hostname> <key> <timestamp> <ns> <value>**.

Un esempio di riga del file di input:

”Linux DB3” db.connections 1429533600 7402561 43

Per maggiori dettagli vedere l’opzione **--input-file**.

-r, **--real-time**
Invia i valori uno per uno non appena vengono ricevuti. Questo può essere usato durante la lettura dallo standard input.

-g, **--group**
Raggruppa i valori per host e li invia a ciascun host in un batch separato.

--tls-connect *value*
Come connettersi al server o al proxy. Valori:

non crittografato
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 della CA o 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 consentito del certificato del server.

--tls-server-cert-subject *cert-subject*
Soggetto consentito del certificato del server.

--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 di identità PSK.

--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 versione di OpenSSL è inferiore 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). Sovrascrive i criteri predefiniti di selezione della suite di cifratura.

-v, **--verbose**
Modalità dettagliata, **-vv** per maggiori dettagli.

-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 i valori sono stati inviati e sono stati elaborati tutti con successo dal server. Se i dati sono stati inviati, ma l'elaborazione di almeno uno dei valori non è riuscito, lo stato di uscita è 2. Se l'invio dei dati è fallito, 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 per l'item **mysql.queries** del host monitorato. Usa il 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 per l'item **mysql.queries** del 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 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 usato 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 con 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 per l'item **mysql.queries** nel host **Monitored Host** al server con IP **192.168.1.113** usando TLS con 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 per l'item **mysql.queries** nel host **Monitored Host** al server con IP **192.168.1.113** usando TLS con chiave precondivisa (PSK).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8), **zabbix_web_service**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

STATO DI USCITA

ESEMPI

VEDI ANCHE

AUTORE

Questo documento è stato creato il: 08:42:39 GMT, June 11, 2021

zabbix_server

Sezione: Comandi di manutenzione (8)
Aggiornato: 2020-09-04
Indice Torna al contenuto principale

NOME

zabbix_server - Demone del server Zabbix

SINOSSI

zabbix_server [**-c** *config-file*]
zabbix_server [**-c** *config-file*] **-R** *runtime-option*
zabbix_server [**-c** *config-file*] **-T**
zabbix_server -h
zabbix_server -V

DESCRIZIONE

zabbix_server è il demone principale del software Zabbix.

OPZIONI

-c, **--config** *config-file*
Usa il file *config-file* alternativo invece di quello predefinito.

-f, **--foreground**
Esegue Zabbix server in primo piano.

-R, **--runtime-control** *runtime-option*
Esegue funzioni amministrative in base a *runtime-option*.

-T, **--test-config**
Convalida il file di configurazione ed esce.

-h, **--help**
Visualizza questa guida ed esce.

-V, **--version**
Mostra le informazioni sulla versione ed esce.

Esempi di esecuzione di Zabbix server con parametri della riga di comando:

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

CONTROLLO DI RUNTIME

Opzioni di controllo di runtime:

config_cache_reload
Ricarica la cache di configurazione.
Ignorato se la cache è attualmente in fase di caricamento.
Verrà utilizzato 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, cancella le proprietà SNMP (engine time, engine boots, engine id, credenziali) per tutti gli host.
Si noti che Zabbix memorizza nella cache le mappature SNMPv3 EngineID→IP e può riutilizzare automaticamente gli EngineID per ridurre il sovraccarico di polling.

housekeeper_execute
Esegue l'housekeeper.
Ignorato se l'housekeeper è attualmente in esecuzione.

trigger_housekeeper_execute
Esegue il trigger housekeeper (rimuove i problemi per i trigger eliminati).
Ignorato se il trigger housekeeper per i servizi è attualmente in esecuzione.

diaginfo[=*section*]
Registra nei log le informazioni diagnostiche interne della sezione specificata.
La sezione può essere *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks*.
Per impostazione predefinita vengono registrate le informazioni diagnostiche di tutte le sezioni.

ha_status
Registra nei log lo stato del cluster ad alta disponibilità (HA).

ha_remove_node[=*target*]
Rimuove il nodo high availability (HA) specificato tramite nome o ID. Si noti che i nodi attivi/in standby non possono essere rimossi.

ha_set_failover_delay[=*delay*]
Imposta il ritardo di failover high availability (HA). Sono supportati i suffissi temporali, ad esempio 10s, 1m.

proxy_config_cache_reload[=*target*] Ricarica la cache di configurazione del proxy.

secrets_reload
Ricarica i segreti da Vault.

service_cache_reload
Ricarica la cache del gestore dei servizi.

prof_enable[=*target*]
Abilita la profilazione. Influisce su tutti i processi se target non è specificato. La profilazione abilitata fornisce dettagli di tutti i rwlock/mutex per nome funzione.

prof_disable[=*target*]
Disabilita la profilazione. Influisce su tutti i processi se target non è specificato.

log_level_increase[=*target*]
Aumenta il livello di log; influisce su tutti i processi se target non è specificato

log_level_decrease[=*target*]
Diminuisce il livello di log; influisce su tutti i processi se target non è specificato

Destinazioni del controllo del livello di log *process-type*
Tutti i processi del tipo specificato (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Tipo di processo e numero (ad esempio, poller,3)

pid
Identificatore di processo, fino a 65535. Per valori più grandi specificare la destinazione come "process-type,N"

Destinazioni del controllo della profilazione

process-type
Tutti i processi del tipo specificato (alerter, alert manager, availability manager, browser poller, configuration syncer, configuration syncer worker, connector manager, connector worker, discovery manager, escalator, ha manager, history poller, history syncer, housekeeper, http poller, icmp pinger, internal poller, ipmi manager, ipmi poller, java poller, odbc poller, poller, agent poller, http agent poller, snmp poller, preprocessing manager, proxy group manager, proxy poller, self-monitoring, service manager, snmp trapper, task manager, timer, trapper, unreachable poller, vmware collector)

process-type,N
Tipo di processo e numero (ad esempio, history syncer,1)

pid
Identificatore di processo, fino a 65535. Per valori più grandi specificare la destinazione come ”process-type,N”

scope
L’ambito di profilazione (rwlock, mutex, processing) può essere utilizzato con il tipo di processo (ad esempio, history syncer,1,processing)

FILE

/usr/local/etc/zabbix_server.conf
Posizione predefinita del file di configurazione di Zabbix server (se non modificata in fase di compilazione).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

FILE

VEDI ANCHE

AUTORE

Questo documento è stato creato il: 16:12:14 GMT, 04 settembre 2020

zabbix_web_service

Sezione: Comandi di manutenzione (8)
Aggiornato: 2019-01-29
Indice Torna al contenuto principale

NOME

zabbix_web_service - Zabbix web service

SINOSSI

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

DESCRIZIONE

zabbix_web_service è un'applicazione per fornire servizi web ai componenti Zabbix.

OPZIONI

-c, **--config** *config-file* Utilizza il *config-file* alternativo al predefinito.

-T, **--test-config** Convalida il file di configurazione ed esce.

-h, **--help** Visualizza questa guida ed esce.

-V, **--version** Visualizza le informazioni sulla versione ed esce.

FILE

/usr/local/etc/zabbix_web_service.conf Posizione predefinita del file di configurazione di Zabbix web service (se non modificata in fase di compilazione).

VEDI ANCHE

Documentazione https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Indice

NOME

SINOSSI

DESCRIZIONE

OPZIONI

FILE

VEDI ANCHE

AUTORE

Questo documento è stato creato il: 12:58:30 GMT, June 11, 2021

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.