Documentation 8.0

ZABBIX

14.04.2026

Contents

Lietotāja rokasgrāmata **6**

Laipni lūdzam Zabbix dokumentācijā **6**

Sāciet darbu ar Zabbix . . . 6
Zabbix Cloud . . . 6
Izstrādātāju centrs . . . 6
Kopiena un citi resursi . . . 6
1 Instalēšana un pirmie soļi . . . 6
1 Tīmekļa saskarnes instalēšana . . . 7
2 Īsās darba sākšanas rokasgrāmatas . . . 13
3 Instalēšana: papildu ceļveži . . . 33
4 Jaunināšana . . . 71
5 Zabbix 8.0 jauninā šanas piezīmes . . . 86
Prasības . . . 87
Zināmās problēmas . . . 96
2 Kas jauns Zabbix 8.0 . . . 104
Veidnes . . . 105
Vienumi . . . 106
Spraudņi . . . 106
Zema līmeņa atklāšana . . . 107
Procesi . . . 107
Autentifikācija . . . 107
Logdaļas . . . 108
Lietotāja saskarne . . . 108
Dokumentācija . . . 110
Kas jauns Zabbix 8.0.x . . . 110
3 Zabbix appliance . . . 110
4 Zabbix procesi . . . 113
1 Serveris . . . 114
2 Aģents . . . 123
3 Aģents 2 . . . 126
4 Starpniekserveris . . . 137
5 Java gateway . . . 141
6 Sūtītājs . . . 145
7 Iegūt . . . 146
8 JS . . . 147
9 Tīmekļa pakalpojums . . . 148
5 Konfigurācija . . . 148
1 Hosti un hostu grupas . . . 158
2 Vienumi . . . 176
3 Problēmu noteikšana ar trigeriem . . . 385
4 Notikumi . . . 434
5 Notikumu korelācija . . . 438
6 Marķēšana . . . 444
7 Vizualizācija . . . 447
8 Veidnes un veidņu grupas . . . 473
9 Gatavas veidnes . . . 473
10 Paziņojumi par notikumiem . . . 486
11 Makrosi . . . 533
12 Lietotāji un lietotāju grupas . . . 548
13 Slepeno datu glabāšana . . . 557

14 Plānotie pārskati . . . 564
15 Datu eksportēšana . . . 567
6 Pakalpojumu uzraudzība . . . 573
1 Pakalpojumu koks . . . 574
2 SLA . . . 577
3 Iestatīšanas piemērs . . . 579
7 Tīmekļa uzraudzība . . . 583
1 Tīmekļa uzraudzības vienumi . . . 591
2 Reālas dzīves scenārijs . . . 593
8 Virtuālo mašīnu uzraudzība . . . 600
1 VMware uzraudzības vienumu atslēgas . . . 602
2 Virtuālās mašīnas atklāšanas atslēgu lauki . . . 620
3 JSON piemēri VMware vienumiem . . . 624
4 VMware uzraudzības iestatīšanas piemērs . . . 628
9 Uzturēšana . . . 633
10 Regulārās izteiksmes . . . 637
11 Problēmas apstiprināšana . . . 643
1 Problēmu apspiešana . . . 645
12 Konfigurācijas eksportēšana/importēšana . . . 646
1 Veidņu grupas . . . 647
2 Hostu grupas . . . 648
3 Veidnes . . . 648
4 Hosti . . . 665
5 Tīkla kartes . . . 681
6 Mediju tipi . . . 689
7 Dashboards . . . 696
13 Atklāšana . . . 707
1 Tīkla atklāšana . . . 707
2 Aktīvā aģenta automātiskā reģistrācija . . . 716
3 Zema līmeņa atklāšana . . . 720
14 Izkliedētā uzraudzība . . . 773
1 Starpniekserveri . . . 774
15 Šifrēšana . . . 782
1 Sertifikātu izmantošana . . . 789
2 Iepriekš koplietotu atslēgu izmantošana . . . 796
3 Problēmu novēršana . . . 798
16 Tīmekļa saskarne . . . 801
1 Izvēlne . . . 801
2 Lietotāja saskarnes sadaļas . . . 809
3 Lietotāja iestatījumi . . . 1062
4 Globālā meklēšana . . . 1067
5 Lietotāja saskarnes uzturēšanas režīms . . . 1069
6 Lapas parametri . . . 1070
7 Definīcijas . . . 1073
8 Savas tēmas izveide . . . 1073
9 Atkļūdošanas režīms . . . 1074
10 Zabbix izmantotās sīkdatnes . . . 1075
11 Laika joslas . . . 1076
12 Paroles atiestatīšana . . . 1077
13 Laika perioda un hostu atlasītāji . . . 1077
17 Labākā prakse . . . 1079
1 Drošības labākā prakse . . . 1079
2 Konfigurācijas labākā prakse . . . 1087
18 API . . . 1088
Metožu atsauce . . . 1093
1. pielikums. Atsauces komentāri . . . 1903
2. pielikums. Izmaiņas no 7.4 uz 8.0 . . . 1909
3. pielikums. Izmaiņas 8.0 versijā . . . 1910
19 Paplašinājumi . . . 1910
1 Ielādējamie moduļi . . . 1912
2 Lietotāja saskarnes moduļi . . . 1921
20 Pielikumi . . . 1922
1 Instalēšana un iestatīšana . . . 1922

2 Procesa konfigurācija . . . 1981
3 Protokoli . . . 2067
4 Vienumi . . . 2092
5 Atbalstītās funkcijas . . . 2127
6 Makrosi . . . 2128
7 Vienību simboli . . . 2162
8 Laika perioda sintakse . . . 2164
9 Komandu izpilde . . . 2165
10 Versiju saderība . . . 2166
12 Zabbix sender dinamisko saišu bibliotēka operētājsistēmai Windows . . . 2167
13 Pakalpojumu uzraudzības jaunināšana . . . 2167
14 Citas problēmas . . . 2168
16 Ekranēšanas piemēri . . . 2169
21 Īsās atsauces rokasgrāmatas . . . 2170
Pārskats . . . 2170
1. Monitor Linux with Zabbix agent . . . 2171
2 Windows uzraudzība ar Zabbix aģentu . . . 2174
3. Monitor Apache via HTTP . . . 2178
4. Monitor MySQL with Zabbix agent 2 . . . 2182
5 VMware uzraudzība ar Zabbix . . . 2190
6 Tīkla datplūsmas uzraudzība ar Zabbix . . . 2193
7 Tīkla datplūsmas uzraudzība, izmantojot aktīvās pārbaudes . . . 2196
8 Uzraugiet tīmekļa vietnes ar pārlūka vienumiem . . . 2199
9 Uzraugiet vietņu sertifikātus ar Zabbix aģents 2 (pasīvais) . . . 2203
10 Tīkla komutatora vai maršrutētāja uzraudzība ar Zabbix . . . 2208
11 Windows notikumu žurnāla uzraudzība, izmantojot aktīvās pārbaudes . . . 2215
Kas ir Zabbix . . . 2219

Zabbix Cloud **2223**

Zabbix Cloud **2223**
Iestatiet un pārvaldiet Zabbix Cloud . . . 2223
Uzziniet par Zabbix Cloud . . . 2223
Zabbix izvietošana mākonī . . . 2224
Mezglu konfigurācija . . . 2228
Pārskats . . . 2228
Pārskata cilne . . . 2229
Piekļuves filtru cilne . . . 2230
Šifrēšanas cilne . . . 2231
Dublējumu cilne . . . 2232
Vēstures cilne . . . 2232
Uzturēšanas cilne . . . 2233
Jaunināšanas cilne . . . 2234
Lietotāju pievienošana . . . 2236
Ievads . . . 2236
Lietotāju pievienošana organizācijām . . . 2236
Lietotāju pievienošana mezgliem . . . 2237
Zabbix Cloud galvenās atšķirības . . . 2238
Ievads . . . 2238

Izstrādātāju centrs **2240**

Izstrādātāju centrs **2240**
Veidojiet uz Zabbix bāzes . . . 2240
Sprauduņa izveide . . . 2240
Sprauduņu saskarnes . . . 2245
Python bibliotēka Zabbix համալպոգai . . . 2247
Instalēšana . . . 2247
Ātrās darba sākšanas ceļvedis . . . 2247
Zabbix API izmantošana . . . 2251
Datu vākšana no Zabbix aģenta . . . 2254
Sūtīt datus uz Zabbix serveri vai starpniekserveri . . . 2255
Atkļūdošanas žurnalēšana . . . 2259
Moduļi . . . 2259

Moduļa failu struktūra . . . 2260
Logrīki . . . 2267
Pamācības . . . 2276
Piemēri . . . 2302
Izmaiņas paplašinājumu izstrādē . . . 2302

Zabbix rokasgrāmatas lapas **2303**
zabbix_agent2 . . . 2303
NOSAUKUMS . . . 2303
KOPSAVILKUMS . . . 2303
IESPĒJAS . . . 2303
DATNES . . . 2304
SKATĪT ARĪ . . . 2304
Saturs . . . 2304
zabbix_agentd . . . 2305
NOSAUKUMS . . . 2305
KOPSAVILKUMS . . . 2305
APRAKSTS . . . 2305
IESPĒJAS . . . 2305
FAILI . . . 2306
SKATIET ARĪ . . . 2306
Saturs . . . 2306
zabbix_get . . . 2306
NOSAUKUMS . . . 2307
KOPSAVILKUMS . . . 2307
APRAKSTS . . . 2307
OPCIJAS . . . 2307
PIEMĒRI . . . 2308
SKATĪT ARĪ . . . 2308
Saturs . . . 2308
zabbix_js . . . 2309
NOSAUKUMS . . . 2309
KOPSAVILKUMS . . . 2309
APRAKSTS . . . 2309
OPCIJAS . . . 2309
PIEMĒRI . . . 2309
SKATĪT ARĪ . . . 2309
Saturs . . . 2310
zabbix_proxy . . . 2310
NOSAUKUMS . . . 2310
KOPSAVILKUMS . . . 2310
APRAKSTS . . . 2310
OPCIJAS . . . 2310
FAILI . . . 2311
SKATIET ARĪ . . . 2311
Indekss . . . 2312
zabbix_sender . . . 2312
NOSAUKUMS . . . 2312
KOPSAVILKUMS . . . 2312
APRAKSTS . . . 2313
OPCIJAS . . . 2313
IZEJAS STATUSS . . . 2315
PIEMĒRI . . . 2315
SKATĪT ARĪ . . . 2315
Saturs . . . 2316
zabbix_server . . . 2316
NOSAUKUMS . . . 2316
KOPSAVILKUMS . . . 2316
APRAKSTS . . . 2316
OPCIJAS . . . 2316
FAILI . . . 2318
SKATĪT ARĪ . . . 2318
Saturs . . . 2318

zabbix_web_service 2318
NOSAUKUMS 2318
KOPSAVILKUMS 2319
APRAKSTS 2319
OPCIJAS 2319
FAILI 2319
SKATĪT ARĪ 2319
Saturs 2319

Autortiesību paziņojums **2319**

Lietotāja rokasgrāmata

Laipni lūdzam Zabbix dokumentācijā

Jūsu galvenais resurss darbam ar Zabbix uzraudzību — no pamata iestatījumiem līdz sarežģītām konfigurācijām. Šī rokasgrāmata aptver visu nepieciešamo, lai instalētu, konfigurētu un darbinātu Zabbix.

Kas jauns Zabbix 8.0

Sāciet darbu ar Zabbix

Instalēšana Soli pa solim norādījumi, kā instalēt Zabbix jūsu izvēlētajā platformā, aptverot dažādas operētājsistēmas un datubāzu konfigurācijas.

Prasības Atbalstīto platformu un programmatūras priekšnosacījumu saraksts, kas palīdzēs jums sagatavot vidi veiksmīgai Zabbix izvietošanai.

Ātrās darba sākšanas rokasgrāmatas Īsas, uzdevumiem orientētas pamācības, kas soli pa solim iepazīstina ar pamatiem — no pirmajiem konfigurēšanas soļiem līdz pirmā problēmas brīdinājuma saņemšanai.

Zabbix Cloud

Sāciet lietot Zabbix Cloud Norādījumi soli pa solim, lai ātri palaistu un sagatavotu darbam savu Zabbix Cloud instanci, tostarp sākotnējo konfigurāciju un pārraugāmo ierīču pievienošanu.

Zabbix Cloud salīdzinājumā ar lokālu izvietojumu Salīdziniet atšķirības pārvaldībā, mērogojamībā un uzturēšanā starp abiem izvietošanas veidiem un uzziniet, kurš no tiem var visvienkāršāk atbalstīt jūsu darbplūsmu.

Iepazīstiet Zabbix Cloud Iegūstiet pārskatu par Zabbix Cloud, tostarp tā galvenajām iespējām, priekšrocībām un to, kas ir iekļauts abonementā. Biežāk uzdotos jautājumus par iestatīšanu un lietošanu skatiet BUJ.

Izstrādātāju centrs

Lietotāja saskarnes moduļi Rokasgrāmatas un atsauces pielāgotu moduļu izveidei, kas paplašina vai modificē Zabbix lietotāja saskarni, lai tā atbilstu konkrētiem lietošanas gadījumiem.

Logrīki Logrīku struktūras un loģikas pārskats, kā arī norādījumi pielāgotu paneļa elementu izveidei atbilstoši jūsu vajadzībām.

Spraudņi Pārskats par to, kā izstrādāt un pārvaldīt Zabbix spraudņus, lai paplašinātu funkcionalitāti vai integrētu ar ārējām sistēmām.

Kopiena un citi resursi

Zabbix forumi Vieta, kur apmainīties ar idejām, atrast risinājumus un dalīties pieredzē visos Zabbix zināšanu līmeņos.

Zabbix emuārs Ziņas, pamācības un gadījumu izpētes, ko atlasījusi Zabbix komanda un kopienas līdzautori.

Pamācību video Videobibliotēka ar demonstrācijām, diskusijām un padomiem, kas palīdzēs jums maksimāli izmantot Zabbix iespējas.

1 Instalēšana un pirmie soļi

Lai iestatītu Zabbix no nulles, izpildiet šo procesu:

1. Pārliecinieties, ka nepieciešamā programmatūra ir instalēta un darbojas.
2. Instalējiet Zabbix.
3. Instalējiet Zabbix tīmekļa lietotāja saskarni.
4. Sāciet uzraudzību.

Tālāk ir sniegta sīkāka informācija par katru soli.

Nepieciešamā programmatūra Zabbix labi darbojas ar šādu programmatūru (to bieži sauc par LAMP steku):

- **Linux operētājsistēma:** Ubuntu, Debian, CentOS vai citas
- **Tīmekļa serveris:** Apache vai NGINX
- **Datu bāzes serveris:** MySQL/MariaDB vai PostgreSQL
- **PHP**

Lielākajā daļā mūsdienu Linux sistēmu PHP jau var būt instalēts. Tādā gadījumā jums, iespējams, būs tikai jāiestata datu bāzes serveris un tīmekļa serveris.

Plašāku informāciju, piemēram, par atbalstītajām programmatūras versijām, skatiet sadaļā Nepieciešamā programmatūra.

Instalējiet Zabbix **Vienkāršākais** veids, kā instalēt Zabbix, ir izmantot **distribūcijas pakotnes**. Pakotnes instalē nelielas programmatūras atkarības.

Precīzas instalēšanas instrukcijas, kas pielāgotas jūsu operētājsistēmai, ir pieejamas **Zabbix vietnē**.

Izvēlieties savu platformu un izpildiet sniegtos norādījumus. Ieteicams sākumā instalēt Zabbix serveri, Zabbix aģents 2 un lietotāja saskarne (tīmekļa saskarni). Vēlāk varat pievienot Zabbix starpniekserveris un citus aģentus.

Daļa no instalēšanas procesa ir datu bāzes izveide (konfigurācijas un vēstures datu glabāšanai) un sākotnējo datu importēšana tajā.

Instalējiet Zabbix tīmekļa saskarni Pabeidziet tīmekļa saskarnes **instalēšanas vedņa** darbības.

Jums tiks prasīta datu bāzes parole (izveidota iepriekšējā solī), lai lietotāja saskarne varētu sazināties ar datu bāzi.

Sāciet uzraudzību Piesakieties Zabbix tīmekļa saskarnē.

Izmantojiet **ātrās sākšanas ceļvežus**, lai sāktu uzraudzīt populārus resursus (Linux, Apache, MySQL vai citus). Šie īsie ceļveži paskaidro, ar kurām veidnēm jums jāstrādā, lai sāktu uzraudzīt populārus resursus.

Iespējams, vēlēsities sākt ar ātrās sākšanas ceļvedi Linux servera uzraudzībai.

1 Tīmekļa saskarnes instalēšana

Šajā sadaļā ir sniegtas soli pa solim instrukcijas Zabbix tīmekļa saskarnes instalēšanai. Veiciet šīs darbības pēc tam, kad esat instalējis Zabbix aizmugursistēmu — Zabbix serveri, aģentu un datubāzi.

Zabbix lietotāja saskarne ir izstrādāta PHP valodā, tāpēc tās darbināšanai ir nepieciešams PHP atbalstīts tīmekļa serveris.

> **Note:**
> Vairāk par SSL iestatīšanu Zabbix lietotāja saskarnei varat uzzināt, iepazīstoties ar šīm labākajām praksēm.

Sveiciena ekrāns

Atveriet Zabbix lietotāja saskarnes URL pārlūkprogrammā. Ja esat instalējis Zabbix no pakotnēm, URL ir:

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

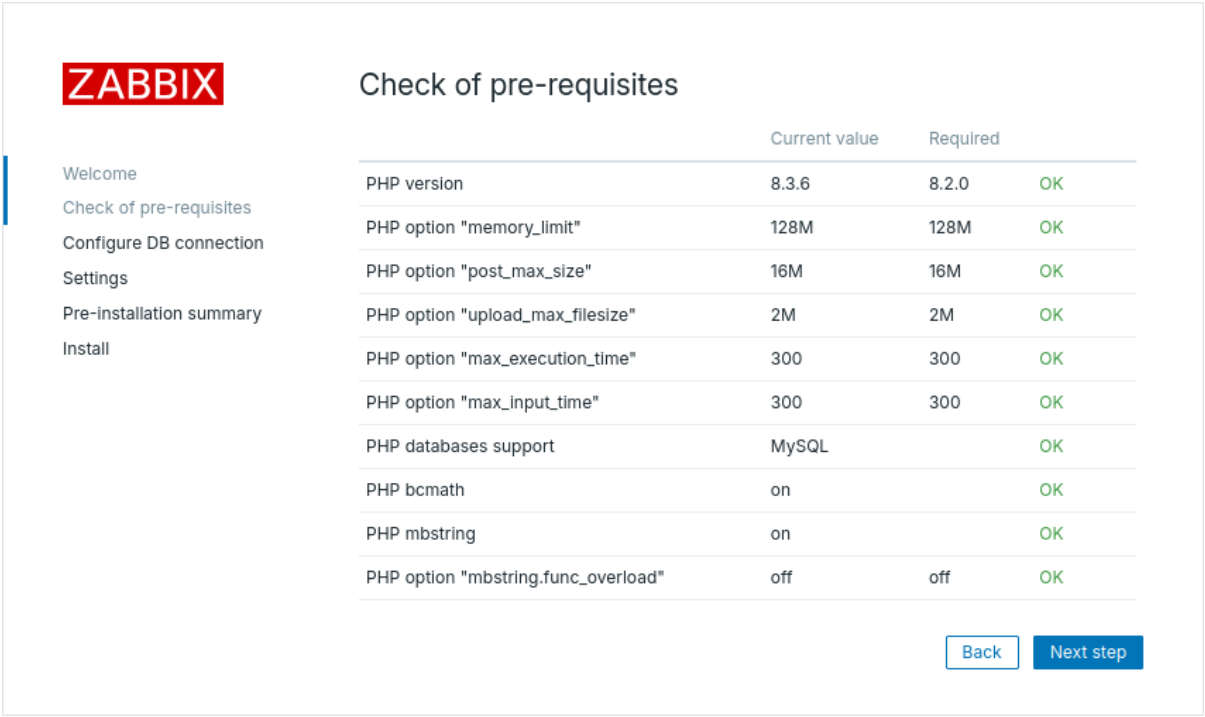
Jums vajadzētu redzēt lietotāja saskarnes instalēšanas vedņa pirmo ekrānu.

Izmantojiet nolaižamo izvēlni *Default language*, lai mainītu sistēmas noklusējuma valodu un turpinātu instalēšanas procesu izvēlētajā valodā (pēc izvēles). Plašāku informāciju skatiet Papildu lietotāja saskarnes valodu instalēšana.

Ņemiet vērā, ka, iestatot valodu uz *English (en_US)*, lietotāja saskarnē tiks iespējots arī ASV laika/datuma formāts.

Priekšnosacījumu pārbaude

Pārliecinieties, ka ir izpildīti visi Zabbix lietotāja saskarnei obligātie priekšnosacījumi.



Priekšnosacījums	Minimālā vērtība	Apraksts
PHP versija	8.2.0	
PHP opcija "memory_limit"	128MB	Failā php.ini: memory_limit = 128M
PHP opcija "post_max_size"	16MB	Failā php.ini: post_max_size = 16M
PHP opcija "upload_max_filesize"	2MB	Failā php.ini: upload_max_filesize = 2M
PHP opcija "max_execution_time"	300 sekundes	Failā php.ini: max_execution_time = 300 (ir atļautas arī vērtības 0 un -1)
PHP opcija "max_input_time"	300 sekundes	Failā php.ini: max_input_time = 300 (ir atļautas arī vērtības 0 un -1)

Priekšnosacījums	Minimālā vērtība	Apraksts
PHP datubāzu atbalsts	Viens no: MySQL, PostgreSQL	Visu obligāto un neobligāto PHP paplašinājumu sarakstu skatiet sadaļā Prasības. Ņemiet vērā, ka neizpildīti neobligātie priekšnosacījumi tiek parādīti ar sarkanu *Brīdinājums* statusu, un iestatīšanas procesu var turpināt arī tad, ja tie nav izpildīti.
PHP bcmath	jābūt iespējotam	
PHP mbstring	jābūt iespējotam	
PHP opcija "mbstring.func_overload"	jābūt atspējotai	Failā php.ini: mbstring.func_overload = 0
PHP opcija "session.auto_start"	jābūt atspējotai	Failā php.ini: session.auto_start = 0
PHP opcija "arg_separator.output"	&	Failā php.ini: arg_separator.output = "&" (ir atļauta arī vērtība "&")

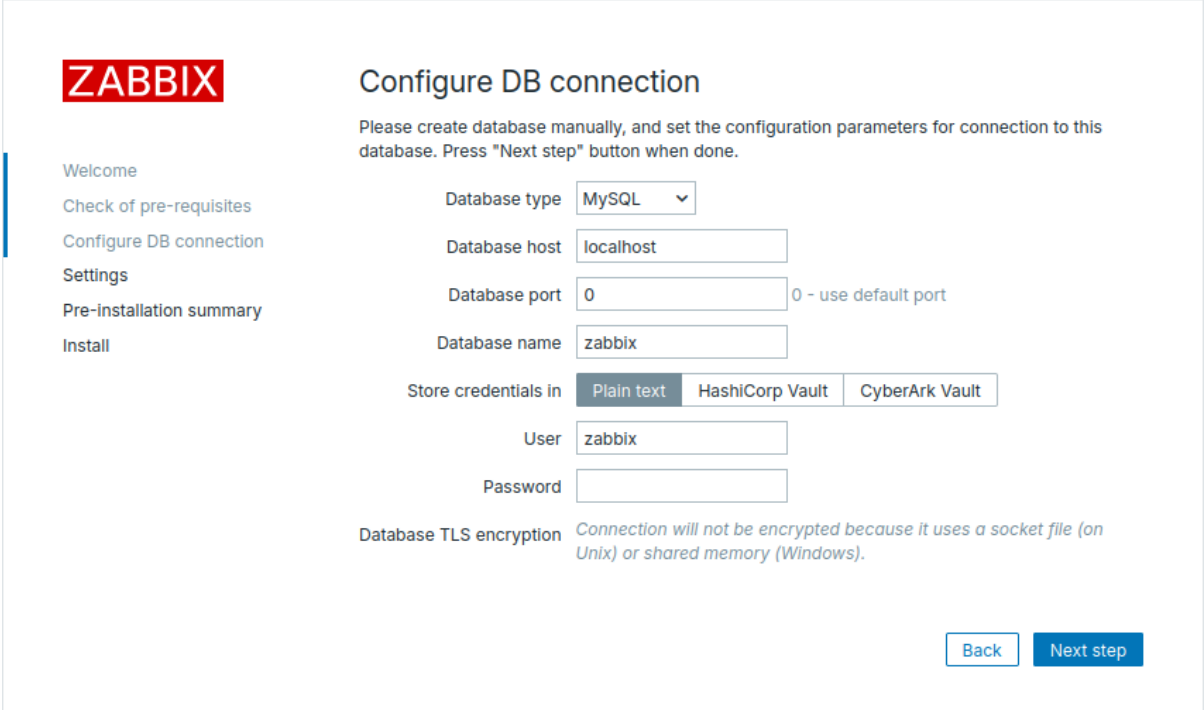
Attention:
Ja ir jāmaina Apache lietotājs vai lietotāju grupa, pārbaudiet sesiju mapes atļaujas; pretējā gadījumā Zabbix iestatīšanu, iespējams, nevarēs turpināt.

Konfigurēt DB savienojumu

Ievadiet informāciju savienojumam ar datubāzi. Zabbix datubāzei jau ir jābūt izveidotai.

MySQL gadījumā, ievadot `localhost` vai atstājot lauku *Database host* tukšu, savienojums tiks izveidots, izmantojot noklusējuma Unix ligzdu. Iestatīšanas forma nenodrošina atsevišķu lauku *Database socket*, tāpēc, lai izmantotu pielāgotu ligzdu, konfigurējiet to Zabbix servera iestatījumos (piemēram, ar `DBSocket=` parametrā zabbix_server.conf). Tas nodrošina, ka lietotāja saskarne ir saskaņota ar servera un datubāzes savienojuma iestatījumiem.

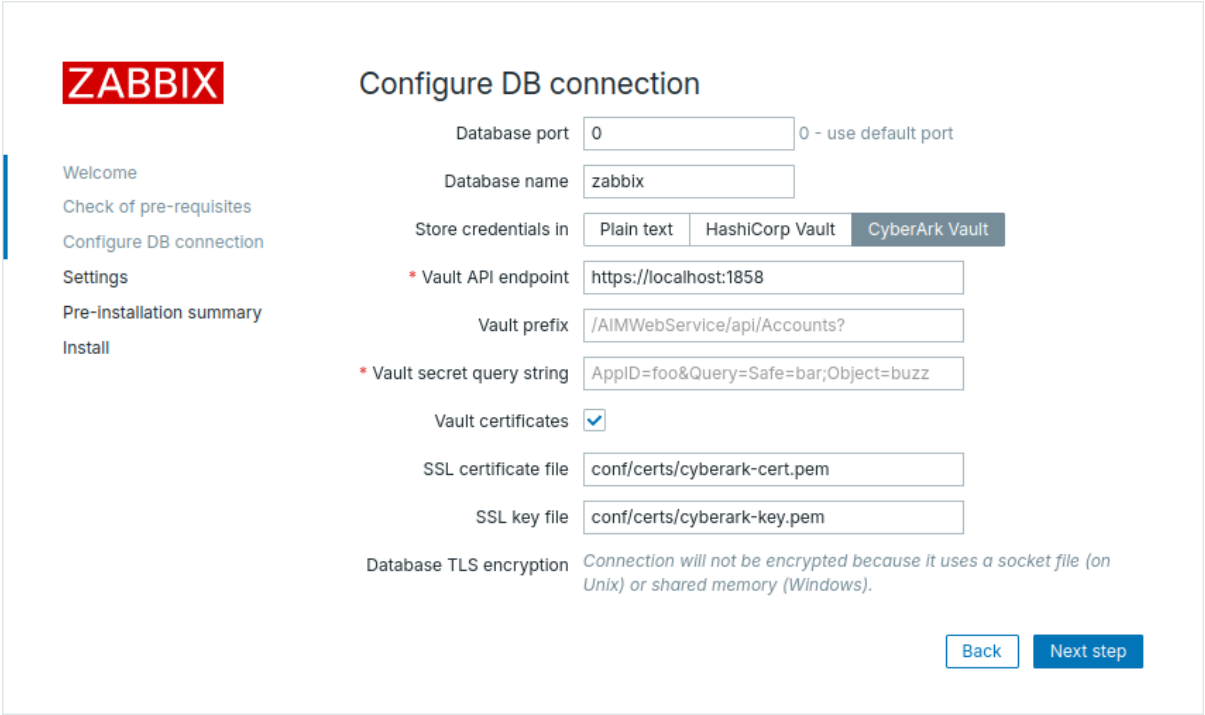
PostgreSQL gadījumā, ja lauks *Database host* ir atstāts tukšs, tiek izmantota noklusējuma Unix domēna ligzda. Ja tiek ievadīts ligzdas ceļš (piemēram, `/var/run/pgbouncer`), tā vietā tiks izmantota šī Unix domēna ligzda.



Ja ir atzīmēta opcija *Database TLS encryption*, formā parādīsies papildu lauki TLS savienojuma konfigurēšanai ar datubāzi (tikai MySQL vai PostgreSQL).

Ja *Store credentials in* ir iestatīts uz HashiCorp Vault vai CyberArk Vault, kļūs pieejami papildu parametri:

- HashiCorp Vault gadījumā: Vault API galapunkts, vault prefikss, slepenā ieraksta ceļš un autentifikācijas marķieris;
- CyberArk Vault gadījumā: Vault API galapunkts, vault prefikss, slepenā vaicājuma virkne un sertifikāti. Atzīmējot izvēles rūtiņu *Vault certificates*, parādīsies divi jauni lauki SSL sertifikāta faila un SSL atslēgas faila ceļu norādīšanai.

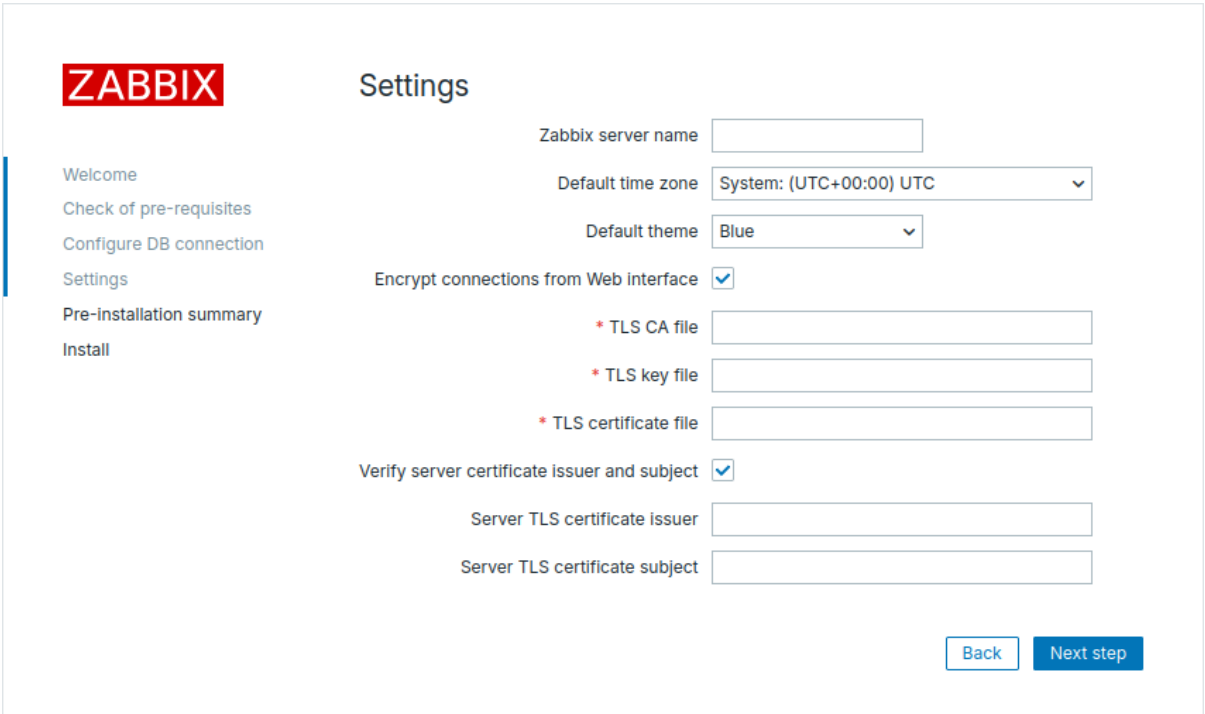


Iestatījumi

Zabbix servera nosaukuma ievadīšana nav obligāta, tomēr, ja tas ir norādīts, tas tiks attēlots izvēlnes joslā un lapu virsrakstos.

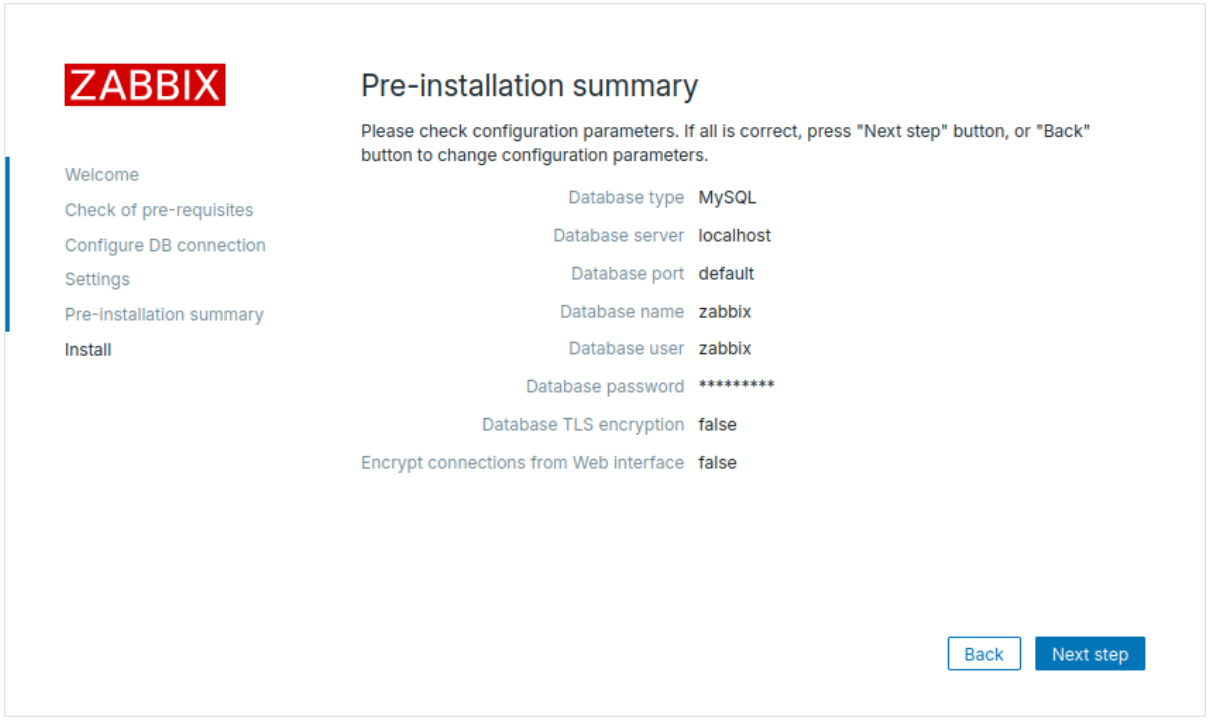
Iestatiet noklusējuma laika joslu un motīvu lietotāja saskarnei.

Ja ir atzīmēta opcija *Šifrēt savienojumus no tīmekļa saskarnes*, tad formā parādās papildu lauki TLS savienojuma konfigurēšanai starp Zabbix serveri un lietotāja saskarni.

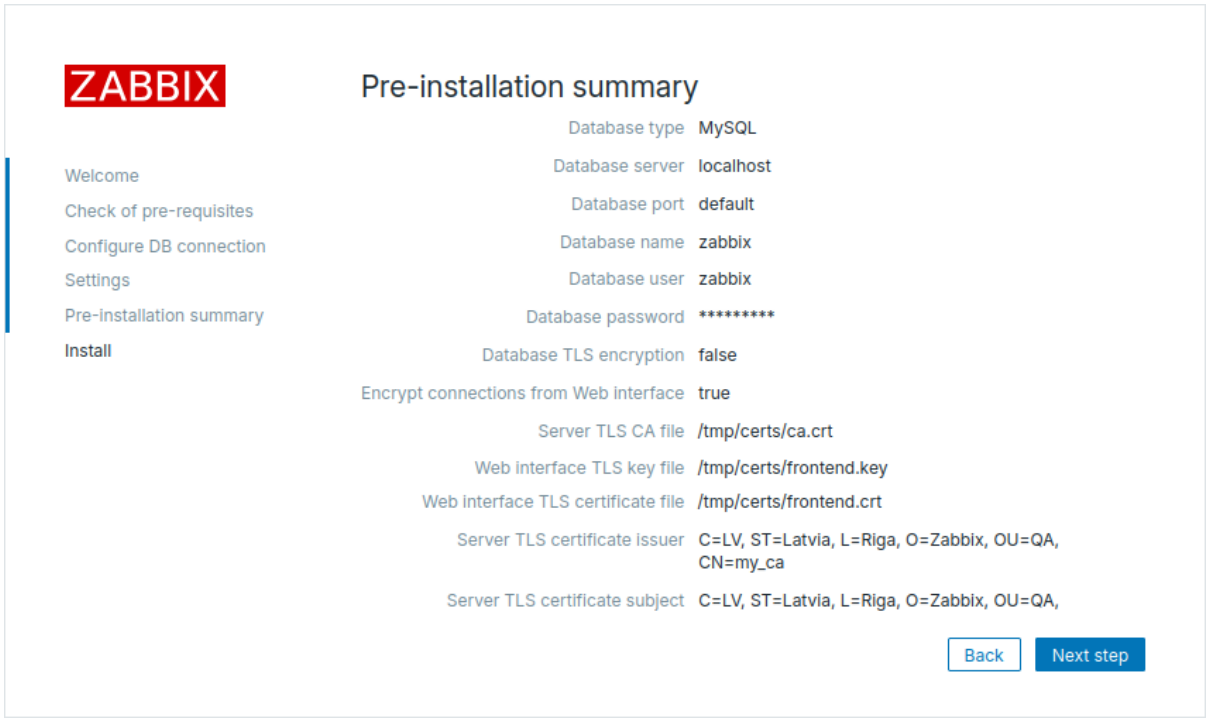


Pirmsinstalēšanas kopsavilkums

Pārskatiet iestatījumu kopsavilkumu.

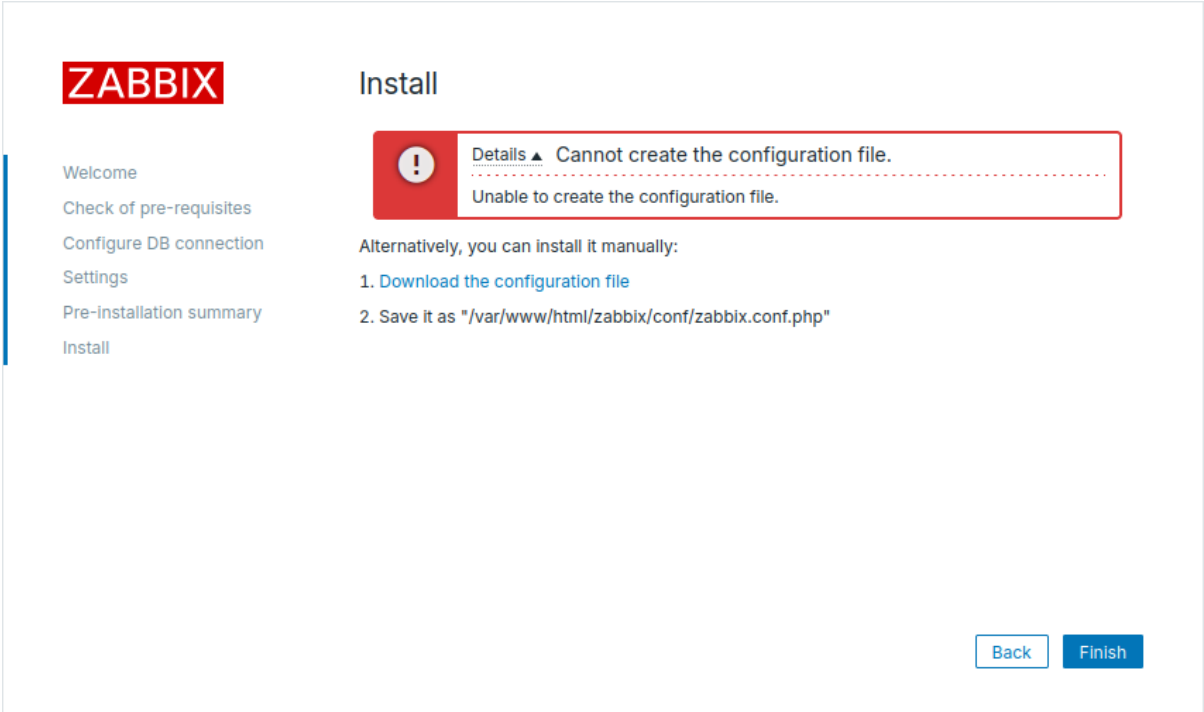


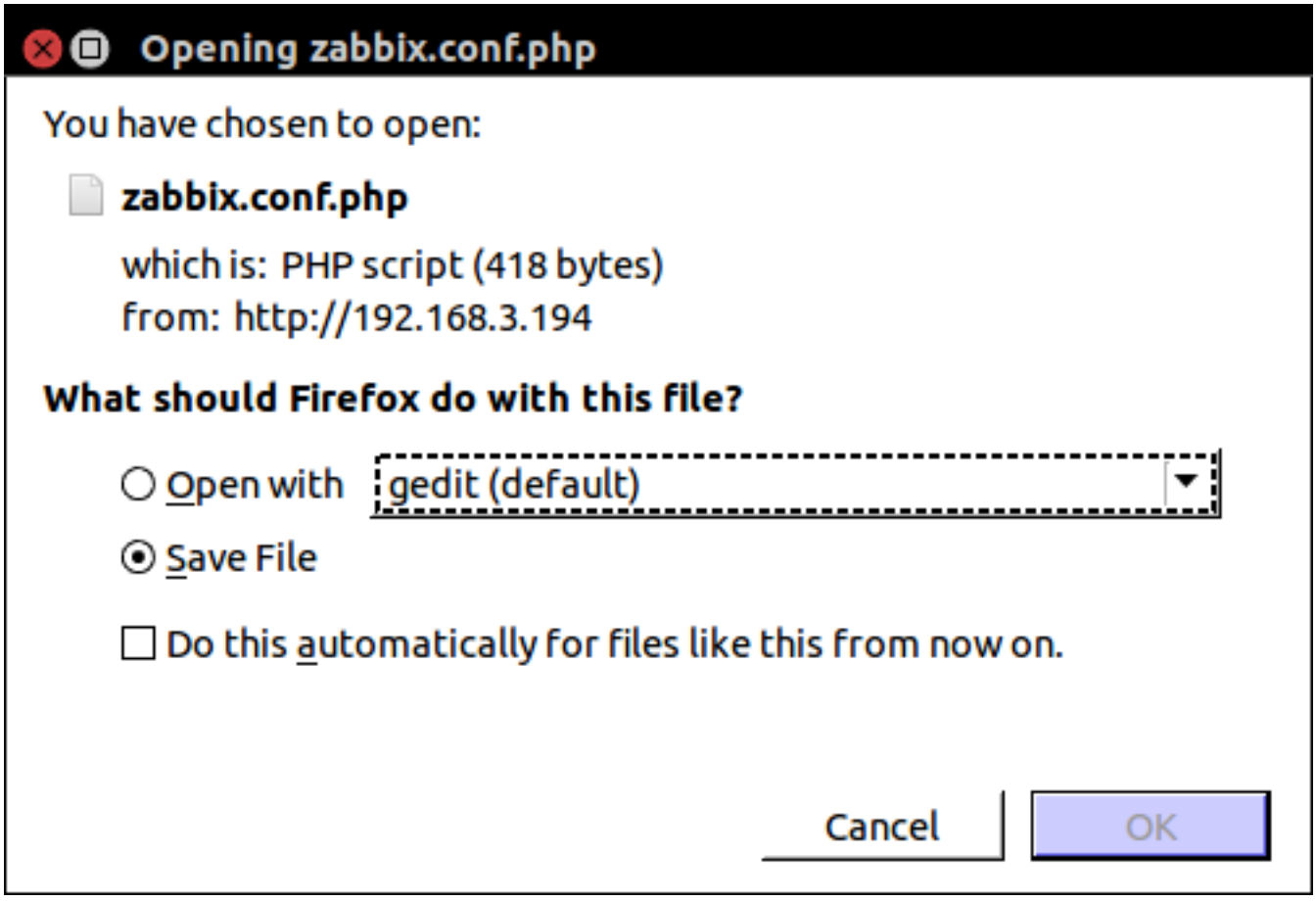
Apakšlapa parādīs datus, ja būs pievienota TLS konfigurācija.



Instalēšana

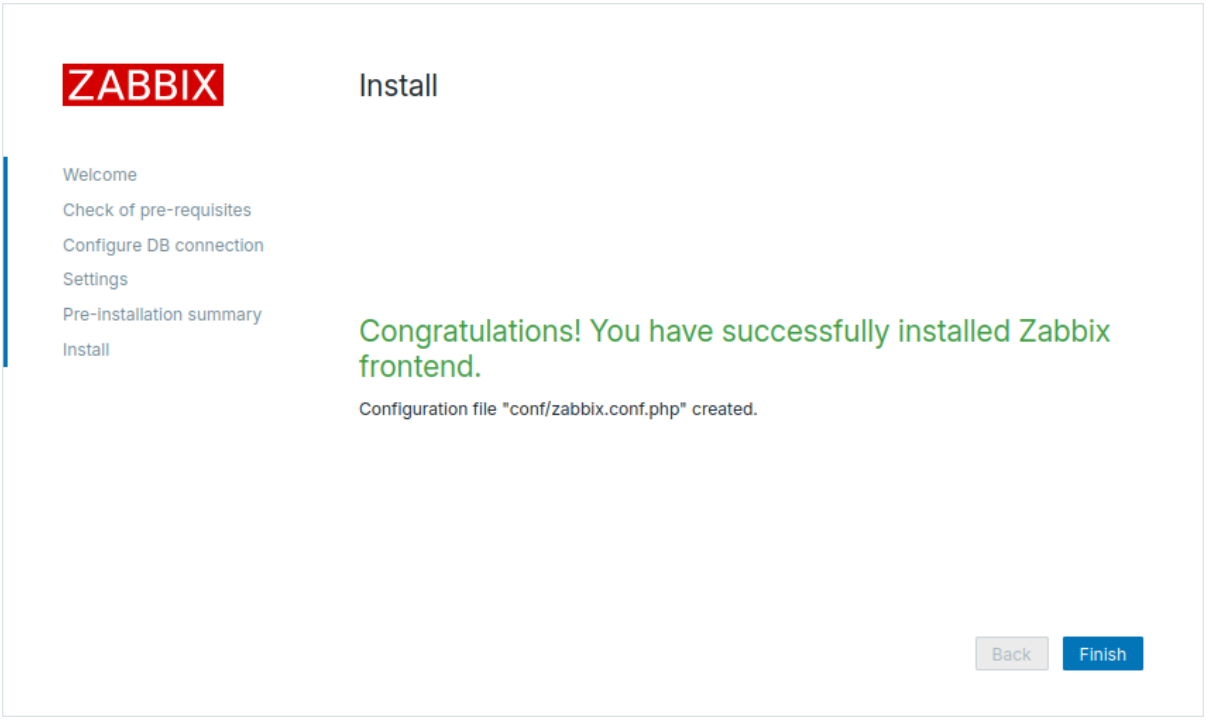
Ja instalējat Zabbix no pirmkoda, lejupielādējiet konfigurācijas failu un ievietojiet to mapē conf/ tīmekļa servera HTML dokumentu apakšdirektorijā, kurā nokopējāt Zabbix PHP failus.





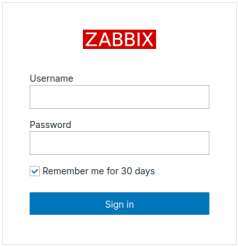
Note:
Ja tīmekļa servera lietotājam ir rakstīšanas piekļuve direktorijam conf/, konfigurācijas fails tiks saglabāts automātiski un būs iespējams nekavējoties pāriet uz nākamo soli.

Pabeidziet instalēšanu.



Pieteikšanās

Zabbix lietotāja saskarne ir gatava! Noklusējuma lietotājvārds ir **Admin**, parole — **zabbix**.



Turpiniet ar Linux uzraudzības ceļvedi.

2 Īsās darba sākšanas rokasgrāmatas

Pēc tam, kad esat instalējis Zabbix un tā lietotāja saskarni, ir laiks sākt uzraudzību.

Izmantojiet šīs īsās rokasgrāmatas, lai sāktu populāru resursu uzraudzību:

- Linux servera uzraudzība
- Apache tīmekļa servera uzraudzība
- MySQL servera uzraudzība

Īsajās darba sākšanas rokasgrāmatās galvenā uzmanība pievērsta oficiālo Zabbix veidņu izmantošanai uzraudzībai. Veidne ir iepriekš konfigurēts uzraudzības elementu kopums. Veidne ļauj mērogot uzraudzību — varat pārbaudīt veidni uz viena uzraudzības mērķa (ierīces, lietotnes u.c.), to pielāgot un pēc tam piemērot vairākiem līdzīgiem uzraudzības mērķiem.

Lai gan veidnes ir piemērotas uzraudzībai uzreiz pēc ieviešanas, jūs varētu interesēt arī to elementu pielāgošana.

Lai labāk izprastu pamatelementus, kas veido veidni, skatiet Pamatkonfigurācija.

Pamata konfigurācija

Šīs rokasgrāmatas iepazīstina ar Zabbix konfigurācijas pamatelementiem:

- Pieteikšanās un lietotāja konfigurēšana
- Jauns hosts
- Jauns vienums
- Jauns trigeris
- Problēmas paziņojuma saņemšana

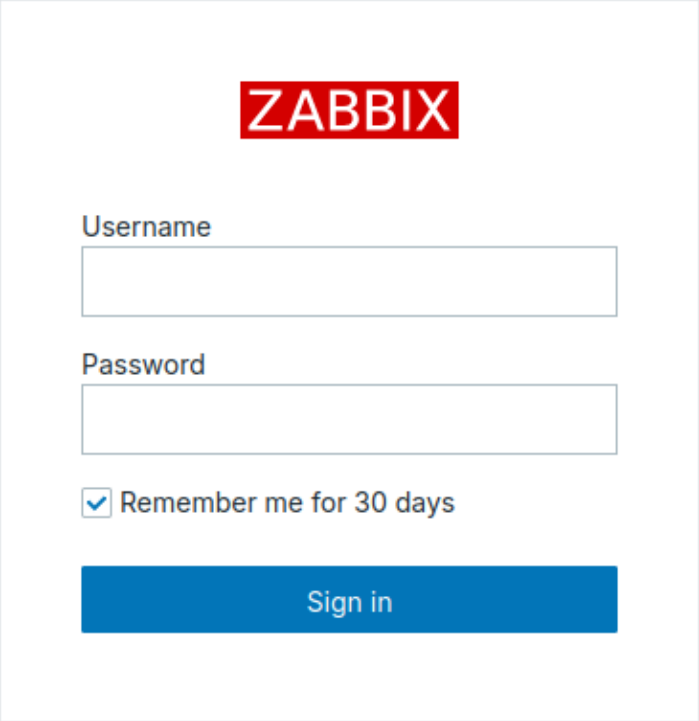
- Jauna veidne

1 Pieteikšanās un lietotāja konfigurēšana

Pārskats

Šajā sadaļā jūs uzzināsiet, kā pieteikties un iestatīt sistēmas lietotāju Zabbix.

Pieteikšanās



Šis ir Zabbix sveiciena ekrāns. Ievadiet lietotājvārdu **Admin** un paroli **zabbix**, lai pieteiktos kā Zabbix superlietotājs. Tiks piešķirta piekļuve visām izvēlnes sadaļām.

Drošības apsvērumu dēļ ir stingri ieteicams nekavējoties pēc pirmās pieteikšanās nomainīt Admin konta noklusējuma paroli.

Pastāvīga pieteikšanās

Lai paliktu pieteicies līdz 30 dienām, pirms noklikšķināt uz *Pierakstīties*, atlasiet *Atcerēties 30 dienas*.

Ja *Atcerēties 30 dienas* ir iespējots:

- Jūsu sesija paliek aktīva 30 dienas.
- *Automātiskā atteikšanās* tiek ignorēta, saglabājot jūs pieteiktu līdz perioda beigām.
- Turpmākajos apmeklējumos 30 dienu laikā jūs tiksiet automātiski pieteikts, atkārtoti neievadot akreditācijas datus.

Ja *Atcerēties 30 dienas* ir atspējots:

- Jebkāda iepriekš iespējota automātiskā pieteikšanās tiek notīrīta.
- Sesijas derīgums beigsies atbilstoši konfigurētajam *Automātiskās atteikšanās* intervālam.

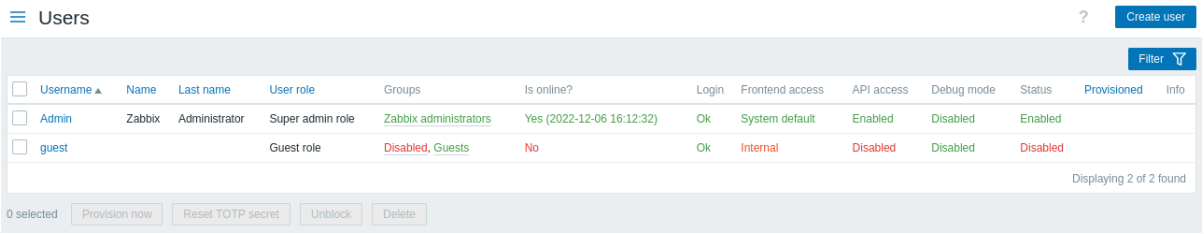
Aizsardzība pret brutāla spēka uzbrukumiem

Ja notiek pieci secīgi neveiksmīgi pieteikšanās mēģinājumi, Zabbix lietotāja saskarne apturēs darbību uz 30 sekundēm, lai novērstu brutāla spēka un vārdnīcas uzbrukumus.

Neveiksmīga pieteikšanās mēģinājuma IP adrese tiks parādīta pēc veiksmīgas pieteikšanās.

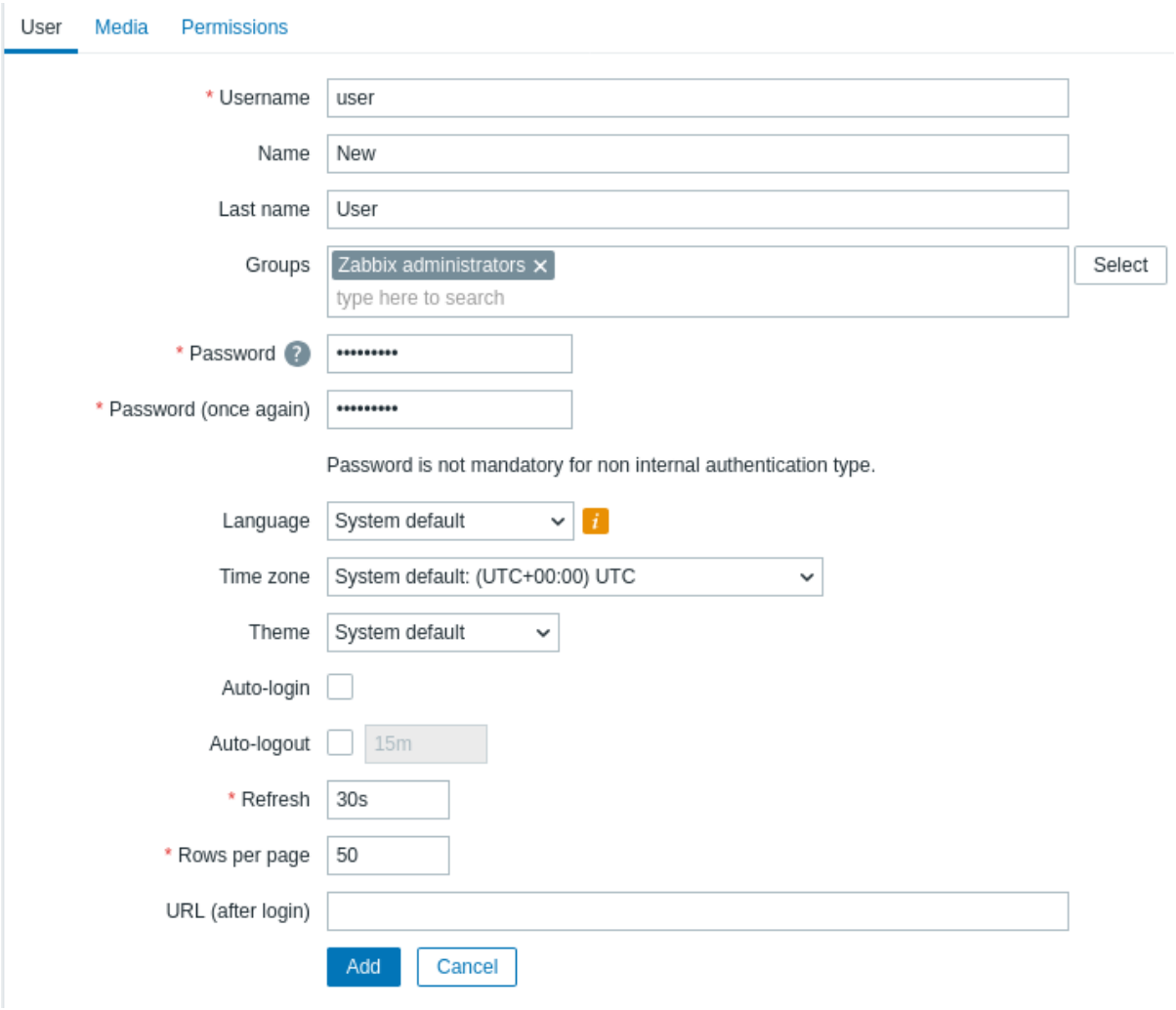
Lietotāja pievienošana

Lai skatītu informāciju par lietotājiem, sānjoslas vertikālajā izvēlnē dodieties uz *Users* > *Users*.



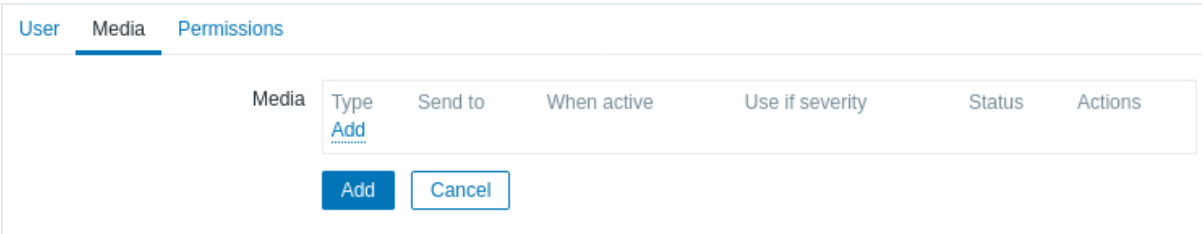
Lai pievienotu jaunu lietotāju, augšējā labajā stūrī atlasiet *Create user*.

Jaunā lietotāja formā noteikti pievienojiet savu lietotāju kādai no esošajām lietotāju grupām, piemēram, 'Zabbix administrators'.



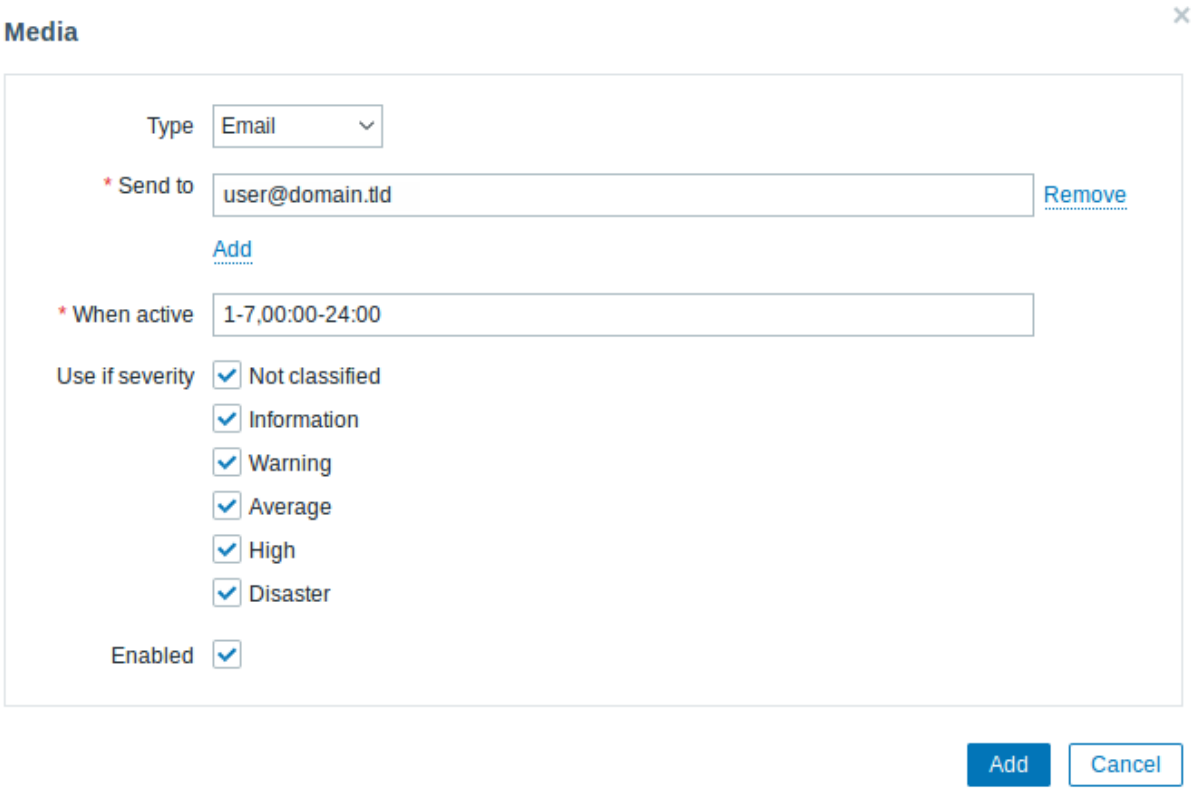
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti. Lai iegūtu sīkāku informāciju par ievades laukiem šajā konfigurācijas formā, skatiet lapu Lietotāja iestatījumi.

Pēc noklusējuma jaunajiem lietotājiem nav definēti mediji (paziņojumu piegādes metodes). Lai izveidotu mediju, dodieties uz cilni 'Media' un noklikšķiniet uz *Add*.



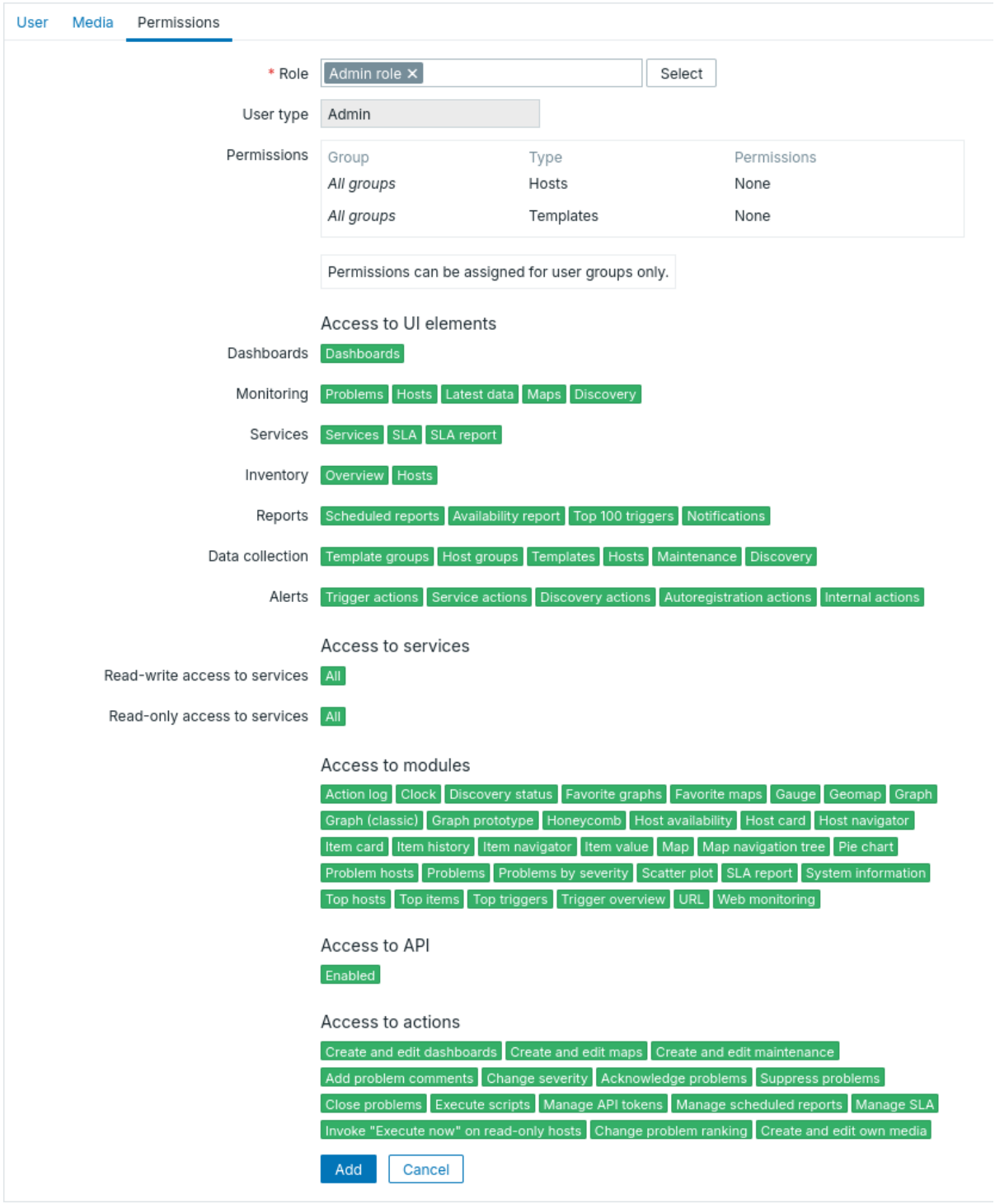
Uznirstošajā logā ievadiet lietotāja e-pasta adresi.

Varat norādīt laika periodu, kurā medijs būs aktīvs (formāta aprakstu skatiet lapā Laika perioda specifikācija). Pēc noklusējuma medijs vienmēr ir aktīvs. Jūs varat arī pielāgot trigera smaguma līmeņus, kuriem medijs būs aktīvs, taču pagaidām atstājiet tos visus iespējotus.



Noklikšķiniet uz *Add*, lai saglabātu mediju, pēc tam dodieties uz cilni Permissions.

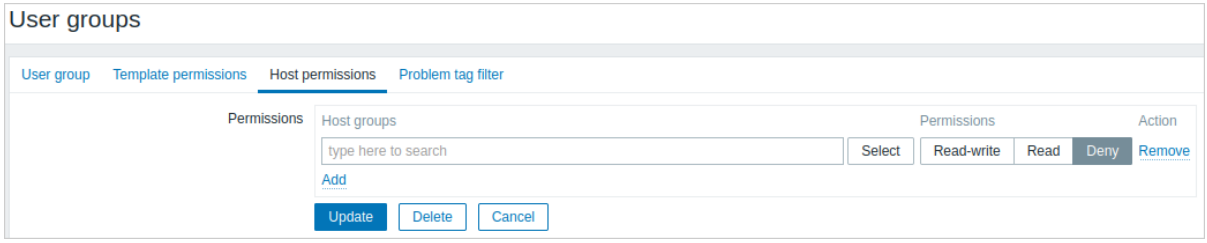
Cilnē Permissions ir obligāts lauks *Role*. Loma nosaka, kurus lietotāja saskarnes elementus lietotājs var skatīt un kādas darbības viņam ir atļauts veikt. Nospiediet Select un atlasiet vienu no lomām sarakstā. Piemēram, atlasiet *Admin role*, lai atļautu piekļuvi visām Zabbix lietotāja saskarnes sadaļām, izņemot Administration. Vēlāk varēsiet mainīt atļaujas vai izveidot vairāk lietotāju lomu. Pēc lomas atlasīšanas atļaujas parādīsies tajā pašā cilnē:



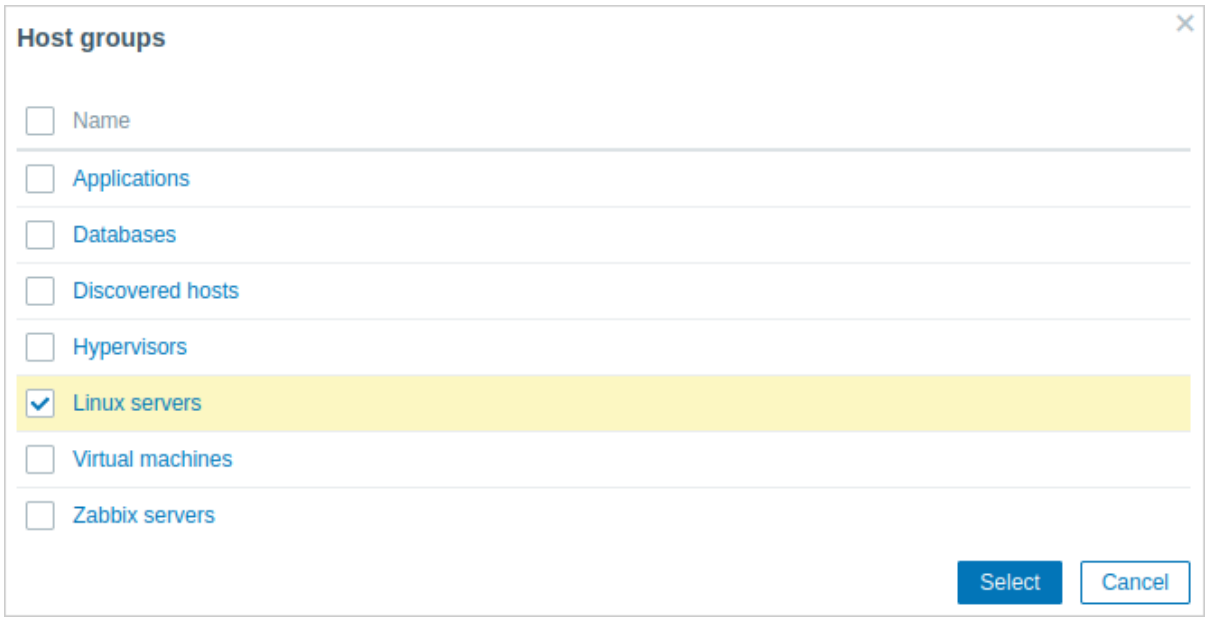
Noklikšķiniet uz *Add* lietotāja rekvizītu formā, lai saglabātu lietotāju. Jaunais lietotājs parādīsies lietotāju sarakstā.

Piekļuves tiesību piešķiršana

Pēc noklusējuma jaunam lietotājam nav piekļuves tiesību hostiem un veidnēm. Lai piešķirtu lietotājam tiesības, noklikšķiniet uz lietotāja grupas kolonnā *Groups* (šajā gadījumā — 'Zabbix administrators'). *User groups* rekvizītu formā atveriet cilni *Host permissions*, lai piešķirtu atļaujas hostu grupām. Noklikšķiniet uz Add, lai tiktu parādīts hostu grupu atlases lauks:



Pēc tam noklikšķiniet uz *Select* blakus laukam, lai skatītu hostu grupu sarakstu. Šim lietotājam jābūt tikai lasīšanas piekļuvei grupai *Linux servers*, tāpēc sarakstā atzīmējiet atbilstošo izvēles rūtiņu un noklikšķiniet uz *Select*, lai apstiprinātu savu izvēli.



Noklikšķiniet uz pogas *Read*, lai iestatītu atļauju līmeni, un pēc tam uz *Update*, lai saglabātu lietotāju grupas konfigurācijā veiktās izmaiņas.

Lai piešķirtu atļaujas veidnēm, jums būs jāpārslēdzas uz cilni *Template permissions* un jānorāda veidņu grupas. Darbības ir identiskas atļauju piešķiršanai hostu grupām. Veidņu pārskats ir pieejams šīs Ātrās darba sākšanas sadaļā New template.

Attention:
Zabbix vidē piekļuves tiesības hostiem un veidnēm tiek piešķirtas lietotāju grupām, nevis atsevišķiem lietotājiem.

Gatavs! Tagad varat mēģināt pieteikties, izmantojot jaunā lietotāja akreditācijas datus.

2 Jauns hosts

Pārskats

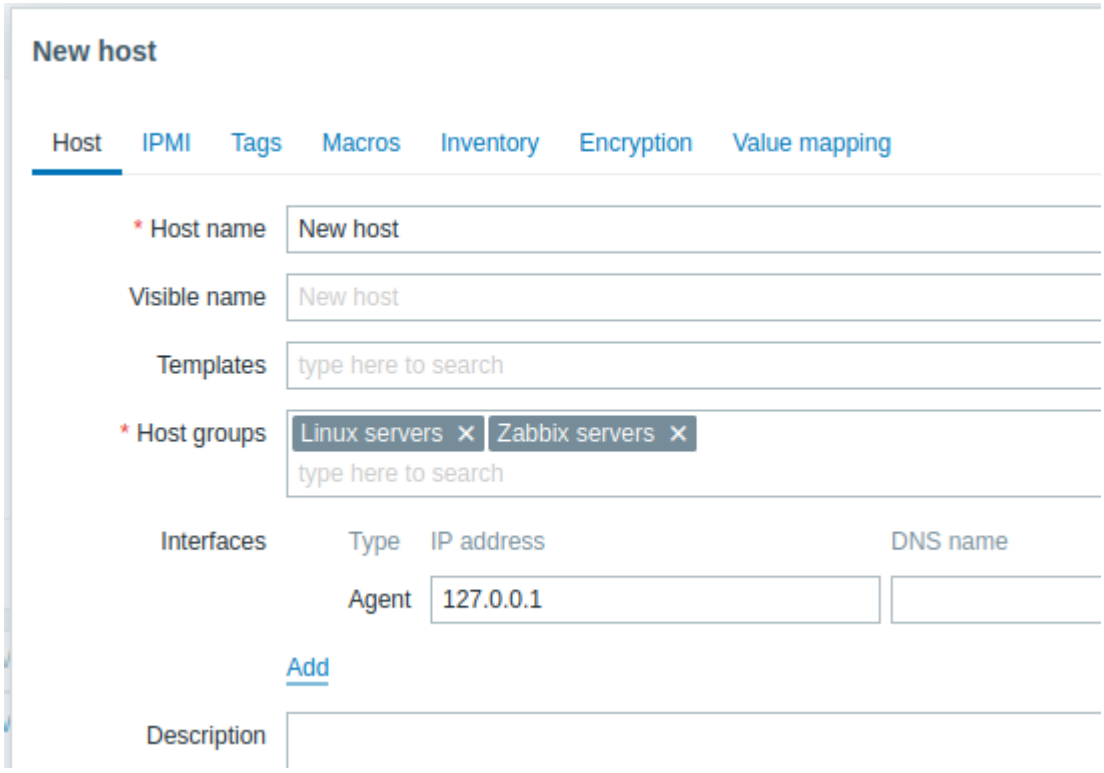
Šajā sadaļā jūs uzzināsiet, kā iestatīt jaunu hostu.

Hosts Zabbix sistēmā ir tīklā pieejama entītija (fiziska, virtuāla), kuru vēlaties uzraudzīt. Definīcija tam, kas Zabbix sistēmā var būt "hosts", ir diezgan elastīga. Tas var būt fizisks serveris, tīkla komutators, virtuāla mašīna vai kāda lietojumprogramma.

Hosta pievienošana

Informācija par konfigurētajiem hostiem Zabbix ir pieejama izvēlnes sadaļās *Datu ievākšana* > *Hosti* un *Monitorings* > *Hosti*. Jau ir viens iepriekš definēts hosts ar nosaukumu "Zabbix server", taču mēs vēlamies iemācīties pievienot vēl vienu.

Lai pievienotu jaunu hostu, noklikšķiniet uz *Izveidot hostu*. Tiks atvērta hosta konfigurācijas forma.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Minimālais nepieciešamais ievadāmais apjoms ir:

Hosta nosaukums

- Ievadiet hosta nosaukumu. Ir atļauti burtciparu simboli, atstarpes, punkti, defises un pasvītrojuma zīmes.

Hostu grupas

- Atlasiet vienu vai vairākas esošās grupas, noklikšķinot uz pogas *Atlasīt*, vai ievadiet neesošas grupas nosaukumu, lai izveidotu jaunu grupu.

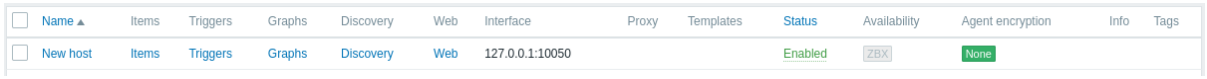
Note:
Visas piekļuves atļaujas tiek piešķirtas hostu grupām, nevis atsevišķiem hostiem. Tāpēc hostam ir jāpieder vismaz vienai grupai.

Saskarnes: IP adrese

- Lai gan tehniski tas nav obligāts lauks, hosta saskarne ir nepieciešama noteiktu metriku ievākšanai. Lai izmantotu Zabbix aģents pasīvās pārbaudes, šajā laukā norādiet aģents IP vai DNS. Ņemiet vērā, ka Zabbix aģents konfigurācijas failā direktīvā 'Server' jums jānorāda arī Zabbix serveris IP vai DNS. Ja Zabbix aģents un Zabbix serveris ir instalēti vienā un tajā pašā datorā, abās vietās jānorāda viens un tas pats IP/DNS.

Citas opcijas var atstāt nemainītas, izmantojot to noklusējuma vērtības.

Kad tas ir izdarīts, noklikšķiniet uz *Pievienot*. Jūsu jaunajam hostam vajadzētu būt redzamam hostu sarakstā.



Kolonnā Pieejamība ir ietverti hosta pieejamības indikatori katrai saskarnei. Mēs esam definējuši Zabbix aģents saskarni, tāpēc varam izmantot aģents pieejamības ikonu (ar 'ZBX' uz tās), lai noteiktu hosta pieejamību:

- ZBX - hosta statuss vēl nav noteikts; metriku pārbaude vēl nav notikusi
- ZBX - hosts ir pieejams, metriku pārbaude ir bijusi veiksmīga
- ZBX - hosts nav pieejams, metriku pārbaude ir neizdevusies (virziet peles kursoru virs ikonas, lai redzētu kļūdas ziņojumu). Iespējams, ir radusies kāda komunikācijas kļūda, ko varētu būt izraisījuši nepareizi saskarnes akreditācijas dati. Pārbaudiet, vai Zabbix serveris darbojas, un vēlāk mēģiniet arī atsvaidzināt lapu.

3 Jauns vienums

Pārskats

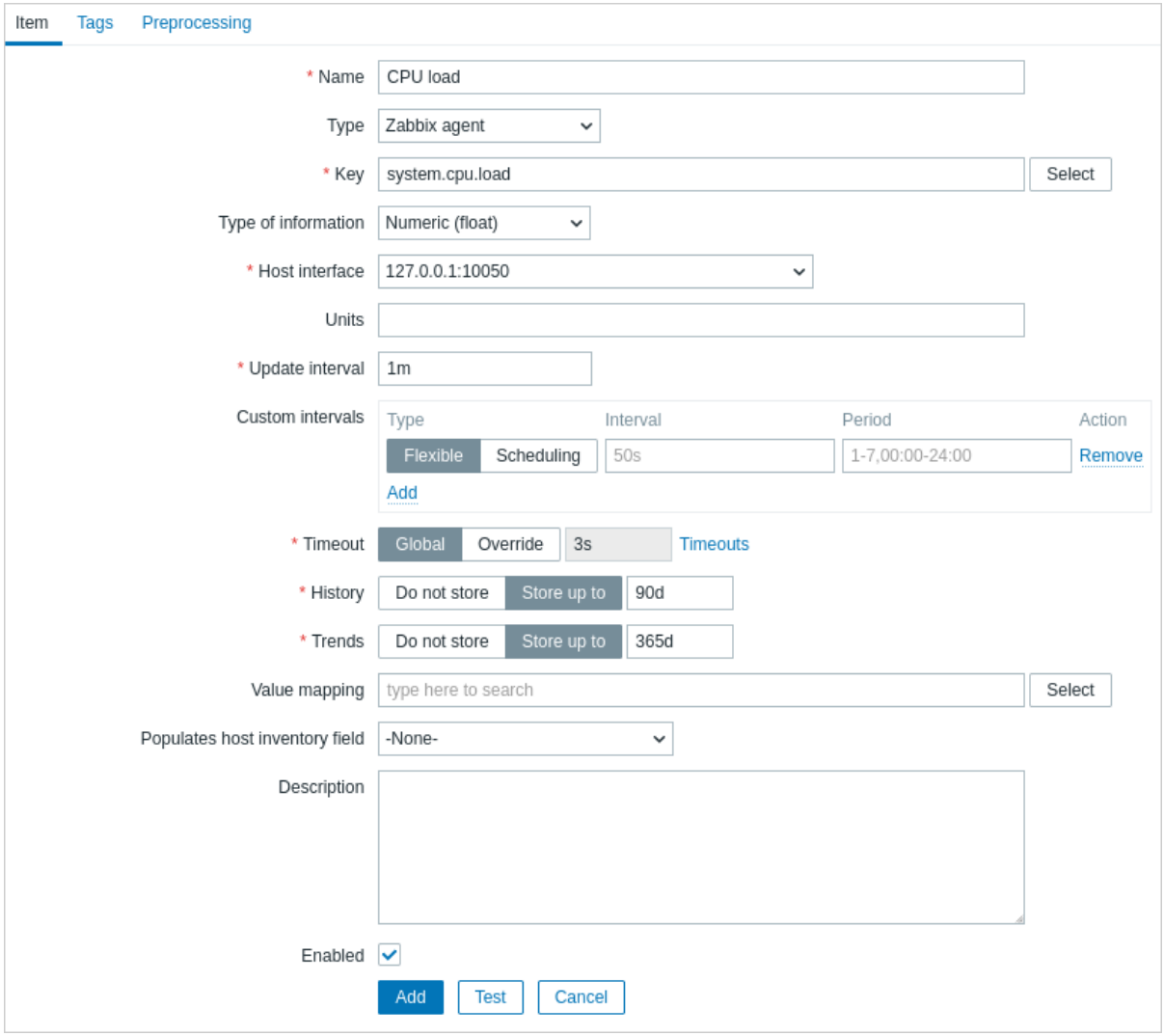
Šajā sadaļā jūs uzzināsiet, kā iestatīt vienumu.

Vienumi ir datu vākšanas pamats Zabbix sistēmā. Bez vienumiem nav datu — jo tikai vienums nosaka vienu metriku vai to, kāda veida datus vākt no hosta.

Vienuma pievienošana

Visi vienumi ir grupēti ap hostiem. Tāpēc, lai konfigurētu parauga vienumu, dodamies uz *Datu ievākšana* > *Hosti* un atrodam iepriekš izveidoto "New host".

Noklikšķiniet uz saites *Items* "New host" rindā un pēc tam noklikšķiniet uz *Create item*. Tiks atvērta vienuma konfigurācijas forma.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Mūsu parauga vienumam būtiskā ievadāmā informācija ir:

Nosaukums

- Ievadiet vērtību *CPU load*. Tas būs vienuma nosaukums, kas tiks attēlots sarakstos un citur.

Atslēga

- Manuāli ievadiet vērtību *system.cpu.load*. Tas ir vienuma tehniskais nosaukums, kas identificē ievācamo informācijas tipu. Konkrētā atslēga ir tikai viena no iepriekš definētajām atslēgām, kas ir pieejamas kopā ar Zabbix aģentu.

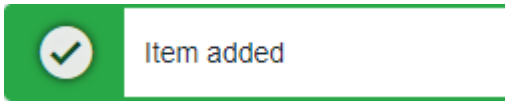
Informācijas tips

- Šis atribūts nosaka sagaidāmo datu formātu. Atslēgai *system.cpu.load* šis lauks tiks automātiski iestatīts uz *Numeric (float)*.

Note:
Iespējams, vēlēsieties arī samazināt dienu skaitu, cik ilgi tiks glabāta vienuma vēsture, līdz 7 vai 14. Tā ir laba prakse, lai atslogotu datubāzi no liela apjoma vēsturisko vērtību glabāšanas.

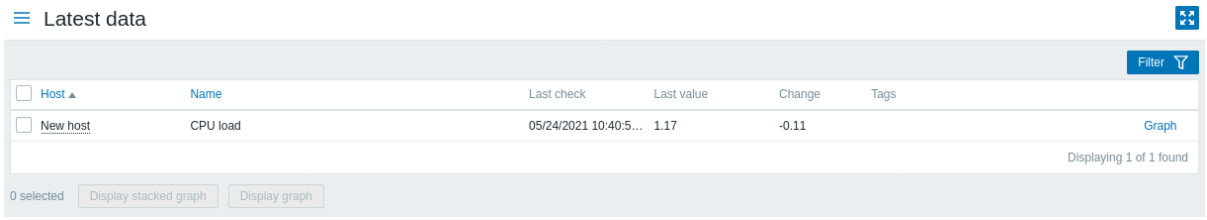
Citās opcijās var neko nemainīt, atstājot to noklusējuma vērtības.

Kad tas ir izdarīts, noklikšķiniet uz *Add*. Jaunajam vienumam vajadzētu parādīties vienumu sarakstā, un jums vajadzētu redzēt veiksmes paziņojumu.



Datu skatīšana

Kad vienums ir definēts, iespējams, vēlaties pārliecināties, vai tas patiešām vāc datus. Lai to izdarītu, dodieties uz *Monitoring* > *Latest data*, filtrā atlasiet 'New host' un noklikšķiniet uz *Apply*.



Ņemiet vērā, ka pirmo datu saņemšana var aizņemt līdz 60 sekundēm. Pēc noklusējuma tieši tik bieži serveris nolasa konfigurācijas izmaiņas un uztver izpildei jaunus vienumus.

Ja kolonnā 'Change' neredzat vērtību, iespējams, līdz šim ir saņemta tikai viena vērtība. Uzgaidiet 30 sekundes, līdz pienāks vēl viena vērtība.

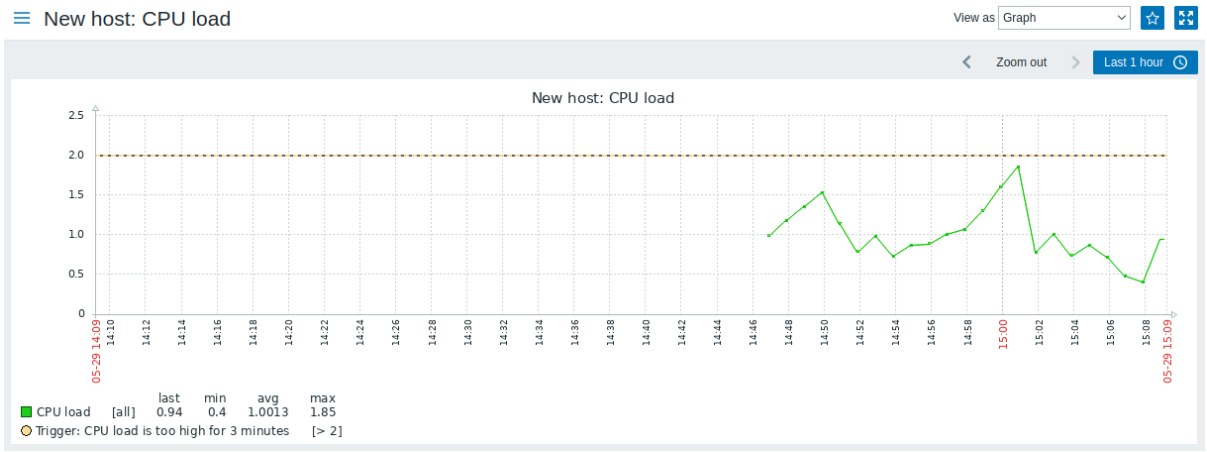
Ja neredzat informāciju par vienumu, kā parādīts ekrānuzņēmumā, pārliecinieties, ka:

- esat aizpildījis vienuma laukus 'Key' un 'Type of information' tieši tā, kā redzams ekrānuzņēmumā;
- gan aģents, gan serveris darbojas;
- hosta statuss ir 'Monitored' un tā pieejamības ikona ir zaļa;
- hostu filtrā atlasītais hosts ir pareizs;
- vienums ir iespējots.

Grafiki

Kad vienums kādu laiku ir darbojies, iespējams, ir pienācis laiks apskatīt kaut ko vizuālu. Vienkāršie grafiki ir pieejami jebkuram uzraudzītam skaitliskam vienumam bez jebkādas papildu konfigurācijas. Šie grafiki tiek ģenerēti izpildlaikā.

Lai skatītu grafiku, dodieties uz *Uzraudzība* > *Jaunākie dati* un noklikšķiniet uz saites 'Grafiks' blakus vienumam.



4 Jauns trigeris

Pārskats

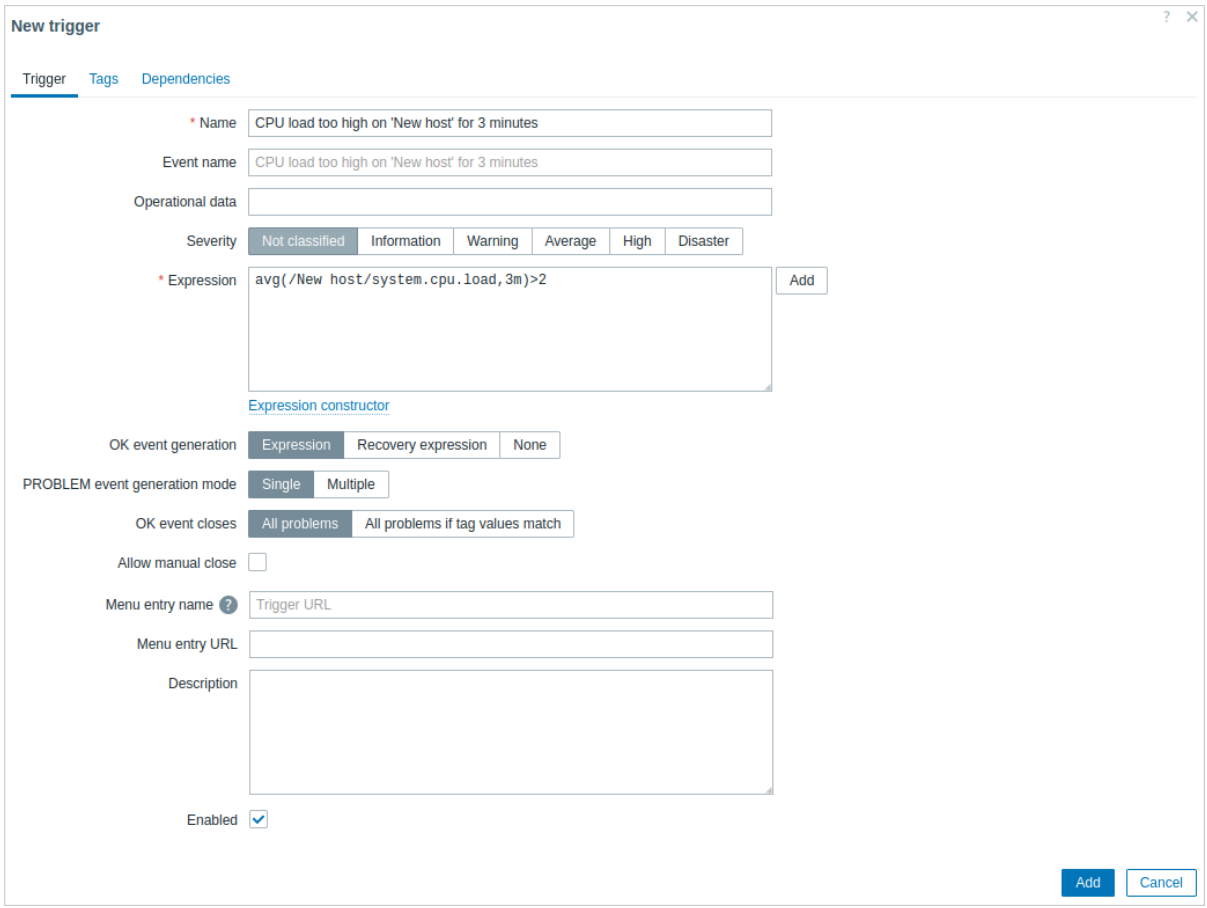
Šajā sadaļā jūs uzzināsiet, kā iestatīt trigeri.

Vienumi tikai apkopo datus. Lai automātiski novērtētu ienākošos datus, mums ir jādefinē trigeri. Trigeris satur izteiksmi, kas nosaka slieksni tam, kāds datu līmenis ir pieņemams.

Ja ienākošie dati pārsniedz šo līmeni, trigeris "nostrādā" vai pāriet 'Problēma' stāvoklī — norādot, ka ir noticis kaut kas, kam var būt nepieciešama uzmanība. Ja līmenis atkal ir pieņemams, trigeris atgriežas 'Labi' stāvoklī.

Trigera pievienošana

Lai konfigurētu trigeri mūsu vienumam, dodieties uz *Datu ievākšana* > *Hosti*. Noklikšķiniet uz saites *Trigeri* rindā ar 'New host' un pēc tam noklikšķiniet uz *Izveidot trigeri*. Tiks atvērta trigera konfigurācijas forma.



Mūsu trigerim šeit jāievada šāda būtiskā informācija:

Nosaukums

- Ievadiet *CPU load too high on 'New host' for 3 minutes* kā vērtību. Tas būs trigera nosaukums, kas tiks parādīts sarakstos un citur.

Izteiksme

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

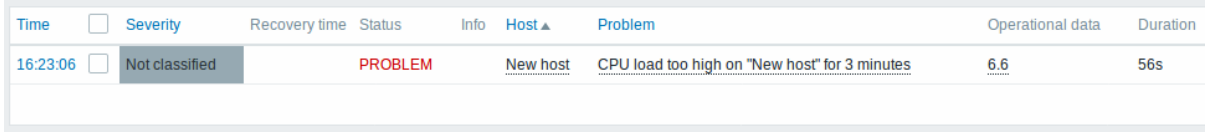
Šī ir trigera izteiksme. Pārliecinieties, ka izteiksme ir ievadīta pareizi līdz pat pēdējam simbolam. Vienuma atslēga šeit (system.cpu.load) tiek izmantota, lai atsauktos uz vienumu. Šī konkrētā izteiksme būtībā nozīmē, ka problēmas slieksnis ir pārsniegts, ja CPU slodzes vidējā vērtība 3 minūšu laikā ir lielāka par 2. Vairāk varat uzzināt par trigeru izteiksmju sintaksi.

Kad pabeigts, noklikšķiniet uz *Pievienot*. Jaunajam trigerim vajadzētu parādīties trigeru sarakstā.

Trigera stāvokļa attēlošana

Kad trigeris ir definēts, jūs, iespējams, vēlēsieties redzēt tā stāvokli.

Ja CPU slodze ir pārsniegusi sliekšņa līmeni, ko definējāt trigerī, problēma tiks attēlota sadaļā *Uzraudzība* > *Problēmas*.



Mirgošana statusa kolonnā norāda uz nesenām trigera stāvokļa izmaiņām, kas notikušas pēdējo 30 minūšu laikā.

5 Problēmas paziņojuma saņemšana

Pārskats

Šajā sadaļā jūs uzzināsiet, kā Zabbix iestatīt brīdināšanu paziņojumu veidā.

Tā kā vienumi vāc datus un trigeri ir paredzēti, lai "nostrādātu" problēmsituācijās, ir lietderīgi ieviest arī kādu brīdināšanas mehānismu, kas paziņo par svarīgiem notikumiem pat tad, ja Zabbix lietotāja saskarne netiek aktīvi pārbaudīta.

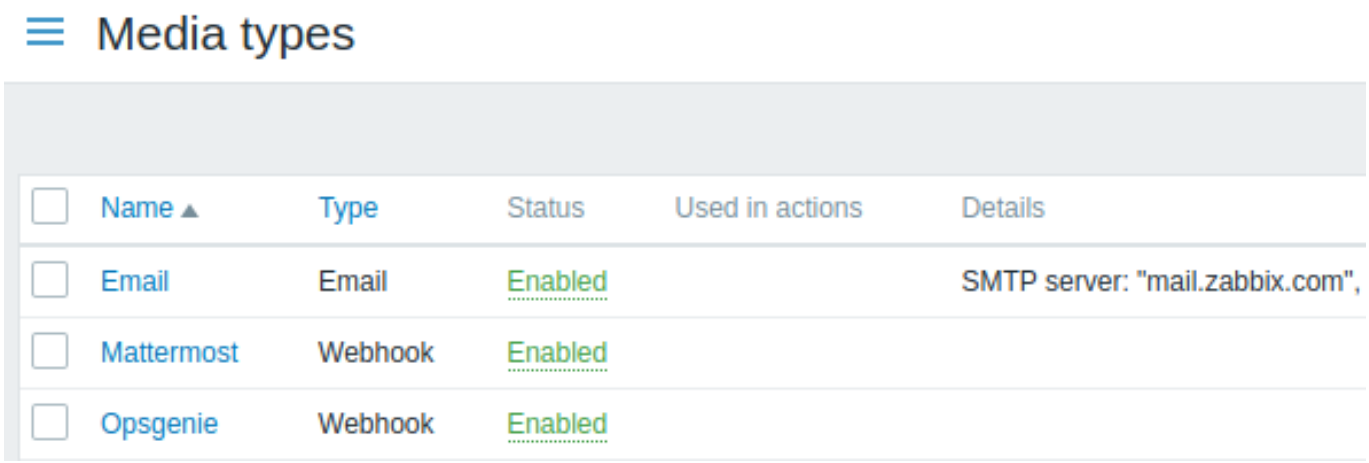
Tieši to nodrošina paziņojumi.

Tā kā e-pasts ir populārākais problēmu paziņojumu piegādes veids, mēs apskatīsim, kā iestatīt e-pasta paziņojumu.

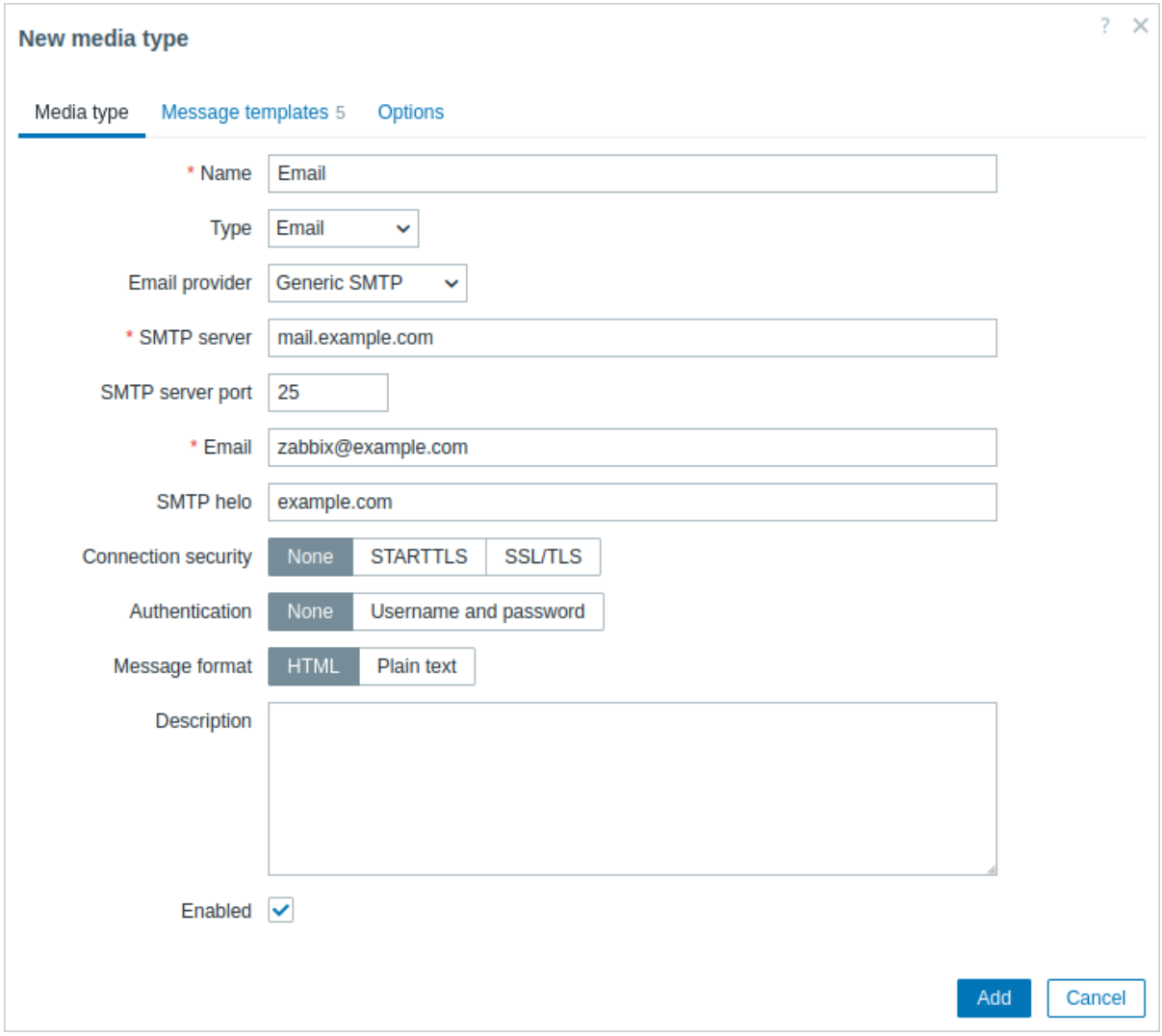
E-pasta iestatījumi

Sākotnēji Zabbix ir pieejamas vairākas iepriekš definētas paziņojumu piegādes metodes. E-pasts ir viena no tām.

Lai konfigurētu e-pasta iestatījumus, dodieties uz *Brīdinājumi* > *Mediju tipi* un iepriekš definēto mediju tipu sarakstā noklikšķiniet uz *E-pasts*.



Tiks atvērta e-pasta iestatījumu konfigurēšanas forma.



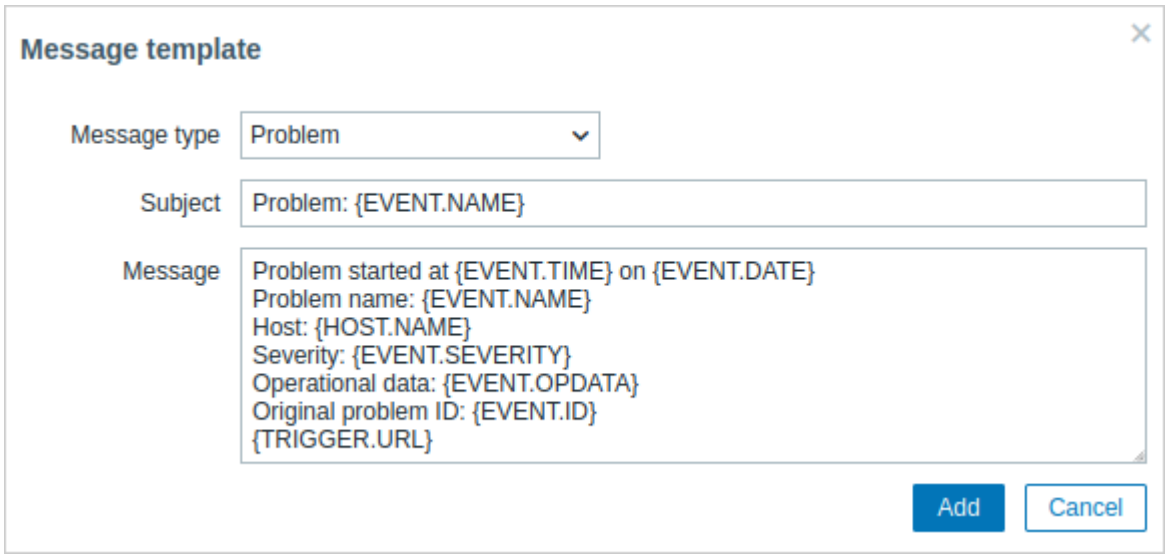
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Cilnē *Mediju tips* iestatiet *SMTP serveris*, *SMTP helo* un *E-pasts* vērtības atbilstoši savai videi.

Note:
Vērtība laukā *E-pasts* tiks izmantota kā 'No' adrese paziņojumiem, kas nosūtīti no Zabbix.

Tālāk ir jādefinē problēmas ziņojuma saturs. Saturs tiek definēts, izmantojot ziņojuma veidni, kas konfigurēta cilnē *Ziņojumu veidnes*.

Noklikšķiniet uz *Pievienot*, lai izveidotu ziņojuma veidni, un atlasiet *Problēma* kā ziņojuma tipu.

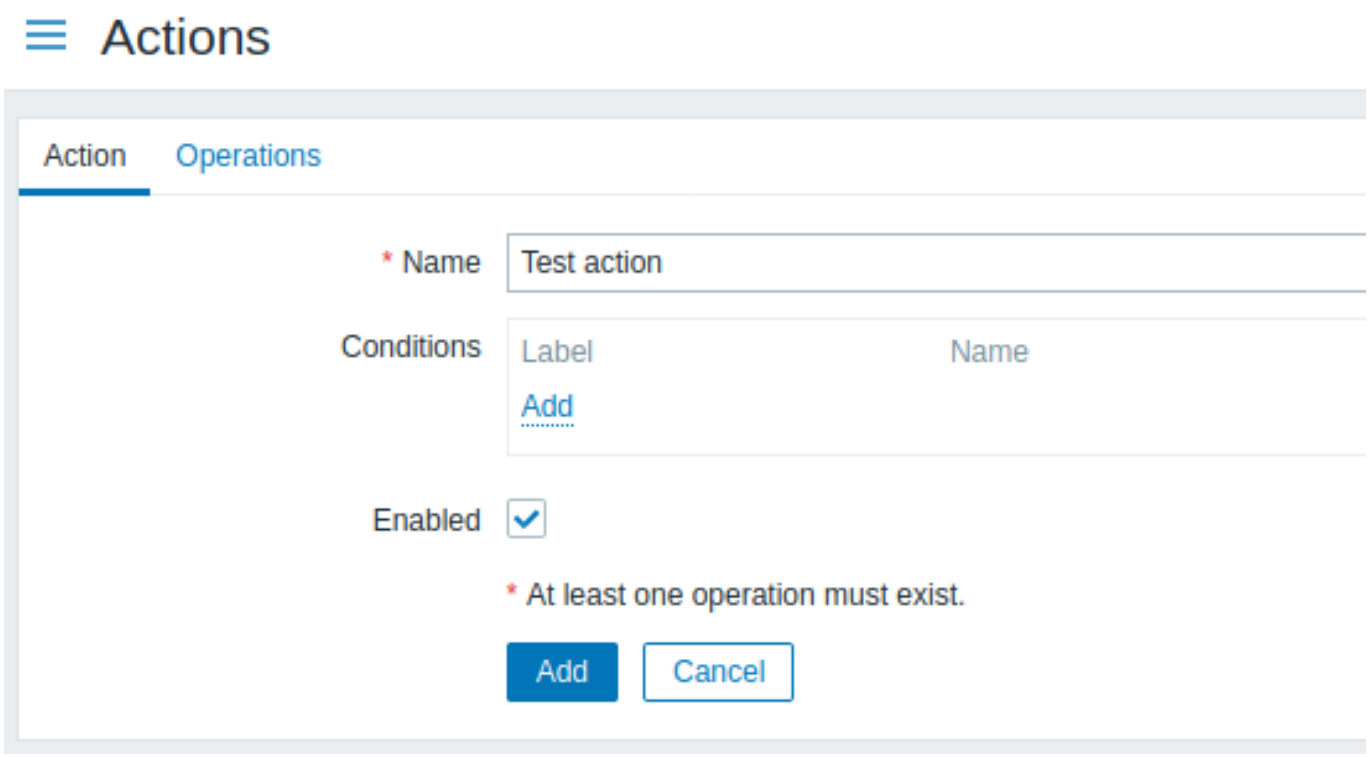


Kad viss ir gatavs, noklikšķiniet uz *Pievienot* un saglabājiet formu.

Tagad *E-pasts* ir konfigurēts kā strādājošs mediju tips. Mediju tips ir arī jāpiesaista lietotājiem, norādot konkrētas piegādes adreses (kā to darījām, konfigurējot jaunu lietotāju), pretējā gadījumā tas netiks izmantots.

Jauna darbība

Paziņojumu piegāde ir viena no lietām, ko Zabbix veic darbības. Tāpēc, lai iestatītu paziņojumu, dodieties uz *Brīdinājumi* > *Darbības* > *Trigeru darbības* un noklikšķiniet uz *Izveidot darbību*.

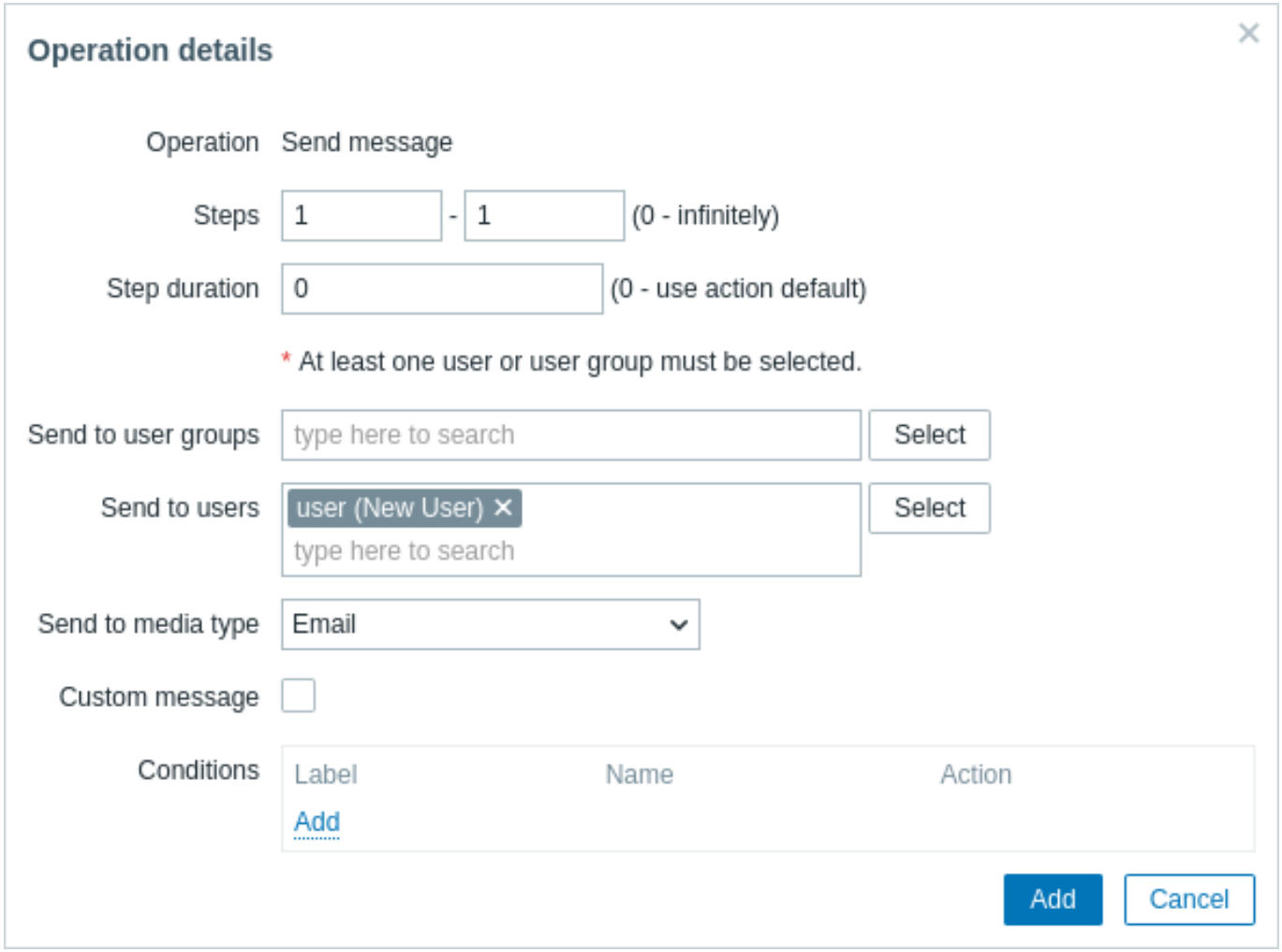


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šajā formā ievadiet darbības nosaukumu.

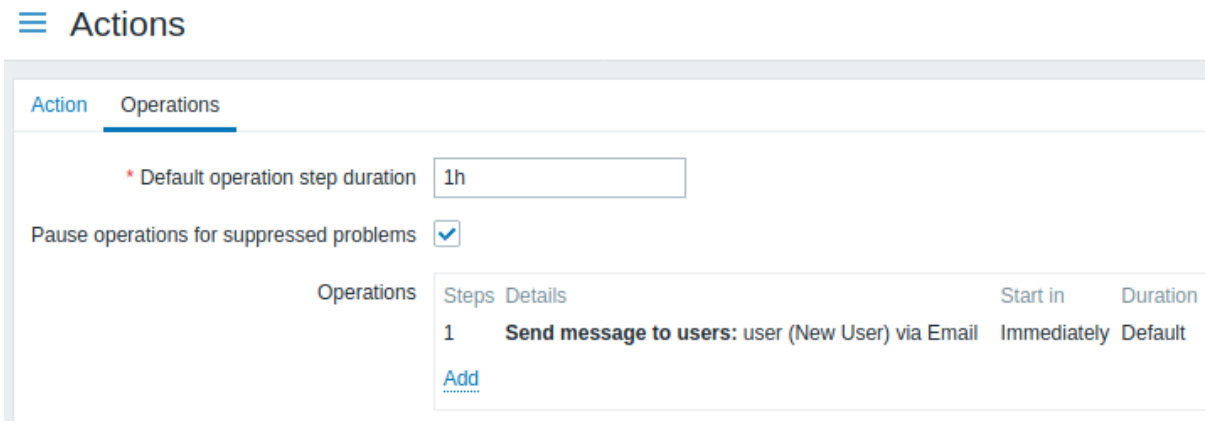
Vienkāršākajā gadījumā, ja nepievienojam nekādus specifiskākus nosacījumus, darbība tiks izpildīta pie jebkurām trigera izmaiņām no 'Ok' uz 'Problem'.

Mums vēl jānosaka, ko darbībai vajadzētu darīt — tas tiek iestatīts cilnē *Darbības*. Noklikšķiniet uz *Pievienot* blokā *Darbības*, kas atvērs jaunu darbības formu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit noklikšķiniet uz *Atlasīt* blokā *Sūtīt lietotājiem* un atlasiet lietotāju ('user'), ko esam definējuši. Atlasiet "Email" kā *Sūtīt uz medija tipu* vērtību. Kad tas ir izdarīts, noklikšķiniet uz *Pievienot*, un darbība būtu jāpievieno:



Tas ir viss vienkāršai darbības konfigurācijai, tāpēc darbības formā noklikšķiniet uz *Pievienot*.

Paziņojuma saņemšana

Tagad, kad paziņojumu piegāde ir konfigurēta, būtu labi arī kādu patiešām saņemt. Lai to panāktu, mēs varam apzināti palielināt slodzi uz mūsu hosta — lai mūsu trigeris "nostrādātu" un mēs saņemtu problēmas paziņojumu.

Atveriet konsoli savā hostā un izpildiet:

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

Varat palaist vienu vai vairākus no šiem procesiem.

Tagad dodieties uz *Monitoring* > *Latest data* un apskatiet, kā ir palielinājušās 'CPU Load' vērtības. Atcerieties — lai mūsu trigeris "nostrādātu", 'CPU Load' vērtībai 3 minūtes nepārtraukti jābūt virs '2'. Kad tas notiks:

- sadaļā *Monitoring* > *Problems* jums vajadzētu redzēt trigeri ar mirgojošu `Problem` stāvokli;
- jums vajadzētu saņemt problēmas paziņojumu savā e-pastā.

Attention:
Ja paziņojumi nedarbojas:
- vēlreiz pārbaudiet, vai gan e-pasta iestatījumi, gan darbība ir konfigurēti pareizi
- pārliecinieties, ka jūsu izveidotajam lietotājam ir vismaz lasīšanas tiesības uz hostu, kas ģenerēja notikumu, kā norādīts solī *Lietotāja pievienošana*. Lietotājam, kas ir daļa no lietotāju grupas 'Zabbix administrators', jābūt vismaz lasīšanas piekļuvei hostu grupai 'Linux servers', kurai pieder mūsu hosts.
- Papildus varat pārbaudīt darbību žurnālu, dodoties uz *Reports* > *Action log*.

6 Jauna veidne

Pārskats

Šajā sadaļā jūs uzzināsiet, kā iestatīt veidni.

Iepriekš mēs uzzinājām, kā iestatīt vienumu, trigeri un kā saņemt problēmas paziņojumu par hostu.

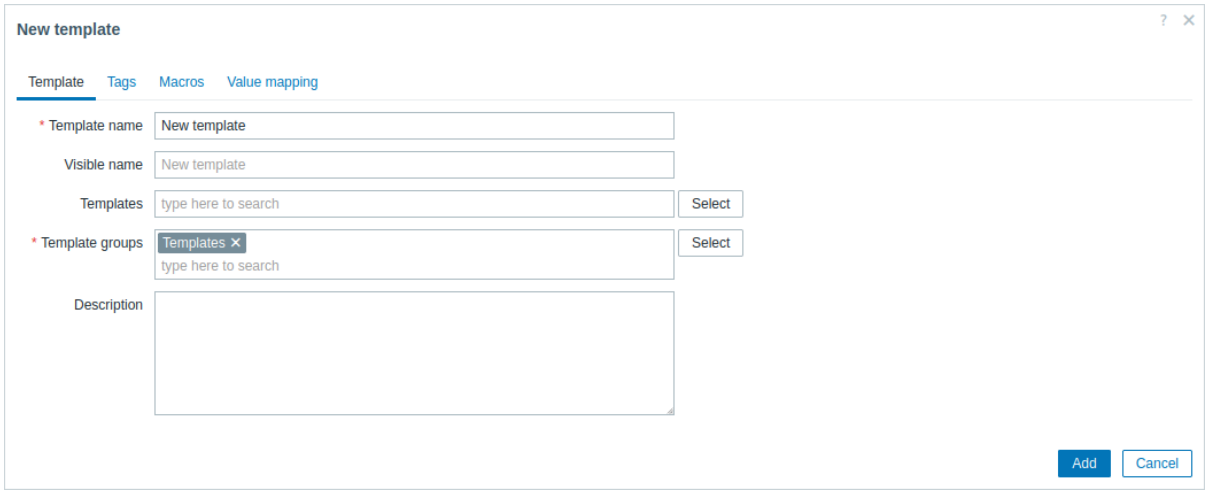
Lai gan visas šīs darbības pašas par sevi piedāvā lielu elastību, ja tas ir nepieciešams, piemēram, tūkstotim hostu, var šķist, ka veicamo darbību ir pārāk daudz. Noderētu zināma automatizācija.

Šeit palīgā nāk veidnes. Veidnes ļauj grupēt noderīgus vienumus, trigerus un citas entītijas, lai tās varētu izmantot atkal un atkal, piemērojot hostiem vienā darbībā.

Kad veidne ir piesaistīta hostam, hosts manto visas veidnes entītijas. Tātad būtībā iepriekš sagatavotu pārbaužu kopu var piemērot ļoti ātri.

Veidnes pievienošana

Lai sāktu strādāt ar veidnēm, vispirms ir jāizveido veidne. Lai to izdarītu, sadaļā *Datu ievākšana* > *Veidnes* noklikšķiniet uz *Izveidot veidni*. Tiks atvērta veidnes konfigurācijas forma.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit jāievada šādi obligātie parametri:

Veidnes nosaukums

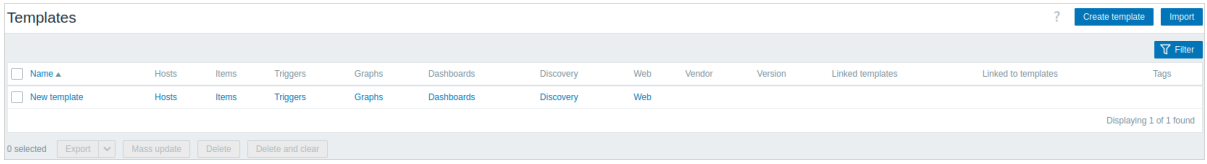
- Ievadiet veidnes nosaukumu. Ir atļauti burtciparu simboli, atstarpes un pasvītrojuma zīmes.

Veidņu grupas

- Atlasiet vienu vai vairākas grupas, noklikšķinot uz pogas *Atlasīt*. Veidnei ir jāpieder grupai.

Note:
Piekļuves tiesības veidņu grupām tiek piešķirtas lietotāju grupas konfigurācijā cilnē **Veidņu atļaujas** tāpat kā hostu atļaujas. Visas piekļuves tiesības tiek piešķirtas grupām, nevis atsevišķām veidnēm, tāpēc veidnes iekļaušana vismaz vienā grupā ir obligāta.

Kad tas ir izdarīts, noklikšķiniet uz *Pievienot*. Jūsu jaunajai veidnei vajadzētu būt redzamai veidņu sarakstā. Varat arī izmantot filtru, lai atrastu savu veidni.



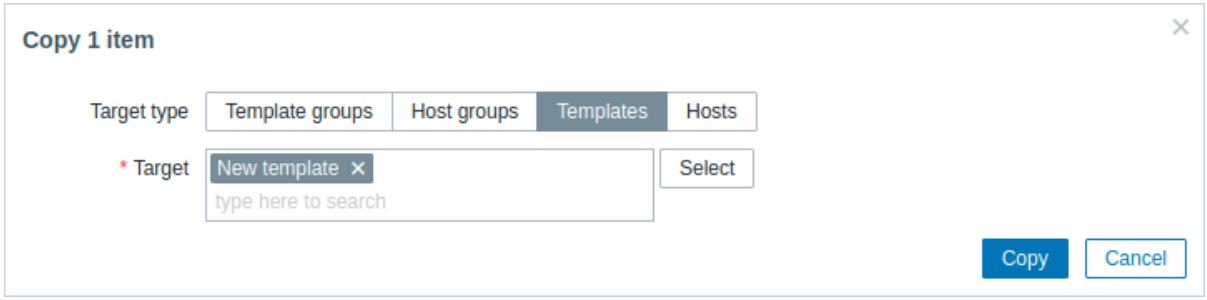
Kā redzat, veidne ir izveidota, taču tajā vēl nekā nav — nav ne vienumu, ne trigeru, ne citu entītiju.

Vienuma pievienošana veidnei

Lai pievienotu vienumu veidnei, atveriet vienumu sarakstu hostam 'New host', dodoties uz *Datu ievākšana* → *Hosti* un noklikšķinot uz *Vienumi* blakus 'New host'.

Pēc tam:

- Atzīmējiet izvēles rūtiņu vienumam 'CPU Load' sarakstā.
- Noklikšķiniet uz *Kopēt* zem saraksta.
- Atlasiet cilni *Veidnes*.
- Atlasiet veidni, uz kuru kopēt vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

- Noklikšķiniet uz *Kopēt*.

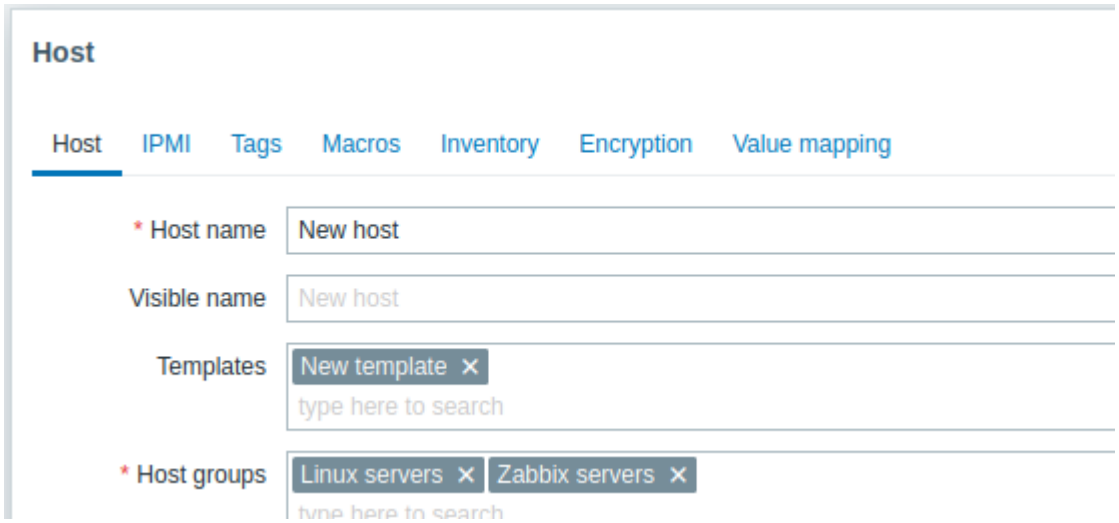
Ja tagad dosieties uz *Datu ievākšana* > *Veidnes*, veidnē 'New template' vajadzētu būt vienam jaunam vienumam.

Pagaidām apstāsimies tikai pie viena vienuma, taču līdzīgā veidā varat pievienot veidnei arī jebkurus citus vienumus, trigerus vai citas entītijas, līdz tā kļūst par pietiekami pilnīgu entītiju kopu konkrētam mērķim (OS uzraudzība, vienas lietotnes uzraudzība).

Veidnes piesaistīšana hostam

Kad veidne ir sagatavota, atliek tikai pievienot to hostam. Lai to izdarītu, dodieties uz *Datu ievākšana* > *Hosti*, noklikšķiniet uz 'Jauns hosts', lai atvērtu tā konfigurācijas veidlapu, un atrodiet lauku **Veidnes**.

Sāciet rakstīt *New template* laukā *Veidnes*. Mūsu izveidotās veidnes nosaukumam vajadzētu parādīties nolaižamajā sarakstā. Ritiniet uz leju, lai to atlasītu. Pārliecinieties, ka tā parādās laukā *Veidnes*.



Noklikšķiniet uz *Atjaunināt* veidlapā, lai saglabātu izmaiņas. Veidne tagad ir pievienota hostam kopā ar visām tajā ietvertajām entītijām.

Šādā veidā to var piemērot arī jebkuram citam hostam. Jebkuras izmaiņas vienumos, trigeros un citās entītijās veidnes līmenī tiks propagētas uz hostiem, kuriem veidne ir piesaistīta.

Iepriekš definētu veidņu piesaistīšana hostiem

Kā, iespējams, jau pamanījāt, Zabbix tiek piegādāts ar iepriekš definētu veidņu kopu dažādām operētājsistēmām, ierīcēm un lietotnēm. Lai ļoti ātri sāktu uzraudzību, varat piesaistīt hostam atbilstošo veidni, taču ņemiet vērā, ka šīs veidnes ir jāpielāgo jūsu videi. Dažas pārbaudes var nebūt nepieciešamas, un aptauju intervāli var būt pārāk bieži.

Ir pieejama plašāka informācija par veidnēm.

Apache tīmekļa servera uzraudzība

Lai sāktu Apache tīmekļa servera uzraudzību, izpildiet šo procesu:

1. Pārliecinieties, ka ir piekļuve Apache statistikai.
2. Pielietojiet Apache uzraudzības veidni.
3. Apskatiet jaunākos datus.

Lai izmantotu šo rokasgrāmatu, jums jābūt installed Zabbix serverim un tīmekļa saskarnei. Jums jābūt pieteikušamies Zabbix kā noklusējuma *Admin* lietotājam.

Piekļuve Apache statistikai

Šī soļa mērķis ir pārliecināties, ka Zabbix var nolasīt Apache tīmekļa servera statistiku.

Apache vidē reāllaika informāciju par servera veiktspēju un aktivitāti nodrošina iebūvētais `mod_status` modulis. Kad tas ir iespējots, tas izveido tīmeklī pieejamu lapu (parasti `/server-status`), kurā tiek rādīti veiktspējas un aktivitātes dati.

Lai pārbaudītu, vai Apache tīmekļa serverim ir iespējots `mod_status`, izpildiet:

Debian/Ubuntu	Uz RHEL balstītās sistēmās
`apache2ctl -M | grep status`	`httpd -M | grep status`

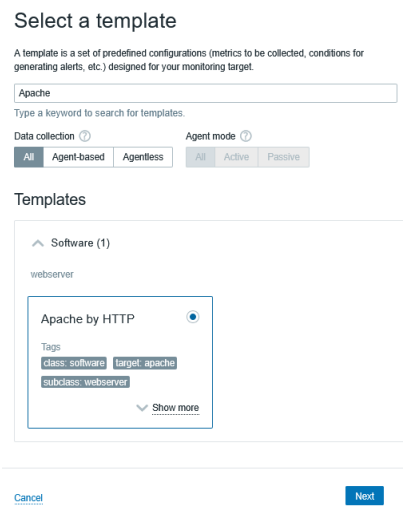
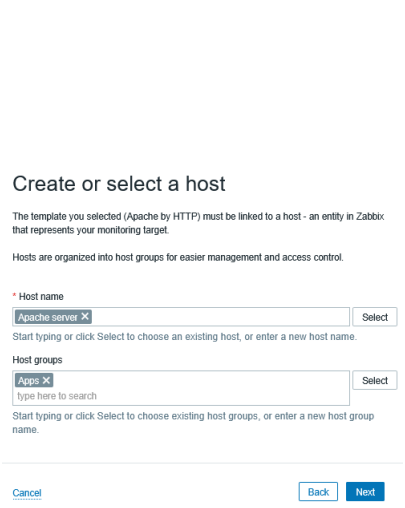
Ja izvadē redzat "`status_module (shared)`", tad statusa modulis jau ir iespējots. Ja nē, tad jums ir jāiespējo statusa modulis.

Pielietojiet Apache uzraudzības veidni

Zabbix šī veidne ir gatava lietošanai. Tā satur iepriekš konfigurētus elementus, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

Lai pielietotu veidni, palaidiet Host Wizard (*Data collection* > *Hosts*):

- **Atlasiet veidni** ("Apache by HTTP")
- Izveidojiet Apache hostu un pievienojiet tam grupu. Šis virtuālais hosts saturēs ar Apache saistītos datus.
- Konfigurējiet hostu. Svarīgi: ievadiet Apache tīmekļa servera adresi (127.0.0.1 lokālajā datorā, IP adresi attālinātā datorā).

1. solis	2. solis	3. solis
		
Veidnes atlasīšana.	Hosta/hostu grupas izveide.	Hosta konfigurēšana.

Pārējos iestatījumus varat atstāt pēc noklusējuma.

Lai pabeigtu Host Wizard, noklikšķiniet uz *Create* un *Finish*.

Apskatiet jaunākos datus

Latest data

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

Apsveicam, Apache tīmekļa servera uzraudzība ir sākta!

Zabbix nodrošina vizualizācijas iespējas un nosūta brīdinājumus, ja rodas kādas problēmas. Lai konfigurētu brīdinājumus e-pastā, skatiet Problēmu paziņojumu saņemšana.

Linux servera uzraudzība

Lai sāktu Linux servera uzraudzību, izpildiet šo procesu:

1. Pielietojiet Linux uzraudzības veidni.
2. Instalējiet Zabbix aģentu serverī.
3. Apskatiet jaunākos datus.

Lai sāktu nelokāla Linux servera uzraudzību, jums jāzina tā IP adrese un Zabbix servera IP adrese.

Linux serverī tā uzraudzībai tiks izmantots Zabbix aģents. Norādījumi par aģenta instalēšanu tiks sniegti procesa gaitā.

Lai izmantotu šo ceļvedi, jums jābūt instalētam Zabbix serverim un tīmekļa saskarnei. Jums jābūt pieteikušamies Zabbix kā noklusējuma *Admin* lietotājam.

Pielietojiet Linux uzraudzības veidni

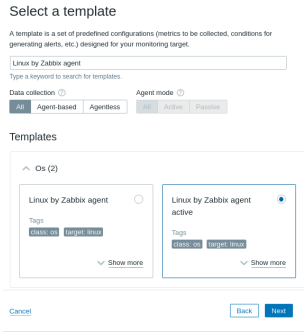
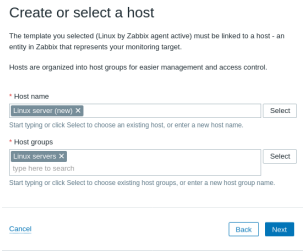
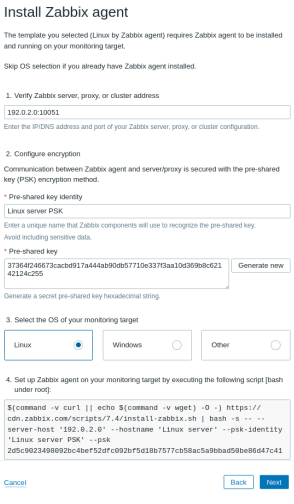
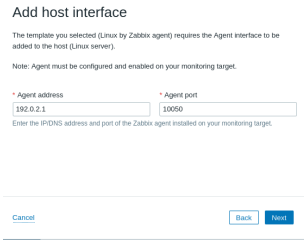
Zabbix šī veidne ir gatava lietošanai. Tā satur iepriekš konfigurētus elementus, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

Lai pielietotu veidni, palaidiet Host Wizard (sadaļā *Data collection* > *Hosts*):

- **Atlasiet veidni** (”Linux by Zabbix agent active”)
- Izveidojiet Linux servera hostu un pievienojiet tam grupu. Šis virtuālais hosts saturēs visus ar Linux servera uzraudzību saistītos datus.
- Konfigurējiet un instalējiet aģentu. Lai instalētu aģentu, Linux serverī izpildiet norādīto komandu. Tā instalēs aģentu un konfigurēs aģenta savienojumu ar Zabbix serveri, izmantojot šifrēšanu.

Ja aģents jau ir instalēts, šo soli varat izlaist (noklikšķiniet *Next*). Tomēr jums tiks lūgts norādīt iepriekš koplietotās atslēgas identitāti un iepriekš koplietoto atslēgu, lai izveidotu savienojumu ar aģentu.

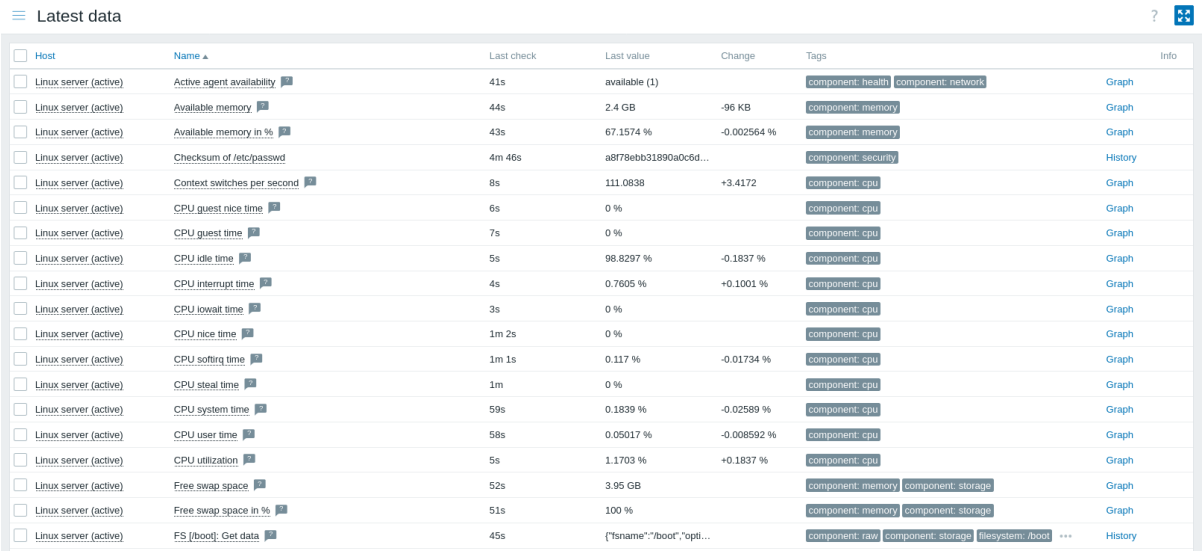
- Pabeidziet hosta konfigurēšanu. Ievadiet aģenta (Linux servera) IP adresi un portu.

1. solis	2. solis	3. solis	4. solis
			
Veidnes atlasīšana.	Hosta/hostu grupas izveide.	Aģenta konfigurēšana un instalēšana hostā.	Linux servera IP adreses/porta pievienošana kā aģenta saskarne.

Pārējos iestatījumus varat atstāt pēc noklusējuma.

Lai pabeigtu Host Wizard, noklikšķiniet uz *Create* un *Finish*.

Apskatiet jaunākos datus



Apsveicam, Linux servera uzraudzība ir sākta!

Zabbix nodrošina vizualizācijas iespējas un nosūta brīdinājumus, ja rodas kādas problēmas. Lai konfigurētu brīdinājumus e-pastā, skatiet Problēmu paziņojumu saņemšana.

MySQL servera uzraudzība

Lai sāktu uzraudzīt MySQL serveri, izpildiet šo procesu:

1. Izveidojiet MySQL lietotāju ar ierobežotām tiesībām.
2. Pielietojiet MySQL uzraudzības veidni.
3. Apskatiet jaunākos datus.

Lai sāktu uzraudzīt MySQL serveri, jums jāzina tā DSN un Zabbix servera IP adrese.

MySQL servera uzraudzībai tiks izmantots Zabbix aģents. Norādījumi par aģenta instalēšanu tiks sniegti procesa gaitā.

Lai izmantotu šo ceļvedi, jums jābūt instalētam Zabbix serverim un tīmekļa saskarnei. Jums jābūt pieteikušamies Zabbix kā noklusējuma *Admin* lietotājam.

Izveidojiet ierobežotu MySQL lietotāju

Ieteicams izveidot MySQL lietotāju, kura atļaujas ir ierobežotas tikai līdz tām, kas nepieciešamas MySQL uzraudzībai.

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

Pielietojiet MySQL uzraudzības veidni

Zabbix šī veidne ir gatava lietošanai. Tā satur iepriekš konfigurētus elementus, kas ir gatavi datu vākšanai, vizualizācijai un analīzei.

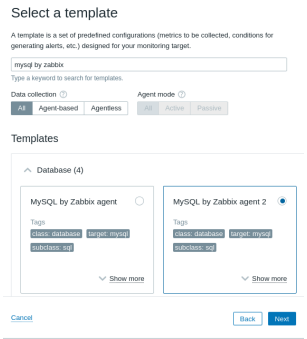
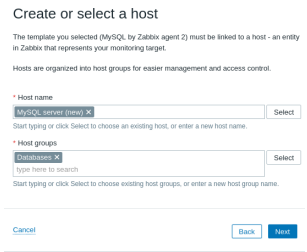
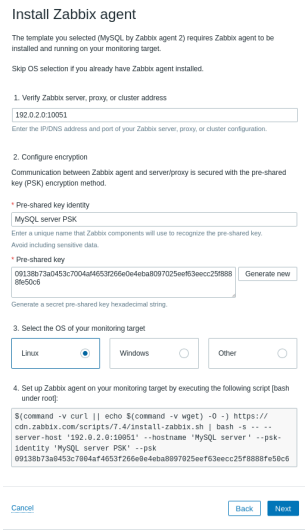
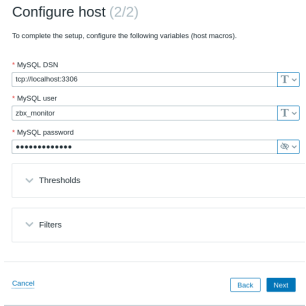
Lai pielietotu veidni, palaidiet Host Wizard (*Data collection* > *Hosts*):

- **Atlasiet veidni** (”MySQL by Zabbix agent 2”)
- Izveidojiet MySQL servera hostu un pievienojiet tam grupu. Šis virtuālais hosts saturēs visus datus, kas saistīti ar MySQL servera uzraudzību.
- Konfigurējiet un instalējiet aģents 2 (Zabbix aģents 2 nodrošina MySQL uzraudzību uzreiz pēc instalēšanas).

Lai instalētu aģentu, izpildiet norādīto komandu datorā, kurā darbojas MySQL serveris. Tā instalēs aģentu un konfigurēs aģenta savienojumu ar Zabbix serveris, izmantojot šifrēšanu.

Ja jums jau ir instalēts aģents 2, šo soli varat izlaist (noklikšķiniet uz *Next*). Tomēr jums tiks lūgts norādīt iepriekš koplietotās atslēgas identitāti un iepriekš koplietoto atslēgu, lai izveidotu savienojumu ar aģentu.

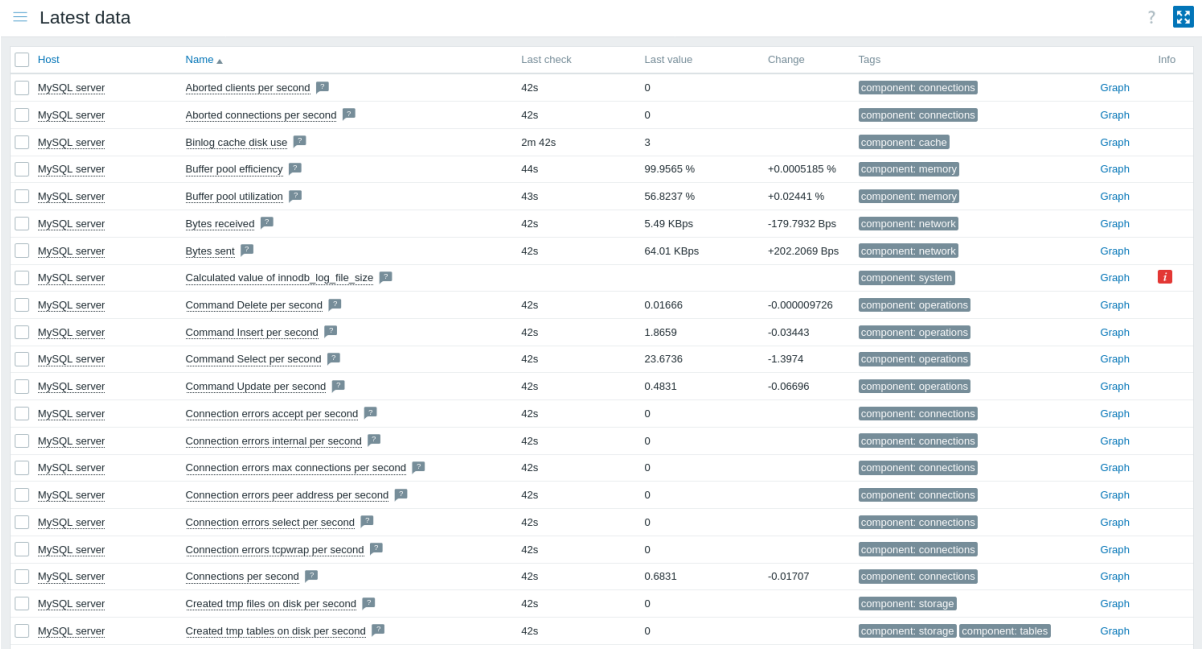
- Pabeidziet hosta konfigurēšanu. Svarīgi, ka jums tiks lūgts ievadīt MySQL servera DSN, MySQL lietotāju un paroli. Norādiet lietotāju `zbx_monitor` ar jūsu izveidoto paroli.

1. solis	2. solis	3. solis	4. solis
			
Veidnes atlasīšana.	Hosta/hostu grupas izveide.	Aģenta konfigurēšana un instalēšana hostā.	MySQL servera DNS, lietotāja un paroles pievienošana.

Pārējiem iestatījumiem varat atstāt noklusējuma vērtības.

Lai pabeigtu Host Wizard, noklikšķiniet uz *Create* un *Finish*.

Apskatiet jaunākos datus



Apsveicam, MySQL servera uzraudzība ir sākta!

Zabbix nodrošina vizualizācijas iespējas un nosūta brīdinājumus, ja rodas kādas problēmas. Lai konfigurētu brīdinājumus e-pastā, skatiet Problēmu paziņojumu saņemšana.

3 Instalēšana: papildu ceļveži

Ja meklējat vienkāršāko Zabbix instalēšanas veidu, lūdzu, skatiet Instalēšana un pirmie soļi.

Tālāk norādītie ceļveži aptver **papildu** vai sistēmai specifiskus Zabbix vai tā komponentu instalēšanas veidus:

- Instalēšana no konteineriem
- Instalēšana no pirmkoda
- Piezīmes par instalēšanu no pakotnēm
- Zabbix aģenta instalēšana no Windows MSI
- Zabbix aģenta 2 būvēšana operētājsistēmā Windows
- Zabbix aģenta būvēšana operētājsistēmā Windows
- macOS aģenta instalēšana no PKG
- Zabbix aģenta būvēšana operētājsistēmā macOS
- Zabbix appliance

Instalēšana no konteineriem

Pārskats

Šajā sadaļā ir aprakstīts, kā izvietot Zabbix ar Docker vai Docker Compose.

Zabbix oficiāli nodrošina:

- Atsevišķus Docker attēlus katram Zabbix komponentam, lai tos darbinātu kā portatīvus un pašpietiekamus konteinerus.
- Compose failus vairāku konteineru Zabbix komponentu definēšanai un darbināšanai Docker vidē.

Attention:
Kopš Zabbix 6.0 deterministiskie trigeri ir jāizveido instalēšanas laikā. Ja MySQL/MariaDB ir iespējota binārā žurnalēšana, tam ir nepieciešamas superlietotāja privilēģijas vai mainīgā/konfigurācijas parametra *log_bin_trust_function_creators = 1* iestatīšana. Norādījumus par to, kā iestatīt šo mainīgo, skatiet sadaļā Database creation scripts.
 Ņemiet vērā, ka, izpildot komandas no konsoles, mainīgais tiks iestatīts tikai uz laiku un tiks atiestatīts, kad Docker tiks restartēts. Šādā gadījumā atstājiet savu SQL servisu darbinātu, apturiet tikai zabbix-server servisu, izpildot 'docker compose down zabbix-server', un pēc tam 'docker compose up -d zabbix-server'.
 Alternatīvi, šo mainīgo var iestatīt konfigurācijas failā.

Avota faili

Docker failu avoti tiek glabāti Zabbix oficiālajā repozitorijā GitHub vietnē, kur varat sekot jaunākajām failu izmaiņām vai atzarot projektu, lai izveidotu savus attēlus.

Docker

Zabbix nodrošina attēlus, kuru pamatā ir dažādi OS bāzes attēli. Lai iegūtu konkrēta Zabbix komponenta atbalstīto bāzes operētājsistēmas attēlu sarakstu, skatiet komponenta aprakstu Docker Hub. Visi Zabbix attēli ir konfigurēti tā, lai pārbūvētu jaunākos attēlus, ja tiek atjaunināti bāzes attēli.

Instalēšana

Lai iegūtu Zabbix komponenta attēlu, izpildiet:

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

Aizstājiet `zabbix/zabbix-server-mysql` ar nepieciešamās Docker repozitorija nosaukumu.

Šī komanda lejupielādēs jaunāko stabilo Zabbix komponenta versiju, kuras pamatā ir Alpine Linux OS. Varat pievienot tagus repozitorija nosaukumam, lai iegūtu attēlu, kura pamatā ir cita operētājsistēma, vai konkrētas Zabbix galvenās vai mazākās versijas attēlu.

Docker Hub ir pieejami šādi repozitoriji:

Komponents		Docker repozitorijs
Zabbix aģents		zabbix/zabbix-agent
Zabbix serveris		
	ar MySQL atbalstu	zabbix/zabbix-server-mysql
	ar PostgreSQL atbalstu	zabbix/zabbix-server-pgsql
Zabbix tīmekļa saskarne		
	uz Apache2 tīmekļa servera bāzes ar MySQL atbalstu	zabbix/zabbix-web-apache-mysql
	uz Apache2 tīmekļa servera bāzes ar PostgreSQL atbalstu	zabbix/zabbix-web-apache-pgsql
	uz Nginx tīmekļa servera bāzes ar MySQL atbalstu	zabbix/zabbix-web-nginx-mysql
	uz Nginx tīmekļa servera bāzes ar PostgreSQL atbalstu	zabbix/zabbix-web-nginx-pgsql
Zabbix starpniekserveris		
	ar SQLite3 atbalstu	zabbix/zabbix-proxy-sqlite3
	ar MySQL atbalstu	zabbix/zabbix-proxy-mysql
Zabbix Java gateway		zabbix/zabbix-java-gateway

Note:
SNMP slazdu atbalsts tiek nodrošināts atsevišķā repozitorijā zabbix/zabbix-snmptraps. To var sasaistīt ar Zabbix serveri un Zabbix starpniekserveri.

Tagi

Oficiālie Zabbix komponentu attēli var saturēt šādus tagus:

Tags	Apraksts	Piemērs
latest	Jaunākā stabilā Zabbix komponenta versija, kas balstīta uz Alpine Linux attēlu.	zabbix-agent:latest

Tags	Apraksts	Piemērs
<OS>-trunk	Jaunākais nakts būvējums tai Zabbix versijai, kas pašlaik tiek izstrādāta konkrētai operētājsistēmai. **<OS>** - bāzes operētājsistēma. Atbalstītās vērtības: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (tikai aģents); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (tikai aģents); *ubuntu* - Ubuntu	zabbix-agent:ubuntu-trunk
<OS>-latest	Jaunākā stabilā Zabbix komponenta versija konkrētā operētājsistēmā. **<OS>** - bāzes operētājsistēma. Atbalstītās vērtības: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (tikai aģents); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (tikai aģents); *ubuntu* - Ubuntu	zabbix-agent:ol-latest
<OS>-X.X-latest	Jaunākā Zabbix komponenta mazākā versija konkrētai galvenajai versijai un operētājsistēmai. **<OS>** - bāzes operētājsistēma. Atbalstītās vērtības: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (tikai aģents); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (tikai aģents); *ubuntu* - Ubuntu **X.X** - galvenā Zabbix versija (piemēram, *6.0*, *7.4*, *8.0*).	zabbix-agent:alpine-8.0-latest
<OS>-X.X.*	Konkrēta Zabbix komponenta mazākā versija konkrētai galvenajai versijai un operētājsistēmai. **<OS>** - bāzes operētājsistēma. Atbalstītās vērtības: *alpine* - Alpine Linux; *ltsc2019* - Windows 10 LTSC 2019 (tikai aģents); *ol* - Oracle Linux; *ltsc2022* - Windows 11 LTSC 2022 (tikai aģents); *ubuntu* - Ubuntu **X.X** - galvenā Zabbix versija (piemēram, *6.0*, *7.4*, *8.0*). * - Zabbix mazākā versija	zabbix-agent:alpine-8.0.1

Sākotnējā konfigurācija

Pēc attēlu lejupielādes palaidiet konteinerus, izpildot komandu `docker run`, kam seko papildu argumenti, lai norādītu nepieciešamos vides mainīgos un/vai montēšanas punktus. Tālāk ir sniegti daži konfigurācijas piemēri.

Note:
Lai iespējotu saziņu starp Zabbix komponentiem, daži porti, piemēram, 10051/TCP Zabbix serverim (trapper), 10050/TCP Zabbix aģentam, 162/UDP SNMP slazdiem un 80/TCP Zabbix tīmekļa saskarnei, tiks atvērti hostsistēmā. Pilns Zabbix komponentu izmantoto noklusējuma portu saraksts ir pieejams lapā Requirements. Zabbix serverim un aģentam noklusējuma portu var mainīt, iestatot ZBX_LISTENPORT vides mainīgo.

Vides mainīgie

Visi Zabbix komponentu attēli nodrošina vides mainīgos konfigurācijas pārvaldībai. Atbalstītie vides mainīgie ir uzskaitīti komponenta repozitorijā.

Šie vides mainīgie ir opcijas no Zabbix konfigurācijas failiem, bet ar atšķirīgu nosaukumu veidošanas metodi. Piemēram, `ZBX_LOGSLOWQUERIES` atbilst `LogSlowQueries` no Zabbix serveris vai Zabbix starpniekserveris konfigurācijas failiem.

Attention:
Dažas konfigurācijas opcijas (piemēram, `PIDFile` un `LogType`) nevar mainīt.

Tālāk norādītie vides mainīgie ir specifiski Docker komponentiem un nepastāv Zabbix konfigurācijas failos:

Mainīgais	Komponenti	Noklusējuma vērtība	Apraksts
`DB_SERVER_HOST`	Serveris Starpniekserveris Tīmekļa saskarne	`mysql-server` priekš MYSQL `postgres-server` priekš PostgreSQL	MySQL vai PostgreSQL servera IP adrese vai DNS nosaukums.
`DB_SERVER_PORT`	Serveris Starpniekserveris Tīmekļa saskarne	3306 priekš MYSQL 5432 priekš PostgreSQL	MySQL vai PostgreSQL servera ports.
`MYSQL_USER`	Serveris Starpniekserveris Tīmekļa saskarne	`zabbix`	MySQL datubāzes lietotājs.
`MYSQL_PASSWORD`	Serveris Starpniekserveris Tīmekļa saskarne	`zabbix`	MySQL datubāzes parole.
`MYSQL_DATABASE`	Serveris Starpniekserveris Tīmekļa saskarne	`zabbix` Zabbix serverim `zabbix_proxy` Zabbix starpniekserverim	Zabbix datubāzes nosaukums.
`POSTGRES_USER`	Serveris Tīmekļa saskarne	`zabbix`	PostgreSQL datubāzes lietotājs.
`POSTGRES_PASSWORD`	Serveris Tīmekļa saskarne	`zabbix`	PostgreSQL datubāzes parole.
`POSTGRES_DB`	Serveris Tīmekļa saskarne	`zabbix` Zabbix serverim `zabbix_proxy` Zabbix starpniekserverim	Zabbix datubāzes nosaukums.
`PHP_TZ`	Tīmekļa saskarne	`Europe/Riga`	Laika josla PHP formātā. Pilns atbalstīto laika joslu saraksts ir pieejams vietnē php.net.
`ZBX_SERVER_NAME`	Tīmekļa saskarne	`Zabbix Docker`	Redzamais Zabbix instalācijas nosaukums zem Zabbix logotipa tīmekļa saskarnes vertikālajā izvēlnē.
`ZBX_JAVAGATEWAY_ENABLE`	Serveris Starpniekserveris	`false`	Iespējo saziņu ar Zabbix Java gateway, lai vāktu ar Java saistītās pārbaudes.
`ZBX_ENABLE_SNMP_TRAPS`	Serveris Starpniekserveris	`false`	Iespējo SNMP trapu funkcionalitāti. Tai ir nepieciešama **zabbix-snmptraps** instance un koplietots sējums */var/lib/zabbix/snmptraps* Zabbix serverim vai Zabbix starpniekserverim.

Sējumi

Attēli ļauj piemontēt sējumus, izmantojot šādus montēšanas punktus:

Sējums	Apraksts
Zabbix aģents	
/etc/zabbix/zabbix_agentd.d	Sējums ļauj iekļaut **.conf* failus un paplašināt Zabbix aģentu, izmantojot `UserParameter` funkcionalitāti
/var/lib/zabbix/modules	Sējums ļauj ielādēt papildu moduļus un paplašināt Zabbix aģentu, izmantojot LoadModule funkcionalitāti
/var/lib/zabbix/enc	Sējums tiek izmantots ar TLS saistītu failu glabāšanai. Šo failu nosaukumi tiek norādīti, izmantojot vides mainīgos `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` un `ZBX_TLSPSKFILE`
Zabbix serveris	

Sējums	Apraksts
	/usr/lib/zabbix/alertscripts Sējums tiek izmantots pielāgotiem brīdinājumu skriptiem. Tas ir parametrs `AlertScriptsPath` failā zabbix_server.conf
	/usr/lib/zabbix/externalscripts Sējumu izmanto ārējās pārbaudes. Tas ir parametrs `ExternalScripts` failā zabbix_server.conf
	/var/lib/zabbix/modules Sējums ļauj ielādēt papildu moduļus un paplašināt Zabbix serveri, izmantojot LoadModule funkcionalitāti
	/var/lib/zabbix/enc Sējums tiek izmantots ar TLS saistītu failu glabāšanai. Šo failu nosaukumi tiek norādīti, izmantojot vides mainīgos `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` un `ZBX_TLSPSKFILE`
	/var/lib/zabbix/ssl/certs Sējums tiek izmantots kā SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai. Tas ir parametrs `SSLCertLocation` failā zabbix_server.conf
	/var/lib/zabbix/ssl/keys Sējums tiek izmantots kā SSL privātās atslēgas failu atrašanās vieta klienta autentifikācijai. Tas ir parametrs `SSLKeyLocation` failā zabbix_server.conf
	/var/lib/zabbix/ssl/ssl_ca Sējums tiek izmantots kā sertifikācijas iestādes (CA) failu atrašanās vieta SSL servera sertifikāta pārbaudei. Tas ir parametrs `SSLCALocation` failā zabbix_server.conf
	/var/lib/zabbix/snmptraps Sējums tiek izmantots kā faila snmptraps.log atrašanās vieta. To var koplietot ar zabbix-snmptraps konteineru un pārmantot, izmantojot Docker opciju volumes_from, veidojot jaunu Zabbix servera instanci. SNMP trapu apstrādes funkcionalitāti var iespējot, izmantojot koplietotu sējumu un pārslēdzot vides mainīgo `ZBX_ENABLE_SNMP_TRAPS` uz 'true'
	/var/lib/zabbix/mibs Sējums ļauj pievienot jaunus MIB failus. Tas neatbalsta apakšdirektorijus, visi MIB ir jānovieto mapē `/var/lib/zabbix/mibs`
Zabbix starpniekserveris	
	/usr/lib/zabbix/externalscripts Sējumu izmanto ārējās pārbaudes. Tas ir parametrs `ExternalScripts` failā zabbix_proxy.conf
	/var/lib/zabbix/db_data Sējums ļauj glabāt datubāzes failus ārējās ierīcēs. Atbalstīts tikai Zabbix starpniekserverim ar SQLite3
	/var/lib/zabbix/modules Sējums ļauj ielādēt papildu moduļus un paplašināt Zabbix serveri, izmantojot LoadModule funkcionalitāti
	/var/lib/zabbix/enc Sējums tiek izmantots ar TLS saistītu failu glabāšanai. Šo failu nosaukumi tiek norādīti, izmantojot vides mainīgos `ZBX_TLSCAFILE`, `ZBX_TLSCRLFILE`, `ZBX_TLSKEY_FILE` un `ZBX_TLSPSKFILE`
	/var/lib/zabbix/ssl/certs Sējums tiek izmantots kā SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai. Tas ir parametrs `SSLCertLocation` failā zabbix_proxy.conf
	/var/lib/zabbix/ssl/keys Sējums tiek izmantots kā SSL privātās atslēgas failu atrašanās vieta klienta autentifikācijai. Tas ir parametrs `SSLKeyLocation` failā zabbix_proxy.conf
	/var/lib/zabbix/ssl/ssl_ca Sējums tiek izmantots kā sertifikācijas iestādes (CA) failu atrašanās vieta SSL servera sertifikāta pārbaudei. Tas ir parametrs `SSLCALocation` failā zabbix_proxy.conf
	/var/lib/zabbix/snmptraps Sējums tiek izmantots kā faila snmptraps.log atrašanās vieta. To var koplietot ar zabbix-snmptraps konteineru un pārmantot, izmantojot Docker opciju volumes_from, veidojot jaunu Zabbix servera instanci. SNMP trapu apstrādes funkcionalitāti var iespējot, izmantojot koplietotu sējumu un pārslēdzot vides mainīgo `ZBX_ENABLE_SNMP_TRAPS` uz 'true'
	/var/lib/zabbix/mibs Sējums ļauj pievienot jaunus MIB failus. Tas neatbalsta apakšdirektorijus, visi MIB ir jānovieto mapē `/var/lib/zabbix/mibs`
Zabbix tīmekļa saskarne uz Apache2 tīmekļa servera bāzes	
	/etc/ssl/apache2 Sējums ļauj iespējot HTTPS Zabbix tīmekļa saskarnei. Sējumā jābūt diviem failiem `ssl.crt` un `ssl.key`, kas sagatavoti Apache2 SSL savienojumiem

Sējums	Apraksts
Zabbix tīmekļa saskarne uz Nginx tīmekļa servera bāzes	
/etc/ssl/nginx	Sējums ļauj iespējot HTTPS Zabbix tīmekļa saskarnei. Sējumā jābūt diviem failiem `ssl.crt`, `ssl.key` un `dhparam.pem`, kas sagatavoti Nginx SSL savienojumiem
Zabbix snmptraps	
/var/lib/zabbix/snmptraps	Sējumā atrodas žurnālfails `snmptraps.log`, kurā tiek ierakstīti saņemtie SNMP trapi
/var/lib/zabbix/mibs	Sējums ļauj pievienot jaunus MIB failus. Tas neatbalsta apakšdirektorijus, visi MIB ir jānovieto mapē `/var/lib/zabbix/mibs`

Papildu informāciju skatiet oficiālajās Zabbix repozitorijās Docker Hub.

Piemēri

1. piemērs

Šis piemērs parāda, kā palaist Zabbix serveris ar MySQL datubāzes atbalstu, Zabbix tīmekļa lietotāja saskarne, kas balstīta uz Nginx tīmekļa serveri, un Zabbix Java gateway.

1. Izveidojiet tīklu, kas paredzēts Zabbix komponentu konteineriem:

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

2. Palaidiet tukšu MySQL servera instanci:

```
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. Palaidiet Zabbix Java gateway instanci:

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

4. Palaidiet Zabbix servera instanci un sasaistiet to ar izveidoto MySQL servera instanci:

```
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. Palaidiet Zabbix tīmekļa lietotāja saskarne un sasaistiet to ar izveidotajām MySQL servera un Zabbix servera instancēm:

```
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
```

2. piemērs

Šis piemērs parāda, kā palaist Zabbix serveris ar PostgreSQL datubāzes atbalstu, Zabbix tīmekļa lietotāja saskarne, kas balstīta uz Nginx tīmekļa serveri, un SNMP trap funkcionalitāti.

1. Izveidojiet tīklu, kas paredzēts Zabbix komponentu konteineriem:

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

2. Palaidiet tukšu PostgreSQL servera instanci:

```
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. Palaidiet Zabbix snmptraps instanci:

```
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. Palaidiet Zabbix servera instanci un sasaistiet to ar izveidoto PostgreSQL servera instanci:

```
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. Palaidiet Zabbix tīmekļa lietotāja saskarne un sasaistiet to ar izveidotajām PostgreSQL servera un Zabbix servera instancēm:

```
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
```

3. piemērs

Šis piemērs parāda, kā palaist Zabbix serveris ar MySQL datubāzes atbalstu, Zabbix tīmekļa lietotāja saskarne, kas balstīta uz Nginx tīmekļa serveri, un Zabbix Java gateway, izmantojot podman Red Hat 8 vidē.

1. Izveidojiet jaunu pod ar nosaukumu zabbix un atvērtajiem portiem (tīmekļa lietotāja saskarne, Zabbix servera trapper):

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

2. (neobligāti) Palaidiet Zabbix aģents konteineru zabbix pod atrašanās vietā:

```
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. Izveidojiet ./mysql/ direktoriju hostā un palaidiet Oracle MySQL serveri 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. Palaidiet Zabbix servera konteineru:

```
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. Palaidiet Zabbix Java Gateway konteineru:

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

6. Palaidiet Zabbix tīmekļa lietotāja saskarne konteineru:

```
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:
Pod `zabbix` publicē 80/TCP portu (HTTP) uz hosta datora no `zabbix-web-mysql` konteinera 8080/TCP porta.

Docker Compose

Alternatīvi, Zabbix var instalēt, izmantojot Docker Compose spraudni. Compose faili vairāku konteineru Zabbix komponentu definēšanai un palaišanai ir pieejami oficiālajā Zabbix Docker repozitorijā GitHub.

Attention:
Oficiālie Zabbix compose faili atbalsta Docker Compose 3. versiju.

Šie compose faili ir pievienoti kā piemēri; tie ir pārslogoti. Piemēram, tie satur starpniekserverus ar gan MySQL, gan SQLite3 atbalstu.

Lai iegūtu Zabbix nodrošinātos Docker compose failus, klonējiet repozitoriju:

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

Pārslēdzieties uz nepieciešamo versiju:

```
git checkout 8.0
```

Izmantojiet compose konfigurācijas failus, lai izveidotu un palaistu konteinerus:

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

Aizstājiet `docker-compose_v3_alpine_mysql_latest.yaml` iepriekš norādītajā komandā ar nepieciešamo konfigurācijas failu.

Ir pieejamas šādas opcijas:

Faila nosaukums	Apraksts
`docker-compose_v3_alpine_mysql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Alpine Linux ar MySQL datubāzes atbalstu.
`docker-compose_v3_alpine_mysql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Alpine Linux ar MySQL datubāzes atbalstu.
`docker-compose_v3_alpine_pgsql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Alpine Linux ar PostgreSQL datubāzes atbalstu.
`docker-compose_v3_alpine_pgsql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Alpine Linux ar PostgreSQL datubāzes atbalstu.
`docker-compose_v3_ol_mysql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Oracle Linux ar MySQL datubāzes atbalstu.
`docker-compose_v3_ol_mysql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Oracle Linux ar MySQL datubāzes atbalstu.
`docker-compose_v3_ol_pgsql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Oracle Linux ar PostgreSQL datubāzes atbalstu.
`docker-compose_v3_ol_pgsql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Oracle Linux ar PostgreSQL datubāzes atbalstu.
`docker-compose_v3_ubuntu_mysql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Ubuntu 24.04 (noble) ar MySQL datubāzes atbalstu.
`docker-compose_v3_ubuntu_mysql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Ubuntu 24.04 (noble) ar MySQL datubāzes atbalstu.
`docker-compose_v3_ubuntu_pgsql_latest.yaml`	Compose fails, lai palaistu jaunāko Zabbix 8.0 komponentu versiju uz Ubuntu 24.04 (noble) ar PostgreSQL datubāzes atbalstu.
`docker-compose_v3_ubuntu_pgsql_local.yaml`	Compose fails, lai lokāli būvētu un palaistu jaunāko Zabbix 8.0 versiju un palaiž Zabbix komponentus uz Ubuntu 24.04 (noble) ar PostgreSQL datubāzes atbalstu.

Krātuve

Compose faili ir konfigurēti tā, lai atbalstītu lokālo krātuvi hosts datorā. Docker Compose izveidos `zbx_env` direktoriju mapē ar compose failu, kad palaidīsiet Zabbix komponentus, izmantojot compose failu. Direktorijā būs tā pati struktūra, kas aprakstīta sadaļā Volumes, kā arī direktorijs datubāzes glabāšanai.

Ir arī sējumi tikai lasīšanas režīmā failiem `/etc/localtime` un `/etc/timezone`.

Vides mainīgie

Mainīgo failiem ir šāda nosaukumu struktūra: `.env_<komponenta tips>`, un tie atrodas *env_vars* direktorijā. Skatiet vides mainīgos, lai iegūtu sīkāku informāciju par mainīgo nosaukumiem un pieejamo izvēli.

Piemēri

1. piemērs

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

Komanda lejupielādēs jaunākos Zabbix 8.0 attēlus katram Zabbix komponentam un palaidīs tos atvienotajā režīmā.

Attention:
Neaizmirstiet lejupielādēt `.env_<type of component>` failus no oficiālā Zabbix repozitorija vietnē github.com kopā ar compose failiem.

2. piemērs

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

Komanda lejupielādēs bāzes attēlu Ubuntu 24.04 (noble), pēc tam lokāli izveidos Zabbix 8.0 komponentus un palaidīs tos atvienotajā režīmā.

Instalēšana no pirmkoda

Jūs varat iegūt pašu jaunāko Zabbix versiju, kompilējot to no pirmkoda. Skatiet arī Zabbix pirmkoda iegūšana.

Šeit ir sniegta soli pa solim pamācība Zabbix instalēšanai no pirmkoda.

Zabbix dēmonu instalēšana

1 Lejupielādējiet pirmkoda arhīvu

Dodieties uz Zabbix lejupielādes lapu un lejupielādējiet pirmkoda arhīvu. Kad lejupielāde ir pabeigta, izpakojiet pirmkodu, izpildot:

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

Note:
Komandā norādiet pareizo Zabbix versiju. Tai ir jāsakrīt ar lejupielādētā arhīva nosaukumu.

2 Izveidojiet lietotāja kontu

Visi Zabbix dēmona procesi darbojas ar nepriviliģētu sistēmas lietotāju tiesībām. Ja Zabbix dēmons tiek palaists no nepriviliģēta lietotāja konta, tas turpinās darboties ar šo lietotāju.

Noklusējuma konfigurācijā, ja dēmons tiek palaists kā `root`, tas pārslēgsies uz `zabbix` lietotāja kontu, kuram jābūt izveidotam. Lai izveidotu `zabbix` sistēmas lietotāju un grupu, izpildiet tālāk norādītās komandas.

Uz RedHat bāzētas sistēmas:

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

Uz Debian bāzētas sistēmas:

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

Nav nepieciešams izveidot atsevišķu lietotāja kontu Zabbix lietotāja saskarnei.

Drošības ieteikums

Ja Zabbix serveris un aģents darbojas vienā un tajā pašā datorā, ieteicams tos darbināt ar **atsevišķiem lietotāju kontiem**. Ja abi darbojas ar vienu un to pašu lietotāju, aģents var piekļūt servera konfigurācijas failam, kas var atklāt sensitīvu informāciju, piemēram, datubāzes paroli, jebkuram Zabbix lietotājam ar Admin līmeņa tiesībām.

Attention:
Zabbix darbināšana kā `root`, `bin` vai jebkurš cits konts ar īpašām tiesībām rada drošības risku.

Mājas direktorijs (neobligāti)

Zabbix procesiem mājas direktorijs nav nepieciešams, tāpēc tā izveide parasti nav ieteicama. Tomēr, ja jums ir nepieciešama funkcionalitāte, kurai vajadzīgs mājas direktorijs (piemēram, MySQL akreditācijas datu glabāšanai $HOME/.my.cnf), varat to izveidot, izmantojot tālāk norādītās komandas.

Uz RedHat bāzētām sistēmām izpildiet:

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

Uz Debian bāzētām sistēmām izpildiet:

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

3 Izveidojiet Zabbix datubāzi

Zabbix serveris un starpniekserveris dēmoniem, kā arī Zabbix lietotāja saskarnei ir nepieciešama datubāze. Tā nav vajadzīga, lai darbinātu Zabbix aģents.

Ir pieejami SQL skripti datubāzes shēmas izveidei un datu kopas ievietošanai. Zabbix starpniekservera datubāzei ir nepieciešama tikai shēma, savukārt Zabbix servera datubāzei papildus shēmai ir nepieciešama arī datu kopa.

Pēc Zabbix datubāzes izveides pārejiet pie nākamajām Zabbix kompilēšanas darbībām.

4 Konfigurējiet pirmkodu

Lai veidotu Zabbix serveri, Zabbix starpniekserveri vai Zabbix aģentu, ir nepieciešams C99 ar GNU paplašinājumiem. Šo versiju var skaidri norādīt, iestatot CFLAGS="-std=gnu99":

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

> **Note:**
> Ja instalējat no Zabbix Git repozitorija, vispirms ir jāpalaiž:
> `./bootstrap.sh`

Konfigurējot pirmkodu Zabbix serverim vai starpniekserverim, jums ir jānorāda izmantojamais datubāzes tips. Vienlaikus ar servera vai starpniekservera procesu var kompilēt tikai vienu datubāzes tipu.

Lai skatītu visas atbalstītās konfigurācijas opcijas, izpakotā Zabbix pirmkoda direktorijā palaidiet:

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

Lai konfigurētu pirmkodu Zabbix serverim un aģentam, varat palaist kaut ko līdzīgu šim:

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

Lai konfigurētu pirmkodu Zabbix serverim (ar PostgreSQL u.c.), varat palaist:

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

Lai konfigurētu pirmkodu Zabbix starpniekserverim (ar SQLite u.c.), varat palaist:

```
./configure --prefix=/usr --enable-proxy --with-net-snmp --with-sqlite3 --with-ssh2
```

Lai konfigurētu pirmkodu Zabbix aģentam, varat palaist:

```
./configure --enable-agent
```

vai Zabbix aģentam 2:

```
./configure --enable-agent2
```

> **Note:**
> Lai veidotu Zabbix aģentu 2, jābūt instalētai atbalstītai Go versijai.

Piezīmes par kompilācijas opcijām:

- `--enable-agent` - kompilē Zabbix aģentu, kā arī komandrindas utilītas Zabbix get un Zabbix sender.
- `--with-libcurl` - nepieciešams virtuālo mašīnu uzraudzībai, SMTP autentifikācijai un `web.page.*` Zabbix aģenta vienumiem. Skatiet arī: Prasības (libcurl).
- `--with-libxml2` - nepieciešams virtuālo mašīnu uzraudzībai.
- `--with-libpcre2[=DIR]` - Zabbix vienmēr tiek kompilēts ar PCRE2 bibliotēku; šī opcija ļauj tikai norādīt pielāgotu PCRE2 instalācijas ceļu.

- `--with-mysql=/path/to/mysql_config` - norāda ceļu uz konkrētas MySQL klienta bibliotēkas konfigurāciju. Noderīgi, ja ir instalētas vairākas MySQL vai MariaDB versijas.
- `--enable-static` - statiski sasaista bibliotēkas (nav atbalstīts Solaris). Izmantojiet šo opciju, ja plānojat izplatīt kompilētos bināros failus sistēmām bez nepieciešamajām bibliotēkām. Nav ieteicams, veidojot Zabbix serveri. Lai veidotu serveri statiski, ir nepieciešama katras ārējās bibliotēkas statiskā versija. Konfigurācijas skripts to automātiski nepārbauda.
- `--with-stacksize=<value>` - iestata katra pavediena steka izmēru kilobaitos (piemēram, `--with-stacksize=512`). Varat palielināt šo vērtību, ja Zabbix avarē vai sastingst steka pārpildes dēļ (piemēram, priekšapstrādes laikā sistēmās ar zemu noklusējuma pavediena steka ierobežojumu).

Attention:
Ja `./configure` neizdodas trūkstošu bibliotēku vai citu problēmu dēļ, lūdzu, pārbaudiet failu `config.log`, lai iegūtu detalizētu informāciju par kļūdu.
Piemēram, ja trūkst `libssl`, tūlītējais kļūdas ziņojums var būt maldinošs:
```
checking for main in -lmysqlclient... no
configure: error: Not found mysqlclient library
```
Šajā gadījumā `config.log` atklāj patieso cēloni:
```
/usr/bin/ld: cannot find -lssl
/usr/bin/ld: cannot find -lcrypto
```

Skatiet arī:

- Zabbix kompilēšana ar šifrēšanas atbalstu
- Zināmās kompilācijas problēmas

5 Izveidojiet un instalējiet visu

Note:
Ja instalējat no Zabbix Git repozitorija, vispirms ir jāpalaiž:
```
$ make dbschema
```

`make install`

Šī darbība jāveic lietotājam ar pietiekamām atļaujām (parasti 'root' vai izmantojot sudo).

Palaižot `make install`, pēc noklusējuma dēmona binārie faili (zabbix_server, zabbix_agentd, zabbix_proxy) tiks instalēti mapē /usr/local/sbin, bet klienta binārie faili (zabbix_get, zabbix_sender) — mapē /usr/local/bin.

Note:
Lai norādītu citu atrašanās vietu nekā /usr/local, iepriekšējā avotu konfigurēšanas darbībā izmantojiet atslēgu --prefix, piemēram, --prefix=/home/zabbix. Šajā gadījumā dēmona binārie faili tiks instalēti mapē <prefix>/sbin, bet utilītas — mapē <prefix>/bin. Man lapas tiks instalētas mapē <prefix>/share.

6 Pārskatiet un rediģējiet konfigurācijas failus

- rediģējiet Zabbix aģenta konfigurācijas failu **/usr/local/etc/zabbix_agentd.conf**

Šis fails ir jākonfigurē katram hostam, kurā ir instalēts zabbix_agentd.

Failā jānorāda Zabbix servera **IP adrese**. Savienojumi no citiem hostiem tiks liegti.

- rediģējiet Zabbix servera konfigurācijas failu **/usr/local/etc/zabbix_server.conf**

Jānorāda datubāzes nosaukums, lietotājs un parole (ja tāda tiek izmantota).

Pārējie parametri ar noklusējuma vērtībām būs piemēroti nelielai instalācijai (līdz desmit uzraudzītiem hostiem). Tomēr, ja vēlaties maksimāli palielināt Zabbix servera (vai starpniekservera) veiktspēju, noklusējuma parametrus ieteicams mainīt.

- ja esat instalējis Zabbix starpniekserveri, rediģējiet starpniekservera konfigurācijas failu **/usr/local/etc/zabbix_proxy.conf**

Jānorāda servera IP adrese un starpniekservera hosta nosaukums (tam jābūt zināmam serverim), kā arī datubāzes nosaukums, lietotājs un parole (ja tāda tiek izmantota).

Note:
Izmantojot SQLite, jānorāda pilns ceļš uz datubāzes failu; DB lietotājs un parole nav nepieciešami.

7 Palaidiet dēmonus

Palaidiet zabbix_server servera pusē.

```
zabbix_server
```

Note:
Pārliecinieties, ka jūsu sistēma ļauj piešķirt 36 MB (vai nedaudz vairāk) koplietojamās atmiņas, pretējā gadījumā serveris var nestartēties un jūs servera žurnālfailā redzēsiet "Cannot allocate shared memory for <type of cache>.". Tas var notikt FreeBSD, Solaris 8 sistēmās.

Palaidiet zabbix_agentd visās uzraugāmajās iekārtās.

```
zabbix_agentd
```

Note:
Pārliecinieties, ka jūsu sistēma ļauj piešķirt 2 MB koplietojamās atmiņas, pretējā gadījumā aģents var nestartēties un jūs aģenta žurnālfailā redzēsiet "Cannot allocate shared memory for collector.". Tas var notikt Solaris 8 sistēmās.

Ja esat instalējis Zabbix starpniekserveri, palaidiet zabbix_proxy.

```
zabbix_proxy
```

Zabbix tīmekļa lietotāja saskarnes instalēšana

PHP failu kopēšana

Zabbix lietotāja saskarne ir rakstīta PHP valodā, tāpēc tās darbināšanai ir nepieciešams PHP atbalstošs tīmekļa serveris. Instalēšana tiek veikta, vienkārši nokopējot PHP failus no ui direktorijas uz tīmekļa servera HTML dokumentu direktoriju.

Apache tīmekļa serveru HTML dokumentu direktoriju biežāk sastopamās atrašanās vietas ir:

- /usr/local/apache2/htdocs (noklusējuma direktorija, instalējot Apache no pirmkoda)
- /srv/www/htdocs (OpenSUSE, SLES)
- /var/www/html (Debian, Ubuntu, Fedora, RHEL)

Ieteicams izmantot apakšdirektoriju, nevis HTML saknes direktoriju. Lai izveidotu apakšdirektoriju un tajā nokopētu Zabbix lietotāja saskarnes failus, izpildiet šādas komandas, aizstājot <htdocs> ar faktisko direktoriju:

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

Ja plānojat izmantot kādu citu valodu, nevis angļu, skatiet Papildu lietotāja saskarnes valodu instalēšana, lai iegūtu norādījumus.

Lietotāja saskarnes instalēšana

Lūdzu, skatiet lapu Tīmekļa saskarnes instalēšana, lai iegūtu informāciju par Zabbix lietotāja saskarnes instalēšanas vedni.

Zabbix aģenta 2 ielādējamo spraudņu instalēšana

Zabbix aģenta 2 ielādējamo spraudņu instalēšana ir nepieciešama tikai tad, ja vēlaties uzraudzīt mērķus, kurus neaptver iebūvētie spraudņi (piemēram, MongoDB serverus vai klasterus, PostgreSQL un tā atvasinājumus u.c.). Skatiet pilnu ielādējamo spraudņu un iebūvēto spraudņu sarakstu.

Attention:
Pirms spraudņa instalēšanas, lūdzu, pārbaudiet tā README failu. Tajā var būt ietvertas īpašas prasības un instalēšanas norādes.

Lai instalētu no pirmkoda, vispirms lejupielādējiet un atarhivējiet ielādējamā spraudņa pirmkoda arhīvu.

Lai kompilētu spraudni, pārejiet uz atarhivētā spraudņa direktoriju un palaidiet `make`:

```
make
```

Note:
Lai izveidotu Zabbix aģenta 2 ielādējamos spraudņus, jābūt instalētai atbalstītai Go versijai.

Spraudņa izpildfailu var novietot jebkur, ja vien to var ielādēt Zabbix aģents 2. Norādiet ceļu uz spraudņa bināro failu spraudņa konfigurācijas failā, piemēram, PostgreSQL spraudņa failā postgresql.conf:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Ceļš uz spraudņa konfigurācijas failu jānorāda Zabbix aģenta 2 konfigurācijas faila parametrā Include:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Lai iegūtu plašāku informāciju par spraudņu konfigurēšanu, turpiniet ar iestatīšanu.

Zabbix nodrošinātie ielādējamie spraudņi izmanto vienkāršus makefile ar šādiem būvēšanas mērķiem:

- `make` - izveidot spraudni
- `make clean` - dzēst visus failus, kas izveidoti spraudņa būvēšanas laikā
- `make check` - palaist pašpārbaudes (nepieciešams reāls uzraudzības mērķis, piemēram, PostgreSQL datubāze)
- `make style` - pārbaudīt Go koda stilu ar `golangci-lint`
- `make format` - formatēt Go kodu ar `go fmt`
- `make dist` - izveidot pirmkoda arhīvu, iekļaujot visas atkarības

Java gateway instalēšana

Java gateway ir jāinstalē tikai tad, ja vēlaties uzraudzīt JMX lietojumprogrammas. Java gateway ir viegls un tam nav nepieciešama datu bāze.

Lai instalētu no pirmkoda, vispirms lejupielādējiet un izpakojiet pirmkoda arhīvu.

Lai kompilētu Java gateway, palaidiet `./configure` skriptu ar `--enable-java` opciju. Ieteicams norādīt `--prefix` opciju, lai pieprasītu instalēšanas ceļu, kas atšķiras no noklusējuma /usr/local, jo Java gateway instalēšana izveidos visu direktoriju koku, nevis tikai vienu izpildāmo failu.

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

Lai kompilētu un sapakotu Java gateway JAR failā, palaidiet `make`. Ņemiet vērā, ka šim solim jūsu ceļā būs nepieciešami `javac` un `jar` izpildāmie faili.

```
make
```

Tagad jums ir fails zabbix-java-gateway-$VERSION.jar direktorijā src/zabbix_java/bin. Ja jums ir pieņemami palaist Java gateway no src/zabbix_java izplatīšanas direktorijā, tad varat turpināt ar norādījumiem par Java gateway konfigurēšanu un palaišanu. Pretējā gadījumā pārliecinieties, ka jums ir pietiekamas tiesības, un palaidiet `make install`.

```
make install
```

Lai iegūtu sīkāku informāciju par Java gateway konfigurēšanu un palaišanu, pārejiet uz iestatīšanu.

Zabbix tīmekļa pakalpojuma instalēšana

Zabbix tīmekļa pakalpojuma instalēšana ir nepieciešama tikai tad, ja vēlaties izmantot ieplānotās atskaites.

Lai instalētu no pirmkoda, vispirms lejupielādējiet un izsaiņojiet pirmkoda arhīvu.

Lai kompilētu Zabbix tīmekļa pakalpojumu, palaidiet `./configure` skriptu ar `--enable-webservice` opciju.

> **Note:**
> Lai izveidotu Zabbix tīmekļa pakalpojumu, jābūt instalētai atbalstītai Go versijai.

Palaidiet zabbix_web_service datorā, kurā ir instalēts tīmekļa pakalpojums:

```
zabbix_web_service
```

Lai iegūtu plašāku informāciju par ieplānoto atskaišu ģenerēšanas konfigurēšanu, pārejiet uz iestatīšanu.

Zabbix pirmkoda iegūšana

Ir vairāki veidi, kā iegūt Zabbix pirmkodu:

- Jūs varat lejupielādēt izlaistās stabilās versijas no oficiālās Zabbix tīmekļvietnes
- Jūs varat lejupielādēt nakts būvējumus no oficiālās Zabbix tīmekļvietnes izstrādātāju lapas
- Jūs varat iegūt jaunāko izstrādes versiju no Git pirmkoda repozitoriju sistēmas:
  - Pilnā repozitorija galvenā atrašanās vieta ir https://git.zabbix.com/scm/zbx/zabbix.git
  - Master un atbalstītie laidieni tiek arī spoguļoti Github vietnē: https://github.com/zabbix/zabbix

Lai klonētu repozitoriju, jābūt instalētam Git klientam. Oficiālās komandrindas Git klienta pakotnes nosaukums distribūcijās parasti ir **git**. Lai to instalētu, piemēram, Debian/Ubuntu sistēmā, izpildiet:

```
sudo apt-get update
sudo apt-get install git
```

Lai iegūtu visu Zabbix pirmkodu, pārejiet uz direktoriju, kurā vēlaties novietot kodu, un izpildiet:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Kompilēšanas problēmas

Šīs ir zināmās problēmas saistībā ar Zabbix kompilēšanu no pirmkoda. Visos pārējos gadījumos skatiet lapu Zināmās problēmas.

Bibliotēka nestandarta atrašanās vietā

Zabbix ļauj norādīt bibliotēku, kas atrodas nestandarta atrašanās vietā. Tālāk redzamajā piemērā Zabbix palaidīs `curl-config` no norādītās nestandarta atrašanās vietas un izmantos tā izvadi, lai noteiktu pareizo izmantojamo libcurl.

```
$ ./configure --enable-server --with-mysql --with-libcurl=/usr/local/bin/curl-config
```

Tas darbosies, ja tā būs vienīgā sistēmā instalētā libcurl, taču var nedarboties, ja standarta atrašanās vietā ir instalēta cita libcurl (piemēram, ar pakotņu pārvaldnieka palīdzību). Tā notiek gadījumos, kad Zabbix nepieciešama jaunāka bibliotēkas versija, bet citām lietotnēm — vecākā.

Tāpēc komponenta norādīšana nestandarta atrašanās vietā ne vienmēr darbosies, ja tas pats komponents pastāv arī standarta atrašanās vietā.

Piemēram, ja izmantojat jaunāku libcurl, kas instalēta `/usr/local`, kamēr libcurl pakotne joprojām ir instalēta, Zabbix var izvēlēties nepareizo versiju un kompilēšana neizdosies:

```
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
```

Šajā gadījumā funkcija `curl_easy_header()` nav pieejama vecākajā `/usr/lib/x86_64-linux-gnu/libcurl.so`, bet ir pieejama jaunākajā `/usr/local/lib/libcurl.so`.

Problēma ir saistīta ar linkera karogu secību, un viens no risinājumiem ir norādīt pilnu ceļu uz bibliotēku mainīgajā LDFLAGS:

```
$ LDFLAGS="-Wl,--no-as-needed /usr/local/lib/libcurl.so" ./configure --enable-server --with-mysql --with-l
```

Ņemiet vērā opciju `-Wl,--no-as-needed`, kas dažās sistēmās var būt nepieciešama (skatiet arī: noklusējuma linkošanas opcijas Debian bāzētās sistēmās).

Pārāk mazs steka izmērs dažās sistēmās

Ja Zabbix avarē vai sastingst steka pārpildes dēļ, varat palielināt steka izmēru katram pavedienam, izmantojot opciju `--with-stacksize`, konfigurējot avota kodu. Šī problēma var rasties sistēmās ar zemu noklusējuma pavediena steka ierobežojumu, īpaši priekšapstrādes laikā, kad tiek izveidoti vairāki pavedieni.

Šajā piemērā steka izmērs tiek iestatīts uz 512 KB katram pavedienam:

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

Linux sistēmās sistēmas pavedienu steka ierobežojumus izpildlaikā var pārbaudīt, izmantojot komandu `ulimit -s`.

Piezīmes par instalēšanu no pakotnēm

Pārskats

Oficiālās Zabbix 8.0 **pirmsizlaides** pakotnes ir pieejamas Zabbix tīmekļvietnē.

Izvēlieties savu operētājsistēmu un Zabbix komponentu, lai ģenerētu jūsu videi piemērotas instalēšanas instrukcijas. Skatiet arī pakotņu instalēšanas piezīmes šajā lapā, kur pieejama svarīga papildu informācija, kā arī sadaļu Installation and setup.

Pakotnes ir pieejamas šādām Linux distribūcijām:

- Red Hat Enterprise Linux un tā atvasinājumiem: 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:**
> Dažas OS distribūcijas (īpaši uz Debian balstītās distribūcijas) nodrošina savas Zabbix pakotnes. Šīs pakotnes Zabbix **neatbalsta**, un tās var būt novecojušas vai tajās var trūkt jaunāko iespēju un kļūdu labojumu. Ieteicams izmantot tikai oficiālās pakotnes no Zabbix Official Repository. Ja iepriekš instalējāt Zabbix no savas operētājsistēmas repozitorija, skatiet darbības sadaļā upgrading Zabbix packages from OS repositories.

Pakotnes atbalsta MySQL/PostgreSQL datubāzi un Apache/Nginx tīmekļa serveri. Ņemiet vērā, ka Zabbix serveris un starpniekserveris nevar koplietot vienu un to pašu datubāzi; izmantojiet atšķirīgus datubāzu nosaukumus, ja abi ir instalēti vienā hostā.

Ja nepieciešams, atsevišķas pakotnes Zabbix aģents/aģents 2, Zabbix get un Zabbix sender ir pieejamas Zabbix Official Repository.

Zabbix nodrošina arī iepriekš kompilētus Zabbix aģents bināros failus operētājsistēmām, kas nav Linux; skatiet:

- Windows aģents instalēšana no MSI
- macOS aģents instalēšana no PKG
- Legacy binaries (vecākām/retāk sastopamām sistēmām, piemēram, HP-UX, NetBSD, Tru64 un vecākām SLES versijām)

Pakotņu instalēšanas piezīmes

Tālāk norādītās piezīmes attiecas uz visām sistēmām:

- Ja izmantojat PostgreSQL, `DBHost=localhost` (vai IP adresi) Zabbix serveris/starpniekserveris konfigurācijā liek PostgreSQL izmantot tīkla ligzdu lokālās UNIX ligzdas vietā; ar to saistītas iestatīšanas instrukcijas skatiet sadaļā SELinux konfigurācija.
- Ja izmantojat TimescaleDB, skatiet papildu TimescaleDB iestatīšanu.
- Ja instalējat Zabbix Java gateway (JMX lietotņu uzraudzībai), skatiet papildu iestatīšanas instrukcijas uz RHEL balstītām sistēmām un uz Debian balstītām sistēmām.
- Lai palaistu Zabbix aģents kā root, skatiet Aģenta palaišana kā root.

Tālāk norādītās piezīmes attiecas uz RHEL un tā atvasinājumiem:

- Ja EL9 sistēmai esat iespējojis EPEL repozitoriju, kas arī nodrošina Zabbix pakotnes, tas pirms oficiālo Zabbix pakotņu instalēšanas ir jāizslēdz no pakotņu atlases; skatiet Nejauša EPEL Zabbix pakotņu instalēšana.
- Lai instalētu Zabbix pakotnes Red Hat UBI vidēs, skatiet Zabbix pakotnes RHEL sistēmām Red Hat UBI vidēs.
- Lai izmantotu ICMP ping vienumus, `fping` pakotnes ir pieejamas arī Zabbix oficiālajā repozitorijā.

SELinux konfigurācija

Zabbix izmanto uz ligzdām balstītu starpprocesu saziņu. Sistēmās, kur ir iespējots Security-Enhanced Linux (SELinux), jums var būt nepieciešams pievienot SELinux noteikumus, lai atļautu Zabbix izveidot/izmantot UNIX domēna ligzdas SocketDir direktorijā. Ligzdu faili tiek izmantoti Zabbix serveris (alerter, priekšapstrāde, IPMI) un Zabbix starpniekserveris (IPMI), un tie pastāv, kamēr process darbojas.

Ja SELinux ir iespējots enforcing režīmā, izpildiet šādas komandas, lai iespējotu saziņu starp Zabbix lietotāja saskarne un serveris:

RHEL 7 (un jaunākām versijām), AlmaLinux, CentOS Stream, Oracle Linux, Rocky Linux 8 (un jaunākām versijām):

```
setsebool -P httpd_can_connect_zabbix on
```

Ja datubāzei piekļūst pa tīklu (ieskaitot `localhost` PostgreSQL gadījumā), atļaujiet arī Zabbix lietotāja saskarnei pieslēgties datubāzei:

```
setsebool -P httpd_can_network_connect_db on
```

RHEL versijām pirms 7:

```
setsebool -P httpd_can_network_connect on
setsebool -P zabbix_can_network on
```

Pēc SELinux iestatījumu piemērošanas pārstartējiet Apache:

```
systemctl restart httpd
```

Pēc izvēles varat instalēt iepriekš definētu `zabbix-selinux-policy` pakotni no Zabbix oficiālās repozitorija. Šī pakotne ir pieejama visām atbalstītajām OS versijām, lai vienkāršotu Zabbix izvēršanu un novērstu situācijas, kad lietotāji konfigurācijas sarežģītības dēļ izslēdz SELinux.

Attention:
Maksimālai drošībai ieteicams iestatīt pielāgotus SELinux iestatījumus.

`zabbix-selinux-policy` pakotne satur pamata SELinux politiku, kas ļauj Zabbix izveidot un izmantot ligzdas, kā arī iespējot HTTPd savienojumu ar PostgreSQL (izmanto lietotāja saskarne).

Avota fails `zabbix_policy.te` satur šādus noteikumus:

```
module zabbix_policy 1.2;

require {
    type zabbix_t;
    type zabbix_port_t;
    type zabbix_var_run_t;
    type postgresql_port_t;
    type httpd_t;
    class tcp_socket name_connect;
```

```
    class sock_file { create unlink };
    class unix_stream_socket connectto;
}

####============= zabbix_t ==============
allow zabbix_t self:unix_stream_socket connectto;
allow zabbix_t zabbix_port_t:tcp_socket name_connect;
allow zabbix_t zabbix_var_run_t:sock_file create;
allow zabbix_t zabbix_var_run_t:sock_file unlink;
allow httpd_t zabbix_port_t:tcp_socket name_connect;

####============= httpd_t ==============
allow httpd_t postgresql_port_t:tcp_socket name_connect;
```

Debuginfo pakotnes

Debuginfo pakotnes satur Zabbix bināro failu atkļūdošanas simbolus. Tās nav nepieciešamas parastai instalēšanai vai darbībai, taču ir noderīgas padziļinātai problēmu novēršanai.

Lai iespējotu `zabbix-debuginfo` repozitoriju:

- RHEL 7 rediģējiet `/etc/yum.repos.d/zabbix.repo` un sadaļai `zabbix-debuginfo` iestatiet `enabled=1`:

```
[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
```

- SUSE rediģējiet `/etc/zypp/repos.d/zabbix.repo` un sadaļai `zabbix-debuginfo` iestatiet `enabled=1`:

```
[zabbix-debuginfo]
name=Zabbix Official Repository debuginfo
type=rpm-md
baseurl=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/$basearch/debuginfo/
gpgcheck=1
gpgkey=https://repo.zabbix.com/zabbix/8.0/stable/sles/15/$basearch/debuginfo/repodata/repomd.xml.key
enabled=0
update=1
```

Kad tas ir iespējots, instalējiet pakotnes:

- RHEL instalējiet vienu pakotni ar atkļūdošanas informāciju visiem Zabbix komponentiem:

```
dnf install zabbix-debuginfo
```

- SUSE instalējiet komponentam specifiskas debuginfo pakotnes:

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

Nestabilu laidienu repozitoriju iespējošana

Tālāk sniegtās instrukcijas ir paredzētas nestabilo Zabbix laidienu repozitoriju iespējošanai (pēc noklusējuma tie ir atspējoti), kas tiek izmantoti nelielu Zabbix versiju laidienu kandidātiem.

Vispirms instalējiet vai atjauniniet uz jaunāko zabbix-release pakotni. Lai savā sistēmā iespējotu rc pakotnes, rīkojieties šādi:

Red Hat Enterprise Linux

Atveriet failu `/etc/yum.repos.d/zabbix.repo` un iestatiet `enabled=1` repozitorijam `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

Atveriet `/etc/apt/sources.list.d/zabbix.list` un noņemiet komentāra zīmi rindai "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

Atveriet failu `/etc/zypp/repos.d/zabbix.repo` un iestatiet `enabled=1` repozitorijam `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
```

Windows aģenta instalēšana no MSI

Pārskats

Zabbix aģentu var instalēt operētājsistēmā Windows, izmantojot 32 bitu vai 64 bitu MSI instalēšanas pakotnes, kas ir pieejamas lejupielādei.

Minimālās OS prasības MSI instalēšanai ir:

- **Zabbix aģentam:** Windows XP (64 bitu) vai Windows Server 2003
- **Zabbix aģentam 2:** Windows 10 (32 bitu) vai Windows Server 2016

32 bitu pakotnes nevar instalēt 64 bitu sistēmās.

Pakotnēs ir iekļauts:

- TLS atbalsts (TLS konfigurācija nav obligāta)
- utilītas Zabbix get un Zabbix sender (tās var instalēt kopā ar Zabbix aģentu/aģentu 2 vai atsevišķi)

> **Attention:**
> Zabbix aģenta 2 pakotnēs nav iekļauti ielādējami spraudņi (MongoDB, PostgreSQL, MSSQL), kas ir jālejupielādē un jāinstalē atsevišķi.

Instalēšanu var veikt, izmantojot iestatīšanas vedni vai komandrindu.

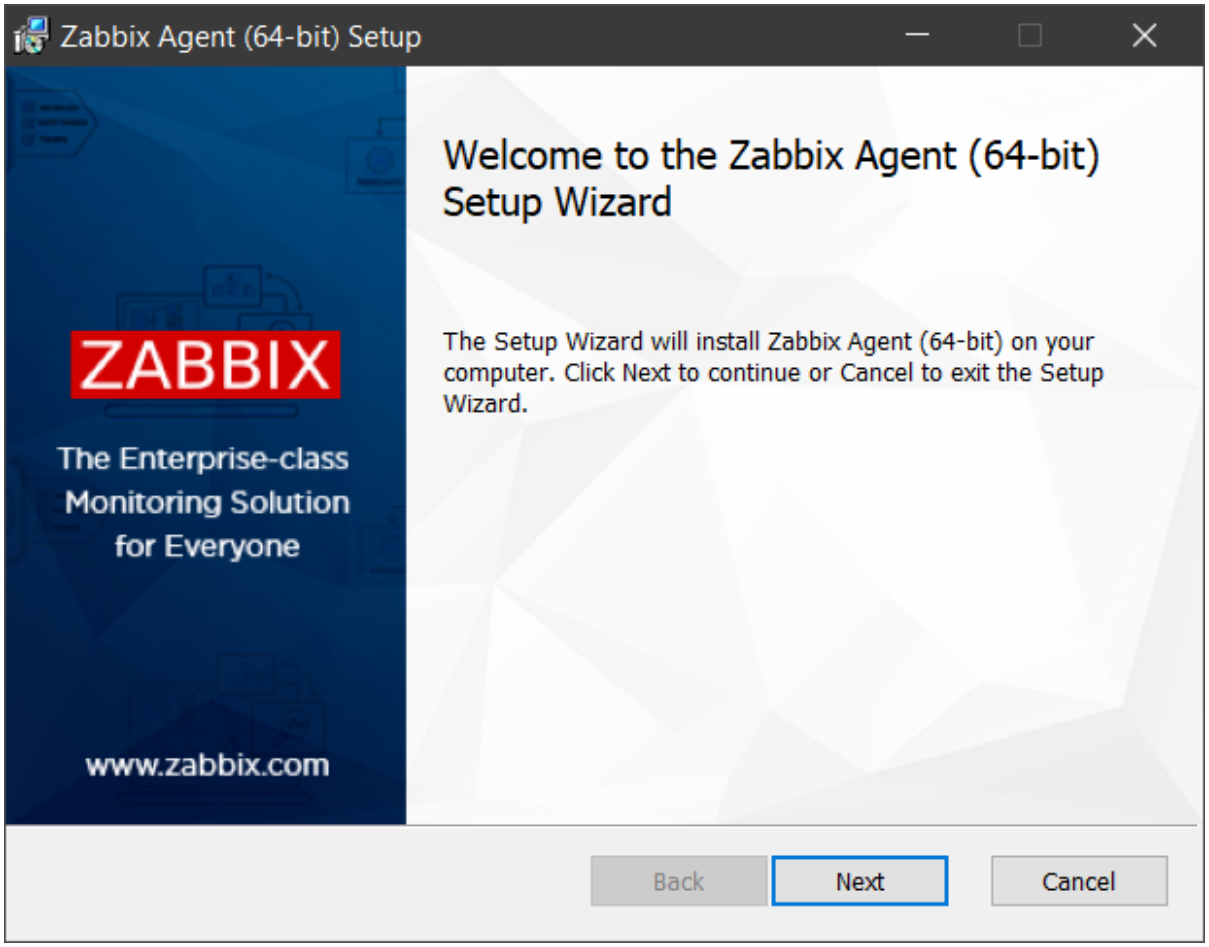
Lai gan instalēšana, izmantojot MSI pakotnes, tiek pilnībā atbalstīta, pareizai kļūdu apstrādei ieteicams instalēt vismaz Microsoft .NET Framework 2.

> **Attention:**
> Ieteicams izmantot instalētāja piedāvāto noklusējuma instalēšanas vietu. Pielāgotas atrašanās vietas izmantošana bez nepieciešamajām atļaujām var apdraudēt instalācijas drošību.

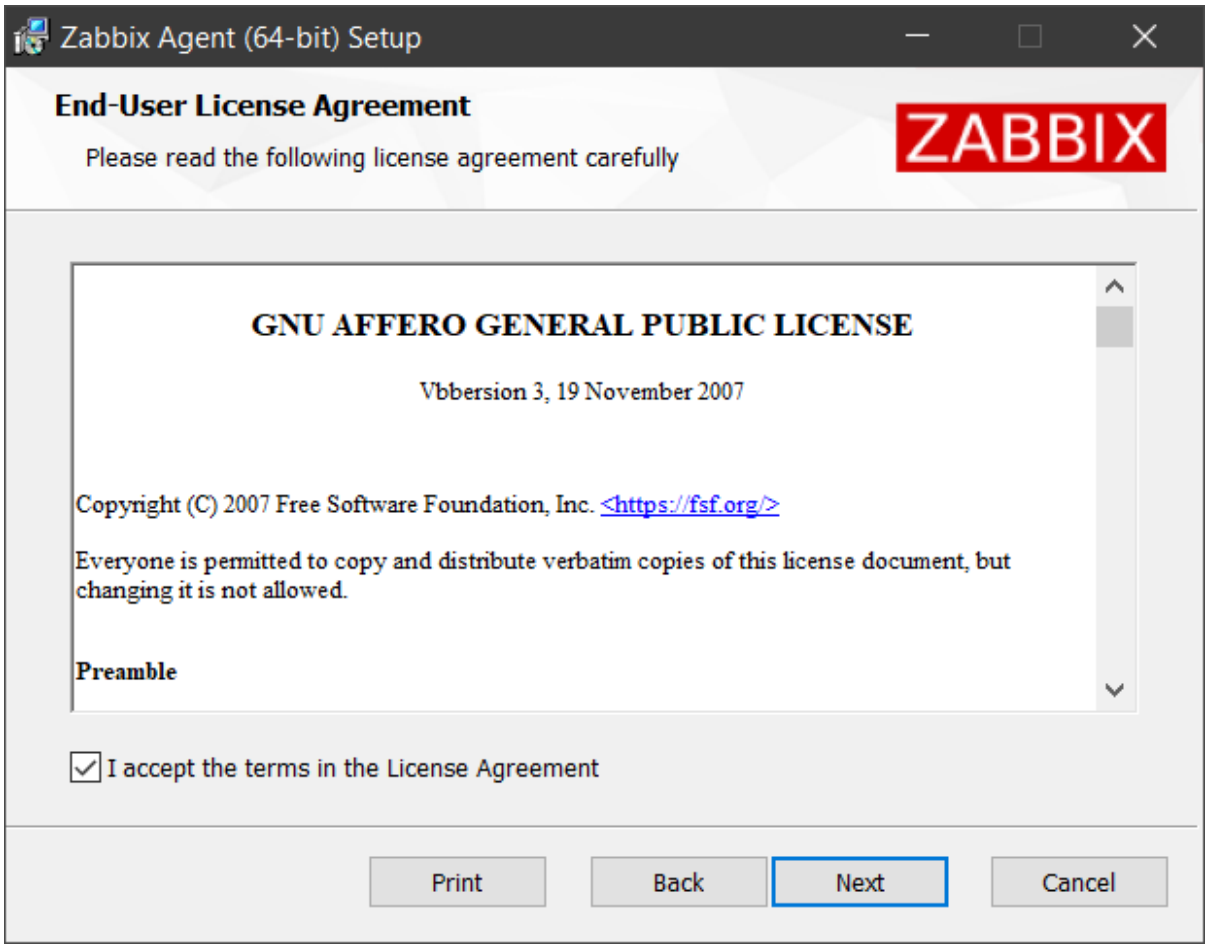
Instalēšana, izmantojot iestatīšanas vedni

Tālāk norādītās instalēšanas darbības attiecas gan uz Zabbix aģents, gan uz Zabbix aģents 2.

1. Veiciet dubultklikšķi uz lejupielādētā MSI faila, lai sāktu instalēšanu:

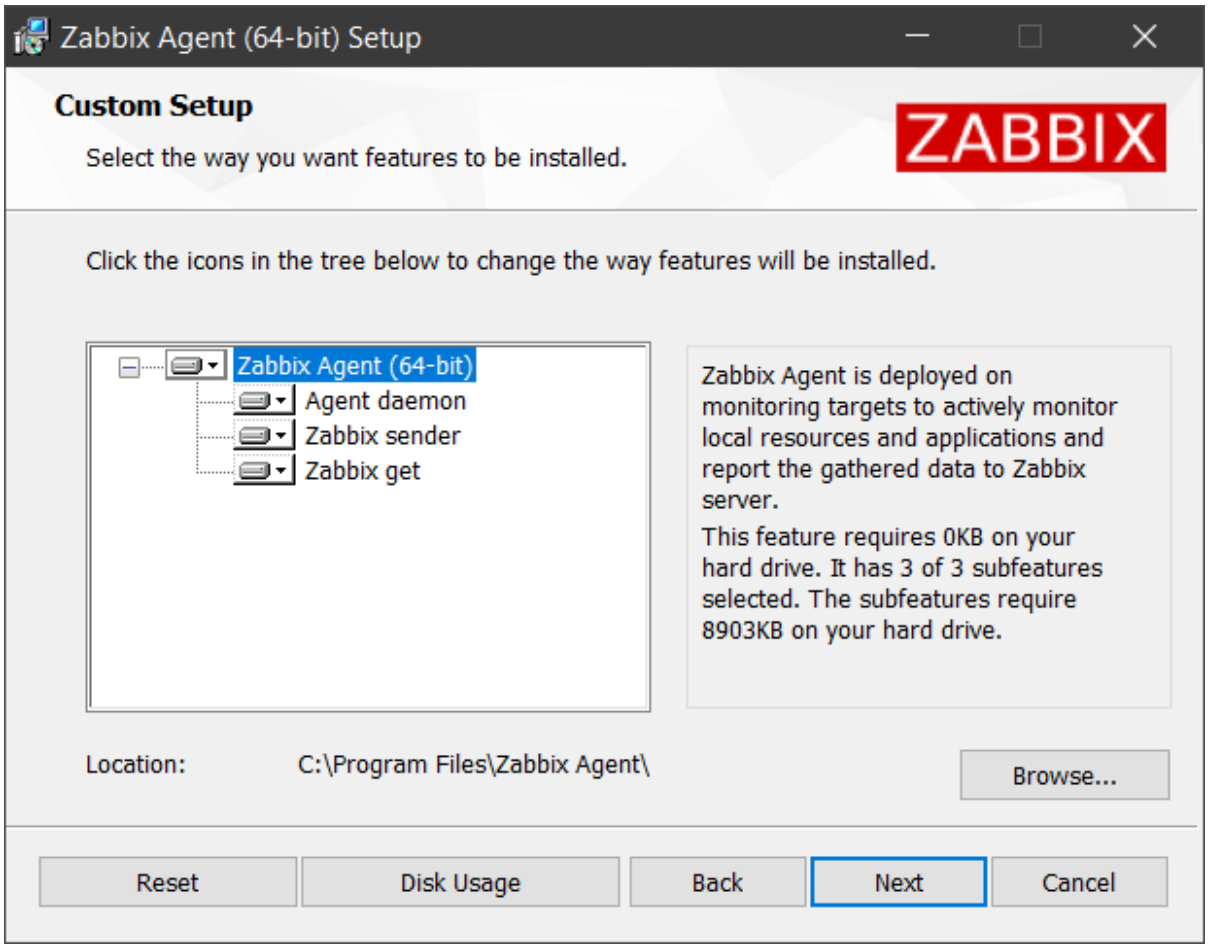


2. Pieņemiet galalietotāja licences līgumu:



3. Atlasiet instalējamos Zabbix komponentus (Aģenta dēmons, Zabbix sender, Zabbix get):

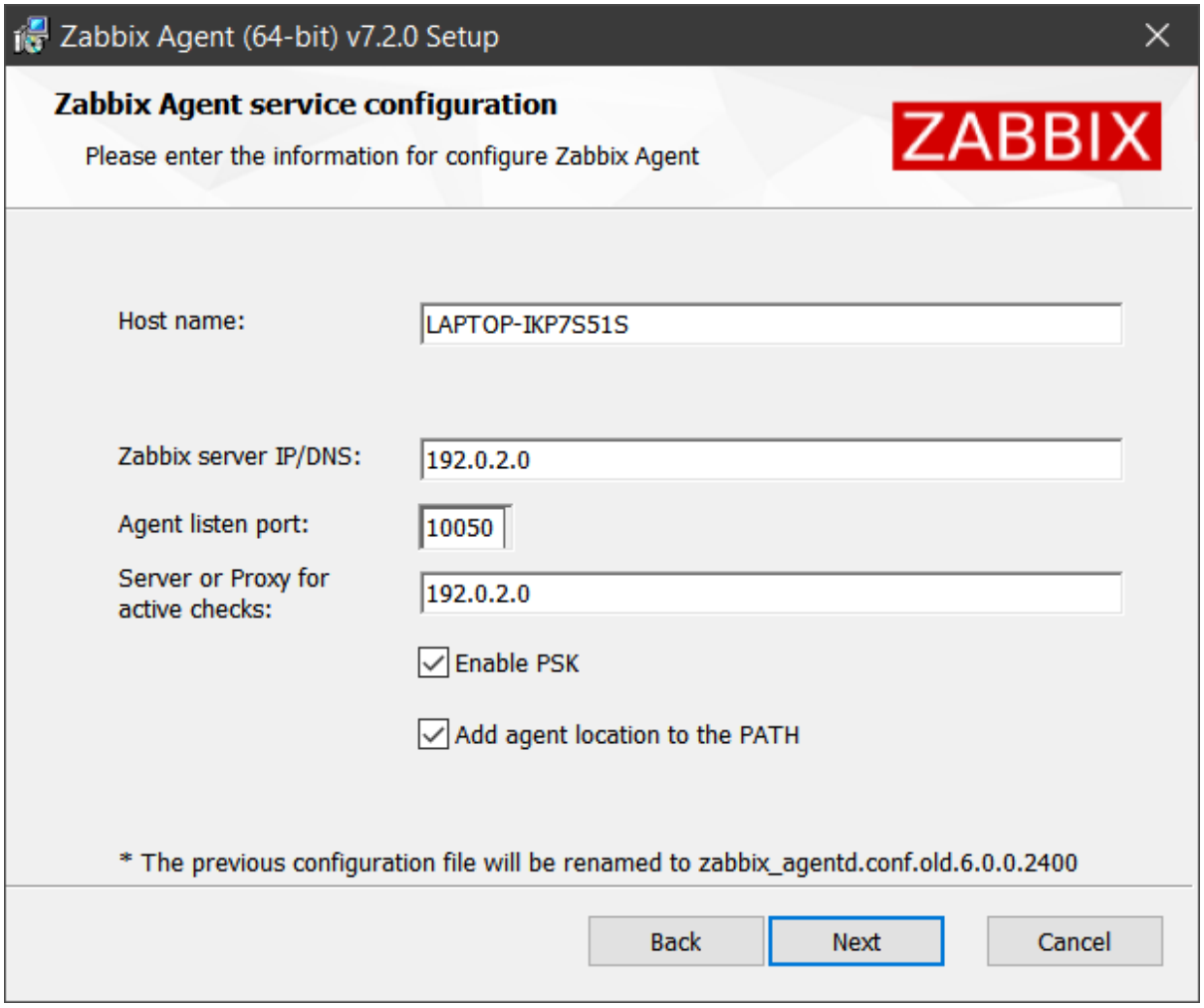
Attention:
Ieteicams izmantot instalētāja piedāvāto noklusējuma instalēšanas atrašanās vietu. Pielāgotas atrašanās vietas izmantošana bez nepieciešamajām atļaujām var apdraudēt instalācijas drošību.



4. Konfigurējiet tālāk norādītos parametrus. To vērtības tiks iestatītas Zabbix aģents konfigurācijas failā:

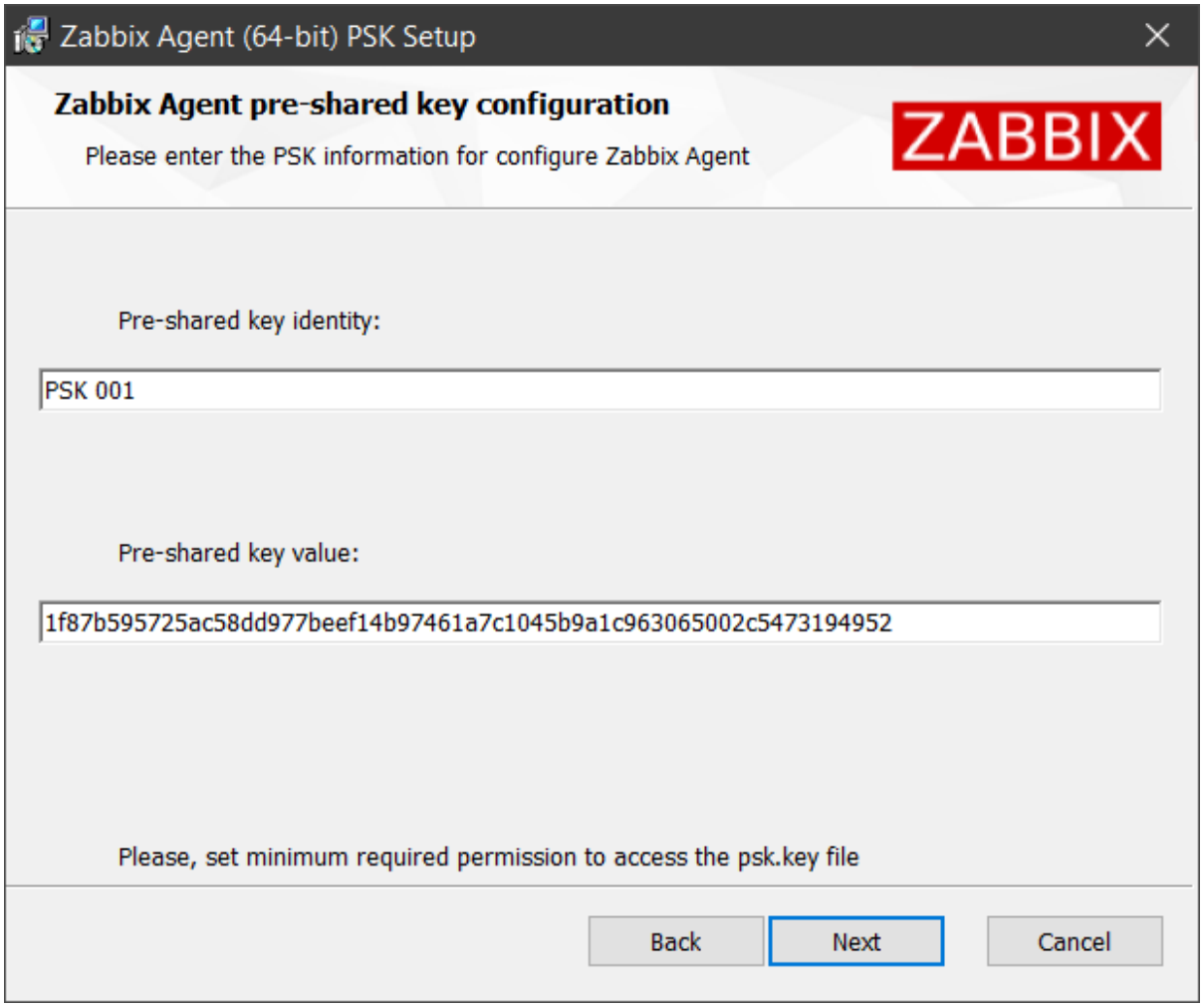
Parameter	Description
Host name	Tās mašīnas hosta nosaukums, kurā tiek instalēts Zabbix aģents. Iestata parametru Hostname.
Zabbix server IP/DNS	Ar komatiem atdalīts Zabbix serveru vai Zabbix starpniekserveru IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Šis parametrs ir **obligāts**. Iestata parametru Server.
Agent listen port	Aģents klausīsies šajā portā savienojumiem no servera. Iestata parametru ListenPort.
Server or Proxy for active checks	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes. Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums un neobligāts ports, kas atdalīts ar kolu. Iestata parametru ServerActive.
Enable PSK	Atzīmējiet izvēles rūtiņu, lai iespējotu TLS atbalstu izmantojot iepriekš koplietotas atslēgas. Iestata parametrus TLSConnect un TLSAccept uz `psk`.
Add agent location to the PATH	Atzīmējiet izvēles rūtiņu, lai pievienotu Zabbix aģents atrašanās vietu sistēmas PATH mainīgajam.

Note:
Ja tiek atrasts esošs Zabbix aģents, tiks parādītas parametru vērtības no tā konfigurācijas faila. Turklāt instalēšanas laikā esošais konfigurācijas fails tiks pārdēvēts, un tiks izveidots jauns konfigurācijas fails.



5. Konfigurējiet PSK parametrus, ja iepriekšējā darbībā atzīmējāt izvēles rūtiņu *Enable PSK*. Šie parametri arī tiks iestatīti Zabbix aģents konfigurācijas failā:

Parameter	Description
Pre-shared key identity	Iepriekš koplietotās atslēgas identitātes virkne. Iestata parametru TLSPSKIdentity.
Pre-shared key value	Iepriekš koplietotās atslēgas virknes vērtība. Izveido failu psk.key, kas satur atslēgu, un iestata parametru TLSPSKFile uz atslēgas atrašanās vietu (noklusējums: `C:\Program Files\Zabbix Agent\psk.key`). Ir **ieteicams** ierobežot piekļuvi iepriekš koplietotās atslēgas failam, pielāgojot faila drošības iestatījumus tā, lai to varētu lasīt tikai Zabbix aģents (vai lietotājs, kas palaiž aģentu).

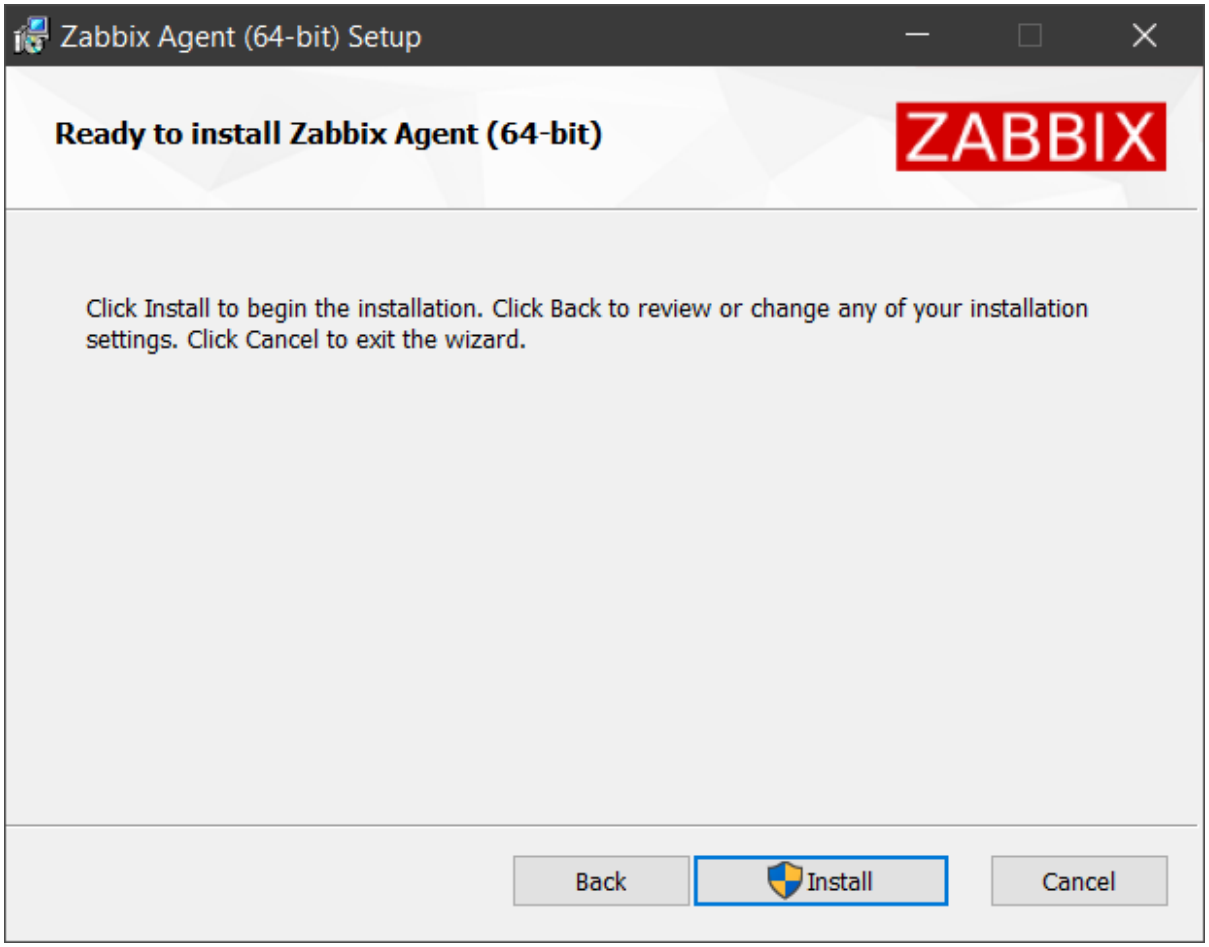


6. Noklikšķiniet uz *Install*, lai sāktu instalēšanu.

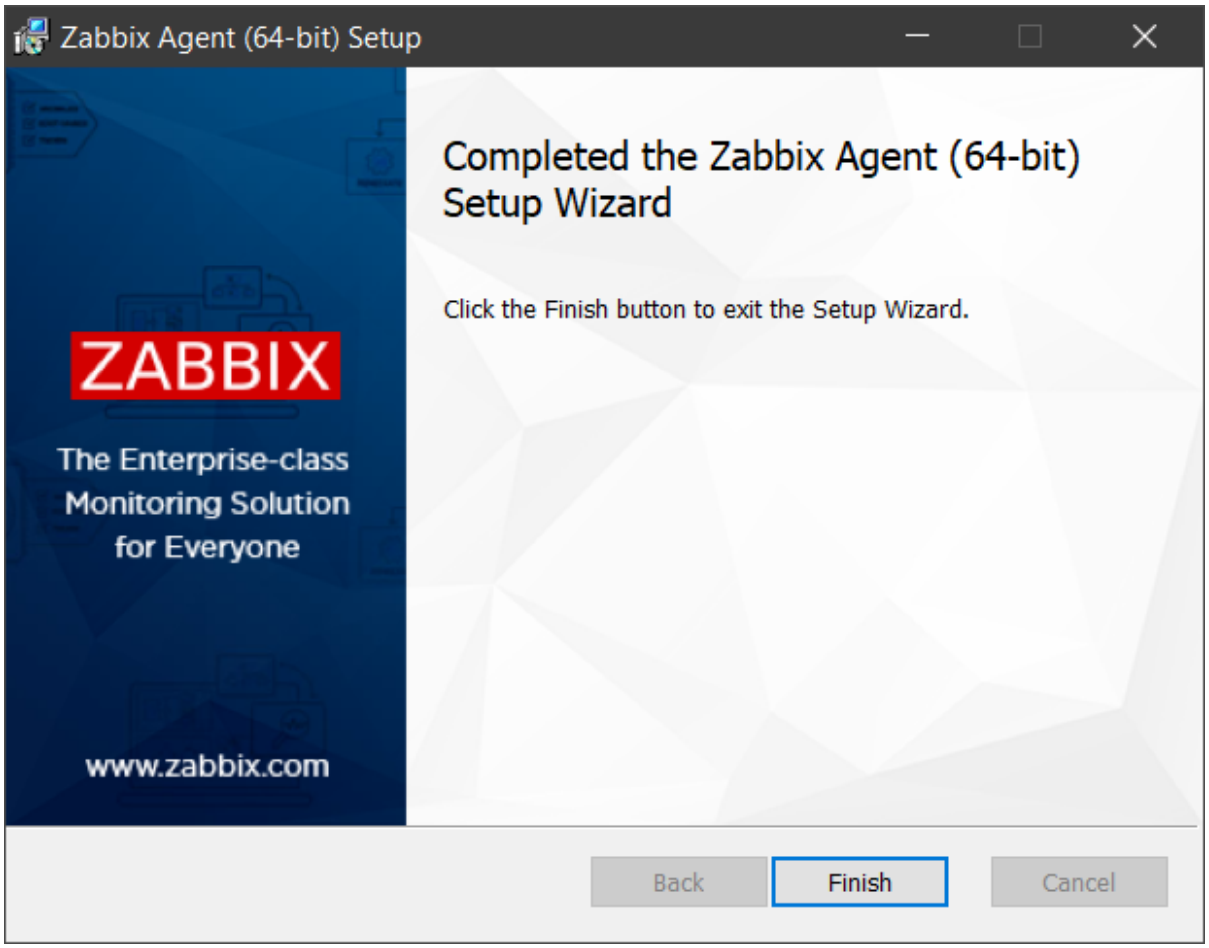
Visi atlasītie Zabbix komponenti un Zabbix aģents konfigurācijas fails tiks instalēti jūsu norādītajā atrašanās vietā (noklusējums: `C:\Program Files\Zabbix Agent`). Tas pats attiecas uz Zabbix aģents 2, izņemot to, ka papildu konfigurācijas faili tā iebūvētajiem spraudņiem tiks instalēti apakšmapē `zabbix_agent2.d\plugins.d`.

Papildus tam zabbix_agentd.exe (vai zabbix_agent2.exe) tiks iestatīts kā Windows pakalpojums ar aizkavētu automātisko palaišanu (vai automātisko palaišanu Windows versijās pirms Windows Vista/Server 2008).

Ja instalēšanas laikā darbojas cita Zabbix aģents versija, jums tiks piedāvāts izvēlēties vai nu aizvērt lietojumprogrammu un mēģināt to restartēt, vai atstāt to atvērtu; tādā gadījumā būs nepieciešama sistēmas pārstartēšana.



7. Noklikšķiniet uz pogas *Finish*, lai izietu no iestatīšanas vedņa.



Instalēšana no komandrindas

Zabbix aģents var tikt instalēts no komandrindas, palaižot MSI instalētāju ar msiexec. Piemēram:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent-8.0.0-windows-amd64-openssl.msi" /qn+ SERVER=192.0.2
```

Šī metode ļauj veikt neuzraudzītas instalācijas un pielāgotas konfigurācijas, izmantojot parametrus.

Atbalstītie parametri

Zabbix aģenta MSI instalēšanas pakotnes atbalsta tālāk norādītos parametrus gan Zabbix aģentam, gan Zabbix aģentam 2.

Note:
Zabbix aģenta/aģenta2 parametri instalēšanas laikā tiek iestatīti konfigurācijas failā. Noklikšķiniet uz parametra nosaukuma, lai skatītu tā detalizētu aprakstu un konfigurācijas piemērus lapā Zabbix aģents (Windows). Zabbix aģentam 2 skatiet lapu Zabbix aģents 2 (Windows).

Parametrs	Apraksts
ADDDEFAULT	Ar komatiem atdalīts komponentu saraksts, kas jāinstalē to noklusējuma konfigurācijā. Papildinformāciju skatiet ADDDEFAULT property. Iespējamās vērtības: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Piemērs: `ADDDEFAULT=AgentProgram,GetProgram`
ADDLOCAL	Ar komatiem atdalīts komponentu saraksts, kas jāinstalē lokāli. Papildinformāciju skatiet ADDLOCAL property. Iespējamās vērtības: `AgentProgram`, `GetProgram`, `SenderProgram`, `ALL` Piemērs: `ADDLOCAL=AgentProgram,SenderProgram`
ALLOWDENYKEY	Ar semikoliem atdalīts AllowKey vai DenyKey parametru saraksts, lai ierobežotu Zabbix aģenta pārbaudes. Ja nepieciešams, izmantojiet atpakaļvērsto slīpsvītru, lai atdalītāju atsoļotu (`\;`). Iestata AllowKey un DenyKey parametrus aģenta konfigurācijas failā. Piemērs: `ALLOWDENYKEY="AllowKey=system.run[more C:\Windows\System32\drivers\etc\hosts\\; echo 'File read complete'];DenyKey=system.run[*]"`
CONF	Pilns ceļš uz Zabbix aģenta veidnes konfigurācijas failu. Instalēšanas laikā šis fails kļūs par aģenta konfigurācijas failu. Failā jābūt vismaz parametriem Server un LogFile. Piemērs: `CONF="C:\full\path\to\example.conf"`
DONOTSTART	Izmantojiet `DONOTSTART=1`, lai neļautu MSI instalētājam palaist Zabbix aģenta servisu.
ENABLEPATH	Izmantojiet `ENABLEPATH=1`, lai pievienotu Zabbix aģenta atrašanās vietu sistēmas PATH mainīgajam.
ENABLEPERSISTENTBUFFER	Tikai Zabbix aģentam 2. Iespējo lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
HOSTINTERFACE	Neobligāts parametrs, kas nosaka hosta saskarni.
HOSTMETADATA	Neobligāts parametrs, kas nosaka hosta metadatus.
HOSTMETADATAITEM	Neobligāts parametrs, kas nosaka vienumu, ko izmantot hosta metadatu iegūšanai.
HOSTNAME	Neobligāts parametrs, kas nosaka hosta nosaukumu.
HOSTNAMEITEM	Neobligāts parametrs, kas nosaka vienumu, ko izmantot hosta nosaukuma iegūšanai.
INCLUDE	Ar semikoliem atdalīts atsevišķu failu vai visu direktorijā esošo failu saraksts, kas jāiekļauj Zabbix aģenta konfigurācijas failā.
INSTALLFOLDER	Pilns ceļš uz mapi, kurā tiks instalēti Zabbix komponenti un Zabbix aģenta konfigurācijas fails. Zabbix aģentam 2 papildu konfigurācijas faili iebūvētajiem spraudņiem tiks instalēti apakšmapē `zabbix_agent2.d\plugins.d`. Piemērs: `INSTALLFOLDER="C:\Program Files\Zabbix Agent"`
LISTENIP	Ar komatiem atdalīts IP adrešu saraksts, kurās aģentam jāgaida savienojumi.
LISTENPORT	Aģents šajā portā gaidīs savienojumus no servera.
LOGFILE	Zabbix aģenta žurnālfaila nosaukums.
LOGTYPE	Žurnāla izvades tips.
NONMSICONFNAME	Pilns ceļš uz Zabbix aģenta pielāgotu konfigurācijas failu. Instalēšanas laikā visi šajā failā esošie derīgie aģenta konfigurācijas parametri (tikai tie, kas uzskaitīti šajā tabulā) tiks ierakstīti jaunizveidotajā aģenta konfigurācijas failā. Failā jābūt vismaz parametram Server. Piemērs: `NONMSICONFNAME="C:\full\path\to\example.conf"`
PERSISTENTBUFFERFILE	Tikai Zabbix aģentam 2. Fails, kurā Zabbix aģentam 2 jāglabā SQLite datubāze.
PERSISTENTBUFFERPERIOD	Tikai Zabbix aģentam 2. Laika periods, cik ilgi dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri.
SERVER	Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Šis parametrs ir **obligāts**, izņemot gadījumu, ja STARTAGENTS ir iestatīts uz 0.
SERVERACTIVE	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes.

Parametrs	Apraksts
SKIP	Izmantojiet `SKIP=fw`, lai neļautu MSI instalētājam pievienot Windows Firewall izņēmuma kārtulu Zabbix aģentam.
SOURCEIP	Izejošo savienojumu avota IP adrese uz Zabbix serveri vai Zabbix starpniekserveri, vai savienojumu izveidei dažu vienumu izpildes laikā (web.page.get, net.tcp.port u.c.).
STARTAGENTS	Iepriekš atzaroto zabbix_agentd instanču skaits, kas apstrādā pasīvās pārbaudes. Ja iestatīts uz 0, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.
STARTUPTYPE	Zabbix aģenta servisa palaišanas tips. Iespējamās vērtības: **automatic** - palaist servisu automātiski Windows startēšanas laikā; **delayed** - *(noklusējums)* aizkavēt servisa palaišanu pēc tam, kad ir pabeigta automātiski palaisto servisu startēšana (pieejams Windows Vista/Server 2008 un jaunākās versijās); **manual** - palaist servisu manuāli (lietotājam vai lietojumprogrammai); **disabled** - atspējot servisu, lai to nevarētu palaist lietotājs vai lietojumprogramma. Piemērs: `STARTUPTYPE=disabled`
STATUSPORT	Tikai Zabbix aģentam 2. Ja iestatīts, aģents šajā portā gaidīs HTTP statusa pieprasījumus (http://localhost:<port>/status).
TIMEOUT	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSACCEPT	Ienākošie savienojumi, kurus pieņemt (izmanto pasīvajām pārbaudēm). Ja iestatīts uz `psk`, tad arī TLSCONNECT tiks iestatīts uz `psk` (ja vien nav norādīts citādi).
TLSCAFILE	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei.
TLSCERTFILE	Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi.
TLSCONNECT	Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim (izmanto aktīvajām pārbaudēm). Ja iestatīts uz `psk`, tad arī TLSACCEPT tiks iestatīts uz `psk` (ja vien nav norādīts citādi).
TLSCRLFILE	Pilns ceļš uz failu, kas satur atsauktos sertifikātus.
TLSKEYFILE	Pilns ceļš uz failu, kas satur Zabbix aģenta privāto atslēgu.
TLSPSKFILE	Pilns ceļš uz failu, kas satur Zabbix aģenta iepriekš koplietoto atslēgu. Ja ir iestatīti gan TLSPSKFILE, gan TLSPSKVALUE, TLSPSKVALUE vērtība tiks ierakstīta failā, kas norādīts TLSPSKFILE. Ir **ieteicams** ierobežot piekļuvi iepriekš koplietotās atslēgas failam, pielāgojot faila drošības iestatījumus tā, lai to varētu lasīt tikai Zabbix aģents (vai lietotājs, kas palaiž aģentu).
TLSPSKIDENTITY	Iepriekš koplietotās atslēgas identitātes virkne.
TLSPSKVALUE	Iepriekš koplietotās atslēgas virknes vērtība. Ja ir iestatīti gan TLSPSKFILE, gan TLSPSKVALUE, TLSPSKVALUE vērtība tiks ierakstīta failā, kas norādīts TLSPSKFILE. Piemērs: `TLSPSKVALUE=1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952`
TLSSERVERCERTISSUER	Atļautais servera (starpniekservera) sertifikāta izdevējs.
TLSSERVERCERTSUBJECT	Atļautais servera (starpniekservera) sertifikāta subjekts.
UNSAFEUSERPARAMETERS	Atļaut visām rakstzīmēm tikt nodotām lietotāja definēto parametru argumentos.

Piemēri

Šis piemērs instalē Zabbix aģents ar pielāgotu konfigurāciju. Tas arī iespējo TLS atbalstu izmantojot iepriekš koplietotas atslēgas.

```
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[*]"
```

Nākamais piemērs instalē jaunāku Zabbix aģents versiju un izmanto **veidnes konfigurācijas failu** (`CONF="C:\agent-template.conf"`). Instalēšanas laikā šis fails kļūs par aģents konfigurācijas failu. Lai mantotu parametrus no vecā konfigurācijas faila, izmantojiet parametru vietturus (piemēram, `[AllowDenyKey]`).

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

#### agent-template.conf example:
```

```
LogFile=[LogFile]
[AllowDenyKey]
Server=192.0.2.8
Hostname=DESKTOP-X9F4A2B
[Include]
[TLSConnect]
[TLSAccept]
[TLSPSKIdentity]
[TLSPSKFile]
```

Alternatīvi varat izmantot **pielāgotu konfigurācijas failu** (`NONMSICONFNAME="C:\agent-custom.conf"`). Instalēšanas laikā visi derīgie aģents konfigurācijas parametri, kas atrodas šajā failā (tikai tie, kas norādīti iepriekš esošajā tabulā), tiks ierakstīti jaunizveidotajā aģents konfigurācijas failā. Lai saglabātu esošo aģents konfigurāciju, definējiet parametrus, kas jāsaglabā.

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

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

Zabbix aģents 2 ielādējamie spraudņi

Zabbix aģenta 2 ielādējamos spraudņus var instalēt operētājsistēmā Windows, izmantojot 64 bitu MSI instalēšanas pakotnes, kas ir pieejamas lejupielādei.

Minimālās OS prasības MSI instalēšanai ir Windows 10 (64 bitu) vai Windows Server 2016.

> **Attention:**
> Pirms spraudņa instalēšanas, lūdzu, pārbaudiet tā README failu. Tajā var būt ietvertas īpašas prasības un instalēšanas norādes.

Līdzīgi kā Zabbix aģentu/agent2, ielādējamos spraudņus var instalēt, izmantojot iestatīšanas vedni vai komandrindu.

Instalēšana no iestatīšanas vedņa

1. Veiciet dubultklikšķi uz lejupielādētā MSI faila, lai sāktu instalēšanu.
2. Pieņemiet gala lietotāja licences līgumu.
3. Atlasiet instalējamos Zabbix aģents 2 ielādējamos spraudņus.

> **Attention:**
> Ieteicams izmantot instalētāja piedāvāto noklusējuma instalēšanas vietu. Pielāgotas atrašanās vietas izmantošana bez nepieciešamajām atļaujām var apdraudēt instalācijas drošību.

4. Noklikšķiniet uz *Install*, lai sāktu instalēšanu.

Visi atlasītie Zabbix aģents 2 ielādējamie spraudņi tiks instalēti jūsu norādītajā vietā (noklusējums: `C:\Program Files\Zabbix Agent 2`), un to konfigurācijas faili tiks instalēti apakšmapē `zabbix_agent2.d`.

5. Noklikšķiniet uz pogas *Finish*, lai izietu no iestatīšanas vedņa.

Instalēšana no komandrindas

Zabbix aģents 2 ielādējamos spraudņus var instalēt no komandrindas, palaižot MSI instalētāju ar msiexec. Piemēram:

```
msiexec.exe /l*v "C:\package.log" /i "C:\zabbix_agent2_plugins-8.0.0-windows-amd64.msi" /qn+
```

Zabbix aģents 2 ielādējamo spraudņu MSI instalētāja pakotnes atbalsta šādus parametrus.

Parametrs	Apraksts
ADDDEFAULT	Ar komatiem atdalīts komponentu saraksts, kas jāinstalē to noklusējuma konfigurācijā. Papildinformāciju skatiet ADDDEFAULT property. Iespējamās vērtības: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Piemērs: `ADDDEFAULT=MongodbPlugin,PostgresqlPlugin`

Parametrs	Apraksts
ADDLOCAL	Ar komatiem atdalīts komponentu saraksts, kas jāinstalē lokāli. Papildinformāciju skatiet ADDLOCAL property. Iespējamās vērtības: `ALL`, `CephPlugin`, `EmberplusPlugin`, `MongodbPlugin`, `MssqlPlugin`, `NvidiagpuPlugin`, `PostgresqlPlugin` Piemērs: `ADDLOCAL=MongodbPlugin,MssqlPlugin`
INSTALLFOLDER	Pilns ceļš uz mapi, kurā tiks instalēti Zabbix komponenti, bet to konfigurācijas faili tiks instalēti apakšmapē `zabbix_agent2.d`. Piemērs: `INSTALLFOLDER="C:\Program Files\Zabbix Agent 2"`

Zabbix aģenta 2 būvēšana operētājsistēmā Windows

Pārskats

Šajā lapā ir parādīts, kā izveidot Zabbix aģents 2 no pirmkoda operētājsistēmā Windows 10 (64 bitu vai 32 bitu).

Gan 32 bitu, gan 64 bitu versijas var izveidot 64 bitu platformā, taču 32 bitu platformā var izveidot tikai 32 bitu versiju.

Zabbix aģents 2 izveidei ir nepieciešams:

- MinGW būvēšanas rīki
- Go programmēšanas valoda
- OpenSSL (Zabbix šifrēšanas funkcijām)
- PCRE2 (Perl Compatible Regular Expressions; regulāro izteiksmju šablonu saskaņošanas funkcijām Zabbix)

Zabbix aģents 2 var izveidot, izmantojot vienu no šīm metodēm:

- Izmantojot vcpkg — automatizēta pieeja, kas vienkāršo atkarību pārvaldību, izmantojot C++ pakotņu pārvaldnieku.
- Manuāla izveide — manuāla pieeja, kurā pirms aģents kompilēšanas ir jāinstalē visas atkarības.

Attention:
Pirms būvēšanas procesa sākšanas, lūdzu, ņemiet vērā:

- Lai izpildītu komandas, izmantojiet Command Prompt, ko palaiž lietotājs ar pietiekamām atļaujām rakstīt aizsargātās mapēs. Tomēr, instalējot OpenSSL un PCRE2, izmantojiet MSYS2 termināli.
- Ieteicams izveidot darba direktoriju `C:\Zabbix` visiem pirmkoda failiem un būvēšanas mapēm. Tomēr kompilētie komponenti jāinstalē mapē `C:\Zabbix\x64` (vai `C:\Zabbix\x86` 32 bitu būvēm).

Zabbix aģenta 2 būvēšana ar vcpkg

Šajā sadaļā ir sniegti norādījumi Zabbix aģenta būvēšanai ar vcpkg — pakotņu pārvaldnieku, kas vienkāršo atkarību pārvaldību un integrāciju ar C++ projektiem.

1. Lejupielādējiet un instalējiet Build Tools for Visual Studio 2022. Instalēšanas laikā noteikti atlasiet *Desktop development with C++* darba slodzi, kas ietver vcpkg pakotņu pārvaldnieku.

2. Lejupielādējiet un instalējiet Go (pieejams kā MSI instalētājs). Instalēšanas laikā noteikti norādiet `C:\Zabbix\Go` kā instalēšanas mapi.

3. Lejupielādējiet MinGW distribūciju, kas izmanto Microsoft Visual C izpildlaika bibliotēku; piemēram:

- 64 bitu būvējumiem: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- 32 bitu būvējumiem: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Pēc tam izpakojiet to uz `C:\Zabbix\mingw64` (vai `C:\Zabbix\mingw32` 32 bitu būvējumiem).

4. Inicializējiet vcpkg un instalējiet atkarības, kas nepieciešamas Zabbix aģenta 2 būvēšanai (ņemiet vērā, ka tas var aizņemt kādu laiku):

```
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

#### 64 bitu būvējumiem:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
```

```
vcpkg install --triplet x64-mingw-static --x-install-root=x64

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

5. Lejupielādējiet Zabbix pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\zabbix-8.0.0`.

6. Pārejiet uz Zabbix būvēšanas direktoriju (`C:\Zabbix\zabbix-8.0.0\build\mingw`) un izveidojiet šādu `build.bat` skriptu:

   - 64 bitu būvējumiem:

```
:: Pievienojiet MinGW un Go sistēmas `PATH` mainīgajam pašreizējai sesijai:
set PATH=C:\Zabbix\mingw64\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Iestatiet vcpkg instalācijas ceļu:
set vcpkg="C:\Zabbix\x64\x64-mingw-static"

:: Iestatiet saistītāja karogus Crypt32 bibliotēkai:
SET CGO_LDFLAGS="-lCrypt32"

:: Palaidiet būvēšanas procesu:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=AMD64 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

   - 32 bitu būvējumiem:

```
:: Pievienojiet MinGW un Go sistēmas `PATH` mainīgajam pašreizējai sesijai:
set PATH=C:\Zabbix\mingw32\bin;%PATH%
set PATH=C:\Zabbix\Go\bin;%PATH%

:: Iestatiet vcpkg instalācijas ceļu:
set vcpkg="C:\Zabbix\x86\x86-mingw-static"

:: Iestatiet saistītāja karogus Crypt32 bibliotēkai:
SET CGO_LDFLAGS="-lCrypt32"

:: Palaidiet būvēšanas procesu:
mingw32-make GOFLAGS="-buildvcs=false" ARCH=x86 ^
    PCRE2="%vcpkg%" ^
    OPENSSL="%vcpkg%" ^
    all
```

7. Kompilējiet Zabbix aģentu 2, izpildot skriptu:

```
build.bat
```

Pēc kompilēšanas Zabbix aģenta 2 binārais fails atradīsies `C:\Zabbix\zabbix-8.0.0\bin\win64` (64 bitu būvējumiem) vai `C:\Zabbix\zabbix-8.0.0\bin\win32` (32 bitu būvējumiem). Zabbix aģenta 2 konfigurācijas faili atrodas `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Lai palaistu aģentu, kopējiet `zabbix_agent2.exe` un tā konfigurācijas failus uz atsevišķu mapi (piemēram, `C:\Zabbix\agent2`) un pēc tam palaidiet aģentu:

```
mkdir C:\Zabbix\agent2

#### 64 bitu būvējumiem:
copy C:\Zabbix\zabbix-8.0.0\bin\win64\zabbix_agent2.exe C:\Zabbix\agent2\

#### 32 bitu būvējumiem:
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
```

Ja nepieciešams, turpiniet ar Zabbix aģenta 2 ielādējamo spraudņu kompilēšanu.

Zabbix aģenta 2 ielādējamo spraudņu kompilēšana

1. Lejupielādējiet Zabbix spraudņa pirmkodu, kas atbilst jūsu Zabbix aģenta 2 versijai (piemēram, `zabbix-agent2-plugin-postgresql-8` un izpakojiet to uz `C:\Zabbix`.

> **Attention:**
> Pirms spraudņa instalēšanas, lūdzu, pārbaudiet tā README failu. Tajā var būt ietvertas īpašas prasības un instalēšanas norādījumi.

2. Pārejiet uz izpakotā spraudņa direktoriju un kompilējiet spraudni:

```
cd C:\Zabbix\zabbix-agent2-plugin-ember-plus-8.0.0

#### 64 bitu būvējumiem:
mingw32-make ARCH=AMD64

#### 32 bitu būvējumiem:
mingw32-make ARCH=x86
```

Pēc kompilēšanas Zabbix aģenta 2 spraudņa binārais fails un tā konfigurācijas fails atradīsies tajā pašā spraudņa direktorijā.

Spraudņa izpildāmo failu var novietot jebkurā vietā, ja vien to var ielādēt Zabbix aģents 2. Norādiet ceļu uz spraudņa bināro failu spraudņa konfigurācijas failā, piemēram, PostgreSQL spraudņa failā postgresql.conf:

```
Plugins.PostgreSQL.System.Path=/path/to/executable/zabbix-agent2-plugin-postgresql
```

Ceļš uz spraudņa konfigurācijas failu jānorāda Zabbix aģenta 2 konfigurācijas faila parametrā Include:

```
Include=/path/to/plugin/configuration/file/postgresql.conf
```

Lai iegūtu plašāku informāciju par spraudņu konfigurēšanu, turpiniet ar iestatīšanu.

Zabbix aģents 2 manuāla būvēšana

> **Attention:**
> Šī Zabbix aģents 2 būvēšanas metode ir piemērota lietotājiem, kuriem nepieciešama pilnīga kontrole pār būvēšanas vidi vai kuri strādā ierobežotā vidē, kur vcpkg izmantošana nav iespējama.

Šajā sadaļā ir sniegti norādījumi Zabbix aģents 2 manuālai būvēšanai, tostarp nepieciešamo būvēšanas rīku un atkarību instalēšanai, kā arī pēc tam aģenta kompilēšanai.

Būvēšanas rīku iestatīšana

1. Lejupielādējiet un instalējiet MSYS2 (pieejams kā MSI instalētājs). Instalēšanas laikā noteikti norādiet `C:\Zabbix\msys64` kā instalēšanas mapi.

2. Lejupielādējiet un instalējiet Go (pieejams kā MSI instalētājs; skatiet pašlaik atbalstītās Go versions). Instalēšanas laikā noteikti norādiet `C:\Zabbix\Go` kā instalēšanas mapi.

3. Lejupielādējiet MinGW distribūciju, kas izmanto Microsoft Visual C izpildlaika bibliotēku; piemēram:

- 64 bitu būvēm: `x86_64-15.1.0-release-win32-seh-msvcrt-rt_v12-rev0.7z`
- 32 bitu būvēm: `i686-15.1.0-release-win32-dwarf-msvcrt-rt_v12-rev0.7z`

Pēc tam izpakojiet to uz `C:\Zabbix\mingw64` (vai `C:\Zabbix\mingw32` 32 bitu būvēm).

OpenSSL instalēšana

> **Note:**
> Lai kompilētu Zabbix aģents bez TLS atbalsta, pārejiet uz sadaļu PCRE2 instalēšana.

1. Atveriet MSYS2 MSYS termināli ar administratora tiesībām un izpildiet šādas komandas:

```
pacman -S perl-Locale-Maketext-Simple
pacman -S nasm
```

```
pacman -S make
pacman -S cmake
```

2. Lejupielādējiet OpenSSL pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\openssl-3.5.0`.

3. Pārejiet uz izpakoto OpenSSL direktoriju un izveidojiet šādu `build.sh` skriptu:

- 64 bitu būvējumiem:

```
####!/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
```

- 32 bitu būvējumiem:

```
####!/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:
Pārliecinieties, ka lietotājiem bez administratora tiesībām ir atņemtas rakstīšanas tiesības uz direktoriju `C:\Zabbix\x64\OpenSSL-Win64-350-static`. Pretējā gadījumā aģents ielādēs SSL iestatījumus no ceļa, ko var mainīt lietotāji bez paaugstinātām tiesībām, radot iespējamu drošības ievainojamību.

- Opcija `no-shared` padara OpenSSL statiskās bibliotēkas libcrypto.lib un libssl.lib pašpietiekamas, tādēļ Zabbix binārie faili ietver OpenSSL bez nepieciešamības pēc ārējām DLL. Tas nozīmē, ka Zabbix bināros failus var kopēt uz citām Windows iekārtām bez OpenSSL bibliotēkām; tomēr, kad tiek izlaista jauna OpenSSL kļūdu labojumu versija, Zabbix aģents būs jāpārkompilē.
- Bez opcijas `no-shared` Zabbix izpildlaikā paļaujas uz OpenSSL DLL. Tas nozīmē, ka OpenSSL atjauninājumiem var nebūt nepieciešama Zabbix aģents pārkompilēšana; tomēr, kopējot to uz citām iekārtām, ir jākopē arī OpenSSL DLL.

Plašāku informāciju par citām OpenSSL konfigurācijas opcijām skatiet OpenSSL dokumentācijā.

4. Konfigurējiet un instalējiet OpenSSL, izpildot skriptu (ņemiet vērā, ka tas var aizņemt kādu laiku):

```
cd /c/Zabbix/openssl-3.5.0
./build.sh
```

PCRE2 instalēšana

1. Lejupielādējiet PCRE2 pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\pcre2-10.45`.

2. Atveriet MSYS2 MSYS termināli ar administratora tiesībām. Pēc tam izveidojiet `build` direktoriju izpakotajā PCRE2 direktorijā un pārejiet uz to:

```
mkdir /c/Zabbix/pcre2-10.45/build
cd /c/Zabbix/pcre2-10.45/build
```

3. Konfigurējiet PCRE2:

```
#### 64 bitu būvējumiem:
export PATH="/c/Zabbix/mingw64/bin:$PATH"
cmake -DCMAKE_C_COMPILER=gcc -DCMAKE_C_FLAGS="-O2 -g" -DCMAKE_INSTALL_PREFIX="/c/Zabbix/x64/PCRE2" ..

#### 32 bitu būvējumiem:
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:
Ja rodas jebkādas kļūdas, pirms atkārtota CMake būvēšanas procesa mēģinājuma ieteicams dzēst CMake kešatmiņu. Kešatmiņa (`CMakeCachecache.txt`) var atrasties izpakotās PCRE2 direktorijas build direktorijā.

4. Instalējiet PCRE2:

```
make install
```

Zabbix aģenta 2 kompilēšana

1. Lejupielādējiet Zabbix pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\zabbix-8.0.0`.

Ja jums ir nepieciešams ģenerēt pirmkoda arhīvu no neapstrādātas pirmkoda repozitorija versijas (piemēram, lai lietotu pielāgotus ielāpus vai veidotu no jaunākā pirmkoda), izpildiet tālāk norādītās komandas **Linux** datorā ar instalētu Go (nepieciešams Zabbix aģenta 2 konfigurēšanai):

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

Tādējādi tiks izveidots pirmkoda arhīvs, kuru pēc tam var kopēt uz Windows datoru.

2. Atveriet komandrindu ar administratora tiesībām. Pēc tam pārejiet uz Zabbix būvēšanas direktoriju un kompilējiet Zabbix aģentu; pārliecinieties, ka ir pareizi norādīti direktoriji, kuros ir instalēti OpenSSL un PCRE2:

- 64 bitu būvēm:

```
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

#### With TLS support:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2" OPENSSL="C:\Zabbix\x64\OpenSSL-Win64-350-static"

#### Without TLS support:
mingw32-make ARCH=AMD64 PCRE2="C:\Zabbix\x64\PCRE2"
```

- 32 bitu būvēm:

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

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

#### Without TLS support:
mingw32-make ARCH=x86 PCRE2="C:\Zabbix\x86\PCRE2"
```

Pēc kompilēšanas Zabbix aģenta 2 binārais fails atradīsies `C:\Zabbix\zabbix-8.0.0\bin\win64` (vai `C:\Zabbix\zabbix-8.0.0\bin\` 32 bitu būvēm). Zabbix aģenta 2 konfigurācijas faili atrodas `C:\Zabbix\zabbix-8.0.0\src\go\conf`.

Lai palaistu aģentu, nokopējiet `zabbix_agent2.exe` bināro failu un tā konfigurācijas failus uz atsevišķu mapi (piemēram, `C:\Zabbix\agent2`) un pēc tam palaidiet aģentu:

```
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
```

Ja nepieciešams, turpiniet ar Zabbix aģenta 2 ielādējamo spraudņu kompilēšanu.

Building Zabbix agent on Windows

Pārskats

Šajā lapā ir parādīts, kā būvēt Zabbix aģents no pirmkoda operētājsistēmā Windows 10 (64 bitu).

Šīs instrukcijas attiecas uz Windows versijām, kas atbalsta Visual Studio 2022.

Lai būvētu Zabbix aģents, ir nepieciešams:

- C kompilators (iekļauts Build Tools for Visual Studio 2022)
- OpenSSL (Zabbix šifrēšanas funkcijām)
- PCRE2 (Perl Compatible Regular Expressions; regulāro izteiksmju šablonu saskaņošanas funkcijām Zabbix)

Jūs varat būvēt Zabbix aģents, izmantojot vienu no šīm metodēm:

- Izmantojot vcpkg — automatizēta pieeja, kas vienkāršo atkarību pārvaldību, izmantojot C++ pakotņu pārvaldnieku.
- Manuāla būvēšana — manuāla pieeja, kurā pirms aģents kompilēšanas ir jāinstalē visas atkarības.

Atkarībā no jūsu uzraudzības vajadzībām var būt nepieciešamas papildu bibliotēkas. Plašāku informāciju skatiet sadaļā Prasības.

Attention:
Pirms būvēšanas procesa sākšanas, lūdzu, ņemiet vērā:

- Lai izpildītu komandas, izmantojiet x64 Native Tools Command Prompt (iekļauts Build Tools for Visual Studio 2022), ko palaiž lietotājs ar pietiekamām atļaujām rakstīšanai aizsargātās mapēs.
- Ieteicams izveidot darba direktoriju `C:\Zabbix` visiem pirmkoda failiem un būvēšanas mapēm. Tomēr kompilētie komponenti jāinstalē mapē `C:\Program Files\Zabbix\x64`.

Zabbix aģenta būvēšana ar vcpkg

Šajā sadaļā ir sniegtas instrukcijas Zabbix aģenta būvēšanai ar vcpkg — pakotņu pārvaldnieku, kas vienkāršo atkarību pārvaldību un integrāciju ar C++ projektiem.

1. Lejupielādējiet un instalējiet Build Tools for Visual Studio 2022. Instalēšanas laikā noteikti atlasiet *Desktop development with C++* darba slodzi, kas ietver rīkus, kuri nepieciešami aģenta būvēšanai ar vcpkg:

- C kompilators (Microsoft Visual C++)
- NMake komandrindas rīks
- vcpkg pakotņu pārvaldnieks
- x64 Native Tools Command Prompt

2. Inicializējiet vcpkg un instalējiet atkarības, kas nepieciešamas Zabbix aģenta būvēšanai (ņemiet vērā, ka tas var aizņemt kādu laiku):

```
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. Lejupielādējiet Zabbix pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\zabbix-8.0.0`.

4. Pārejiet uz Zabbix būvēšanas direktoriju (`C:\Zabbix\zabbix-8.0.0\build\win32\project`) un izveidojiet šādu `build.bat` skriptu; noteikti pareizi norādiet direktorijus, kuros ir instalēti OpenSSL un 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. Kompilējiet Zabbix aģentu, izpildot skriptu:

```
build.bat
```

Pēc kompilēšanas Zabbix komponentu binārie faili atradīsies `C:\Zabbix\zabbix-8.0.0\bin\win64`. Zabbix aģenta konfigurācijas fails atrodas `C:\Zabbix\zabbix-8.0.0\conf`.

Lai palaistu aģentu, nokopējiet `zabbix_agent.exe` un tā konfigurācijas failu uz atsevišķu mapi (piemēram, `C:\Zabbix\agent`) un pēc tam palaidiet aģentu:

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

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

Zabbix aģenta manuāla būvēšana

Attention:
Šī Zabbix aģenta būvēšanas metode ir piemērota lietotājiem, kuriem nepieciešama pilnīga kontrole pār būvēšanas vidi vai kuri strādā ierobežotā vidē, kur vcpkg izmantošana nav iespējama.

Šajā sadaļā ir sniegtas instrukcijas Zabbix aģenta manuālai būvēšanai, kas ietver nepieciešamo būvēšanas rīku un atkarību (Perl, OpenSSL, PCRE2) instalēšanu un pēc tam aģenta kompilēšanu.

Būvēšanas rīku instalēšana

1. Lejupielādējiet un instalējiet Build Tools for Visual Studio 2022. Instalēšanas laikā noteikti atlasiet *Desktop development with C++* darba slodzi, kas ietver rīkus, kuri nepieciešami, lai manuāli būvētu aģents:

- C kompilators (Microsoft Visual C++)
- NMake komandrindas rīks
- x64 Native Tools Command Prompt

OpenSSL instalēšana

Note:
Lai kompilētu Zabbix aģents bez TLS atbalsta, pārejiet uz sadaļu PCRE2 instalēšana.

1. Lejupielādējiet un instalējiet Strawberry Perl (pieejams kā MSI instalētājs). Instalēšanas laikā noteikti norādiet `C:\Zabbix\Strawberry` kā instalēšanas mapi.

2. Instalējiet `Text::Template` Perl moduli:

```
cpanm Text::Template
```

3. Pārbaudiet, vai Netwide Assembler (NASM; nepieciešams OpenSSL kompilēšanai) tika kompilēts Strawberry Perl instalēšanas laikā:

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

Ja NASM nav kompilēts, instalējiet to manuāli. Papildinformāciju skatiet NASM dokumentācijā.

4. Lejupielādējiet OpenSSL pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\openssl-3.5.0`.

5. Pārejiet uz izpakoto direktoriju un konfigurējiet OpenSSL, piemēram:

```
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:
Ja, kompilējot Zabbix aģents operētājsistēmā Windows, OpenSSL izvēlaties pielāgotu direktoriju (piemēram, `C:\zabbix` vai `C:\openssl-64bit`), noteikti atņemiet rakstīšanas piekļuvi šim direktorijam lietotājiem bez administratora tiesībām. Pretējā gadījumā aģents ielādēs SSL iestatījumus no ceļa, ko var mainīt lietotāji bez paaugstinātām tiesībām, radot iespējamu drošības ievainojamību.

- Opcija `no-shared` padara libcrypto.lib un libssl.lib OpenSSL statiskās bibliotēkas pašpietiekamas, tāpēc Zabbix binārie faili ietver OpenSSL bez nepieciešamības pēc ārējiem DLL. Tas nozīmē, ka Zabbix bināros failus var kopēt uz citām Windows iekārtām bez OpenSSL bibliotēkām; tomēr, kad tiek izlaista jauna OpenSSL kļūdu labojumu versija, Zabbix aģents būs jāpārkompilē.
- Bez opcijas `no-shared` Zabbix izpildlaikā paļaujas uz OpenSSL DLL. Tas nozīmē, ka OpenSSL atjauninājumiem var nebūt nepieciešama Zabbix aģents pārkompilēšana; tomēr, kopējot to uz citām iekārtām, būs jākopē arī OpenSSL DLL.

Papildinformāciju par citām OpenSSL konfigurācijas opcijām skatiet OpenSSL dokumentācijā.

6. Kompilējiet OpenSSL un palaidiet testus (ņemiet vērā, ka tas var aizņemt kādu laiku):

> **Attention:**
> Palaidiet testus bez administratora privilēģijām; pretējā gadījumā tas var izraisīt neparedzētus rezultātus vai drošības riskus. Ja daži testi neizdodas, problēmu novēršanai skatiet OpenSSL dokumentāciju.

```
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. Instalējiet OpenSSL:

```
nmake install
```

Lai instalētu tikai programmatūras komponentus (bibliotēkas, galveņu failus, bet ne dokumentāciju), varat izmantot `nmake install_sw`.

PCRE2 instalēšana

1. Lejupielādējiet un instalējiet CMake (pieejams kā MSI instalētājs). Instalēšanas laikā noteikti norādiet `C:\Zabbix\CMake` kā instalēšanas mapi un atlasiet opciju *Add CMake to the PATH environment variable*.

2. Lejupielādējiet PCRE2 pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\pcre2-10.45`.

3. Izveidojiet `build` direktoriju izpakotajā PCRE2 direktorijā un pārejiet uz to:

```
mkdir C:\Zabbix\pcre2-10.45\build
cd C:\Zabbix\pcre2-10.45\build
```

4. Konfigurējiet PCRE2:

```
cmake -G "NMake Makefiles" -DPCRE_SUPPORT_UNICODE_PROPERTIES=ON -DCMAKE_BUILD_TYPE=Release -DCMAKE_INSTALL
```

> **Note:**
> Ja rodas kādas kļūdas, pirms mēģināt atkārtot CMake būvēšanas procesu, ieteicams dzēst CMake kešatmiņu. Kešatmiņa (`CMakeCachecache.txt`) var atrasties izpakotā PCRE2 direktorija build direktorijā.

5. Būvējiet PCRE2, izmantojot NMake:

```
nmake
```

6. Instalējiet PCRE2:

```
cmake --install .
```

Zabbix aģenta kompilēšana

1. Lejupielādējiet Zabbix pirmkoda arhīvu un izpakojiet to uz `C:\Zabbix\zabbix-8.0.0`.

Ja jums ir nepieciešams ģenerēt pirmkoda arhīvu no neapstrādātas pirmkoda repozitorija versijas (piemēram, lai lietotu pielāgotus ielāpus vai veidotu no jaunākā pirmkoda), izpildiet šādas komandas **Linux** datorā:

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

Tādējādi tiks izveidots pirmkoda arhīvs, kuru pēc tam var nokopēt uz Windows datoru.

2. Pārejiet uz Zabbix būvēšanas direktoriju un kompilējiet Zabbix aģentu (vai citus komponentus); pārliecinieties, ka pareizi norādāt direktorijus, kuros ir instalēti OpenSSL un PCRE2:

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

#### Ar TLS atbalstu:
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
```

```
#### Bez TLS atbalsta:
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
```

Pēc kompilēšanas Zabbix komponentu binārie faili atradīsies `C:\Zabbix\zabbix-8.0.0\bin\win64`. Zabbix aģenta konfigurācijas fails atrodas `C:\Zabbix\zabbix-8.0.0\conf`.

Lai palaistu aģentu, nokopējiet `zabbix_agent.exe` un tā konfigurācijas failu uz atsevišķu mapi (piemēram, `C:\Zabbix\agent`) un pēc tam palaidiet aģentu:

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

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

macOS aģenta instalēšana no PKG

Pārskats

Zabbix aģents var tikt instalēts macOS, izmantojot PKG instalēšanas pakotnes, kas ir pieejamas lejupielādei.

Zabbix aģenta pakotnes ir pieejamas ar vai bez šifrēšanas.

Aģenta instalēšana

Aģentu var instalēt, izmantojot grafisko lietotāja saskarni vai komandrindu, piemēram:

```
sudo installer -pkg zabbix_agent-8.0.0-macos-arm64-openssl.pkg -target /
```

Pārliecinieties, ka komandā izmantojat pareizo Zabbix pakotnes versiju. Tai jāsakrīt ar lejupielādētās pakotnes nosaukumu.

Aģenta palaišana

Aģents tiks automātiski palaists pēc instalēšanas vai restartēšanas.

Ja nepieciešams, varat rediģēt konfigurācijas failu `/usr/local/etc/zabbix/zabbix_agentd.conf`.

Lai palaistu aģentu manuāli, varat izpildīt:

```
sudo launchctl start com.zabbix.zabbix_agentd
```

Lai apturētu aģentu manuāli:

```
sudo launchctl stop com.zabbix.zabbix_agentd
```

Atjaunināšanas laikā esošais konfigurācijas fails netiek pārrakstīts. Tā vietā tiek izveidots jauns fails `zabbix_agentd.conf.NEW`, ko var izmantot esošā konfigurācijas faila pārskatīšanai un atjaunināšanai, ja nepieciešams. Atcerieties pēc manuālām izmaiņām konfigurācijas failā restartēt aģentu.

Problēmu novēršana un aģenta noņemšana

Šajā sadaļā ir uzskaitītas dažas noderīgas komandas, kuras var izmantot Zabbix aģenta instalācijas problēmu novēršanai un noņemšanai.

Pārbaudiet, vai Zabbix aģents darbojas:

```
ps aux | grep zabbix_agentd
```

Pārbaudiet, vai Zabbix aģents ir instalēts no pakotnēm:

```
pkgutil --pkgs | grep zabbix
com.zabbix.pkg.ZabbixAgent
```

Apskatiet failus, kas tika instalēti no instalētāja pakotnes (ņemiet vērā, ka šajā skatā sākotnējā `/` netiek parādīta):

```
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
```

Apturiet Zabbix aģentu, ja tas tika palaists ar `launchctl`:

```
sudo launchctl unload /Library/LaunchDaemons/com.zabbix.zabbix_agentd.plist
```

Noņemiet failus (ieskaitot konfigurāciju un žurnālus), kas tika instalēti ar instalētāja pakotni:

```
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
```

Noņemiet informāciju par to, ka Zabbix aģents ir bijis instalēts:

```
sudo pkgutil --forget com.zabbix.pkg.ZabbixAgent
```

Building Zabbix agent on macOS

Pārskats

Šajā sadaļā ir parādīts, kā no pirmkoda izveidot Zabbix macOS aģents bināros failus ar vai bez TLS.

Priekšnosacījumi

Jums būs nepieciešami komandrindas izstrādātāja rīki (Xcode nav nepieciešams), Automake, pkg-config un PCRE (v8.x) vai PCRE2 (v10.x). Ja vēlaties būvēt aģents bināros failus ar TLS, jums būs nepieciešams arī OpenSSL vai GnuTLS.

Lai instalētu Automake un pkg-config, jums būs nepieciešams Homebrew pakotņu pārvaldnieks no https://brew.sh/. Lai to instalētu, atveriet termināli un izpildiet šo komandu:

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

Pēc tam instalējiet Automake un pkg-config:

```
brew install automake
brew install pkg-config
```

PCRE, OpenSSL un GnuTLS bibliotēku sagatavošana ir atkarīga no tā, kā tās tiks piesaistītas aģents.

Ja plānojat palaist aģents bināros failus macOS datorā, kurā šīs bibliotēkas jau ir pieejamas, varat izmantot Homebrew nodrošinātās iepriekš kompilētās bibliotēkas. Parasti tie ir macOS datori, kuros Homebrew tiek izmantots Zabbix aģents bināro failu būvēšanai vai citiem mērķiem.

Ja aģents binārie faili tiks izmantoti macOS datoros, kuros nav bibliotēku koplietojamās versijas, jums vajadzētu kompilēt statiskās bibliotēkas no pirmkoda un sasaistīt Zabbix aģents ar tām.

Aģenta bināro failu veidošana ar koplietojamām bibliotēkām

Instalējiet PCRE2 (ja nepieciešams, tālāk norādītajās komandās aizstājiet *pcre2* ar *pcre*):

```
brew install pcre2
```

Veidojot ar TLS, instalējiet OpenSSL un/vai GnuTLS:

```
brew install openssl
brew install gnutls
```

Lejupielādējiet Zabbix pirmkodu:

```
git clone https://git.zabbix.com/scm/zbx/zabbix.git
```

Veidojiet aģentu bez TLS:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6
make
make install
```

Veidojiet aģentu ar OpenSSL:

```
cd zabbix
./bootstrap.sh
./configure --sysconfdir=/usr/local/etc/zabbix --enable-agent --enable-ipv6 --with-openssl=/usr/local/opt/
make
make install
```

Veidojiet aģentu ar 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
```

Aģenta bināro failu veidošana ar statiskajām bibliotēkām bez TLS

Pieņemsim, ka PCRE statiskās bibliotēkas tiks instalētas mapē `$HOME/static-libs`. Mēs izmantosim PCRE2 10.39.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
```

Lejupielādējiet un izveidojiet PCRE ar Unicode īpašību atbalstu:

```
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
```

Lejupielādējiet Zabbix pirmkodu un izveidojiet aģentu:

```
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
```

Aģenta bināro failu veidošana ar statiskajām bibliotēkām un OpenSSL

Veidojot OpenSSL, pēc veiksmīgas veidošanas ir ieteicams palaist `make test`. Pat ja veidošana bija veiksmīga, testi dažkārt neizdodas. Ja tā notiek, problēmas ir jāizpēta un jānovērš, pirms turpināt.

Pieņemsim, ka PCRE un OpenSSL statiskās bibliotēkas tiks instalētas mapē `$HOME/static-libs`. Mēs izmantosim PCRE2 10.39 un OpenSSL 1.1.1a.

```
PCRE_PREFIX="$HOME/static-libs/pcre2-10.39"
OPENSSL_PREFIX="$HOME/static-libs/openssl-1.1.1a"
```

Izveidosim statiskās bibliotēkas mapē `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Lejupielādējiet un izveidojiet PCRE ar Unicode īpašību atbalstu:

```
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 ..
```

Lejupielādējiet un izveidojiet 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 ..
```

Lejupielādējiet Zabbix pirmkodu un izveidojiet aģentu:

```
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
```

Aģenta bināro failu veidošana ar statiskajām bibliotēkām, izmantojot GnuTLS

GnuTLS ir atkarīgs no Nettle kriptogrāfijas aizmugures un GMP aritmētikas bibliotēkas. Pilnas GMP bibliotēkas vietā šajā ceļvedī tiks izmantots mini-gmp, kas ir iekļauts Nettle.

Veidojot GnuTLS un Nettle, pēc veiksmīgas izveides ieteicams palaist `make check`. Pat ja izveide bija veiksmīga, testi dažkārt neizdodas. Ja tā notiek, problēmas ir jāizpēta un jānovērš, pirms turpināt.

Pieņemsim, ka PCRE, Nettle un GnuTLS statiskās bibliotēkas tiks instalētas mapē `$HOME/static-libs`. Mēs izmantosim PCRE2 10.39, Nettle 3.4.1 un 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"
```

Izveidosim statiskās bibliotēkas mapē `static-libs-source`:

```
mkdir static-libs-source
cd static-libs-source
```

Lejupielādējiet un izveidojiet 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 ..
```

Lejupielādējiet un izveidojiet 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 ..
```

Lejupielādējiet Zabbix pirmkodu un izveidojiet aģentu:

```
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 Jaunināšana

Pirms jaunināšanas ir ļoti ieteicams iepazīties ar visām attiecīgajām jaunināšanas piezīmēm.

Būs noderīgi arī zināt, kuri komponenti jums ir jājaunina.

Pārskats

Soli pa solim jaunināšanas norādījumi ir pieejami šādām platformām::

- Red Hat Enterprise Linux (izmantojot pakotnes)
- Debian/Ubuntu (izmantojot pakotnes)
- Jaunināšana, izmantojot konteinerus
- Jaunināšana, izmantojot avota kodu

Saistītie norādījumi:

- Serveriem augstas pieejamības (HA) klasterī skatiet HA klastera jaunināšana
- TimescaleDB datubāzei skatiet TimescaleDB shēmas jaunināšana

Komponenti, kas jājaunina

Ieteicams jaunināt Zabbix aģentus, taču tas nav obligāti.

Ir ļoti ieteicams jaunināt Zabbix starpniekserverus. Zabbix serveris pilnībā atbalsta starpniekserverus, kuriem ir tā pati galvenā versija kā serverim. Zabbix serveris atbalsta arī starpniekserverus, kas nav **vecāki** par iepriekšējo Zabbix servera LTS laidiena versiju, taču ar ierobežotu funkcionalitāti (datu vākšana, izpilde attālinātajām komandām, tūlītējām vienuma vērtību pārbaudēm). Konfigurācijas atjaunināšana arī ir atspējota, un novecojuši starpniekserveri darbosies tikai ar veco konfigurāciju.

Attention:
Starpniekserveri, kas ir vecāki par iepriekšējo Zabbix servera LTS laidiena versiju vai jaunāki par Zabbix servera galveno versiju, netiek atbalstīti. Zabbix serveris ignorēs datus no neatbalstītiem starpniekserveriem, un visa saziņa ar Zabbix serveri neizdosies ar brīdinājumu. Plašāku informāciju skatiet Versiju saderība.

Lai jaunināšanas laikā samazinātu dīkstāvi un datu zudumu, ieteicams apturēt, jaunināt un palaist Zabbix serveri un pēc tam apturēt, jaunināt un palaist Zabbix starpniekserverus vienu pēc otra. Servera dīkstāves laikā darbojošies starpniekserveri turpinās datu vākšanu. Kad serveris atkal darbosies, novecojuši starpniekserveri nosūtīs datus uz jaunāko serveri (tomēr starpniekservera konfigurācija netiks atjaunināta) un paliks daļēji funkcionāli. Jebkuri paziņojumi par problēmām Zabbix servera dīkstāves laikā tiks ģenerēti tikai pēc tam, kad jaunināatais serveris būs palaists.

Ja Zabbix starpniekserveris tiek palaists pirmo reizi un trūkst SQLite datubāzes faila, starpniekserveris to izveido automātiski.

Ņemiet vērā, ka, ja Zabbix starpniekserveris izmanto SQLite3 un palaišanas laikā konstatē, ka esošā datubāzes faila versija ir vecāka par nepieciešamo, tas **automātiski izdzēsīs datubāzes failu** un izveidos jaunu. Tādēļ SQLite datubāzes failā saglabātie vēstures dati tiks zaudēti. Ja Zabbix starpniekservera versija ir vecāka par datubāzes faila versiju, Zabbix reģistrēs kļūdu žurnālā un beigs darbu.

Atkarībā no datubāzes lieluma datubāzes jaunināšana uz versiju 8.0 var aizņemt ilgu laiku.

Jaunināšanas piezīmes

Tieša jaunināšana uz Zabbix 8.0.x tiek atbalstīta no **2.0**.x versijām. Lai jaunināatu no agrākām versijām, skatiet Zabbix dokumentāciju par 2.0 un agrākām versijām.

Note:
Lūdzu, ņemiet vērā, ka pēc jaunināšanas var tikt ietekmētas dažas trešo pušu programmatūras integrācijas Zabbix vidē, ja ārējā programmatūra nav saderīga ar jaunināto Zabbix versiju.

Ir pieejamas šādas jaunināšanas piezīmes:

Jaunināšana no	Lasīt pilnās jauninājumu piezīmes	Svarīgākās izmaiņas starp versijām
7.4.x	Priekš: Zabbix 8.0	Paaugstinātas minimāli nepieciešamās datubāzes versijas. Minimāli nepieciešamā PHP versija paaugstināta no 8.0.0 uz 8.2.0. Rakstzīme % ir pievienota `UnsafeUserParameters` sarakstam Zabbix aģents un Zabbix aģents 2.
7.2.x	Priekš: Zabbix 7.4 Zabbix 8.0	PCRE bibliotēkas atbalsts ir pārtraukts.
7.0.x	Priekš: Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Oracle DB atbalsts ir pārtraukts.
6.4.x	Priekš: Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimāli nepieciešamā PHP versija paaugstināta no 7.4.0 uz 8.0.0. Asinhronie aptaujātāji aģenta, HTTP aģenta, SNMP walk[OID] un SNMP get[OID] pārbaudēm. Atsevišķa datubāzes tabula starpniekserveriem. Mainīta noklusējuma atrašanās vieta Windows aģenta konfigurācijas failam. Oracle DB ir novecojis. Vecais skaitliskais (peldošā komata) vērtības tips ir noņemts.
6.2.x	Priekš: Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimāli nepieciešamā MySQL versija paaugstināta no 8.0.0 uz 8.0.30. Zabbix serverim/starpniekserverim ir nepieciešama bibliotēka 'libevent_pthreads'. Pēc pirmās palaišanas pēc jauninājumiem Zabbix starpniekserveris ar SQLite3 automātiski dzēš veco datubāzes versiju (ar visu vēsturi) un izveido jaunu.
6.0.x LTS	Priekš: Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Minimāli nepieciešamā PHP versija paaugstināta no 7.2.5 uz 7.4.0. Pakalpojumu uzraudzība ir būtiski pārstrādāta. Deterministiski trigeri ir jāizveido jauninājumu laikā. Ja MySQL/MariaDB ir iespējota binārā žurnalēšana, tam ir nepieciešamas superlietotāja privilēģijas vai mainīgā/konfigurācijas parametra *log_bin_trust_function_creators = 1* iestatīšana. Skatiet Datubāzes izveides skripti, lai uzzinātu, kā iestatīt šo mainīgo.
5.4.x	Priekš: Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Paaugstinātas minimāli nepieciešamās datubāzes versijas. serveris/starpniekserveris netiks palaists, ja datubāze ir novecojusi. Datubāzes struktūras izmaiņu dēļ tiek zaudēti audita žurnāla ieraksti.
5.2.x	Priekš: Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	Paaugstinātas minimāli nepieciešamās datubāzes versijas. Apkopotie vienumi ir noņemti kā atsevišķs tips.
5.0.x LTS	Priekš: 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	Minimāli nepieciešamā PHP versija paaugstināta no 7.2.0 uz 7.2.5. Paroļu jaukšanas algoritms ir mainīts no MD5 uz bcrypt.

Jaunināšana no	Lasīt pilnās jauninājanas piezīmes	Svarīgākās izmaiņas starp versijām
4.4.x	Priekš:	IBM DB2 atbalsts ir pārtraukts.
	Zabbix 5.0	Minimāli nepieciešamā PHP versija paaugstināta no 5.4.0 uz 7.2.0.
	Zabbix 5.2	Paaugstinātas minimāli nepieciešamās datubāzes versijas.
	Zabbix 5.4	Mainīts Zabbix PHP failu direktorijs.
	Zabbix 6.0	
	Zabbix 6.2	
	Zabbix 6.4	
	Zabbix 7.0	
	Zabbix 7.2	
	Zabbix 7.4	
	Zabbix 8.0	
4.2.x	Priekš:	Jabber un Ez Texting mediju tipi ir noņemti.
	Zabbix 4.4	
	Zabbix 5.0	
	Zabbix 5.2	
	Zabbix 5.4	
	Zabbix 6.0	
	Zabbix 6.2	
	Zabbix 6.4	
	Zabbix 7.0	
	Zabbix 7.2	
	Zabbix 7.4	
	Zabbix 8.0	
4.0.x LTS	Priekš:	Vecāki starpniekserveri vairs nevar nosūtīt datus uz jaunināt serveri.
	Zabbix 4.2	Jaunāki aģenti vairs nevarēs darboties ar vecāku Zabbix serveri.
	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	
3.4.x	Priekš:	Bibliotēkas ’libpthread’ un ’zlib’ tagad ir obligātas.
	Zabbix 4.0	Vienkāršā teksta protokola atbalsts ir pārtraukts, un galvene ir obligāta.
	Zabbix 4.2	Zabbix aģenti pirms 1.4 versijas vairs netiek atbalstīti.
	Zabbix 4.4	Parametrs Server pasīvā starpniekservera konfigurācijā tagad ir obligāts.
	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	

Jauninašana no	Lasīt pilnās jauninašanas piezīmes	Svarīgākās izmaiņas starp versijām
3.2.x	Priekš: Zabbix 3.4 Zabbix 4.0 Zabbix 4.2 Zabbix 4.4 Zabbix 5.0 Zabbix 5.2 Zabbix 5.4 Zabbix 6.0 Zabbix 6.2 Zabbix 6.4 Zabbix 7.0 Zabbix 7.2 Zabbix 7.4 Zabbix 8.0	SQLite atbalsts kā aizmugures datubāzei Zabbix serverim/lietotāja saskarnei ir pārtraukts. POSIX paplašināto regulāro izteiksmju vietā tiek atbalstītas Perl Compatible Regular Expressions (PCRE). Bibliotēkas 'libpcre' un 'libevent' ir obligātas Zabbix serverim. Lietotāja parametriem, attālinātajām komandām un system.run[] vienumiem bez karoga 'nowait', kā arī Zabbix servera izpildītajiem skriptiem ir pievienotas izejas koda pārbaudes. Zabbix Java gateway ir jājaunina, lai atbalstītu jauno funkcionalitāti.
3.0.x LTS	Priekš: 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	Datubāzes jauninašana var būt lēna atkarībā no vēstures tabulas lieluma.
2.4.x	Priekš: 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	Minimāli nepieciešamā PHP versija paaugstināta no 5.3.0 uz 5.4.0. Jānorāda aģenta parametrs LogFile.

Jaunināšana no	Lasīt pilnās jauninašanas piezīmes	Svarīgākās izmaiņas starp versijām
2.2.x LTS	Priekš: 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	Uz mezgliem balstīta sadalītā uzraudzība ir noņemta.
2.0.x	Priekš: 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	Minimāli nepieciešamā PHP versija paaugstināta no 5.1.6 uz 5.3.0. Zabbix servera pareizai darbībai ir nepieciešama MySQL datubāze ar reģistrjutīgu salīdzināšanu; lai Zabbix serveris pareizi darbotos ar MySQL datubāzi, ir nepieciešama rakstzīmju kopa utf8 un utf8_bin kolācija. Skatiet database creation scripts. PHP paplašinājums ’mysqli’ ir nepieciešams ’mysql’ vietā.

Upgrade from packages

Pārskats

Šajā sadaļā ir sniegti soļi, kas nepieciešami veiksmīgai jauninašanai, izmantojot oficiālās RPM un DEB pakotnes, ko Zabbix nodrošina šādām sistēmām:

- Red Hat Enterprise Linux
- Debian/Ubuntu

Zabbix pakotnes no OS repozitorijiem

Daži OS distributīvi (jo īpaši uz Debian balstīti distributīvi) nodrošina savas Zabbix pakotnes. Šīs pakotnes **Zabbix neatbalsta**, un tās var būt novecojušas vai tajās var trūkt jaunāko funkciju un kļūdu labojumu. Ieteicams izmantot tikai oficiālās pakotnes no Zabbix Official Repository.

Ja veicat jauninašanu no pakotnēm, ko nodrošina OS distributīvi (vai tās kādā brīdī bija instalētas), izpildiet šo procedūru, lai pārietu uz oficiālajām Zabbix pakotnēm:

1. Atinstalējiet vecās pakotnes.
2. Pārbaudiet, vai nav palikuši faili no iepriekšējās instalācijas, un noņemiet tos.
3. Instalējiet oficiālās Zabbix pakotnes, ievērojot Zabbix sniegtās instalēšanas instrukcijas.

Neveiciet tiešu jauninašanu, jo tas var izraisīt bojātu instalāciju.

Debian/Ubuntu

Pārskats

Šajā sadaļā ir sniegti norādījumi, kā veikt jauninášanu no Zabbix **7.4.x** uz jaunāko Zabbix **8.0.x** versiju, izmantojot oficiālās Zabbix pakotnes Debian/Ubuntu sistēmām.

> **Warning:**
> Pirms jauninášanas, lūdzu, pārskatiet attiecīgās jauninášanas piezīmes un pārliecinieties, ka jūsu sistēma atbilst Zabbix 8.0 prasībām.

> **Note:**
> Apsveriet iespēju jauninášanas laikā izmantot divas paralēlas SSH sesijas: vienu jauninášanas darbību izpildei un otru servera/starpniekservera žurnālu uzraudzībai. Piemēram, otrajā sesijā palaidiet `tail -f zabbix_server.log` vai `tail -f zabbix_proxy.log`, lai reāllaikā skatītu jaunākos žurnāla ierakstus un iespējamās kļūdas. Tas var būt kritiski svarīgi produkcijas instancēm.

Norādījumus par jauninášanu starp Zabbix 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3) skatiet sadaļā Jauninášana starp mazajām versijām.

Jauninášanas procedūra

1 Apturiet Zabbix procesus

Apturiet Zabbix serveri, lai pārliecinātos, ka datubāzē netiek ievietoti jauni dati:

```
systemctl stop zabbix-server
```

Ja jaunināt Zabbix starpniekserveri, aģentu vai aģentu 2, apturiet arī šos komponentus:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Izveidojiet Zabbix datubāzes dublējumu

Izveidojiet esošās Zabbix datubāzes dublējumu, lai pasargātu sevi no jauninášanas kļūmēm (piemēram, diska vietas problēmām, elektroapgādes pārtraukuma vai neparedzētām problēmām).

3 Izveidojiet Zabbix konfigurācijas failu, PHP failu un Zabbix bināro failu dublējumkopiju

Izveidojiet esošo Zabbix konfigurācijas failu, PHP failu un Zabbix bināro failu dublējumkopiju.

Konfigurācijas failiem izpildiet:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/apache2/conf-enabled/zabbix.conf /opt/zabbix-backup/
```

PHP failiem un Zabbix binārajiem failiem izpildiet:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Atjauniniet repozitorija konfigurācijas pakotni

Pirms turpināt jauninášanu, atinstalējiet pašreizējo Zabbix repozitorija pakotni:

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

Jums var būt nepieciešams arī manuāli noņemt visas vecās Zabbix pakotnes no darba direktorija (piemēram, `rm zabbix-release_latest+d` pirms jaunās pakotnes lejupielādes, lai novērstu to, ka pakotņu pārvaldnieks jauninášanas procesa laikā atkārtoti izmanto novecojušu versiju.

Pēc tam instalējiet jaunāko repozitorija konfigurācijas pakotni, lai nodrošinātu saderību ar jaunākajām pakotnēm un iekļautu nesenos drošības ielāpus vai kļūdu labojumus.

Uz **Debian 12** izpildiet:

```
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:
Vecākām Debian versijām aizstājiet iepriekš norādīto saiti ar pareizo no Zabbix repozitorija. Tomēr ņemiet vērā, ka šo versiju pakotnes var neietvert visus Zabbix komponentus, un, lai šos komponentus jauninātu no pakotnēm, apsveriet iespēju jaunināt savu OS. Iekļauto komponentu sarakstu skatiet sadaļā Zabbix pakotnes.

Uz **Ubuntu 24.04** izpildiet:

```
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
```

Uz **Ubuntu 22.04** izpildiet:

```
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:
Vecākām Ubuntu versijām aizstājiet iepriekš norādīto saiti ar pareizo no Zabbix repozitorija. Tomēr ņemiet vērā, ka šo versiju pakotnes var neietvert visus Zabbix komponentus, un, lai šos komponentus jauninātu no pakotnēm, apsveriet iespēju jaunināt savu OS. Iekļauto komponentu sarakstu skatiet sadaļā Zabbix pakotnes.

Jūs varat redzēt uzvedni par Zabbix repozitorija konfigurāciju:

```
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] ?
```

Ievadiet `Y` (vai `I`), lai instalētu pakotnes uzturētāja Zabbix repozitorija konfigurācijas versiju.

Pēc tam atjauniniet repozitorija informāciju:

```
apt update
```

5 Zabbix komponentu jaunināšana

Lai jauninātu Zabbix komponentus, izpildiet:

```
apt install --only-upgrade zabbix-server-mysql zabbix-frontend-php zabbix-agent
```

- Ja izmantojat PostgreSQL, komandā aizstājiet `mysql` ar `pgsql`.
- Ja jaunināt starpniekserveri, komandā aizstājiet `server` ar `proxy`.
- Ja jaunināt Zabbix aģents 2, komandā aizstājiet `zabbix-agent` ar `zabbix-agent2 zabbix-agent2-plugin-*`.

Attention:
Jauninot Zabbix aģents 2 ar komandu `apt install zabbix-agent2`, var rasties kļūda. Plašāku informāciju skatiet sadaļā *Zināmās problēmas*.

Jūs varat redzēt uzvedni par Zabbix serveris (vai starpniekserveris) konfigurāciju:

```
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] ?
```

Ievadiet opciju, kas vislabāk atbilst jūsu situācijai. Piemēram, ievadiet `D`, lai salīdzinātu pašreizējo un jauno konfigurāciju, pēc tam izlemiet, vai vēlaties instalēt pakotnes uzturētāja versiju (`Y` vai `I`).

Pēc tam, lai jauninātu Zabbix lietotāja saskarne ar Apache un restartētu Apache, izpildiet:

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

6 Pārskatiet komponentu konfigurācijas parametrus

Pārskatiet atbilstošās jaunināšanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros ir nepieciešamas kādas izmaiņas.

Informāciju par jaunajiem neobligātajiem parametriem skatiet lapā Kas jauns.

7 Palaidiet Zabbix procesus

Palaidiet atjaunināto Zabbix komponentus:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Notīriet tīmekļa pārlūka sīkfailus un kešatmiņu

Pēc jaunināšanas jums var būt nepieciešams notīrīt tīmekļa pārlūka sīkfailus un tīmekļa pārlūka kešatmiņu, lai Zabbix lietotāja saskarne darbotos pareizi.

Jaunināšana starp mazajām versijām

Ir iespējams jaunināt starp Zabbix 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3).

Vispirms atjauniniet repozitorija informāciju:

```
apt update
```

Pēc tam, lai jauninātu visus Zabbix komponentus, izpildiet:

```
apt install --only-upgrade 'zabbix*'
```

- Lai jauninātu tikai Zabbix serveri, komandā aizstājiet `'zabbix*'` ar `'zabbix-server*'`.
- Lai jauninātu tikai Zabbix starpniekserveri, komandā aizstājiet `'zabbix*'` ar `'zabbix-proxy*'`.
- Lai jauninātu tikai Zabbix aģentu, komandā aizstājiet `'zabbix*'` ar `'zabbix-agent*'`.
- Lai jauninātu tikai Zabbix aģentu 2, komandā aizstājiet `'zabbix*'` ar `'zabbix-agent2*'`.

Red Hat Enterprise Linux

Pārskats

Šajā sadaļā ir sniegti norādījumi, kā veikt jaunināšanu no Zabbix **7.4.x** uz jaunāko Zabbix **8.0.x** versiju, izmantojot oficiālās Zabbix pakotnes Red Hat Enterprise Linux vai tā atvasinājumiem — AlmaLinux, CentOS Stream, Oracle Linux un Rocky Linux.

> **Warning:**
> Pirms jaunināšanas, lūdzu, pārskatiet attiecīgās jaunināšanas piezīmes un pārliecinieties, ka jūsu sistēma atbilst Zabbix 8.0 prasībām.

> **Note:**
> Apsveriet iespēju jaunināšanas laikā izmantot divas paralēlas SSH sesijas: vienu jaunināšanas darbību izpildei un otru servera/starpniekservera žurnālu uzraudzībai. Piemēram, otrajā sesijā palaidiet `tail -f zabbix_server.log` vai `tail -f zabbix_proxy.log`, lai reāllaikā skatītu jaunākos žurnāla ierakstus un iespējamās kļūdas. Tas var būt kritiski svarīgi produkcijas vidēm.

Norādījumus par jaunināšanu starp Zabbix 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3) skatiet sadaļā Jaunināšana starp mazajām versijām.

Jaunināšanas procedūra

1 Apturiet Zabbix procesus

Apturiet Zabbix serveri, lai pārliecinātos, ka datubāzē netiek ievietoti jauni dati:

```
systemctl stop zabbix-server
```

Ja jaunināt Zabbix starpniekserveri, aģentu vai aģentu 2, apturiet arī šīs komponentes:

```
systemctl stop zabbix-proxy
systemctl stop zabbix-agent
systemctl stop zabbix-agent2
```

2 Izveidojiet Zabbix datubāzes dublējumu

Izveidojiet esošās Zabbix datubāzes dublējumu, lai pasargātos no jauninašanas kļūmēm (piemēram, diska vietas problēmām, elektroapgādes pārtraukuma vai neparedzētām problēmām).

3 Izveidojiet Zabbix konfigurācijas failu, PHP failu un Zabbix bināro failu dublējumkopiju

Izveidojiet esošo Zabbix konfigurācijas failu, PHP failu un Zabbix bināro failu dublējumkopiju.

Konfigurācijas failiem izpildiet:

```
mkdir /opt/zabbix-backup/
cp /etc/zabbix/zabbix_server.conf /opt/zabbix-backup/
cp /etc/httpd/conf.d/zabbix.conf  /opt/zabbix-backup/
```

PHP failiem un Zabbix binārajiem failiem izpildiet:

```
cp -R /usr/share/zabbix/ /opt/zabbix-backup/
cp -R /usr/share/zabbix-* /opt/zabbix-backup/
```

4 Atjauniniet repozitorija konfigurācijas pakotni

Pirms turpināt jauninašanu, atjauniniet pašreizējo repozitorija pakotni uz jaunāko versiju, lai nodrošinātu saderību ar jaunākajām pakotnēm un iekļautu jaunākos drošības ielāpus vai kļūdu labojumus.

Uz **RHEL 10** izpildiet:

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

Uz **RHEL 9** izpildiet:

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

Note:
Vecākām RHEL versijām vai to atvasinājumiem aizstājiet iepriekš norādīto saiti ar pareizo no Zabbix repozitorija. Tomēr ņemiet vērā, ka šo versiju pakotnes var neietvert visas Zabbix komponentes, un, lai jauninātu šīs komponentes no pakotnēm, apsveriet iespēju jaunināt savu OS. Iekļauto komponenšu sarakstu skatiet sadaļā Zabbix pakotnes.

Pēc tam iztīriet `dnf` pakotņu pārvaldnieka kešatmiņu (ieskaitot galvenes, metadatus un pakotņu failus, kas lejupielādēti iepriekšējo instalāciju vai atjauninājumu laikā):

```
dnf clean all
```

Nākamās `dnf` darbības laikā `dnf` lejupielādēs svaigus metadatus no repozitorijiem, jo vecie metadati ir notīrīti.

Skatiet arī: Zināmās problēmas par repozitorija konfigurācijas pakotnes atjauninašanu uz RHEL.

5 Atjauniniet Zabbix komponentus

Lai atjauninātu Zabbix komponentus, izpildiet:

```
dnf install zabbix-server-mysql zabbix-web-mysql zabbix-agent
```

- Ja izmantojat PostgreSQL, komandā aizstājiet `mysql` ar `pgsql`.
- Ja atjaunināt starpniekserveri, komandā aizstājiet `server` ar `proxy`.
- Ja atjaunināt aģents 2, aizstājiet `zabbix-agent` ar `zabbix-agent2 zabbix-agent2-plugin-*` komandā.

Attention:
Zabbix aģents 2 atjauninašana ar komandu `dnf install zabbix-agent2` var izraisīt kļūdu. Papildinformāciju skatiet sadaļā *Zināmās problēmas*.

Pēc tam, lai atjauninātu Zabbix lietotāja saskarne ar Apache un restartētu Apache, izpildiet:

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

6 Pārskatiet komponenta konfigurācijas parametrus

Pārskatiet attiecīgās jauninašanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros ir nepieciešamas kādas izmaiņas.

Par jaunajiem neobligātajiem parametriem skatiet lapu Kas jauns.

7 Palaidiet Zabbix procesus

Palaidiet atjauninātos Zabbix komponentus:

```
systemctl start zabbix-server
systemctl start zabbix-proxy
systemctl start zabbix-agent
systemctl start zabbix-agent2
```

8 Notīriet tīmekļa pārlūka sīkfailus un kešatmiņu

Pēc jauninašanas jums var būt nepieciešams notīrīt tīmekļa pārlūka sīkfailus un tīmekļa pārlūka kešatmiņu, lai Zabbix tīmekļa lietotāja saskarne darbotos pareizi.

Jauninašana starp mazajām versijām

Ir iespējams jaunināt starp Zabbix 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3).

Lai jauninātu visus Zabbix komponentus, izpildiet:

```
dnf upgrade 'zabbix-*'
```

- Lai jauninātu tikai Zabbix serveri, aizstājiet `'zabbix-*'` ar `'zabbix-server-*'` komandā.
- Lai jauninātu tikai Zabbix starpniekserveri, aizstājiet `'zabbix-*'` ar `'zabbix-proxy-*'` komandā.
- Lai jauninātu tikai Zabbix aģentu, aizstājiet `'zabbix-*'` ar `'zabbix-agent-*'` komandā.
- Lai jauninātu tikai Zabbix aģentu 2, aizstājiet `'zabbix-*'` ar `'zabbix-agent2-*'` komandā.

Jauninašanas problēmas

Ar atsoļošanu saistīts Pārsniegts funkciju parametru rakstzīmju ierobežojums pēc jauninašanas

Zabbix 7.0 vēstures funkciju virkņu parametros ir pievienota korekta atpakaļvērsto slīpsvītru ekrānēšana.

Tā kā jauninašanas laikā no Zabbix versijām pirms 7.0 tiek pievienotas papildu atpakaļvērstās slīpsvītras, parametri kļūst garāki, un tas var izraisīt bojātas trigeru funkcijas, ja parametra garums pārsniedz maksimālo datu izmēru — 255 rakstzīmes.

Lai parādītu, kādas problēmas var rasties datubāzes jauninašanas laikā, kāda ir to ietekme, kā tās atpazīt un novērst, aplūkosim piemēra konfigurāciju ar 3 vienumiem un 2 trigeriem.

Iestatīšana

1. Zabbix 6.0 versija.
2. `/tmp/ONE` fails, kas satur 114 atpakaļvērstās slīpsvītras ar 'a' beigās:

```
\\\
```

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

```
\\\
```

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

4. Noņemiet konteineru:

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

5. Palaidiet atjaunināto konteineru, izpildot `docker run` komandu, kam seko papildu argumenti, lai norādītu nepieciešamos vides mainīgos un/vai montēšanas punktus.

Konfigurācijas piemēri

Zabbix serveris ar 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 serveris ar PostgreSQL:

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

MySQL serveris:

```
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 \
```

Vairāk konfigurācijas piemēru, tostarp piemēri citiem Zabbix komponentiem, ir pieejami lapā Instalēšana no konteineriem.

6. Pārbaudiet atjauninājumu:

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

Compose faili

Izpildiet šajā sadaļā sniegtos jaunināšanas norādījumus, ja instalējāt Zabbix, izmantojot compose failu.

1. Pārbaudiet pašreizējo attēla versiju:

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

2. Iegūstiet jaunākos atjauninājumus no GitHub repozitorija un pārslēdzieties uz nepieciešamo zaru:

```
git pull
git checkout 8.0
```

3. Palaidiet Zabbix komponentus, izmantojot jauno compose failu:

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

4. Pārbaudiet atjauninājumu:

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

Plašāku informāciju, tostarp atbalstīto vides mainīgo un sējumu montēšanas punktu sarakstus, skatiet sadaļā Instalēšana no konteineriem.

Upgrade from sources

Pārskats

Šajā sadaļā ir aprakstītas darbības, kas nepieciešamas veiksmīgai jaunināšanai no Zabbix **7.4**.x uz Zabbix **8.0**.x, izmantojot oficiālos Zabbix avotus.

> **Warning:**
> Pirms jaunināšanas noteikti izlasiet atbilstošās **jaunināšanas piezīmes**!

Iespējams, vēlēsieties arī pārskatīt 8.0 prasības.

> **Note:**
> Jaunināšanas laikā var būt noderīgi izmantot divas paralēlas SSH sesijas: vienā izpildīt jaunināšanas darbības, bet otrā uzraudzīt servera/starpniekservera žurnālus. Piemēram, otrajā SSH sesijā palaidiet `tail -f zabbix_server.log` vai `tail -f zabbix_proxy.log`, lai reāllaikā redzētu jaunākos žurnālfaila ierakstus un iespējamās kļūdas. Tas var būt kritiski svarīgi produkcijas instancēm.

Servera jaunināšanas process

1 Apturiet serveri

Apturiet Zabbix serveri, lai pārliecinātos, ka datubāzē netiek ievietoti jauni dati.

2 Izveidojiet esošās Zabbix datubāzes dublējumu

Šis ir ļoti svarīgs solis. Pārliecinieties, ka jums ir jūsu datubāzes dublējums. Tas palīdzēs, ja jaunināšanas procedūra neizdosies (diska vietas trūkums, elektroapgādes pārtraukums, jebkāda neparedzēta problēma).

3 Dublējiet konfigurācijas failus, PHP failus un Zabbix bināros failus

Izveidojiet Zabbix bināro failu, konfigurācijas failu un PHP failu direktorija rezerves kopiju.

4 Instalējiet jaunus servera bināros failus

Izmantojiet šīs instrukcijas, lai kompilētu Zabbix serveri no pirmkoda.

5 Pārskatiet servera konfigurācijas parametrus

Noteikti pārskatiet Jaunināšanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros nav nepieciešamas kādas izmaiņas.

Informāciju par jauniem neobligātajiem parametriem skatiet lapā Kas jauns.

6 Palaidiet jaunus Zabbix bināros failus

Palaidiet jaunus bināros failus. Pārbaudiet žurnālfailus, lai redzētu, vai binārie faili ir veiksmīgi palaisti.

Zabbix serveris automātiski jauninās datubāzi. Palaišanas laikā Zabbix serveris ziņo par pašreizējo (obligāto un neobligāto) un nepieciešamo datubāzes versiju. Ja pašreizējā obligātā versija ir vecāka par nepieciešamo versiju, Zabbix serveris automātiski izpilda nepieciešamos datubāzes jauninašanas ielāpus. Datubāzes jauninašanas sākums un norises līmenis (procentos) tiek ierakstīts Zabbix servera žurnālfailā. Kad jauninašana ir pabeigta, žurnālfailā tiek ierakstīts ziņojums "database upgrade fully completed". Ja kāds no jauninašanas ielāpiem neizdodas, Zabbix serveris netiks palaists. Zabbix serveris arī netiks palaists, ja pašreizējā obligātā datubāzes versija ir jaunāka par nepieciešamo. Zabbix serveris tiks palaists tikai tad, ja pašreizējā obligātā datubāzes versija atbilst nepieciešamajai obligātajai versijai.

```
8673:20161117:104750.259 current database version (mandatory/optional): 03040000/03040000
8673:20161117:104750.259 required mandatory version: 03040000
```

Pirms servera palaišanas:

- Pārliecinieties, ka datubāzes lietotājam ir pietiekamas atļaujas (create table, drop table, create index, drop index)
- Pārliecinieties, ka jums ir pietiekami daudz brīvas vietas diskā.

7 Instalējiet jauno Zabbix tīmekļa lietotāja saskarni

Minimālā nepieciešamā PHP versija ir 8.2.0. Ja nepieciešams, atjauniniet to un sekojiet instalēšanas norādījumiem.

8 Notīriet tīmekļa pārlūka sīkdatnes un kešatmiņu

Pēc jauninašanas jums, iespējams, būs jānotīra tīmekļa pārlūka sīkdatnes un tīmekļa pārlūka kešatmiņa, lai Zabbix tīmekļa lietotāja saskarne darbotos pareizi.

Starpniekservera jauninašanas process

1 Apturiet starpniekserveri

Apturiet Zabbix starpniekserveri.

2 Dublējiet konfigurācijas failus un Zabbix starpniekservera binārос failus

Izveidojiet Zabbix starpniekservera binārā faila un konfigurācijas faila rezerves kopiju.

3 Instalējiet jaunus starpniekservera binārос failus

Izmantojiet šīs instrukcijas, lai kompilētu Zabbix starpniekserveri no pirmkoda.

4 Pārskatiet starpniekservera konfigurācijas parametrus

Noteikti pārskatiet Jauninašanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros nav nepieciešamas kādas izmaiņas.

Par jaunajiem neobligātajiem parametriem skatiet lapu Kas jauns.

5 Palaidiet jaunu Zabbix starpniekserveri

Palaidiet jauno Zabbix starpniekserveri. Pārbaudiet žurnālfailus, lai redzētu, vai starpniekserveris ir veiksmīgi palaists.

Zabbix starpniekserveris automātiski jauninās datubāzi. Datubāzes jauninašana notiek līdzīgi kā, palaižot Zabbix serveri.

Aģenta jauninašanas process

> **Attention:**
> Aģentu jauninašana nav obligāta. Aģenti ir jājaunina tikai tad, ja tas ir nepieciešams, lai piekļūtu jaunajai funkcionalitātei.

Šajā sadaļā aprakstīto jauninašanas procedūru var izmantot, lai jauninātu gan Zabbix aģentu, gan Zabbix aģentu 2.

1 Apturiet aģentu

Apturiet Zabbix aģentu.

2 Dublējiet konfigurācijas failus un Zabbix aģenta binārос failus

Izveidojiet Zabbix aģenta binārā faila un konfigurācijas faila rezerves kopiju.

3 Instalējiet jaunus aģenta binārос failus

Izmantojiet šīs instrukcijas, lai kompilētu Zabbix aģentu no pirmkoda.

Alternatīvi, jūs varat lejupielādēt iepriekš kompilētus Zabbix aģentus no Zabbix lejupielādes lapas.

4 Pārskatiet aģenta konfigurācijas parametrus

Noteikti pārskatiet Jauninašanas piezīmes, lai pārbaudītu, vai konfigurācijas parametros ir nepieciešamas kādas izmaiņas.

Informāciju par jaunajiem neobligātajiem parametriem skatiet lapā Kas jauns.

5 Palaidiet jauno Zabbix aģentu

Palaidiet jauno Zabbix aģentu. Pārbaudiet žurnālfailus, lai redzētu, vai aģents ir veiksmīgi palaists.

Jauninašana starp mazajām versijām

Jauninot starp 8.0.x mazajām versijām (piemēram, no 8.0.1 uz 8.0.3), ir jāveic tās pašas darbības attiecībā uz serveri/starpniekserveri/aģentu kā jauninašanas laikā starp galvenajām versijām. Vienīgā atšķirība ir tāda, ka, jauninot starp mazajām versijām, datubāzē netiek veiktas nekādas izmaiņas.

5 Zabbix 8.0 jaunināšanas piezīmes

Šīs piezīmes ir paredzētas jaunināšanai no Zabbix 7.4.x uz Zabbix 8.0.0.

Visas piezīmes ir sagrupētas šādi:

- **Nesaderīgas izmaiņas** - izmaiņas, kas var izjaukt esošās instalācijas, un cita kritiska informācija, kas saistīta ar jaunināšanas procesu
- **Cits** - visa pārējā informācija, kas apraksta izmaiņas Zabbix funkcionalitātē

Skatiet arī:

- Jaunināšanas procedūra visai attiecīgajai informācijai par jaunināšanu no versijām pirms Zabbix 8.0.0;
- HA klastera jaunināšana norādījumiem par serveru jaunināšanu **augstas pieejamības** (HA) klasterī.

Nesaderīgas izmaiņas Datubāzu versijas

Minimālās nepieciešamās datubāzu versijas ir paaugstinātas:

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

Spraudņi

Ceph spraudnis Zabbix aģents 2 tagad ir ielādējams spraudnis, un tam ir nepieciešamas papildu instalēšanas darbības. Sīkāku informāciju skatiet Ceph spraudņa readme.

Shēma `tcp://` ir novecojusi MongoDB spraudnī, taču tā tiek saglabāta atpakaļsaderībai ar esošajām konfigurācijām.

Minimālā nepieciešamā PHP versija

Minimālā nepieciešamā PHP versija ir paaugstināta no 8.0.0 uz 8.2.0.

Paplašināts UnsafeUserParameters saraksts

Rakstzīme % ir pievienota `UnsafeUserParameters` sarakstam Zabbix aģentam un Zabbix aģentam 2.

Citi Novecojušās makrokomandas noņemtas

Atbalsts šādām iebūvētajām makrokomandām tagad ir noņemts:

Noņemta	Izmantojiet tā vietā
{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}

Noņemta JSON un XML validācija HTTP aģenta vienumiem

JSON un XML validācija ir noņemta no HTTP aģenta vienumu lauka *Pieprasījuma pamatteksts*. Lauks vairs nepārbauda, vai vērtība ir derīgs XML vai JSON, un, izvēloties *XML dati* kā *Pieprasījuma pamatteksta tipu*, vairs nav nepieciešama libxml2 bibliotēka.

Atspējotās saites ir paslēptas uzraudzīto hostu sarakstā

Atspējotās *Graphs*, *Dashboards* un *Web* saites vairs netiek rādītas uzraudzīto hostu sarakstā.

JSON datu tips

Zabbix tagad atbalsta JSON kā datu tipu vienuma vērtībām. Ja izmantojat TimescaleDB, jaunā `history_json` hipertabula (ko izmanto JSON vērtību glabāšanai) ir jākonfigurē manuāli. Ja izmantojat Elasticsearch, `HistoryStorageTypes` Zabbix servera konfigurācijas parametra noklusējuma vērtība tagad ietver `json`.

Noņemta radio poga jaunā birkas filtra formā

Radio poga, kas, veidojot jaunu birkas filtru, prasīja izvēlēties starp *Visas birkas* un *Birku saraksts*, ir noņemta.

Vienumu vērtības, kas attēlotas kā HTML, izmantojot iframe

Vienuma dati logrīkā Vienuma vēsture, ja tie tiek attēloti kā HTML formatēts teksts, tagad ir izolēti iframe.

Veidņu saderība ar hostu vedni

Visas gatavās veidnes tagad ir saderīgas ar Host Wizard. Lai tās jauninātu, skatiet Veidņu jaunināšana.

Veidņu izmaiņas

Šajā lapā ir uzskaitītas visas izmaiņas standarta veidnēs, kas tiek piegādātas kopā ar Zabbix.

Jauninot Zabbix, esošās veidnes netiek atjauninātas automātiski, lai izvairītos no pielāgoto izmaiņu pārrakstīšanas. Lai jauninātu veidnes vai pievienotu jaunas, skatiet Veidņu jaunināšana.

> **Note:**
> Lūdzu, ņemiet vērā, ka kopš Zabbix 6.0 visas veidnes izmanto atjauninātu formātu, kas var ietekmēt pirms 6.0 izveidoto veidņu importēšanu. Papildinformāciju skatiet sadaļā Veidņu izmaiņas versijā 6.0.

Izmaiņas 8.0.0

Zabbix 8.0.x jaunināšanas piezīmes

Šajā lapā ir apkopotas galvenās Zabbix versijas mazāko laidienu jaunināšanas piezīmes.

Skatiet arī galvenās versijas jaunināšanas piezīmes.

Jaunināšanas piezīmes versijai 8.0.1 Šī versija vēl nav izlaista.

Prasības

Aparatūra

Atmiņa

Zabbix nepieciešama gan fiziskā, gan diska atmiņa. Nepieciešamais diska atmiņas apjoms acīmredzami ir atkarīgs no uzraugāmo hostu un parametru skaita. Ja plānojat ilgstoši glabāt uzraudzīto parametru vēsturi, jums vajadzētu rēķināties vismaz ar dažiem gigabaitiem, lai datubāzē pietiktu vietas vēstures glabāšanai. Katram Zabbix dēmona procesam ir nepieciešami vairāki savienojumi ar datubāzes serveri. Savienojumam piešķirtais atmiņas apjoms ir atkarīgs no datubāzes dzinēja konfigurācijas.

> **Note:**
> Jo vairāk fiziskās atmiņas jums ir, jo ātrāk darbojas datubāze (un līdz ar to arī Zabbix).

CPU

Zabbix un jo īpaši Zabbix datubāzei var būt nepieciešami ievērojami CPU resursi atkarībā no uzraudzīto parametru skaita un izvēlētā datubāzes dzinēja.

Cita aparatūra

Lai izmantotu SMS paziņojumu atbalstu Zabbix, ir nepieciešams seriālās komunikācijas ports un seriālais GSM modems. Derēs arī USB-uz-seriālo pārveidotājs.

Aparatūras konfigurācijas piemēri

Tabulā sniegti aparatūras konfigurācijas piemēri, pieņemot, ka tiek izmantota **Linux/BSD/Unix** platforma.

Šie ir izmēra un aparatūras konfigurācijas piemēri, ar ko sākt. Katra Zabbix instalācija ir unikāla. Noteikti veiciet sava Zabbix sistēmas veiktspējas testēšanu sagatavošanas vai izstrādes vidē, lai pirms Zabbix instalācijas izvietošanas ražošanas vidē jūs varētu pilnībā izprast savas prasības.

Instalācijas lielums	Uzraudzītās metrikas[1]	CPU/vCPU kodoli	Atmiņa (GiB)	Datubāze	Amazon EC2[2]
Maza	1 000	2	8	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.large/m6g.large
Vidēja	10 000	4	16	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.xlarge/m6g.xlarge
Liela	100 000	16	64	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.4xlarge/m6g.4xlarge
Ļoti liela	1 000 000	32	96	MySQL Server, Percona Server, MariaDB Server, PostgreSQL	m6i.8xlarge/m6g.8xlarge

[1] 1 metrika = 1 vienums + 1 trigeris + 1 grafiks
 [2] Piemērs ar Amazon vispārējas nozīmes EC2 instancēm, izmantojot ARM64 vai x86_64 arhitektūru; Zabbix instalācijas novērtēšanas un testēšanas laikā pirms instalēšanas ražošanas vidē jāizvēlas atbilstošs instances tips, piemēram, Compute/Memory/Storage optimised.

Note:
Faktiskā konfigurācija lielā mērā ir atkarīga no aktīvo vienumu skaita un atsvaidzināšanas biežuma (sīkāku informāciju skatiet šīs lapas sadaļā database size). Lielām instalācijām ir ļoti ieteicams darbināt datubāzi uz atsevišķa servera.

Atbalstītās platformas

Drošības prasību un monitoringa servera kritiski svarīgā rakstura dēļ UNIX ir vienīgā operētājsistēma, kas var konsekventi nodrošināt nepieciešamo veiktspēju, kļūdu noturību un darbības nepārtrauktību. Zabbix darbojas tirgū vadošajās versijās.

Zabbix komponenti ir pieejami un testēti šādām platformām:

Platforma	Serveris	Aģents	Aģents 2	Komentāri
Linux	x	x	x	
Windows	-	x	x	Zabbix aģents tiek atbalstīts visās darbvirsmas un serveru versijās, sākot no Windows XP (64 bitu)/Server 2003. Zabbix aģents 2 tiek atbalstīts visās darbvirsmas un serveru versijās, sākot no Windows 10 (32 bitu)/Server 2016, jo tas tiek kompilēts tikai ar atbalstītu Go versiju, lai novērstu kritiskas drošības ievainojamības. Kopš Go 1.21 minimālās prasītās Windows versijas ir paaugstinātas, tādēļ Windows 10/Server 2016 ir minimālā versija Zabbix aģentam 2.
macOS	x	x	-	
IBM AIX	x	x	-	Zabbix aģents nedarbojas AIX platformās ar versijām zemākām par 6.1 TL07 / 7.1 TL01.
FreeBSD	x	x	-	
OpenBSD	x	x	-	
Solaris	x	x	-	
NetBSD	x	x	-	
HP-UX	x	x	-	

Note:
Zabbix serveris/aģents var darboties arī citās Unix tipa operētājsistēmās.

Attention:
Zabbix atspējo core dump failu izveidi, ja tas ir kompilēts ar šifrēšanu, un nestartējas, ja sistēma neļauj atspējot core dump failu izveidi.

Nepieciešamā programmatūra

Zabbix ir veidots, izmantojot mūsdienīgus tīmekļa serverus, vadošās datubāzu pārvaldības sistēmas un PHP skriptu valodu.

Trešo pušu ārējā apkārtējā programmatūra

Ja norādīts kā obligāts, nepieciešamā programmatūra/bibliotēka ir stingri nepieciešama. Izvēles komponenti ir vajadzīgi noteiktu specifisku funkciju atbalstam.

Programmatūra	Obligātuma statuss	Atbalstītās versijas	Komentāri
MySQL/Percona	Viena no	8.4.0-9.6.X	Nepieciešams, ja MySQL (vai Percona) tiek izmantota kā Zabbix aizmugursistēmas datubāze. Ir nepieciešams InnoDB dzinis. Mēs iesakām servera/starpniekservera būvēšanai izmantot bibliotēku C API (libmysqlclient).
MariaDB		10.11.00-12.2.X	Ir nepieciešams InnoDB dzinis. Ieteicamā versija ir 11.4. Mēs iesakām servera/starpniekservera būvēšanai izmantot bibliotēku MariaDB Connector/C. Skatiet arī: Iespējamie strupceļi ar MariaDB un Piekļuve UI elementiem ar MariaDB 10.5.1-10.5.9.
PostgreSQL		15.0-18.X	Nepieciešams, ja PostgreSQL tiek izmantota kā Zabbix aizmugursistēmas datubāze. Atkarībā no instalācijas apjoma var būt nepieciešams palielināt PostgreSQL *work_mem* konfigurācijas rekvizītu (4MB ir noklusējuma vērtība), lai datubāzes izmantotais atmiņas apjoms konkrētai darbībai būtu pietiekams un vaicājumu izpilde neaizņemtu pārāk daudz laika.
TimescaleDB for PostgreSQL		2.20.X-2.26.X	Nepieciešams, ja TimescaleDB tiek izmantots kā PostgreSQL datubāzes paplašinājums. Pārliecinieties, ka instalējat TimescaleDB Community Edition, kas atbalsta saspiešanu. Ņemiet vērā, ka PostgreSQL 15 tiek atbalstīts kopš TimescaleDB 2.10. Varat arī skatīt TimescaleDB documentation, lai iegūtu detalizētu informāciju par PostgreSQL un TimescaleDB versiju saderību.
SQLite	Izvēles	3.3.5-3.34.X	SQLite tiek atbalstīts tikai ar Zabbix starpniekserveriem. Nepieciešams, ja SQLite tiek izmantota kā Zabbix starpniekservera datubāze.
Elasticsearch		7.X	Elasticsearch tiek atbalstīts tikai ar Zabbix serveriem un tikai vēstures datu glabāšanai. Elasticsearch atbalsts pašlaik ir eksperimentāls. Skatiet arī nepieciešamo programmatūru serverim/starpniekserverim.
smartmontools		7.1 vai jaunāka	Nepieciešams Zabbix aģentam 2.
who			Nepieciešams lietotāju skaita spraudnim.
dpkg			Nepieciešams system.sw.packages spraudnim.
pkgtool			Nepieciešams system.sw.packages spraudnim.
rpm			Nepieciešams system.sw.packages spraudnim.
pacman			Nepieciešams system.sw.packages spraudnim.
q applets			`qlist` un `qsize` kā daļa no q applets ir nepieciešami system.sw.packages spraudnim uz Gentoo Linux.

Note:
Lai gan Zabbix var darboties ar operētājsistēmās pieejamām datubāzēm, vislabākajai lietošanas pieredzei mēs iesakām izmantot datubāzes, kas instalētas no oficiālajām datubāzu izstrādātāju repozitorijām.

Lietotāja saskarne

Minimālais atbalstītais ekrāna platums Zabbix lietotāja saskarnei ir 1200px.

Ja norādīts kā obligāts, nepieciešamā programmatūra/bibliotēka ir stingri nepieciešama. Izvēles komponenti ir vajadzīgi, lai atbalstītu dažas specifiskas funkcijas.

Programmatūra	Obligātuma statuss	Atbalstītās versijas	Komentāri
PHP	Jā	8.2.0 - 8.5.X	
Apache	Viena no	2.4 vai jaunāka	
Nginx		1.20 vai jaunāka	
MySQL	Viena no	Skatiet Trešo pušu ārējo apkārtējās vides programmatūru	
PostgreSQL			
PHP paplašinājumi			
mysqli	Jā		Nepieciešams, ja MySQL tiek izmantots kā Zabbix aizmugursistēmas datubāze.
pgsql			Nepieciešams, ja PostgreSQL tiek izmantots kā Zabbix aizmugursistēmas datubāze.
bcmath			php-bcmath (*--enable-bcmath*)
mbstring			php-mbstring (*--enable-mbstring*)
sockets			php-net-socket (*--enable-sockets*); nepieciešams lietotāja skriptu atbalstam.
gd		2.0.28 vai jaunāka	php-gd (ja izplatītājs to nodrošina kā atsevišķu pakotni); PHP GD paplašinājumam jāatbalsta PNG attēli (*--with-png-dir*), JPEG attēli (*--with-jpeg-dir*) un FreeType 2 (*--with-freetype-dir*). Lai izvairītos no iespējamas teksta pārklāšanās grafikos dažām lietotāja saskarnes valodām, var būt nepieciešama versija 2.3.0 vai jaunāka.
libxml		2.6.15 vai jaunāka	php-xml (ja izplatītājs to nodrošina kā atsevišķu pakotni)
xmlwriter			php-xmlwriter (ja izplatītājs to nodrošina kā atsevišķu pakotni)
xmlreader			php-xmlreader (ja izplatītājs to nodrošina kā atsevišķu pakotni)
ctype			php-ctype (*--enable-ctype*)
session			php-session (ja izplatītājs to nodrošina kā atsevišķu pakotni)
ldap	Nē		php-ldap; nepieciešams LDAP autentifikācijai.
openssl			php-openssl; nepieciešams SAML autentifikācijai.
gettext			php-gettext (*--with-gettext*); nepieciešams tulkojumiem.
cURL		7.19.4 vai jaunāka	php-curl; nepieciešams Duo Universal Prompt MFA un SMTP autentifikācijai.

Trešo pušu lietotāja saskarnes bibliotēkas, kas tiek piegādātas kopā ar Zabbix:

Bibliotēka	Obligātuma statuss	Piegādātā versija	Komentāri
jQuery JavaScript Library	Jā	3.6.0	JavaScript bibliotēka, kas vienkāršo starppārlūku izstrādes procesu.
jQuery UI		1.12.1	Lietotāja saskarnes mijiedarbību, efektu, logrīku un motīvu kopums, kas veidots uz jQuery bāzes.

Bibliotēka	Obligātuma statuss	Piegādātā versija	Komentāri
SAML PHP Toolkit		4.3.1	PHP rīkkopa, kas pievieno SAML 2.0 autentifikācijas atbalstu, lai varētu pierakstīties Zabbix.
Symfony Yaml Component		5.1.0	Pievieno atbalstu Zabbix konfigurācijas elementu eksportēšanai un importēšanai YAML formātā.

Note:
Zabbix var darboties arī ar iepriekšējām Apache, MySQL un PostgreSQL versijām.

Attention:
Citām fontu saimēm nekā noklusējuma DejaVu var būt nepieciešama PHP funkcija imagerotate. Ja tās nav, šie fonti var tikt attēloti nepareizi, kad tiek parādīts grafiks. Šī funkcija ir pieejama tikai tad, ja PHP ir kompilēts ar iekļauto GD, kas nav gadījums Debian un citās distribūcijās.

Trešo pušu bibliotēkas, ko izmanto Zabbix lietotāja saskarnes koda rakstīšanai un atkļūdošanai:

Bibliotēka	Obligātuma statuss	Minimālā versija	Apraksts
Composer	Nē	2.4.1	Lietojumprogrammu līmeņa PHP pakotņu pārvaldnieks, kas nodrošina standarta formātu PHP programmatūras atkarību un nepieciešamo bibliotēku pārvaldībai.
PHPUnit		8.5.29	PHP vienību testēšanas ietvars Zabbix lietotāja saskarnes testēšanai.
SASS		3.4.22	Priekšapstrādes skriptu valoda, kas tiek interpretēta un kompilēta kaskadējošajās stilu lapās (CSS).

Tīmekļa pārlūks klienta pusē

Jābūt iespējotiem sīkfailiem un JavaScript.

Tiek atbalstītas jaunākās stabilās Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari un Opera versijas.

Warning:
Tiek īstenota vienas izcelsmes politika IFrame elementiem, kas nozīmē, ka Zabbix nevar ievietot rāmjos citā domēnā.

 Tomēr lapām, kas ievietotas Zabbix rāmī, būs piekļuve Zabbix lietotāja saskarnei (izmantojot JavaScript), ja rāmī ievietotā lapa un Zabbix lietotāja saskarne atrodas vienā un tajā pašā domēnā. Tādai lapai kā `http://secure-zabbix.com/cms/page.html`, ja tā ir ievietota paneļos vietnē `http://secure-zabbix.com/zabbix/`, būs pilna JS piekļuve Zabbix.

Serveris/starpniekserveris

Ja norādīts kā obligāts, nepieciešamā programmatūra/bibliotēka ir stingri nepieciešama. Neobligātās ir vajadzīgas noteiktu specifisku funkciju atbalstam.

Prasība	Obligātuma statuss	Apraksts
libpcre2	Jā	PCRE2 bibliotēka ir nepieciešama Perl Compatible Regular Expression (PCRE) atbalstam. Tiek atbalstīta PCRE2 v10.x.
libevent		Nepieciešama starpprocesu saziņai. Versija 2.0.10 vai jaunāka.
libevent-pthreads		Nepieciešama starpprocesu saziņai.
libpthread		Nepieciešama mutex un lasīšanas-rakstīšanas bloķēšanas atbalstam (var būt daļa no libc).
libresolv		Nepieciešama DNS izšķiršanai (var būt daļa no libc).
libiconv		Nepieciešama teksta kodējuma/formāta pārveidošanas atbalstam (var būt daļa no libc). Obligāta Zabbix serverim Linux vidē.
libz		Nepieciešama saspiešanas atbalstam.
libm		Matemātikas bibliotēka. Nepieciešama tikai Zabbix serverim.
libmysqlclient	Viena no	Nepieciešama, ja tiek izmantots MySQL.

Prasība	Obligātuma statuss	Apraksts
libmariadb		Nepieciešama, ja tiek izmantots MariaDB.
libpq5		Nepieciešama, ja tiek izmantots PostgreSQL; *libpq5* versijai jāsakrīt ar izmantotās PostgreSQL datubāzes versiju vai jābūt jaunākai.
libsqlite3		Nepieciešama, ja tiek izmantots SQLite. Nepieciešama tikai Zabbix starpniekserverim.
libOpenIPMI	Nē	Nepieciešama IPMI atbalstam. Nepieciešama tikai Zabbix serverim.
libssh2 vai *libssh*		Nepieciešama SSH pārbaudēm. Versija 1.8.0 vai jaunāka (libssh2); 0.9.0 vai jaunāka (libssh).
libcurl		Nepieciešama šādām iespējām: - Tīmekļa uzraudzība, VMware uzraudzība un HTTP aģenta vienumi (visiem: versija 7.19.1 vai jaunāka); - Zabbix aģenta web.page.* vienumi (versija 7.19.1 vai jaunāka; skatiet arī aģents prasības); - SMTP autentifikācija (Basic: versija 7.20.0 vai jaunāka; OAuth: versija 7.33 vai jaunāka; skatiet arī lietotāja saskarne prasības); - Elasticsearch (versija 7.28.0 vai jaunāka). Visām iespējām ieteicama versija 7.28.0 vai jaunāka. Lai izmantotu atjauninātās cURL iespējas `web.page.*` vienumiem, pārstartējiet Zabbix serveri/starpniekserveri. SMTP autentifikācijai izpildlaikā izmantojiet `libcurl-full` pakotni.
libxml2		Nepieciešama VMware uzraudzībai un XML XPath priekšapstrādei.
net-snmp		Nepieciešama SNMP atbalstam. Versija 5.3.0 vai jaunāka. Spēcīgu šifrēšanas protokolu (AES192/AES192C, AES256/AES256C) atbalsts ir pieejams, sākot ar net-snmp bibliotēkas versiju 5.8; sistēmās, kuru pamatā ir RHEL 8+, ieteicams izmantot net-snmp 5.8.15 vai jaunāku.
libunixodbc		Nepieciešama datubāzes uzraudzībai.
libgnutls vai *libopenssl*		Nepieciešama, izmantojot šifrēšanu. Minimālās versijas: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1
libldap		Nepieciešama LDAP atbalstam.
fping		Nepieciešama ICMP ping vienumiem.
c-ares		Nepieciešama asinhronai DNS izšķiršanai, ja Zabbix ir konfigurēts ar `--with-ares` opciju. Pretējā gadījumā tiks izmantots *libevent*. Minimālā versija: 1.16.0. DNS vaicājumu kešošanai (vaicājumu kešatmiņa ir iespējota pēc noklusējuma) nepieciešams c-ares 1.26.0 vai jaunāks.

Aģents

Prasība	Obligātuma statuss	Apraksts
libpcre2	Jā	PCRE2 bibliotēka ir nepieciešama Perl Compatible Regular Expression (PCRE) atbalstam. Tiek atbalstīta PCRE2 v10.x. Nepieciešama žurnālfailu uzraudzībai. Nepieciešama arī Windows vidē.
libpthread		Nepieciešama mutex un lasīšanas-rakstīšanas bloķēšanas atbalstam (var būt daļa no libc). Nav nepieciešama Windows vidē.
libresolv		Nepieciešama DNS atrisināšanai (var būt daļa no libc). Nav nepieciešama Windows vidē.
libiconv		Nepieciešama teksta kodējuma/formāta pārveidošanai uz UTF-8 žurnāla vienumos, faila satura, faila regex un regmatch vienumos (var būt daļa no libc). Nav nepieciešama Windows vidē.
libgnutls vai *libopenssl*	Nē	Nepieciešama, ja tiek izmantota šifrēšana. Minimālās versijas: *libgnutls* - 3.1.18, *libopenssl* - 1.0.1 Microsoft Windows vidē ir nepieciešams OpenSSL 1.1.1 vai jaunāks.
libldap		Nepieciešama, ja tiek izmantots LDAP. Netiek atbalstīta Windows vidē.

Prasība	Obligātuma statuss	Apraksts
libcurl		Nepieciešama paplašinātam Zabbix aģenta web.page.* vienumu atbalstam. Bez libcurl ir pieejama pamatfunkcionalitāte (piemēram, web.page.get[http://example.com]). Ar libcurl aģents atbalsta papildu iespējas, piemēram, HTTP URL ar akreditācijas datiem (piemēram, http://user:password@example.com) un HTTPS URL. Nepieciešama versija 7.19.1 vai jaunāka (ieteicama 7.28.0 vai jaunāka). Lai izmantotu atjauninātās cURL iespējas, pārstartējiet Zabbix aģentu.
libmodbus		Nepieciešama tikai tad, ja tiek izmantota Modbus uzraudzība. Versija 3.0 vai jaunāka.

Aģents 2

Prasība	Obligātuma statuss	Apraksts
Go	Jā	Nepieciešams, lai no pirmkoda izveidotu Zabbix aģentu 2 un tā spraudņus. Tiek atbalstīts Go 1.24.10 vai jaunāks. Instalēšanas norādījumus skatiet go.dev. Go bibliotēkas, ko izmanto Zabbix aģents 2 un tā spraudņi, ir uzskaitītas Zabbix Git repozitorijā (bibliotēkas, kas repozitorijā atzīmētas kā `indirect`, ir citu nepieciešamo bibliotēku atkarības): - Zabbix aģents 2 - Spraudņu atbalsts - PostgreSQL - MongoDB - MSSQL - Ember+ - NVIDIA GPU - Spraudņa piemērs
libpcre2	Jā	PCRE2 bibliotēka ir nepieciešama Perl Compatible Regular Expression (PCRE) atbalstam. Tiek atbalstīta PCRE2 v10.x. Nepieciešama žurnālu uzraudzībai. Nepieciešama arī operētājsistēmā Windows.
libopenssl	Nē	Nepieciešams, ja tiek izmantota šifrēšana. UNIX platformās ir nepieciešams OpenSSL 1.0.1 vai jaunāks. OpenSSL bibliotēkā jābūt iespējotam PSK atbalstam. LibreSSL netiek atbalstīts. Microsoft Windows sistēmās ir nepieciešams OpenSSL 1.1.1 vai jaunāks.

Tīmekļa pakalpojums

Jaunākā stabilā Google Chrome versija tiek atbalstīta ieplānoto atskaišu ģenerēšanai, izmantojot Zabbix tīmekļa pakalpojumu.

Nepieciešamā Go versija tīmekļa pakalpojuma būvēšanai atbilst tai, kas tiek izmantota Zabbix aģents 2.

Java gateway

Ja Zabbix ieguvāt no pirmkoda repozitorija vai arhīva, tad nepieciešamās atkarības jau ir iekļautas pirmkoda kokā.

Ja Zabbix ieguvāt no savas distribūcijas pakotnes, tad nepieciešamās atkarības jau nodrošina pakotņu pārvaldības sistēma.

Abos iepriekš minētajos gadījumos programmatūra ir gatava lietošanai, un papildu lejupielādes nav nepieciešamas.

Tomēr, ja vēlaties nodrošināt savas šo atkarību versijas (piemēram, ja gatavojat pakotni kādai Linux distribūcijai), zemāk ir norādīts bibliotēku versiju saraksts, ar kurām Java gateway, kā zināms, darbojas. Zabbix var darboties arī ar citām šo bibliotēku versijām.

Šajā tabulā ir uzskaitīti JAR faili, kas pašlaik ir iekļauti Java gateway oriģinālajā kodā:

Bibliotēka	Obligātuma statuss	Iekļautā versija	Komentāri
android-json	Jā	4.3r1	JSON (JavaScript Object Notation) ir viegls datu apmaiņas formāts. Šī ir ar org.json saderīga Android implementācija, kas iegūta no Android SDK.
logback-classic		1.5.16	
logback-core		1.5.16	

Bibliotēka	Obligātuma statuss	Iekļautā versija	Komentāri
slf4j-api		2.0.16	

Java gateway var kompilēt, izmantojot gan Oracle Java, gan atvērtā pirmkoda OpenJDK (versija 1.6 vai jaunāka). Zabbix nodrošinātās pakotnes ir kompilētas, izmantojot OpenJDK. Šajā tabulā ir uzskaitītas OpenJDK pakotnes, kas tiek izmantotas Zabbix pakotņu būvēšanai atkarībā no distribūcijas:

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

Noklusējuma portu numuri

Tālāk redzamais atvērto portu saraksts katram komponentam ir piemērojams noklusējuma konfigurācijai:

Zabbix komponents	Porta numurs	Protokols	Savienojuma veids
Zabbix aģents	10050	TCP	pēc pieprasījuma
Zabbix aģents 2	10050	TCP	pēc pieprasījuma
Zabbix serveris	10051	TCP	pēc pieprasījuma
Zabbix starpniekserveris	10051	TCP	pēc pieprasījuma
Zabbix Java gateway	10052	TCP	pēc pieprasījuma
Zabbix tīmekļa pakalpojums	10053	TCP	pēc pieprasījuma
Zabbix lietotāja saskarne	80	HTTP	pēc pieprasījuma
	443	HTTPS	pēc pieprasījuma
Zabbix trapper	10051	TCP	pēc pieprasījuma

Note:
Lai nodrošinātu Zabbix saziņu, šiem portu numuriem jābūt atvērtiem ugunsmūrī. Izejošajiem TCP savienojumiem parasti nav nepieciešami īpaši ugunsmūra iestatījumi.

Datubāzes lielums

Zabbix konfigurācijas datiem ir nepieciešams noteikts diska vietas apjoms, un tie būtiski nepalielinās.

Zabbix datubāzes lielums galvenokārt ir atkarīgs no šiem mainīgajiem, kas nosaka saglabāto vēsturisko datu apjomu:

- Apstrādāto vērtību skaits sekundē

Tas ir vidējais jauno vērtību skaits, ko Zabbix serveris saņem katru sekundi. Piemēram, ja mums ir 3000 vienumi uzraudzībai ar atjaunināšanas intervālu 60 sekundes, vērtību skaits sekundē tiek aprēķināts šādi: 3000/60 = **50**.

Tas nozīmē, ka Zabbix datubāzei katru sekundi tiek pievienotas 50 jaunas vērtības.

- Housekeeper iestatījumi vēsturei

Zabbix saglabā vērtības noteiktu laika periodu, parasti vairākas nedēļas vai mēnešus. Katrai jaunai vērtībai ir nepieciešams noteikts diska vietas apjoms datiem un indeksam.

Tātad, ja mēs vēlamies saglabāt 30 dienu vēsturi un saņemam 50 vērtības sekundē, kopējais vērtību skaits būs aptuveni (**30***24*3600)* **50** = 129.600.000 jeb aptuveni 130M vērtību.

Atkarībā no izmantotās datubāzes dzinēja, saņemto vērtību tipa (peldošā komata skaitļi, veseli skaitļi, virknes, žurnālfaili utt.) diska vieta vienas vērtības glabāšanai var svārstīties no 40 baitiem līdz simtiem baitu. Parasti skaitliskiem vienumiem tas ir apmēram 90 baiti uz vērtību[2]. Mūsu gadījumā tas nozīmē, ka 130M vērtību būs nepieciešami 130M * 90 baiti = **10.9GB** diska vietas.

Note:
Teksta/žurnāla vienumu vērtību lielumu nav iespējams precīzi paredzēt, taču var rēķināties ar aptuveni 500 baitiem uz vērtību.

- Housekeeper iestatījums tendencēm

Zabbix tabulā **trends** katram vienumam saglabā 1 stundas max/min/avg/count vērtību kopu. Šie dati tiek izmantoti tendenču analīzei un ilgtermiņa grafikiem. Vienas stundas periodu nevar pielāgot.

Zabbix datubāzei atkarībā no datubāzes tipa ir nepieciešami aptuveni 90 baiti katram šādam ierakstam. Pieņemsim, ka mēs vēlamies saglabāt tendenču datus 5 gadus. Vērtībām no 3000 vienumiem būs nepieciešami 3000*24*365* **90** = **2.2GB** gadā jeb **11GB** 5 gados.

- Housekeeper iestatījumi notikumiem

Katram Zabbix notikumam ir nepieciešami aptuveni 250 baiti diska vietas[1]. Ir grūti novērtēt, cik notikumu Zabbix ģenerē dienā. Sliktākajā gadījumā var pieņemt, ka Zabbix ģenerē vienu notikumu sekundē.

Katram atjaunotajam notikumam tiek izveidots event_recovery ieraksts. Parasti lielākā daļa notikumu tiks atjaunoti, tāpēc var pieņemt vienu event_recovery ierakstu uz vienu notikumu. Tas nozīmē papildu 80 baitus uz notikumu.

Pēc izvēles notikumiem var būt birkas, un katram birkas ierakstam ir nepieciešami aptuveni 100 baiti diska vietas[1]. Birkas skaits uz notikumu (#tags) ir atkarīgs no konfigurācijas. Tātad katram notikumam papildus būs nepieciešams #tags * 100 baitu diska vietas.

Tas nozīmē, ka, ja mēs vēlamies saglabāt 3 gadu notikumus, tam būs nepieciešama 3*365*24*3600* (250+80+#tags*100) = **~30GB**+#tags*100B diska vieta[2].

Note:
[1] Vairāk, ja notikumu nosaukumos, birkās un vērtībās ir rakstzīmes, kas nav ASCII.
 [2] Lieluma aptuvenie aprēķini ir balstīti uz MySQL un citām datubāzēm var atšķirties.

Tabulā ir formulas, kuras var izmantot, lai aprēķinātu Zabbix sistēmai nepieciešamo diska vietu:

Parameter	Formula for required disk space (in bytes)
Zabbix configuration	Fiksēts lielums. Parasti 10MB vai mazāk.
History	days*(items/refresh rate)*24*3600*bytes items : vienumu skaits days : dienu skaits, cik ilgi saglabāt vēsturi refresh rate : vienumu vidējais atjaunināšanas intervāls bytes : baitu skaits, kas nepieciešams vienas vērtības glabāšanai; atkarīgs no datubāzes dzinēja, parasti ~90 baiti.
Trends	days*(items/3600)*24*3600*bytes items : vienumu skaits days : dienu skaits, cik ilgi saglabāt vēsturi bytes : baitu skaits, kas nepieciešams vienas tendences glabāšanai; atkarīgs no datubāzes dzinēja, parasti ~90 baiti.

Parameter	Formula for required disk space (in bytes)
Events	days*events*24*3600*bytes events : notikumu skaits sekundē. Sliktākajā gadījumā viens (1) notikums sekundē. days : dienu skaits, cik ilgi saglabāt vēsturi bytes : baitu skaits, kas nepieciešams viena notikuma glabāšanai; atkarīgs no datubāzes dzinēja, parasti ~330 + vidējais birku skaits uz notikumu * 100 baiti.

Tātad kopējo nepieciešamo diska vietu var aprēķināt šādi:

Configuration + History + Trends + Events

Diska vieta NETIKS izmantota uzreiz pēc Zabbix instalēšanas. Datubāzes lielums pieaugs, un pēc tam kādā brīdī pārstās pieaugt, atkarībā no housekeeper iestatījumiem.

Laika sinhronizācija

Ir ļoti svarīgi nodrošināt precīzu sistēmas laiku serverī, kurā darbojas Zabbix. ntpd ir vispopulārākais dēmons, kas sinhronizē hosta laiku ar citu iekārtu laiku. Ir stingri ieteicams uzturēt sinhronizētu sistēmas laiku visās sistēmās, kurās darbojas Zabbix komponenti.

Tīkla prasības

Tālāk norādītais atvērto portu saraksts katram komponentam attiecas uz noklusējuma konfigurāciju.

Komponenti	Ports
Lietotāja saskarne	http uz 80, https uz 443
Serveris	10051 (lietošanai ar aktīvu starpniekserveri/aģentiem)
Aktīvs starpniekserveris	10051
Pasīvs starpniekserveris	10051
Aģents2	10050
Trapper	
Java gateway	10052
WebService	10053

Note:
Portu numuriem jābūt atvērtiem ugunsmūrī, lai iespējotu ārējo saziņu ar Zabbix. Izejošajiem TCP savienojumiem parasti nav nepieciešami īpaši ugunsmūra iestatījumi.

Zināmās problēmas

Skatiet arī: Kompilēšanas problēmas.

Jaunināšana

SQL režīma iestatījums veiksmīgai jaunināšanai

MySQL/MariaDB iestatījumam `sql_mode` jābūt iestatītam režīmam "STRICT_TRANS_TABLES". Ja tas nav iestatīts, Zabbix datubāzes jaunināšana neizdosies (skatiet arī ZBX-19435).

Jaunināšana ar MariaDB 10.2.1 un vecākām versijām

Zabbix jaunināšana var neizdoties, ja datubāzes tabulas tika izveidotas ar MariaDB 10.2.1 un vecākām versijām, jo šajās versijās noklusējuma rindas formāts ir compact. To var novērst, mainot rindas formātu uz dynamic (skatiet arī ZBX-17690).

Veidnes

Veidņu saderība dual-stack (IPv4/IPv6) vidēs

Dual-stack vidēs (sistēmās, kas konfigurētas darbam gan ar IPv4, gan IPv6) resursdatora nosaukums `localhost` parasti tiek atrisināts gan uz IPv4, gan IPv6 adresēm. Tā kā daudzas operētājsistēmas un DNS atrisinātāji bieži piešķir prioritāti IPv6 salīdzinājumā ar IPv4, Zabbix veidnes var nedarboties pareizi, ja uzraugāmais pakalpojums ir konfigurēts klausīties tikai uz IPv4.

Pakalpojumi, kas nav konfigurēti klausīties uz IPv6 adresēm, var kļūt nepieejami, izraisot uzraudzības kļūmes. Lietotāji var pareizi konfigurēt piekļuvi IPv4, taču joprojām saskarties ar savienojamības problēmām, jo pēc noklusējuma prioritāte tiek piešķirta IPv6.

Viens no risinājumiem ir nodrošināt, ka pakalpojumi (Nginx, Apache, PostgreSQL u.c.) ir konfigurēti klausīties gan uz IPv4, gan IPv6 adresēm un ka Zabbix serverim/aģentam ir atļauta piekļuve, izmantojot IPv6. Turklāt Zabbix veidnēs un konfigurācijās izmantojiet tieši `localhost`, nevis `127.0.0.1`, lai nodrošinātu saderību gan ar IPv4, gan IPv6.

Piemēram, uzraugot PostgreSQL ar veidni PostgreSQL by Zabbix agent 2, iespējams, būs jāredifģē fails `pg_hba.conf`, lai atļautu savienojumus lietotājam `zbx_monitor`. Ja dual-stack vide piešķir prioritāti IPv6 (sistēma atrisina localhost uz `::1`) un jūs konfigurējat `localhost`, bet pievienojat tikai IPv4 ierakstu (`127.0.0.1/32`), savienojums neizdosies, jo nebūs atbilstoša IPv6 ieraksta.

Šis `pg_hba.conf` faila piemērs nodrošina, ka lietotājs `zbx_monitor` var pieslēgties jebkurai datubāzei no lokālās mašīnas, izmantojot gan IPv4, gan IPv6 adreses ar dažādām autentifikācijas metodēm:

```
### 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
```

Ja nepieciešams, konfigurējot veidnes PostgreSQL by Zabbix agent 2 makrosu savienojuma virknei, varat arī tieši izmantot IPv4 adresi (`127.0.0.1`).

Nejauša EPEL Zabbix pakotņu instalēšana

Ja EPEL repozitorijs ir instalēts un iespējots, instalējot Zabbix pakotnes, oficiālo Zabbix pakotņu vietā var tikt instalētas EPEL versijas. Lai to novērstu:

1. Noņemiet visas no EPEL instalētās Zabbix pakotnes:

```
dnf remove zabbix-server-mysql
```

2. Izslēdziet Zabbix pakotnes no EPEL, pievienojot šādu rindu failam `/etc/yum.repos.d/epel.repo`:

```
[epel]
...
excludepkgs=zabbix*
```

3. Atkārtoti instalējiet oficiālo Zabbix serveris pakotni:

```
dnf install zabbix-server-mysql
```

Instalēšanas laikā oficiālo Zabbix pakotņu versijas virknē ir iekļauts vārds `release` (piemēram, `7.0.0-release1.el8`), kas tās atšķir no EPEL pakotnēm.

Zabbix pakotnes RHEL sistēmām Red Hat UBI vidēs

Instalējot Zabbix no Red Hat Enterprise Linux pakotnēm Red Hat Universal Base Image vidēs, pārliecinieties, ka ir piekļuve nepieciešamajiem repozitorijiem un atkarībām. Zabbix pakotnes ir atkarīgas no `libOpenIPMI.so` un `libOpenIPMIposix.so` bibliotēkām, kuras nenodrošina neviena pakotne noklusējuma pakotņu pārvaldnieka repozitorijos, kas ir iespējoti UBI sistēmās, un tas izraisīs instalēšanas kļūmes.

`libOpenIPMI.so` un `libOpenIPMIposix.so` bibliotēkas ir pieejamas `OpenIPMI-libs` pakotnē, ko nodrošina `redhat-#-for-<arch>-ap` repozitorijs. Piekļuvi šim repozitorijam regulē abonementi, kas UBI vidēs tiek nodoti, montējot RHEL hosta repozitoriju konfigurācijas un slepeno datu direktorijus konteinera failu sistēmas nosaukumvietā.

Papildinformāciju skatiet ZBX-24291.

Beigusies RHEL pakotņu parakstīšanas atslēgas derīguma termiņš

Jauninot Zabbix uz Red Hat Enterprise Linux vai tā atvasinājumiem, var rasties problēma ar beigušos pakotņu parakstīšanas atslēgas derīguma termiņu Zabbix repozitorijā. Kad parakstīšanas atslēgas derīguma termiņš beidzas, mēģinājumi pārbaudīt pakotņu parakstus izraisa kļūdu, kas norāda, ka sertifikāts vai atslēga vairs nav derīga. Piemēram:

```
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
```

Lai novērstu šādas problēmas, manuāli pārinstalējiet jaunāko `zabbix-release` pakotni savai konkrētajai RHEL versijai (aizstājiet tālāk norādīto saiti ar pareizo no Zabbix repozitorija).

Piemēram, uz **RHEL 10** izpildiet:

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

Pēc tam atjauniniet repozitorija informāciju:

```
dnf update
```

Papildinformāciju skatiet ZBX-24761.

Timescale DB: liels atmiņas patēriņš ar lielu nodalījumu skaitu

PostgreSQL versijas 9.6-12 izmanto pārāk daudz atmiņas, atjauninot tabulas ar lielu nodalījumu skaitu. Šī problēma izpaužas, kad Zabbix atjaunina trendus sistēmās ar TimescaleDB, ja trendi ir sadalīti salīdzinoši mazos (piemēram, 1 dienas) gabalos. Tas noved pie simtiem gabalu trends tabulās ar noklusējuma housekeepinga iestatījumiem — apstākļos, kuros PostgreSQL, visticamāk, pietrūks atmiņas.

Problēma ir novērsta kopš Zabbix 5.0.1 jaunām instalācijām ar TimescaleDB, bet, ja TimescaleDB tika iestatīts ar Zabbix pirms tam, lūdzu, skatiet ZBX-16347 migrācijas piezīmes.

Timescale DB 2.5.0: saspiešanas politika var neizdoties tabulām, kas satur veselus skaitļus

Šī problēma izpaužas, ja tiek izmantots TimescaleDB 2.5.0/2.5.1. Tā ir novērsta kopš TimescaleDB 2.5.2.

Papildinformāciju skatiet TimescaleDB Issue #3773.

Datubāzes TLS savienojums ar MariaDB

Datubāzes TLS savienojums netiek atbalstīts ar opciju 'verify_ca' DBTLSConnect parametram, ja tiek izmantots MariaDB.

Iespējamās strupceļa situācijas ar MySQL/MariaDB

Darbojoties augstas slodzes apstākļos un ja ir iesaistīts vairāk nekā viens LLD darbinieks, ir iespējams saskarties ar strupceļa situāciju, ko izraisa InnoDB kļūda, kas saistīta ar rindu bloķēšanas stratēģiju (skatiet upstream bug). Kļūda ir novērsta MySQL kopš 8.0.29 versijas, bet ne MariaDB. Papildinformāciju skatiet ZBX-21506.

Globālā notikumu korelācija

Notikumi var netikt korekti korelēti, ja laika intervāls starp pirmo un otro notikumu ir ļoti mazs, t.i., puse sekundes vai mazāk.

NetBSD 8.0 un jaunākas versijas

Dažādi Zabbix procesi var nejauši avarēt startēšanas laikā NetBSD 8.X un 9.X versijās. Tas notiek pārāk maza noklusējuma steka izmēra (4MB) dēļ, kas jāpalielina, izpildot:

```
ulimit -s 10240
```

Papildinformāciju skatiet saistītajā problēmas ziņojumā: ZBX-18275.

Regulāro izteiksmju ierobežojumi Zabbix aģents 2

Zabbix aģents 2 neatbalsta lookahead un lookbehind konstrukcijas regulārajās izteiksmēs standarta Go regexp bibliotēkas ierobežojumu dēļ.

IPMI pārbaudes

IPMI pārbaudes nedarbosies ar standarta OpenIPMI bibliotēkas pakotni Debian versijās pirms 9 (stretch) un Ubuntu versijās pirms 16.04 (xenial). Lai to novērstu, pārkompilējiet OpenIPMI bibliotēku ar iespējotu OpenSSL, kā aprakstīts ZBX-6139.

IPMI — neuzticami hosti var izraisīt OpenIPMI avāriju

OpenIPMI bibliotēkā, ko Zabbix izmanto IPMI datu aptaujāšanai, ir kļūda, kuru var aktivizēt ar īpaši izveidotām atbildēm no neuzticamas ierīces.
Neuzticama IPMI ierīce var nosūtīt īpaši izveidotus datus, kas izraisa OpenIPMI bibliotēkas avāriju, un tas savukārt var izraisīt tā Zabbix servera procesa pārtraukšanu, kas veic IPMI aptaujāšanu.

SSH pārbaudes

- Dažas Linux distribūcijas, piemēram, Debian un Ubuntu, neatbalsta šifrētas privātās atslēgas (ar paroli), ja libssh2 bibliotēka ir instalēta no pakotnēm. Lūdzu, skatiet ZBX-4850, lai iegūtu sīkāku informāciju.
- Izmantojot libssh 0.9.x dažās Linux distribūcijās ar OpenSSH 8, SSH pārbaudes reizēm var ziņot "Cannot read data from SSH server". To izraisa libssh problēma (detalizētāks ziņojums). Paredzams, ka kļūda ir novērsta stabilajā libssh 0.9.5 laidienā. Sīkāku informāciju skatiet arī ZBX-17756.
- Caurules simbola "|" izmantošana SSH skriptā var izraisīt kļūdu "Cannot read data from SSH server". Šādā gadījumā ieteicams atjaunināt libssh bibliotēkas versiju. Sīkāku informāciju skatiet arī ZBX-21337.

ODBC pārbaudes

- MySQL unixODBC draiveri nevajadzētu izmantot ar Zabbix serveri vai Zabbix starpniekserveri, kas kompilēts pret MariaDB savienotāja bibliotēku, un otrādi; ja iespējams, labāk arī izvairīties no tā paša savienotāja izmantošanas kā draivera augšupplūsmas kļūdas dēļ. Ieteicamā konfigurācija:

  PostgreSQL, SQLite vai Oracle savienotājs → MariaDB vai MySQL unixODBC draiveris
MariaDB savienotājs → MariaDB unixODBC draiveris
MySQL savienotājs → MySQL unixODBC draiveris

  Plašāku informāciju un pieejamos apiešanas risinājumus skatiet ZBX-7665.

- XML dati, kas vaicāti no Microsoft SQL Server, Linux un UNIX sistēmās var tikt dažādos veidos saīsināti.

- Ir novērots, ka ODBC pārbaužu izmantošana Oracle datubāzu uzraudzībai, izmantojot dažādas Oracle Instant Client for Linux versijas, izraisa Zabbix servera avāriju.
 Skatiet arī: ZBX-18402, ZBX-20803.

- Ja izmantojat FreeTDS UnixODBC draiveri, SQL vaicājumam jāpievieno priekšā priekšraksts 'SET NOCOUNT ON' (piemēram, `SET NOCOUNT ON DECLARE @strsql NVARCHAR(max) SET @strsql = ....`). Pretējā gadījumā datubāzes uzraudzības vienums Zabbix nevarēs izgūt informāciju un tiks parādīta kļūda "SQL query returned empty result".
 Plašāku informāciju skatiet ZBX-19917.

Nepareizs pieprasījuma metodes parametrs vienumos

Pieprasījuma metodes parametrs, kas tiek izmantots tikai HTTP pārbaudēs, pēc jauninašanas no Zabbix versijas, kas ir vecāka par 4.0, visiem vienumiem var būt nepareizi iestatīts uz '1', kas ir nestandarta vērtība. Lai iegūtu informāciju par to, kā novērst šo situāciju, skatiet ZBX-19308.

Tīmekļa uzraudzība un HTTP aģents

Dažās Linux distribūcijās Zabbix serveris izraisa atmiņas noplūdi augšupplūsmas kļūdas dēļ, ja tīmekļa scenārijos vai HTTP aģentā ir iespējota opcija "SSL verify peer". Lūdzu, skatiet ZBX-10486, lai iegūtu vairāk informācijas un pieejamos risinājumus.

Vienkāršās pārbaudes

fping versijās, kas ir vecākas par v3.10, ir kļūda, kuras dēļ tiek nepareizi apstrādātas dublētas echo reply paketes. Tas var izraisīt negaidītus rezultātus `icmpping`, `icmppingloss`, `icmppingsec` vienumiem. Ieteicams izmantot jaunāko **fping** versiju. Lūdzu, skatiet ZBX-11726, lai iegūtu sīkāku informāciju.

Kļūdas ar fping izpildi rootless konteineros

Ja konteineri darbojas rootless režīmā vai vidē ar specifiskiem ierobežojumiem, veicot ICMP pārbaudes, var rasties ar fping izpildi saistītas kļūdas, piemēram, `fping: Operation not permitted`, vai arī var tikt zaudētas visas paketes uz visiem resursiem.

Lai novērstu šo problēmu, pievienojiet `--cap-add=net_raw` komandām "docker run" vai "podman run".

Papildus tam fping izpilde vidēs bez root tiesībām var prasīt sysctl modifikāciju, piemēram:

```
sudo sysctl -w "net.ipv4.ping_group_range=0 1995"
```

kur "1995" ir zabbix GID. Plašāku informāciju skatiet ZBX-22833.

SNMP pārbaudes

Ja tiek izmantota OpenBSD operētājsistēma, use-after-free kļūda Net-SNMP bibliotēkā līdz 5.7.3 versijai var izraisīt Zabbix serveris avāriju, ja Zabbix serveris konfigurācijas failā ir iestatīts parametrs SourceIP. Kā pagaidu risinājumu, lūdzu, neiestatiet parametru SourceIP. Tā pati problēma attiecas arī uz Linux, taču tā neizraisa Zabbix serveris darbības pārtraukšanu. Lokāls labojums net-snmp pakotnei OpenBSD sistēmā tika ieviests un tiks izlaists kopā ar OpenBSD 6.3.

SNMP datu pīķi

Ir novēroti SNMP datu pīķi, kas var būt saistīti ar noteiktiem fiziskiem faktoriem, piemēram, sprieguma lēcieniem elektrotīklā. Plašāku informāciju skatiet ZBX-14318.

SNMP trapi

Pakotne "net-snmp-perl", kas nepieciešama SNMP trapiem, ir noņemta RHEL 8.0-8.2; atkārtoti pievienota RHEL 8.3.

Tāpēc, ja izmantojat RHEL 8.0-8.2, labākais risinājums ir jaunināt uz RHEL 8.3.

Lūdzu, skatiet arī ZBX-17192, lai iegūtu vairāk informācijas.

Alerter procesa avārija RHEL 7

RHEL 7 vidē ir konstatēti Zabbix servera alerter procesa avārijas gadījumi. Lūdzu, skatiet ZBX-10461, lai iegūtu sīkāku informāciju.

Zabbix aģenta 2 jauninašana (6.0.5 vai vecāka versija)

Jauninot Zabbix aģentu 2 (versiju 6.0.5 vai vecāku) no pakotnēm, var rasties ar spraudni saistīta failu konflikta kļūda. Lai novērstu kļūdu, izveidojiet sava aģenta 2 konfigurācijas dublējumkopiju (ja nepieciešams), atinstalējiet aģentu 2 un instalējiet to no jauna.

Sistēmās, kuru pamatā ir RHEL, izpildiet:

```
dnf remove zabbix-agent2
dnf install zabbix-agent2
```

Sistēmās, kuru pamatā ir Debian, izpildiet:

```
apt remove zabbix-agent2
apt install zabbix-agent2
```

Papildinformāciju skatiet ZBX-23250.

Lietotāja saskarnes lokalizāciju pārslēgšanās

Ir novērots, ka lietotāja saskarnes lokalizācijas var pārslēgties bez acīmredzamas loģikas, t. i., dažas lapas (vai lapu daļas) tiek attēlotas vienā valodā, kamēr citas lapas (vai lapu daļas) — citā valodā. Parasti problēma var parādīties, ja ir vairāki lietotāji, no kuriem daži izmanto vienu lokalizāciju, bet citi — citu.

Zināms pagaidu risinājums ir atspējot daudzpavedienu darbību PHP un Apache.

Problēma ir saistīta ar to, kā lokalizācijas iestatīšana darbojas in PHP: lokalizācijas informācija tiek uzturēta katram procesam, nevis katram pavedienam. Tāpēc daudzpavedienu vidē, ja vienā un tajā pašā Apache procesā darbojas vairāki projekti, ir iespējams, ka lokalizācija tiek mainīta citā pavedienā, un tas ietekmē to, kā dati tiek apstrādāti Zabbix pavedienā.

Plašāku informāciju skatiet saistītajos problēmu ziņojumos:

- ZBX-10911 (Problēma ar lietotāja saskarnes lokalizāciju pārslēgšanos)
- ZBX-16297 (Problēma ar skaitļu apstrādi grafikos, izmantojot BC Math funkciju `bcdiv`)

Grafiki

Grafiku (klasisko) problēmas

Ja saskaraties ar problēmām ar klasiskajiem grafikiem, ieteicams atjaunināt GD bibliotēku (libgd) uz versiju 2.3.3-13 vai jaunāku, un PHP uz versiju 8.0.19, 8.1.33, 8.2.29, 8.3.25, 8.4.12 vai jaunāku.

Vasaras laiks

Pārejas uz vasaras laiku (DST) izmaiņas izraisa neatbilstības X ass etiķešu attēlošanā (datumu dublēšanās, datuma trūkums u.c.).

Summas agregācija

Izmantojot summas agregāciju grafikā periodam, kas ir mazāks par vienu stundu, grafikos tiek attēlotas nepareizas (reizinātas) vērtības, ja dati nāk no trendiem.

Teksta pārklāšanās

Dažām lietotāja saskarne valodām (piemēram, japāņu) lokālie fonti var izraisīt teksta pārklāšanos grafika leģendā. Lai no tā izvairītos, izmantojiet PHP GD paplašinājuma versiju 2.3.0 (vai jaunāku).

Žurnālfaila uzraudzība

`log[]` un `logrt[]` vienumi atkārtoti pārlasa žurnālfailu no sākuma, ja failu sistēma ir 100% pilna un žurnālfails tiek papildināts (vairāk informācijas skatiet ZBX-10884).

Lēni MySQL vaicājumi

Zabbix serveris ģenerē lēnus `SELECT` vaicājumus gadījumos, kad vienumiem nepastāv vērtības. Šī problēma ir zināma MySQL 5.6/5.7 versijās (plašāku apspriedi skatiet ZBX-10652), un noteiktos gadījumos tā var rasties arī jaunākās MySQL versijās. Kā pagaidu risinājumu var atspējot MySQL optimizatoru `index_condition_pushdown` vai `prefer_ordering_index`. Tomēr ņemiet vērā, ka šis pagaidu risinājums var nenovērst visas ar lēniem vaicājumiem saistītās problēmas.

Pastāvīgie filtra iestatījumi no saitēm

Atverot saiti uz Zabbix lietotāja saskarnes lapu, kas satur filtra iestatījumus, tostarp laika atlasītāju, filtrs tiek automātiski saglabāts datubāzē attiecīgajam lietotājam, aizstājot iepriekš saglabāto filtru un/vai laika atlasītāja iestatījumus šai lapai. Šie iestatījumi paliek aktīvi, līdz lietotājs tos manuāli atjaunina vai atiestata.

IPv6 adreses problēma SNMPv3 slazdos

Net-SNMP kļūdas dēļ IPv6 adrese var netikt attēlota pareizi, izmantojot SNMPv3 SNMP slazdos. Plašāku informāciju un iespējamo apiešanas risinājumu skatiet ZBX-14541.

Saīsināta gara IPv6 IP adrese neveiksmīgas pieteikšanās informācijā

Ziņojumā par neveiksmīgu pieteikšanās mēģinājumu tiks parādītas tikai pirmās 39 saglabātās IP adreses rakstzīmes, jo tas ir rakstzīmju ierobežojums datubāzes laukā. Tas nozīmē, ka IPv6 IP adreses, kas ir garākas par 39 rakstzīmēm, tiks parādītas nepilnīgi.

Zabbix aģenta pārbaudes operētājsistēmā Windows

Nepastāvoši DNS ieraksti Zabbix aģenta konfigurācijas faila (zabbix_agentd.conf) parametrā `Server` var palielināt Zabbix aģenta atbildes laiku operētājsistēmā Windows. Tas notiek tāpēc, ka Windows DNS kešošanas dēmons nekešo negatīvās atbildes IPv4 adresēm. Tomēr IPv6 adresēm negatīvās atbildes tiek kešotas, tāpēc viens no iespējamiem risinājumiem ir atspējot IPv4 hostā.

YAML eksports/imports

Ir zināmas dažas problēmas ar YAML eksportu/importu:

- Kļūdu ziņojumi nav tulkojami;
- Derīgu JSON ar .yaml faila paplašinājumu dažkārt nevar importēt;
- Cilvēkam lasāmi datumi bez pēdiņām tiek automātiski pārveidoti par Unix laikspiedoliem.

Iestatīšanas vednis SUSE vidē ar NGINX un php-fpm

Lietotāja saskarnes iestatīšanas vednis nevar saglabāt konfigurācijas failu SUSE vidē ar NGINX + php-fpm. To izraisa iestatījums vienībā /usr/lib/systemd/system/php-fpm.service, kas neļauj Zabbix rakstīt mapē /etc. (ieviests PHP 7.4).

Ir pieejamas divas apiešanas iespējas:

- Iestatīt ProtectSystem opciju uz 'true', nevis 'full' php-fpm systemd vienībā.
- Manuāli saglabāt failu /etc/zabbix/web/zabbix.conf.php.

Authorization galvenes pārsūtīšana

Dažos gadījumos Apache vai NGINX var neļaut API pieprasījumos esošajai Authorization galvenei sasniegt Zabbix. Tas var izraisīt autentifikācijas problēmas, izmantojot Zabbix API vai vienotās pierakstīšanās (SSO) pakalpojumus, piemēram, SAML ar Okta.

Lai to novērstu, atjauniniet sava tīmekļa servera konfigurāciju.

Apache gadījumā, ja to izmantojat kā reverso starpniekserveri (ne-CGI iestatījumā), pievienojiet šādu direktīvu failam `/etc/httpd/conf/httpd.conf` (uz RHEL bāzētās sistēmās) vai `/etc/apache2/apache2.conf` (uz Debian/Ubuntu):

```
SetEnvIfNoCase ^Authorization$ "(.+)" HTTP_AUTHORIZATION=$1
```

Ja Apache tieši izpilda skriptus pieprasījumu apstrādei (piemēram, izmantojot mod_cgi), tā vietā pievienojiet šādu direktīvu:

```
CGIPassAuth On
```

Savukārt **NGINX** Authorization galveni apstrādā automātiski. Tomēr, ja NGINX darbojas kā reversais starpniekserveris, varat skaidri pārsūtīt Authorization galveni, pievienojot šādas direktīvas failam `/etc/nginx/nginx.conf` (jūsu Zabbix lietotāja saskarne atrašanās vietai):

```
...
location / {
...
    proxy_set_header Authorization $http_authorization;
    proxy_pass http://backend_server;
...
}
```

Pēc konfigurācijas atjaunināšanas restartējiet savu tīmekļa serveri.

Plašāku informāciju skatiet:

- ZBX-22952
- Apache 2.4 + PHP-FPM and Authorization headers
- SetEnvIfNoCase un CGIPassAuth direktīvas
- NGINX Reverse Proxy

Chromium Zabbix tīmekļa servisam uz Ubuntu 20

Lai gan vairumā gadījumu Zabbix tīmekļa serviss var darboties ar Chromium, Ubuntu 20.04 vidē Chromium izmantošana izraisa šādu kļūdu:

```
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
```

Šī kļūda rodas tāpēc, ka `/var/lib/zabbix` tiek izmantots kā lietotāja 'zabbix' mājas direktorijs.

MySQL pielāgotie kļūdu kodi

Kad Zabbix konstatē, ka aizmugursistēmas datubāze nav pieejama, tas nosūta paziņojumu un turpina mēģināt izveidot savienojumu. Noteiktiem datubāzu dzinējiem tiek atpazīti specifiski kļūdu kodi. MySQL gadījumā šie atpazītie kļūdu kodi ietver:

- 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

Turklāt, izmantojot Zabbix ar MySQL instalāciju Azure vidē, Zabbix žurnālos var parādīties vispārīgs kļūdas ziņojums *[9002] Some errors occurred*. Šo ziņojumu datubāze nosūta Zabbix serverim vai starpniekserverim. Lai noteiktu kļūdas cēloni, lūdzu, skatiet Azure žurnālus.

Nederīgas regulārās izteiksmes pēc pārslēgšanās uz PCRE2

Zabbix 6.0 versijā ir pievienots PCRE2 atbalsts. Lai gan PCRE joprojām tiek atbalstīts, Zabbix instalācijas pakotnes RHEL 7 un jaunākām versijām, SLES (visām versijām), Debian 9 un jaunākām versijām, Ubuntu 16.04 un jaunākām versijām ir atjauninātas, lai izmantotu PCRE2. Lai gan tas sniedz daudzas priekšrocības, pāreja uz PCRE2 var izraisīt to, ka noteikti esošie PCRE regexp šabloni kļūst nederīgi vai darbojas citādi. Jo īpaši tas attiecas uz šablonu *^[\w-\.]*. Lai šo regexp atkal padarītu derīgu, neietekmējot semantiku, nomainiet izteiksmi uz *^[-\w\.]* . Tas notiek tāpēc, ka PCRE2 domuzīmes zīmi apstrādā kā atdalītāju, izveidojot diapazonu rakstzīmju klasē.

Geomap logrīka kļūda

Kartes Geomap logrīkā var netikt ielādētas pareizi, ja esat veicis jaunināšanu no vecākas Zabbix versijas ar NGINX un jaunināšanas laikā nepārslēdzāties uz jauno NGINX konfigurācijas failu.

Lai novērstu šo problēmu, varat atmest veco konfigurācijas failu, izmantot konfigurācijas failu no pašreizējās versijas pakotnes un pārkonfigurēt to, kā aprakstīts lejupielādes instrukcijās sadaļā *e. Configure PHP for Zabbix frontend*.

Alternatīvi, varat manuāli rediģēt esošu NGINX konfigurācijas failu (parasti, */etc/zabbix/nginx.conf*). Lai to izdarītu, atveriet failu un atrodiet šādu bloku:

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

Pēc tam aizstājiet šo bloku ar:

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

location /vendor {
        deny            all;
        return          404;
}
```

Priekšapstrāde — globālie mainīgie nav droši

Priekšapstrādes JavaScript tiek izpildīts katram pieprasījumam atsevišķi, taču piešķīrumi nedeklarētiem identifikatoriem (piemēram, `secret = value`) izveido netiešus globālos mainīgos, kas var saglabāties arī pēc pašreizējās izpildes. Sensitīvu datu (tokenu, paroļu u.c.) glabāšana netiešos globālajos mainīgajos palielina nejaušas atklāšanas vai atkārtotas izmantošanas risku nākamajās priekšapstrādes izpildēs vai citās integrācijās, kas darbojas tajā pašā vidē.

Nepaļaujieties uz netiešiem globālajiem mainīgajiem. Vienmēr deklarējiet mainīgos ar `var` vai `const`, un nepiesaistiet noslēpumus globāliem objektiem (piemēram, `globalThis` vai `window`). Nav atbalstīta veida, kā no priekšapstrādes pārrakstīt iebūvētos globālos objektus.

Drošs piemērs:

```
var apiToken = payload.token;
var count = 1;
return JSON.stringify({ token: apiToken, calls: count });
```

Procesoru grupas operētājsistēmā Windows

Microsoft dokumentācijā ir norādīts, ka sistēmām ar mazāk nekā 64 loģiskajiem procesoriem vienmēr ir tikai viena procesoru grupa — 0. grupa. Tomēr Zabbix lietotāji ir ziņojuši par retu kļūdu ZBX-20260, kad sistēmās ar 64 vai mazāk loģiskajiem procesoriem ir divas procesoru grupas. Tā rezultātā "\Processor(n)" veiktspējas skaitītāji bija pieejami tikai vienai no divām procesoru grupām. Šīs kļūdas patiesais pamatcēlonis nav zināms. Tomēr līdzīgs gadījums ir aprakstīts vietnē stackoverflow.com, un tur pamatcēlonis bija BIOS un Windows savstarpējā mijiedarbība.

Filtrēšanas ierobežojumi ar utf8mb4 kolācijām

Filtri (piemēram, sadaļā *Datu ievākšana* > *Uzturēšana*) var nedarboties pareizi, ja tie tiek piemēroti entītijām, kas satur noteiktas unikoda rakstzīmes (piemēram, 𝒸, 𝓅). Šī problēma rodas tādēļ, ka MySQL vai MariaDB datubāzēs noklusējuma utf8mb4_bin kolācija apstrādā unikoda rakstzīmju kārtošanu un salīdzināšanu noteiktā veidā.

Lai novērstu šo ierobežojumu, lietotāji var mainīt datubāzes kolonnu kolāciju uz alternatīvām, piemēram, utf8mb4_0900_bin, utf8mb4_0900_ai_ci vai utf8mb4_unicode_520_ci. Tomēr ņemiet vērā, ka kolācijas maiņa var izraisīt neparedzētu darbību tukšo atstarpju apstrādē, kā arī citu rakstzīmju kārtošanā un filtrēšanā.

Plašāku informāciju par kolāciju maiņu skatiet MySQL documentation vai MariaDB documentation. Informāciju par kolāciju atšķirībām skatiet Unicode Character Sets MySQL dokumentācijā.

Piekļuve lietotāja saskarnes elementiem ar MariaDB 10.5.1–10.5.9

Piekļūstot Zabbix tīmekļa lietotāja saskarnei ar lomu, kas nav Super Admin, var tikt parādīts ziņojums: "Radās sistēmas kļūda. Lūdzu, sazinieties ar Zabbix administratoru.". Šī problēma ietekmē instalācijas, kurās tiek izmantotas MariaDB versijas no 10.5.1 līdz 10.5.9.

Lai izvairītos no šīs problēmas, atjauniniet MariaDB uz versiju, kas ir jaunāka par 10.5.9. Papildinformāciju skatiet ZBX-25746.

Pārmērīga atmiņas lietojuma profilēšana ar tcmalloc

Ja jums ir aizdomas, ka jūsu Zabbix instalācija izmanto pārāk daudz atmiņas, varat izmantot tcmalloc atmiņas profilēšanas iespēju, lai izmeklētu Zabbix servera/starpniekservera atmiņas patēriņu.

1. Instalējot Zabbix no pirmkoda, norādiet papildu karogus:

```
export CFLAGS="-std=gnu99 -g -O0"
```

Karogs `-std=gnu99` ir nepieciešams, lai būvētu Zabbix serveri, Zabbix starpniekserveri vai Zabbix aģentu. Karogs `-g` pievieno papildu atkļūdošanas informāciju, savukārt `-O0` atspējo optimizācijas, kas var traucēt tcmalloc profilēšanai.

2. Pirms Zabbix servera palaišanas iestatiet šādus vides mainīgos. Šie mainīgie norāda tcmalloc, kā izsekot atmiņas lietojumu un par to ziņot:

```
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. Aktivizējiet profila izmetni, nosūtot mērķa procesam 5. signālu. Aizstājiet 1234 ar faktisko procesa ID (PID):

```
kill -5 1234
```

4. Izdrukājiet ģenerēto profilu:

```
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
```

Šajā piemērā zbx_malloc2 ir atbildīgs par gandrīz visām atmiņas alokācijām.

Skatiet arī:

- ZBX-25050 un ZBX-25584 saistītos problēmu ziņojumus.
- GCC Option Summary par kompilēšanas opcijām (`-std=gnu99`, `-g`, `-O0` u.c.).
- Gperftools Heap Profiler dokumentāciju par vides mainīgajiem tcmalloc profilēšanai.

MySQL 8.0 Group Replication vairāku primāro mezglu režīmā

Izmantojot MySQL 8.0 Group Replication vairāku primāro mezglu režīmā, darījumu apstiprināšanas laikā var rasties kļūda, kas ir līdzīga šādai:

```
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
```

Šī kļūda, šķiet, tiek izraisīta problēmu dēļ ar atcelšanas darbībām, kas saistītas ar ārējās atslēgas ierobežojumiem.

Skatiet arī:

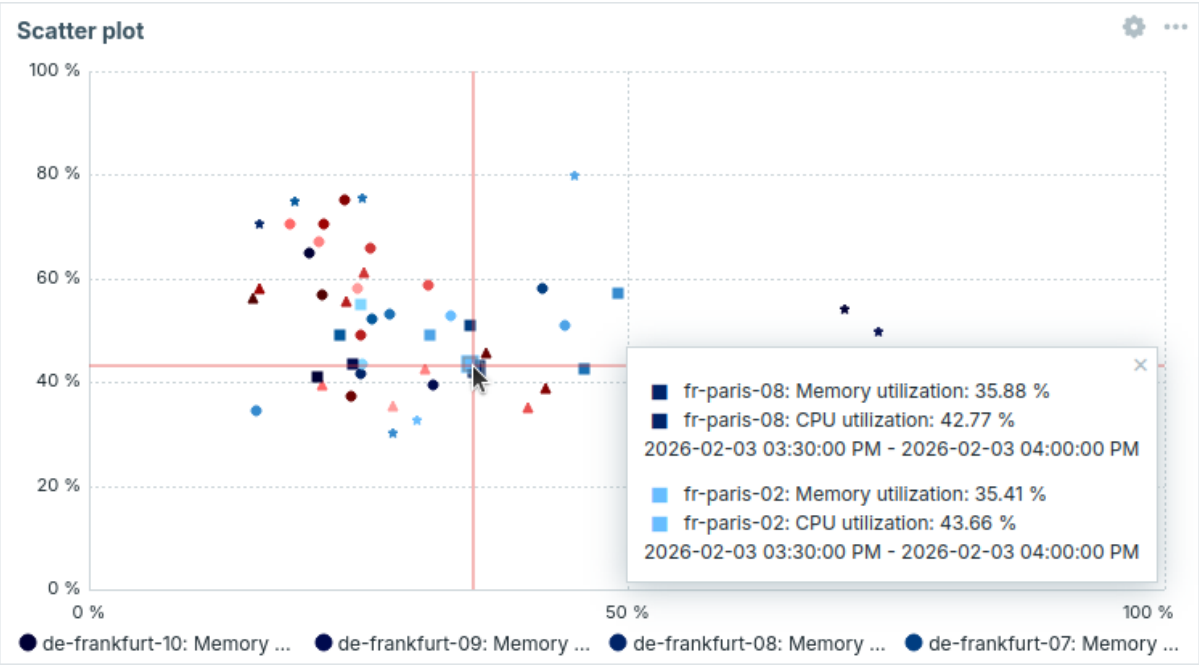
- ZBX-26060 saistītās problēmas ziņojumam.
- MySQL Bug #96758 "Rollbacks with Foreign Keys on single node" augšupējai problēmai.

2 Kas jauns Zabbix 8.0

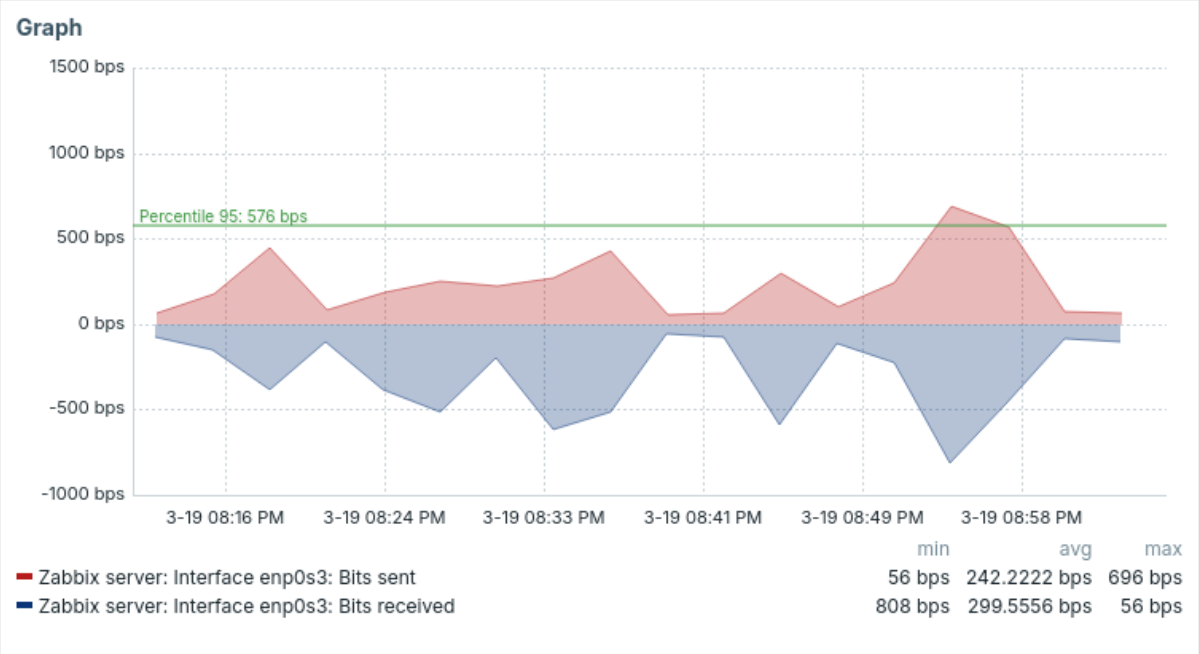
Zabbix 8.0.0 turpina Zabbix 7.4.0 versiju, pievienojot jaunas iespējas un uzlabojumus.

Skatiet šīs versijas nesaderīgās izmaiņas.

Izkliedes diagrammas logrīks Izkliedes diagrammas logrīks ir pievienots paneļa logrīkiem. Tas attēlo saistību starp diviem rādītājiem, izvietojot atsevišķus datu punktus gar X un Y asi. Tas palīdz atklāt datu kopā modeļus, kopas, korelācijas un izņēmuma vērtības.



Y ass apvēršana grafika logrīkam Grafikus grafika logrīkā tagad var attēlot ar apvērstām Y ass vērtībām. Jauns iestatījums *Apvērst vērtības* ļauj reizināt Y ass vērtības ar −1, nemainot sākotnējos datus.



Skatiet arī: Grafika logrīka uzlabojumi.

Veidnes

Jaunas veidnes

- Aruba CX 8300s by SNMP, veidne, kas nodrošina uz SNMP balstītu Aruba CX 8300 komutatoru sērijas uzraudzību.
- Veidņu kopa AWS by HTTP ir papildināta ar veidni *AWS Backup Vault by HTTP*.
- Veidņu kopa Azure by HTTP ir papildināta ar veidni *Azure Sentinel by HTTP*.
- Ciena 3906 by SNMP, veidne Ciena 3906 ierīču uzraudzībai.
- Cisco Secure Firewall Threat Defense by HTTP, veidne, kas nodrošina Cisco Secure Firewall Threat Defense ierīču uzraudzības iespējas, izmantojot REST API.
- Cradlepoint NCM v2 by HTTP, veidņu kopa Cradlepoint NCM v2 un tā ierīču uzraudzībai, izmantojot HTTP.
- Domain RDAP by HTTP, veidne domēna reģistrācijas datu uzraudzībai, izmantojot RDAP protokolu.
- Huawei AR600 by SNMP, veidne Huawei AR600 maršrutētāju uzraudzībai, tostarp CoS un NQA metrikām.
- IBM Maximo Service Request, uz webhook balstīta veidne, kas ļauj sasaistīt Zabbix ar IBM Maximo.
- MariaDB by ODBC, veidne MariaDB datubāzu uzraudzībai, izmantojot ODBC.
- Microsoft Hyper-V Failover Cluster by SSH un Microsoft Hyper-V Standalone by SSH, veidnes Microsoft Hyper-V klasteru un atsevišķu hostu uzraudzībai, izmantojot SSH.
- OpenAI Platform by HTTP, veidne OpenAI izstrādātāju platformas uzraudzībai.
- Percona by ODBC, veidne Percona datubāzu uzraudzībai, izmantojot ODBC.
- Ribbon SBC Edge by HTTP, veidne Ribbon SBC Edge (iepriekš SWe Lite) ierīču uzraudzībai, izmantojot HTTP.
- Ribbon SBC SWe Core by HTTP, veidne SBC SWe Core ierīču uzraudzībai, izmantojot HTTP, un Ribbon SBC SWe CE by HTTP, veidne Ribbon SBC SWe Call Engine (CE) instanču uzraudzībai, izmantojot HTTP.
- Stormshield SNS by SNMP, veidne Stormshield Network Security (SNS) ierīču uzraudzībai, izmantojot SNMP.
- VeloCloud SD-WAN Edge by HTTP, veidne VeloCloud SD-WAN Edge ierīču uzraudzībai, izmantojot HTTP.
- Vyatta Virtual Router by SNMP, veidne Vyatta 1908e virtuālā maršrutētāja uzraudzībai.

Atjauninātas veidnes

- Ciena 3906 by SNMP ir atjaunināta, iekļaujot failu sistēmas un CPU slodzes vienumus.
- GCP by HTTP veidņu kopa ir papildināta ar veidni *GCP Cloud Run Service by HTTP*.
- GitHub organization by HTTP ir atjaunināta, iekļaujot vienumus Microsoft Copilot uzraudzībai.
- GLPI webhook media type ir atjaunināts, lai atbalstītu RESTful API v2 ar OAuth2 autorizāciju (iespējota pēc noklusējuma), izvēles mantotā REST API v1 atbalstu, izmantojot konfigurācijas karogu, biļešu izveidi problēmu notikumiem un lietotāja konfigurējamu steidzamības kartēšanu.
- Microsoft 365 reports by HTTP ir atjaunināta, iekļaujot vienumus Microsoft Copilot uzraudzībai.
- MySQL by Zabbix agent, MySQL by Zabbix agent 2, MySQL by Zabbix agent active, MySQL by Zabbix agent 2 active un MySQL by ODBC ir atjauninātas, lai atbalstītu gan `SHOW SLAVE STATUS` (vecā sintakse), gan `SHOW REPLICA STATUS` (jaunā sintakse).
- MySQL by ODBC ir atjaunināta arī ar jaunām metrikām, tabulu un repliku atklāšanas kārtulām, kā arī uzlabotu paneli.
- PostgreSQL by ODBC ir atjaunināta ar jaunām metrikām, atklāšanas kārtulām un paneļiem, kā arī vispārīgiem veiktspējas un novērojamības uzlabojumiem.

- Proxmox VE by HTTP ir atjaunināta ar ligzdotās LLD funkcionalitāti. Turklāt skaidrības labad ir pielāgots vienību formāts vienumiem, kas attēlo procentu vērtības.
- VeloCloud SD-WAN by HTTP, iepriekš *VMWare SD-WAN VeloCloud by HTTP*, ir pārdēvēta un atjaunināta, lai atbilstu jaunākajai VeloCloud SD-WAN platformai.
- Vyatta Virtual Router by SNMP ir atjaunināta ar jauniem OID vienumiem un paneļa uzlabojumiem.

Vienumi

JSON datu tips Zabbix tagad atbalsta JSON kā datu tipu vienumu vērtībām.

Iepriekš JSON vērtības tika ievāktas ar teksta vienumiem un glabātas kā virknes ar 64KB ierobežojumu. Tagad Zabbix var glabāt JSON vērtības natīvi ar 128MiB ierobežojumu, kā arī noraidīt nederīgas JSON vērtības (piemēram, ar neiekļautām atslēgām pēdiņās, liekiem komatiem beigās, neatbilstošām iekavām).

JSON datu tips tiek atbalstīts visiem vienumu tipiem un vienumu prototipiem (izņemot *Calculated*), un ir pieejams reāllaika datu eksportā un savienotājos. JSON vērtības var glabāt visās atbalstītajās datubāzēs un Elasticsearch. Ja izmantojat TimescaleDB, lūdzu, skatiet jauninašanas piezīmes.

Ņemiet vērā, ka JSON vienumus nevar izmantot trigeros; tomēr varat izvilkt JSON laukus ar atkarīgajiem vienumiem, kuriem ir datu tips, kas nav JSON, un izmantot tos trigeros.

Vienumi, kas atgriež JSON virkni (`net.if.discovery`, `vfs.file.get` u.c.), joprojām ir teksta vienumi; tomēr, ja nepieciešams, varat tos mainīt uz JSON.

Plašāku informāciju un JSON datu ierobežojumus skatiet vienuma konfigurācijā.

Parametrs type S.M.A.R.T. disku atklāšanai Vienums smart.disk.discovery (Zabbix aģents 2 S.M.A.R.T. spraudnis) tagad pieņem neobligātu parametru **type**, lai norādītu vērtību, pēc kuras skenēt diskus.

Spraudņi

Ceph spraudnis Šis spraudnis tagad darbojas divos režīmos:

- **native** — šajā režīmā tiek izmantota go-ceph bibliotēka, lai sazinātos tieši ar Ceph klasteri, izmantojot vietējo Ceph API (msgr2 protokolu). Šis ir ieteicamais režīms modernām Ceph instalācijām, taču tas tiek atbalstīts tikai **Linux** vidē un начиная ar Ceph 16.
- **restful** (novecojis) — šajā režīmā saziņai tiek izmantots Ceph RESTful API. Tas ir noklusējuma režīms atpakaļsaderībai, taču tas nedarbosies ar Ceph versiju 20 (Tentacle) vai jaunāku, jo ir noņemts mgr/restful modulis.

Izmantojamais režīms tiek noteikts pēc parametra mode vērtības (native/restful):

- `Plugins.Ceph.Default.Mode=native` — iestata spraudnim native režīmu
- `Plugins.Ceph.Sessions.<SessionName>.Mode=native` — iestata native režīmu nosauktajai sesijai

Ņemiet vērā, ka lietotāja akreditācijas datu kopa katram režīmam atšķiras, un tie nav savstarpēji saderīgi. Parametrs `Plugins.Ceph.InsecureSkipVerify` native režīmā tiek ignorēts, jo savienojuma drošība tiek definēta Ceph klastera pusē ar msgr2 protokolu (pēc noklusējuma drošs).

Ņemiet vērā, ka Ceph spraudnis Zabbix aģents 2 tagad ir ielādējams spraudnis un tam ir nepieciešamas papildu instalēšanas darbības. Tas ir saistīts ar librados pakotnes prasību (native režīmam). Sīkāku informāciju skatiet Ceph spraudņa readme.

MongoDB spraudnis Zabbix aģents 2 MongoDB spraudnis piedāvā vairākus uzlabojumus: - Pilns atbalsts standarta MongoDB URI parsēšanai, atbalstot gan `mongodb://`, gan `mongodb+srv://` shēmas. - Automātiska MongoDB repliku kopas mezglu atklāšana, uzlabojot iespējas uzraudzīt MongoDB klasterus. - x509 autentifikācijas atbalsts, kas ļauj izveidot drošus savienojumus ar MongoDB, izmantojot klienta sertifikātus.

Oracle spraudnis Zabbix aģents 2 Oracle spraudnis tagad atbalsta šifrētus savienojumus ar Oracle datubāzēm, izmantojot TCPS (TLS) protokolu. Tas nodrošina iespēju uzraudzīt Oracle instances, izmantojot ar TLS aizsargātas ligzdas, tādējādi uzlabojot attālās uzraudzības drošību.

Redis spraudnis — TLS atbalsts un validācija palaišanas laikā Redis spraudnim ir pievienots TLS atbalsts Zabbix aģentam 2.

Ir ieviesta spraudņa TLS konfigurācijas validācija palaišanas laikā un uzlaboti validācijas/kļūdu ziņojumi. Nederīga konfigurācijas loģika (piemēram: savienojuma tipa `verify_full` izmantošana, nenorādot `TLSCAFile`) var neļaut Zabbix aģentam 2 palaisties.

Ielādējamo spraudņu testa izpildes režīms Ielādējamie spraudņi tagad var tikt palaisti testa režīmā, izmantojot `-t` (`--test`) karogu un kā argumentu norādot vienuma atslēgu. Šajā režīmā spraudnis darbojas atkļūdošanas un izstrādes nolūkos, un spraudņa konfigurācijas faili tiek ignorēti.

Zema līmeņa atklāšana

Atzīmēšanas rūtiņa Konvertēt uz JSON atklāšanas formās Zema līmeņa atklāšanas kārtulas formā un atklāšanas prototipa formā ir pievienota jauna atzīmēšanas rūtiņa *Konvertēt uz JSON*, kas tiek parādīta, ja nolaižamajā sarakstā *Tips* ir atlasīts "HTTP aģents". Šī opcija ļauj atzīmēt saņemtos datus automātiskai konvertēšanai uz JSON pirms turpmākas apstrādes.

Makro atbalsts ligzdotai zema līmeņa atklāšanai Zema līmeņa atklāšanas makro tagad tiek atbalstīti ligzdotajos zema līmeņa atklāšanas noteikumos šādos gadījumos:

- JSONPath priekšapstrādes parametros
- JSONPath laukā pielāgotajiem LLD makro

Rediģējamas birkas trigeriem, kas izveidoti ar zema līmeņa atklāšanu Trigeriem, kas izveidoti no trigeru prototipiem, tagad var manuāli pievienot birkas. Birkas, kas mantotas no trigeru prototipiem, joprojām tiek piemērotas automātiski. Manuāli pievienotās birkas var tikt modificētas atklātajos trigeros, un tās tiks iekļautas notikumu birku masīvos un būs pieejamas funkcijām, kas izmanto birkas, piemēram, filtriem, paneļa logrīkiem un paziņojumu makrosiem.

Procesi

SNMPv3 EngineID kešošana un atkārtota izmantošana Zabbix tagad kešo SNMPv3 EngineID → IP kartējumus un mēģinās atkārtoti izmantot kešotos EngineID turpmākajām SNMPv3 pārbaudēm, samazinot zondēšanas datplūsmu un uzlabojot aptaujātāja veiktspēju. Ja atkārtoti izmantots EngineID neatbild, aptaujātājs atgriežas pie EngineID zondēšanas un var noņemt novecojušus ierakstus pēc saskarnes izmaiņām vai pastāvīgu kļūmju gadījumā.

Uzlabota starpniekservera droselēšana vēstures kešatmiņas atkopšanas laikā Starpniekservera droselēšanas loģika ir uzlabota, lai vēstures kešatmiņas atkopšanas laikā uzlabotu servera stabilitāti. Kad vēstures kešatmiņas lietojums sasniedz droselēšanas slieksni, serveris turpina pārtraukt starpniekservera datu pieņemšanu, kā iepriekš. Kad kešatmiņas lietojums samazinās līdz 60%, serveris sāk apstrādāt droselēšanas sarakstu, taču joprojām var noraidīt starpniekservera augšupielādes, kas satur ļoti lielas paketes (aptuveni vairāk nekā 10k ierakstu), līdz kešatmiņas noslodze vēl vairāk samazinās. Šīs izmaiņas samazina atkārtotas kešatmiņas pārslodzes risku, kamēr serveris atkopjas.

Palielināts maksimālais noildzes laiks utilītām zabbix_get un zabbix_js Komandrindas utilītu zabbix_get un zabbix_js parametra `timeout` maksimālā vērtība ir palielināta līdz 600 sekundēm.

Optimizēta uzturēšana Uzturēšana ir optimizēta, pārvietojot uzturēšanas uzdevumu izveidi uz datubāzes trigeriem. Kad tiek dzēsti vienumi (tostarp zema līmeņa atklāšanas kārtulas), trigeri, pakalpojumi vai tīkla atklāšanas kārtulas, datubāzes trigeri (nevis tieši lietojumprogrammas līmeņa izsaukumi) tagad aizpilda `housekeeper` tabulu ar tīrīšanas uzdevumiem. Sīkāku informāciju skatiet uzturēšanas procedūrā.

Uzturēšanas process tagad noņem arī tīkla atklāšanas notikumus, kas ģenerēti no atklātajiem hostiem vai pakalpojumiem, kuri kopš tā laika ir dzēsti, kā arī visus notikumus, kas saistīti ar problēmām trigeriem, kuri kopš tā laika ir dzēsti (iepriekš tika noņemtas tikai pašas problēmas; saistītie notikumi tika noņemti tikai pēc tam, kad uzturēšanas procesa *Trigeru datu glabāšanas periods* bija beidzies).

Manuāla aizvēršana — trigera birkas, ko manto atkopšanas notikumi Atkopšanas notikumi, kas izveidoti pēc manuālas aizvēršanas, manto trigera birkas papildus vienuma un hosta birkām. Šīs birkas ir ietvertas notikuma birku masīvā un ir pieejamas paziņojumu makrosiem, piemēram, `{EVENT.RECOVERY.TAGS}` un `{EVENT.RECOVERY.TAGSJSON}`.

DNS vaicājumu kešošana Zabbix dēmoniem Zabbix serveris, Zabbix starpniekserveris un Zabbix aģents saņēma atbalstu c-ares atrisinātāja izmantošanai visiem DNS pieprasījumiem, nodrošinot DNS vaicājumu kešošanu un uzlabotu atrisinātāja pārslēgšanos kļūmes gadījumā, ja Zabbix ir būvēts ar `--with-ares`. DNS vaicājumu kešošanai ir nepieciešams c-ares 1.26.0 vai jaunāka versija.

c-ares atbalsta izveide operētājsistēmā Windows Zabbix aģents tagad var tikt izveidots, izmantojot c-ares atrisinātāju operētājsistēmā Microsoft Windows. c-ares var instalēt, izmantojot vcpkg, un aģenta izveide atbalsta vai nu `ARES=<vcpkg prefix>`, vai atsevišķus `ARESINCDIR`/`ARESLIBDIR` ceļus iekļaušanas un bibliotēku direktorijiem.

Autentifikācija

Importējami SAML sertifikāti vienotajai pierakstīšanās funkcijai Super admin lietotāji tagad var importēt sertifikātus un privātās atslēgas tieši lietotāja saskarnē SAML konfigurācijai. Sadaļā *Users* > *Authentication* > *SAML settings* ir pievienoti trīs jauni lauki:

- *IdP certificate* - identitātes nodrošinātāja uzrādītais X.509 sertifikāts

- *SP certificate* - pakalpojuma nodrošinātāja sertifikāts, ko izmanto SAML apmaiņai
- *SP private key* - privātā atslēga, kas atbilst SP sertifikātam

Šīs vadīklas ļauj pievienot jaunas vērtības vai mainīt esošās SAML iestatījumu cilnē. Pirms saglabāšanas izvēlētajā glabāšanas aizmugursistēmā sertifikāti un privātās atslēgas tiek validēti; nederīgas vai nepareizi formatētas vērtības tiek noraidītas ar paskaidrojošu kļūdas ziņojumu.

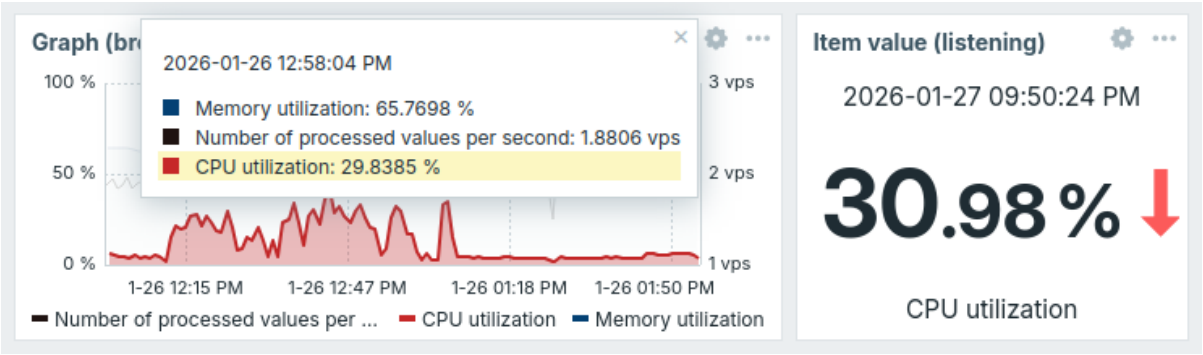
Logdaļas

Geomap hostu marķieru grupēšana Geomap logrīks tagad atbalsta parametru *Clustering*, lai kontrolētu, kā tuvumā esošie hostu marķieri tiek apvienoti vienā marķierī ar skaitu. Iestatot kartes tālummaiņas līmeni grupēšanai, varat saglabāt lielu karšu pārskatāmību un nodrošināt precīzu redzamību, kad karte ir pietuvināta.

Apkopotās kolonnas Top items Top items logrīks tagad var grupēt vienumu šablonus vienā apkopotā kolonnā vai rindā, tādējādi atvieglojot apvienotu metriku attēlošanu (piemēram: kopējā ienākošā datplūsma visās hosta tīkla saskarnēs). Konfigurējot vienumu kolonnas, tika pievienotas trīs jaunas opcijas:*Apkopot kolonnas*, *Kolonnu apkopošanas funkcija* un *Apvienotās kolonnas nosaukums*.

Graph logrīka uzlabojumi Graph logrīks piedāvā vairākus lietojamības uzlabojumus:

- Vienumi grafika datu uznirstošajā paskaidrojumā, kas parādās, novietojot kursoru virs grafika, tiek kārtoti dilstošā secībā pēc vērtības.
- Novietojot kursoru virs vienuma uznirstošajā paskaidrojumā, tā grafiks tiek izcelts, bet pārējie tiek aptumšoti.
- Atlasot vienumu uznirstošajā paskaidrojumā, tā dati tiek apraidīti uz citiem logrīkiem, kas atbalsta to saņemšanu.
- Ir noņemts ierobežojums uznirstošajā paskaidrojumā attēlojamo vienumu skaitam.
- Jauns iestatījums *Host names in labels* ļauj izvēlēties, vai hostu nosaukumi tiek attēloti uznirstošajā paskaidrojumā un grafika leģendā.



- Jauns iestatījums *Invert values* ļauj apvērst grafiku.

Palielināts rakstzīmju skaita ierobežojums logrīku apraksta laukiem Rakrūžu skaita ierobežojums laukam *Description* logrīkos Gauge un Item value ir palielināts no 2048 līdz 65535 rakstzīmēm.

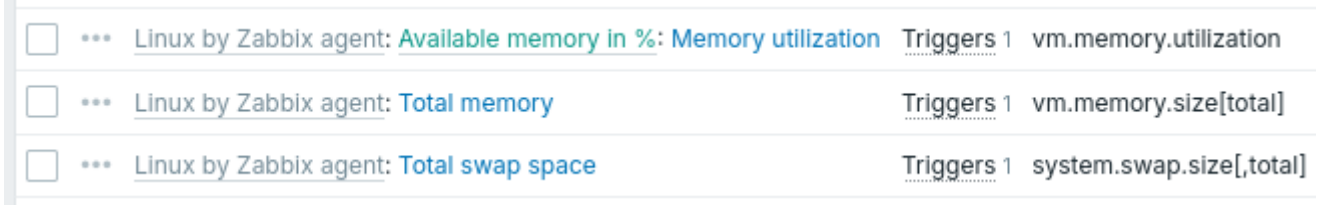
Dashboard export and import Dashboards can now be exported and imported, making it easier to migrate dashboards between Zabbix instances or reuse them in different environments.

A new *Export* option allows saving selected dashboards to YAML, JSON, or XML files. These files can later be imported through the *Import* functionality to recreate the dashboards with their pages, widgets, and configuration.

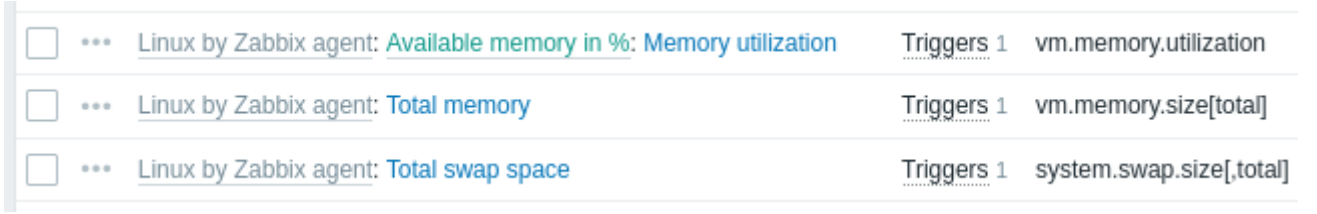
Lietotāja saskarne

Jaunas un iegultas fontu saimes Zabbix tagad ietver jaunas un iegultas fontu saimes, kas uzlabo lasāmību, ielādējas ātrāk un attēlo papildu lietotāja saskarne valodas ar minimālu ietekmi uz izkārtojumu. Tā kā šīs fontu saimes ir iegultas, tās arī nodrošina vienotu izskatu dažādās sistēmās.

Jaunas fontu saimes:



Mantotās fontu saimes:



Jaunās fontu saimes tiek izmantotas gandrīz visās tēmās. Ja nepieciešams, ir pieejamas nesen pievienotās *Blue (classic)* un *Dark (classic)* tēmas ar mantotajām fontu saimēm.

Monospace satura un grafiku fontu saime paliek nemainīga.

Iekļautā validācija Lietotāja saskarnē šādas formas ir pievienotas to formu kopai, kas atbalsta iekļauto validāciju:

- API pilnvara
- Audita žurnāls
- Autentifikācija
- Automātiskā reģistrācija
- Savienotājs
- Ģeogrāfiskās kartes
- GUI
- Hostu grupa
- Hostu prototipi
- Iztīrīšana
- Ikonu kartēšana
- Attēli
- Uzturēšana
- Medija tips
- Paziņojumi
- Citi konfigurācijas parametri
- Profils
- Starpniekserveris
- Regulārās izteiksmes
- Pakalpojums
- SLA
- Veidņu grupa
- Noildzes
- Trigeru attēlošanas opcijas
- Trigera izteiksmes konstruktors
- Atjaunināt problēmu
- Globālie lietotāja makrosi
- Lietotāju lomas
- Lietotāji

Ievades kļūdas tiek parādītas uzreiz pēc lauku aizpildīšanas, uzlabojot lietojamību un samazinot konfigurācijas kļūdu skaitu.

Modālie logi Zema līmeņa atklāšanas iestatīšanā hosta prototipa konfigurācijas forma tagad tiek atvērta modālajā (uznirstošajā) logā.

Mantotās birkas redzamas hostos, veidnēs, vienumos un trigeros Mantotās birkas tagad tiek attēlotas un atgrieztas konsekventi visās veidnēs, hostos, vienumos, tīmekļa scenārijos un trigeros. Birkas, kas mantotas no veidņu/hostu ķēdēm, ir redzamas sadaļā *Monitoring > Latest data*. Filtrēšana pēc mantotajām birkām ir pieejama visur, kur tiek atbalstīta filtrēšana pēc birkām — tostarp sadaļās *Monitoring* > *Latest data* un *Data collection*, kā arī visos paneļa logrīkos, kuros ir atļauta filtrēšana pēc hosta, vienuma, trigera vai tīmekļa scenārija birkām — tādēļ atlase un apakšfiltrēšana pēc birkām darbojas vienādi neatkarīgi no tā, kur birka tika definēta.

Cilne *Tags* veidnes, hosta un hosta prototipa konfigurācijas formās tagad piedāvā radio elementu, lai izvēlētos, kā birkas tiek attēlotas: veidnēs tiek rādīts *Template tags* / *Inherited and template tags*, bet hostos un hostu prototipos — *Host tags* / *Inherited and host tags*. Mantotās birkas ir vizuāli atšķirtas ar jaunu kontūrētu dokumenta ikonu blakus birkas etiķetei.

Graph un Pie chart logrīki ietver jaunu iestatījumu *Item tags*.

Uznirstošo paskaidrojumu novietojums Uznirstošos paskaidrojumus tagad var pārvietot uz citu vietu, velkot tos uz jaunu novietojumu. Tas attiecas, piemēram, uz Graph uznirstošo paskaidrojumu, kā arī uz uznirstošajiem paskaidrojumiem ar aprakstiem sadaļās Latest Data vai Problems.

Dokumentācija

Konsolidētas dokumentācijas lapas mazajiem laidieniem Lielā Zabbix laidiena mazo versiju laidienu dokumentācija tagad tiks apkopota atsevišķās dokumentācijas lapās par jaunajām iespējām un jauninašanas piezīmēm.

Kas jauns Zabbix 8.0.x

Šajā lapā ir apkopota informācija par jaunajām iespējām, kas iekļautas galvenās Zabbix versijas mazākajos laidienos.

Skatiet arī galvenās versijas sadaļu Kas jauns.

Informāciju par izmaiņām esošajās veidnēs un par jaunām veidnēm dažādām ierīcēm, pakalpojumiem un partneru risinājumiem skatiet sadaļā Veidņu izmaiņas.

Kas jauns Zabbix 8.0.1 Šī versija vēl nav izlaista.

3 Zabbix appliance

Pārskats Zabbix appliance nodrošina iespēju nekavējoties izvietot Zabbix serveri un lietotāja saskarni, nevis iestatīt tos manuāli vai atkārtoti izmantot esošu serveri Zabbix vajadzībām.

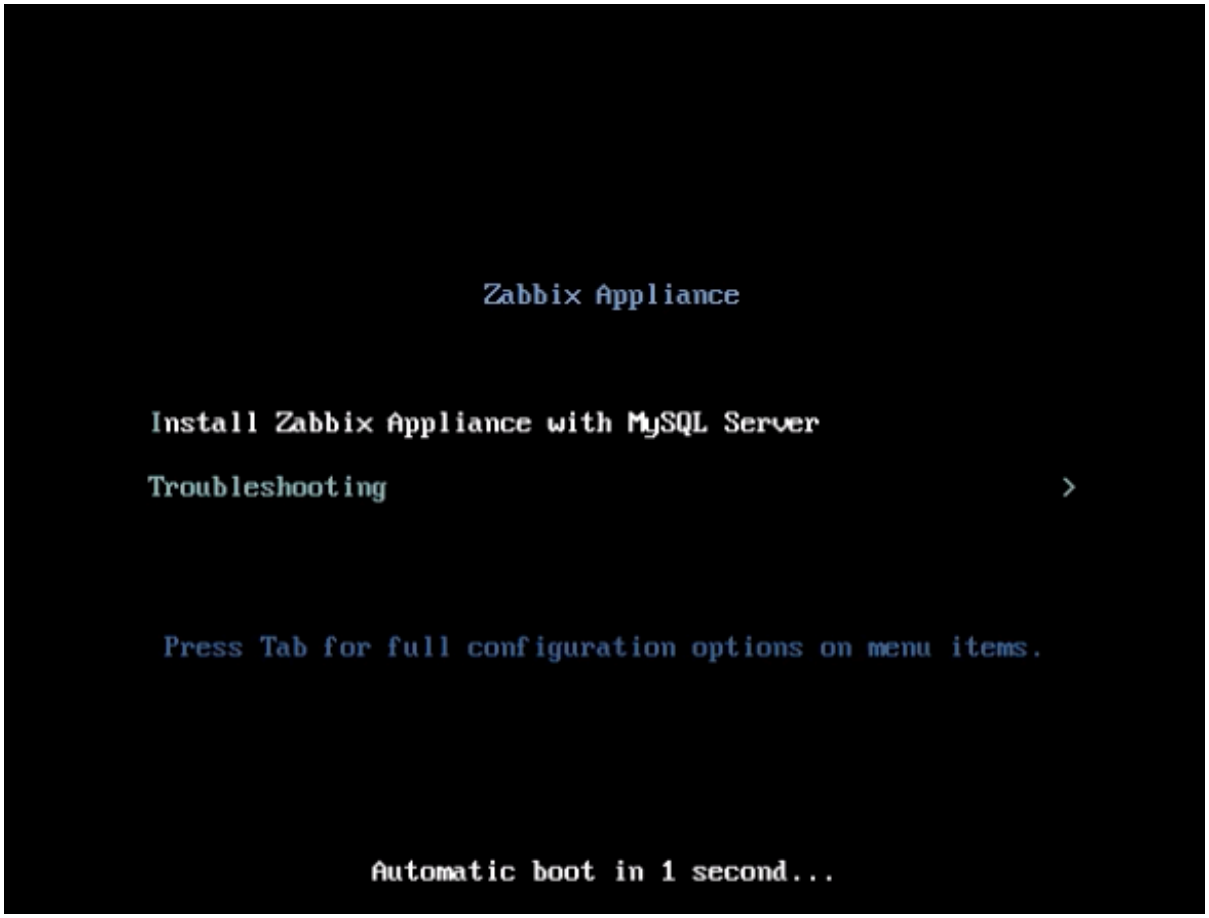
Appliance ir balstīts uz AlmaLinux 8 (x86_64) un ietver iepriekš konfigurētu Zabbix serveri, kas darbojas uz MySQL, un lietotāja saskarni, kas darbojas uz Nginx tīmekļa servera.

> **Attention:**
> Šis appliance ir paredzēts Zabbix novērtēšanai. Tā izmantošana nopietnās produkcijas vidēs nav ieteicama.

Appliance attēli ir pieejami lejupielādei šādos formātos:

- Instalācijas CD/DVD (.iso)
- VMware (.vmx) - skatiet piezīmes
- Atvērtās virtualizācijas formāts (.ovf)
- Microsoft Hyper-V (.vhd/.vhdx) - skatiet piezīmes
- KVM, Parallels, QEMU, USB zibatmiņa, VirtualBox, Xen (.raw) - skatiet piezīmes
- KVM, QEMU (.qcow2)

Zabbix instalācijas CD/DVD sāknēšanas izvēlne:



Ātrā darba sākšana Priekšnosacījumi

1. Pārliecinieties, ka hosta datoram ir pietiekami resursi, lai izpildītu virtuālās mašīnas sistēmas prasības:

- *RAM*: 4 GB
- *Diska vieta*: virtuālajai mašīnai jāpiešķir vismaz 8 GB
- *CPU*: vismaz 2 kodoli

2. Ja vēl nav instalēta, instalējiet virtualizācijas programmatūru appliance attēla palaišanai (piemēram, VirtualBox).
3. Lejupielādējiet appliance formātā, ko atbalsta jūsu virtualizācijas programmatūra.
4. Pārbaudiet tīkla iestatījumus, lai pārliecinātos, ka hosta datorā ir iespējots DHCP.

Instalēšana

1. Palaidiet appliance virtuālo mašīnu no lejupielādētā attēla.
2. Konfigurējiet virtuālās mašīnas tīkla iestatījumus, lai atļautu piekļuvi no pārlūkprogrammas hostsistēmā. To var panākt, iespējojot *tilta režīmu*.
3. Piesakieties virtuālajā mašīnā, izmantojot noklusējuma sistēmas akreditācijas datus.
4. Lai iegūtu IP adresi, virtuālajā mašīnā izpildiet šādu komandu:

```
ip addr show
```

5. Atveriet pārlūkprogrammu hostsistēmā un norādiet tajā uz IP adresi, ko appliance ir saņēmis, izmantojot DHCP.
6. Piesakieties Zabbix, izmantojot noklusējuma akreditācijas datus, un sāciet uzraudzību.

Konfigurācija Šajā sadaļā ir aprakstīti bieži nepieciešamie noklusējuma konfigurācijas iestatījumi, kā arī pieejamās pielāgošanas iespējas.

Akreditācijas dati

Sistēma

- Lietotājvārds: root
- Parole: zabbix

Zabbix lietotāja saskarne

- Lietotājvārds: Admin
- Parole: zabbix

Pēc pieteikšanās varat mainīt noklusējuma paroli lietotāja profila iestatījumos vai izveidot jaunus lietotājus un dzēst noklusējuma lietotāju.

Datu bāze

Paroles visiem datubāzes lietotājiem instalēšanas procesa laikā tiek ģenerētas nejauši. Datu bāzei ir definēti šādi lietotāji:

Sakne:

- Lietotājvārds: root
- Parole: parole tiek glabāta failā `/root/.my.cnf`. root kontam paroles ievadīšana nav nepieciešama.

Zabbix serveris:

- Lietotājvārds: zabbix_srv
- Parole: parole tiek glabāta failā `/etc/zabbix/zabbix\_server.conf`

Zabbix lietotāja saskarne:

- Lietotājvārds: zabbix_web
- Parole: parole tiek glabāta failā `/etc/zabbix/web/zabbix.conf.php`

Lai mainītu datubāzes lietotāja paroli, modificējiet to MySQL un attiecīgajā konfigurācijas failā.

Lietotāja saskarnes piekļuve

Zabbix lietotāja saskarnei var piekļūt vietnē *http://<virtual machine's IP>*.

Pēc noklusējuma piekļuve ir atļauta no jebkuras vietas. Lai ierobežotu piekļuvi, modificējiet `/etc/nginx/conf.d/zabbix.conf`. Pēc rediģētā faila saglabāšanas restartējiet Nginx, piesakoties caur SSH kā root user un izpildot:

```
systemctl restart nginx
```

Statiska IP adrese

Pēc noklusējuma ierīce izmanto DHCP, lai iegūtu IP adresi. Lai iestatītu statisku IP adresi:

- Piesakieties kā root lietotājs.
- Izpildiet tālāk norādītās komandas, aizstājot vērtības ar savām IP adresēm:

```
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
```

Alternatīvi varat atvērt failu `/etc/NetworkManager/system-connections/eth0.nmconnection` un pievienot šādas rindas:

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

Pēc rediģētā faila saglabāšanas izpildiet komandu `systemctl restart network`.

Ugunsmūra konfigurācija

Lai pārvaldītu ugunsmūra iestatījumus, iekārta izmanto iptables ar iepriekš definētiem noteikumiem:

- Atvērts SSH ports (22 TCP)
- Atvērti Zabbix aģenta (10050 TCP) un Zabbix trapper (10051 TCP) porti
- Atvērti HTTP (80 TCP) un HTTPS (443 TCP) porti
- Atvērts SNMP trap ports (162 UDP)
- Atvērti izejošie savienojumi uz NTP portu (123 UDP)
- ICMP paketes ir ierobežotas līdz 5 paketēm sekundē
- Visi pārējie ienākošie savienojumi tiek bloķēti

Lai atvērtu papildu portus, modificējiet failu `/etc/sysconfig/iptables` un pārlādējiet ugunsmūra noteikumus:

```
systemctl reload iptables
```

Repozitoriji

Zabbix appliance izmanto *zabbix-release* pakotni no Zabbix repozitorija. Repozitoriji ir konfigurēti direktorijā `/etc/yum.repos.d/*`.

Laika josla

Pēc noklusējuma ierīce sistēmas pulkstenim izmanto UTC. Lai mainītu laika joslu, nokopējiet atbilstošo failu no `/usr/share/zoneinfo` uz `/etc/localtime`, piemēram:

```
cp /usr/share/zoneinfo/Europe/Riga /etc/localtime
```

> **Note:**
> Zabbix lietotāja saskarnes laika josla tiek iestatīta atsevišķi, un to var mainīt lietotāja saskarnes iestatījumos. Zabbix lietotāja saskarnes noklusējuma laika josla ir Europe/Riga.

Failu atrašanās vietas

- Konfigurācijas faili atrodas `/etc/zabbix`
- Zabbix servera, starpniekservera un aģenta žurnālfaili atrodas `/var/log/zabbix`
- Zabbix lietotāja saskarne atrodas `/usr/share/zabbix`
- Lietotāja *zabbix* mājas direktorijs ir `/var/lib/zabbix`

Sistēmas servisi

Systemd servisi ir pieejami. Lai skatītu Zabbix servisu sarakstu, virtuālajā mašīnā izpildiet šādu komandu:

```
systemctl list-units zabbix*
```

Attēla formātam specifiskas piezīmes VMware

Attēlus *vmdk* formātā var izmantot tieši VMware Player, Server un Workstation produktos. Lai tos izmantotu ESX, ESXi un vSphere vidēs, tie jākonvertē, izmantojot VMware vCenter Converter (lejupielādei nepieciešama autentifikācija). Ja izmantojat VMware vCenter Converter, var rasties problēmas ar hibrīdo tīkla adapteri. Tādā gadījumā konvertēšanas procesa laikā varat mēģināt norādīt E1000 adapteri. Alternatīvi pēc konvertēšanas pabeigšanas varat dzēst esošo adapteri un pievienot E1000 adapteri.

HDD/flash attēls (raw)

Lai sāknētu attēlu, izpildiet:

```
dd if=./zabbix_appliance_8.0.0.raw of=/dev/sdc bs=4k conv=fdatasync
```

Aizstājiet `/dev/sdc` ar sava Flash/HDD diska ierīces ceļu.

Hyper-V

Ja ierīce Hyper-V vidē neizdodas palaist, mēģiniet nospiest **Ctrl+Alt+F2**, lai pārslēgtos uz TTY sesiju.

Problēmu novēršana Ja, mēģinot pieteikties lietotāja saskarnē, redzat kļūdas ziņojumu *Access denied for user 're-place_user'@'localhost' (using password: YES)*, tas var norādīt, ka instalēšana joprojām turpinās.

Ja kļūda saglabājas arī pēc dažu minūšu gaidīšanas vai novērojat jebkādu citu neparedzētu darbību, visticamāk, tas nozīmē, ka instalēšanas process nav veiksmīgi pabeigts. Šādā gadījumā iesakām dzēst pašreizējo ierīci un izvietot to atkārtoti, ievērojot tās pašas instalēšanas instrukcijas. Šī darbība parasti atrisina problēmu.

Ņemiet vērā, ka nav ieteicams mēģināt manuāli labot bojātu instalāciju, jo tas var radīt papildu sarežģījumus.

4 Zabbix procesi

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam Zabbix procesu sadaļā.

1 Serveris

Pārskats

Zabbix serveris ir Zabbix programmatūras centrālais process.

Serveris veic datu aptaujāšanu un saņemšanu, aprēķina trigerus, sūta paziņojumus lietotājiem. Tas ir centrālais komponents, kuram Zabbix aģenti un starpniekserveri ziņo datus par sistēmu pieejamību un integritāti. Serveris pats var arī attālināti pārbaudīt tīkla pakalpojumus (piemēram, tīmekļa serverus un pasta serverus), izmantojot vienkāršas pakalpojumu pārbaudes.

Serveris ir centrālā krātuve, kurā tiek glabāti visi konfigurācijas, statistikas un darbības dati, un tā ir Zabbix vienība, kas aktīvi brīdinās administratorus, kad kādā no uzraudzītajām sistēmām radīsies problēmas.

Pamata Zabbix servera darbība ir sadalīta trīs atsevišķos komponentos; tie ir: Zabbix serveris, tīmekļa lietotāja saskarne un datubāzes krātuve.

Visa Zabbix konfigurācijas informācija tiek glabāta datubāzē, ar kuru mijiedarbojas gan serveris, gan tīmekļa lietotāja saskarne. Piemēram, kad izveidojat jaunu vienumu, izmantojot tīmekļa lietotāja saskarni (vai API), tas tiek pievienots datubāzes tabulai items. Pēc tam aptuveni reizi minūtē Zabbix serveris vaicā tabulu items, lai iegūtu aktīvo vienumu sarakstu, kas pēc tam tiek saglabāts kešatmiņā Zabbix serverī. Tāpēc var paiet līdz divām minūtēm, līdz jebkuras Zabbix lietotāja saskarnē veiktās izmaiņas parādās jaunāko datu sadaļā.

Servera palaišana

Ja instalēts kā pakotne

Zabbix serveris darbojas kā dēmona process. Serveri var palaist, izpildot:

```
systemctl start zabbix-server
```

Tas darbosies lielākajā daļā GNU/Linux sistēmu. Citās sistēmās jums var būt nepieciešams izpildīt:

```
/etc/init.d/zabbix-server start
```

Līdzīgi, lai apturētu/pārstartētu/apskatītu statusu, izmantojiet šādas komandas:

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

Palaist manuāli

Ja iepriekš minētais nedarbojas, tas ir jāpalaiž manuāli. Atrodiet ceļu uz zabbix_server bināro failu un izpildiet:

```
zabbix_server
```

Ar Zabbix serveris var izmantot šādus komandrindas parametrus:

```
-c --config <file>              ceļš uz konfigurācijas failu (noklusējums ir /usr/local/etc/zabbix_server.
-f --foreground                 palaist Zabbix serveris priekšplānā
-R --runtime-control <option>   veikt administratīvās funkcijas
-T --test-config                pārbaudīt konfigurācijas failu un iziet
-h --help                       parādīt šo palīdzību
-V --version                    parādīt versijas numuru
```

Zabbix serveris palaišanas piemēri ar komandrindas parametriem:

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

Izpildlaika vadība

Izpildlaika vadības opcijas:

Opcija	Apraksts	Mērķis
config_cache_reload	Pārlādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta.	

Opcija	Apraksts	Mērķis
history_cache_clear[=<**target**>]	Notīrīt vēstures kešatmiņu vienumam, kas norādīts ar tā ID. Ietekmē visas vienuma vērtības, izņemot pirmo un pēdējo vērtību.	**target** - vienuma ID
diaginfo[=<**section**>]	Apkopot diagnostikas informāciju servera žurnālfailā.	**historycache** - vēstures kešatmiņas statistika **valuecache** - vērtību kešatmiņas statistika **preprocessing** - priekšapstrādes pārvaldnieka statistika **alerting** - brīdinājumu pārvaldnieka statistika **lld** - LLD pārvaldnieka statistika **locks** - muteksu saraksts (ir tukšs *BSD* sistēmās) **connector** - statistika savienotājiem ar lielāko rindu
ha_status	Reģistrēt augstas pieejamības (HA) klastera statusu žurnālā.	
ha_remove_node=<**target**>	Noņemt augstas pieejamības (HA) mezglu, kas norādīts ar tā nosaukumu vai ID. Ņemiet vērā, ka aktīvos/gaidstāves mezglus nevar noņemt.	**target** - mezgla nosaukums vai ID (var iegūt, izpildot ha_status)
ha_set_failover_delay=<**delay**>	Iestatīt augstas pieejamības (HA) pārslēgšanās aizturi. Tiek atbalstīti laika sufiksi, piemēram, 10s, 1m.	
proxy_config_cache_reload[=<**target**>]	Pārlādēt starpniekserverа konfigurācijas kešatmiņu.	**target** - ar komatiem atdalīts starpniekserveru nosaukumu saraksts Ja mērķis nav norādīts, pārlādēt konfigurāciju visiem starpniekserveriem
secrets_reload	Pārlādēt noslēpumus no Vault.	
service_cache_reload	Pārlādēt pakalpojumu pārvaldnieka kešatmiņu.	
snmp_cache_reload	Pārlādēt SNMP kešatmiņu — notīrīt SNMP dzinēja īpašības (dzinēja laiks, dzinēja palaišanas reizes, dzinēja ID, akreditācijas dati) visiem hostiem. Izmantojiet, lai piespiestu globālu kešatmiņas notīrīšanu SNMP problēmu novēršanas laikā.	
housekeeper_execute	Sākt tīrīšanas procedūru. Tiek ignorēts, ja tīrīšanas procedūra pašlaik tiek izpildīta.	
trigger_housekeeper_execute	Sākt trigeru tīrīšanas procedūru. Tiek ignorēts, ja trigeru tīrīšanas procedūra pašlaik tiek izpildīta.	
log_level_increase[=<**target**>]	Palielināt žurnālošanas līmeni; ietekmē visus procesus, ja mērķis nav norādīts. Netiek atbalstīts *BSD* sistēmās.	**process type** - visi norādītā tipa procesi (piemēram, poller) Skatiet visus servera procesu tipus. **process type,N** - procesa tips un numurs (piemēram, poller,3) **pid** - procesa identifikators (1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'.
log_level_decrease[=<**target**>]	Samazināt žurnālošanas līmeni; ietekmē visus procesus, ja mērķis nav norādīts. Netiek atbalstīts *BSD* sistēmās.	

Opcija	Apraksts	Mērķis
prof_enable[=<**target**>]	Iespējot profilēšanu. Ietekmē visus procesus, ja mērķis nav norādīts. Iespējota profilēšana sniedz detalizētu informāciju par visiem rwlock/mutex pēc funkcijas nosaukuma.	**process type** - visi norādītā tipa procesi (piemēram, history syncer) Atbalstītie procesu tipi kā profilēšanas mērķi: 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** - procesa tips un numurs (piemēram, history syncer,1) **pid** - procesa identifikators (1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` var izmantot kopā ar procesa tipu un numuru (piemēram, history syncer,1,processing) vai visiem šī tipa procesiem (piemēram, history syncer,rwlock)
prof_disable[=<**target**>]	Atspējot profilēšanu. Ietekmē visus procesus, ja mērķis nav norādīts.	**process type** - visi norādītā tipa procesi (piemēram, history syncer) Atbalstītie procesu tipi kā profilēšanas mērķi: skatiet `prof_enable` **process type,N** - procesa tips un numurs (piemēram, history syncer,1) **pid** - procesa identifikators (1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'.

Piemērs, kā izmantot izpildlaika vadību, lai pārlādētu servera konfigurācijas kešatmiņu:

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

Piemēri, kā izmantot izpildlaika vadību, lai pārlādētu starpniekservera konfigurāciju:

```
### Pārlādēt visu starpniekserveru konfigurāciju:
zabbix_server -R proxy_config_cache_reload

### Pārlādēt Proxy1 un Proxy2 konfigurāciju:
zabbix_server -R proxy_config_cache_reload=Proxy1,Proxy2
```

Piemērs, kā izmantot izpildlaika vadību, lai notīrītu vēstures kešatmiņu vienumam:

```
zabbix_server -c /usr/local/etc/zabbix_server.conf -R history_cache_clear=42243
```

Piemēri, kā izmantot izpildlaika vadību, lai iegūtu diagnostikas informāciju:

```
### Iegūt visu pieejamo diagnostikas informāciju servera žurnālfailā:
zabbix_server -R diaginfo

### Iegūt vēstures kešatmiņas statistiku servera žurnālfailā:
zabbix_server -R diaginfo=historycache
```

Piemērs, kā izmantot izpildlaika vadību, lai pārlādētu SNMP kešatmiņu:

```
zabbix_server -R snmp_cache_reload
```

Attention:
Kad SNMPv3 saskarne tiek atjaunināta, izmantojot Zabbix UI, Zabbix vairumā gadījumu automātiski pārlādēs šīs saskarnes jaunos SNMPv3 akreditācijas datus; izmantojiet `-R snmp_cache_reload` tikai tad, ja aptauja joprojām neizdodas pēc akreditācijas datu izmaiņām (piemēram, engineBoots/engineID neatbilstību vai RFC neatbilstošu ierīču dēļ), vai ja problēmu novēršanas nolūkos nepieciešams piespiedu kārtā notīrīt globālo SNMP kešatmiņu.

Piemērs, kā izmantot izpildlaika vadību, lai aktivizētu housekeeper izpildi:

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

Piemēri, kā izmantot izpildlaika vadību, lai mainītu žurnāla līmeni:

```
### Palielināt žurnāla līmeni visiem procesiem:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase

### Palielināt žurnāla līmeni otrajam poller procesam:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2

### Palielināt žurnāla līmeni procesam ar PID 1234:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234

### Samazināt žurnāla līmeni visiem http poller procesiem:
zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"
```

Piemērs, kā iestatīt HA pārslēgšanās aizkavi uz minimālajām 10 sekundēm:

```
zabbix_server -R ha_set_failover_delay=10s
```

Procesa lietotājs

Zabbix serveris ir paredzēts darbībai kā lietotājam bez root tiesībām. Tas darbosies ar jebkuru lietotāju bez root tiesībām, ar kuru tas ir palaists. Tāpēc serveri varat bez problēmām palaist kā jebkuru lietotāju bez root tiesībām.

Ja mēģināsiet to palaist kā 'root', tas pārslēgsies uz stingri definētu 'zabbix' lietotāju, kuram jābūt izveidotam jūsu sistēmā. Serveri var palaist kā 'root' tikai tad, ja attiecīgi modificējat parametru 'AllowRoot' servera konfigurācijas failā.

Ja Zabbix serveris un aģents darbojas vienā un tajā pašā datorā, ieteicams servera darbināšanai izmantot citu lietotāju nekā aģenta darbināšanai. Pretējā gadījumā, ja abi darbojas ar vienu un to pašu lietotāju, aģents var piekļūt servera konfigurācijas failam, un jebkurš Zabbix administrators ar Admin līmeņa tiesībām var diezgan viegli iegūt, piemēram, datubāzes paroli.

Konfigurācijas fails

Detalizētu informāciju par zabbix_server konfigurēšanu skatiet konfigurācijas faila opcijās.

Startēšanas skripti

Skripti tiek izmantoti, lai sistēmas palaišanas/izslēgšanas laikā automātiski palaistu/apturētu Zabbix procesus. Skripti atrodas direktorijā misc/init.d.

Servera procesu tipi un pavedieni

- `agent poller` - asinhrons aģenta aptaujātāja process pasīvajām pārbaudēm ar darba pavedienu
- `alert manager` - brīdinājumu rindas pārvaldnieks
- `alert syncer` - brīdinājumu DB rakstītājs
- `alerter` - process paziņojumu sūtīšanai
- `availability manager` - process hostu pieejamības atjaunināšanai
- `browser poller` - aptaujātājs pārlūka vienumu pārbaudēm
- `configuration syncer` - process konfigurācijas datu atmiņā esošās kešatmiņas pārvaldībai
- `configuration syncer worker` - process lietotāja makro vērtību atrisināšanai un sinhronizēšanai vienumu nosaukumos
- `connector manager` - savienotāju pārvaldnieka process
- `connector worker` - process pieprasījumu apstrādei no savienotāju pārvaldnieka
- `discovery manager` - pārvaldnieka process ierīču atklāšanai
- `discovery worker` - process atklāšanas uzdevumu apstrādei no atklāšanas pārvaldnieka
- `escalator` - process darbību eskalācijai
- `ha manager` - process augstas pieejamības pārvaldībai
- `history poller` - process aprēķināto pārbaužu apstrādei, kurām nepieciešams datubāzes savienojums
- `history syncer` - vēstures DB rakstītājs
- `housekeeper` - process novecojušu datu (vienumu vēstures un tendenču, lietotāju sesiju, notikumu u.c.), kā arī pēc dzēstiem objektiem palikušo datu noņemšanai

- `http agent poller` - asinhrons HTTP aģenta aptaujātāja process HTTP pārbaudēm ar darba pavedienu
- `http poller` - tīmekļa uzraudzības aptaujātājs
- `icmp pinger` - aptaujātājs icmpping pārbaudēm
- `internal poller` - aptaujātājs iekšējām pārbaudēm
- `ipmi manager` - IPMI aptaujātāju pārvaldnieks
- `ipmi poller` - aptaujātājs IPMI pārbaudēm
- `java poller` - aptaujātājs Java pārbaudēm
- `lld manager` - zema līmeņa atklāšanas uzdevumu pārvaldnieka process
- `lld worker` - zema līmeņa atklāšanas uzdevumu darba process
- `odbc poller` - aptaujātājs ODBC pārbaudēm
- `poller` - parasts aptaujātājs pasīvajām pārbaudēm
- `preprocessing manager` - priekšapstrādes uzdevumu pārvaldnieks ar priekšapstrādes darba pavedieniem
- `preprocessing worker` - pavediens datu priekšapstrādei
- `proxy poller` - aptaujātājs pasīvajiem starpniekserveriem
- `proxy group manager` - starpniekserveru slodzes līdzsvarošanas un augstas pieejamības pārvaldnieks
- `report manager`- plānoto atskaišu ģenerēšanas uzdevumu pārvaldnieks
- `report writer` - process plānoto atskaišu ģenerēšanai
- `self-monitoring` - process iekšējās servera statistikas apkopošanai
- `service manager` - process pakalpojumu pārvaldībai, saņemot informāciju par problēmām, problēmu birkām un problēmu atkopšanu no history syncer, task manager un alert manager
- `snmp poller` - asinhrons SNMP aptaujātāja process SNMP pārbaudēm ar darba pavedienu (tikai `walk[OID]` un `get[OID]` vienumi)
- `snmp trapper` - uztvērējs SNMP slazdiem
- `task manager` - process citu komponentu pieprasīto uzdevumu attālinātai izpildei (piemēram, problēmas aizvēršana, problēmas apstiprināšana, vienuma vērtības pārbaude tagad, attālināto komandu funkcionalitāte)
- `timer` - taimeris uzturēšanas periodu apstrādei
- `trapper` - uztvērējs aktīvajām pārbaudēm, slazdiem, starpniekservera saziņai
- `trigger housekeeper` - process problēmu un notikumu noņemšanai, ko ģenerējuši trigeri, kuri kopš tā laika ir dzēsti
- `unreachable poller` - aptaujātājs nesasniedzamām ierīcēm
- `vmware collector` - VMware datu savācējs, kas atbild par datu iegūšanu no VMware pakalpojumiem

Servera žurnālfailu var izmantot, lai novērotu šos procesu tipus.

Servera žurnālfails tiek izveidots ar lasīšanas-rakstīšanas atļaujām tikai faila īpašniekam. Papildus tam fails ir lasāms īpašnieka grupai. Visas pārējās atļaujas ir liegtas.

Dažādus Zabbix servera procesu tipus var uzraudzīt, izmantojot iekšējo **zabbix[process,<type>,<mode>,<state>]** vienumu.

Vēstures sinhronizētāja transakciju statistika

Vēstures sinhronizētāja procesa nosaukums parāda detalizētu statistiku par vēstures sinhronizētāja transakcijām:

```
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
```

Sadaļā "A+B trigeri":

- A - trigeri, kas apstrādāti vēstures vērtību dēļ;
- B - trigeri, kas apstrādāti taimeru dēļ.

Laiki sadaļā "processed...in N (<timings>) sec" ir:

- Laiks, kas pavadīts, ierakstot vienumu vērtības datubāzē;
- Laiks, kas pavadīts, atjauninot vienumu datus (stāvokli, kļūdas, hosta inventāru utt.);
- Laiks, kas pavadīts, saglabājot tendenču datus datubāzē;
- Laiks, kas pavadīts, aprēķinot trigerus;
- Laiks, kas pavadīts, apstrādājot notikumus un darbības.

Housekeeping procedūra

`housekeeper` process periodiski noņem novecojušus datus (vienumu vēsturi un tendences, lietotāju sesijas, notikumus utt.), kā arī datus, kas palikuši pēc dzēstiem objektiem. Tas darbojas ciklos, kuru biežumu un vienā ciklā dzēšamo ierakstu limitu nosaka `HousekeepingFrequency` un `MaxHousekeeperDelete`. Visi dati, kas netiek noņemti vienā ciklā, tiek pārcelti uz nākamo. Automātisko housekeeping var iespējot un konfigurēt sadaļā *Administration* > *Housekeeping*.

Lai noņemtu datus, kas palikuši pēc dzēstiem objektiem, `housekeeper` process izmanto uzdevumus no `housekeeper` tabulas, kas tiek aizpildīta ikreiz, kad objekts tiek dzēsts. Piemēram, dzēšot hostu, Zabbix dzēš arī tā vienumus, bet ne to vēsturi, tendences vai problēmas. Tā vietā datubāzes trigeri aizpilda `housekeeper` tabulu ar uzdevumiem, kas sastāv no šiem laukiem:

- `housekeeperid` - uzdevuma ID
- `object` - objekta tips (0 - vienums; 1 - trigeris; 2 - pakalpojums; 3 - atklātais hosts; 4 - atklātais pakalpojums)
- `objectid` - objekta ID (palīdz housekeeper atrast ar objektu saistītos datus)

Piemēram, dzēšot hostu ar diviem vienumiem un vienu trigeri, `housekeeper` tabula tiek aizpildīta šādi:

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

Datubāzes trigeri aizpilda `housekeeper` tabulu, nepārbaudot, vai pastāv ar objektu saistīti dati; šo pārbaudi veic `housekeeper` process.

Katrs uzdevums rada vienu vai vairākas `housekeeper` darbības, kas ir atkarīgas no objekta tipa:

- Vienumiem (ieskaitot LLD kārtulas) - noņem datus no visām vēstures un tendenču tabulām (`history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint`), kurās ir šo vienumu vērtības. Tāpat pārbauda `problems` tabulu un noņem novecojušus iekšējos notikumus, kas saistīti ar šiem vienumiem.
- Trigeriem - pārbauda ar notikumiem saistītās tabulas (`problem`, `event_symptom`, `event_recovery`, `events`) un noņem novecojušus notikumus, kas ir saistīti ar šiem trigeriem, kā arī paziņo `service manager` procesam par noņemtajiem notikumiem.

Note:
Atsevišķs `trigger housekeeper` process veic šaurāku uzdevumu — noņem problēmas un notikumus, kuriem nav zināma avota trigeris. Tā palaišanas biežumu kontrolē `ProblemHousekeepingFrequency`.

 Līdz trigger housekeeping procedūras palaišanai problēmas, ko izraisījuši vēlāk dzēsti trigeri, joprojām var ģenerēt pakalpojumu problēmas un piesaistīt tās pakalpojumiem. Ja jūsu vidē ir daudz pakalpojumu statusa aprēķināšanas kārtulu, kas balstītas uz bieži atklātiem/slēptiem trigeriem, apsveriet iespēju palielināt housekeeping procedūras biežumu, pielāgojot servera konfigurācijas parametru `ProblemHousekeepingFrequency`.

- Pakalpojumiem - pārbauda `problems` tabulu un noņem novecojušus pakalpojumu notikumus, kā arī novecojušas pakalpojumu problēmas, tādējādi tās atrisinot housekeeping izpildes laikā.
- Tīkla atklāšanai - noņem novecojušus atklāšanas notikumus no `problems` tabulas.

Housekeeper noņem tikai tos notikumus, kas nav saistīti ar problēmām. Piemēram, novecojis problēmas/atkopšanas notikums netiks noņemts, ja tas ir saistīts ar atvērtu problēmu. Kad housekeeper noņem novecojušas entītijas, tas vispirms noņem problēmas un pēc tam notikumus.

Note:
Tabulas, kas izmanto `partition` režīmu (TimescaleDB particionētās tabulas), tiek izlaistas; tiek apstrādātas tikai tabulas, kas izmanto `regular` režīmu.

Atbalstītās platformas

Drošības prasību un servera darbības kritiski svarīgā rakstura dēļ UNIX ir vienīgā operētājsistēma, kas spēj pastāvīgi nodrošināt nepieciešamo veiktspēju, kļūdu noturību un darbības nepārtrauktību. Zabbix darbojas tirgū vadošajās versijās.

Zabbix serveris ir testēts šādās platformās:

- Linux
- Solaris
- AIX
- HP-UX
- Mac OS X
- FreeBSD
- OpenBSD
- NetBSD
- SCO Open Server

Note:
Zabbix var darboties arī citās Unix tipa operētājsistēmās.

Lokalizācija

Ņemiet vērā, ka serveris pieprasa UTF-8 lokalizāciju, lai daži teksta vienumi varētu tikt pareizi interpretēti. Lielākajā daļā mūsdienu Unix tipa sistēmu UTF-8 lokalizācija ir iestatīta pēc noklusējuma, tomēr ir sistēmas, kurās tā ir jānorāda īpaši.

1 Augsta pieejamība

Pārskats

Augsta pieejamība (HA) parasti ir nepieciešama kritiskās infrastruktūrās, kurās praktiski nevar atļauties dīkstāvi. Tāpēc jebkuram pakalpojumam, kas var atteikt, ir jābūt ieviestai pārslēgšanās uz rezerves risinājumam, lai tas pārņemtu darbību, ja pašreizējais pakalpojums atteiktu.

Zabbix piedāvā **iebūvētu** augstas pieejamības risinājumu, ko ir viegli iestatīt un kam nav nepieciešama iepriekšēja HA pieredze. Iebūvētā Zabbix HA var būt noderīga kā papildu aizsardzības slānis pret Zabbix serveris programmatūras/aparatūras atteicēm vai lai samazinātu dīkstāvi apkopes laikā.

Zabbix augstas pieejamības režīmā vairāki Zabbix serveri darbojas kā mezgli klasterī. Kamēr viens Zabbix serveris klasterī ir aktīvs, pārējie atrodas gaidstāvē un ir gatavi pārņemt darbību, ja tas būs nepieciešams.

Pāreja uz Zabbix HA nav saistoša. Jūs jebkurā brīdī varat pārslēgties atpakaļ uz savrupas darbības režīmu.

Skatiet arī: Ieviešanas informācija

Augstas pieejamības iespējošana

Zabbix servera palaišana kā klastera mezglam

Lai palaistu Zabbix serveri kā klastera mezglu, servera konfigurācijā ir nepieciešami divi parametri:

- **HANodeName** parametrs jānorāda katram Zabbix serverim, kas būs HA klastera mezgls.

Tas ir unikāls mezgla identifikators (piemēram, `zabbix-node-01`), ar kuru uz serveri tiks atsaukts aģenta un starpniekservera konfigurācijās. Ja nenorādīsiet HANodeName, serveris tiks palaists savrupā režīmā.

- **NodeAddress** parametrs jānorāda katram mezglam.

NodeAddress parametrs (adrese:ports) tiks izmantots Zabbix lietotāja saskarnei, lai izveidotu savienojumu ar aktīvo servera mezglu. NodeAddress jāatbilst attiecīgā Zabbix servera IP adresei vai FQDN nosaukumam.

Pēc izmaiņu veikšanas konfigurācijas failos restartējiet visus Zabbix serverus. Tagad tie tiks palaisti kā klastera mezgli. Jauno serveru statusu var redzēt sadaļā *Atskaites* → *Sistēmas informācija*, kā arī izpildot:

```
zabbix_server -R ha_status
```

Šī izpildlaika komanda ierakstīs pašreizējo HA klastera statusu Zabbix servera žurnālā (un arī 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
```

Lietotāja saskarnes sagatavošana

Pārliecinieties, ka Zabbix servera adrese:ports lietotāja saskarnes konfigurācijā **nav definēta** (atrodama lietotāja saskarnes failu direktorijā esošajā `conf/zabbix.conf.php`).

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

Zabbix lietotāja saskarne automātiski noteiks aktīvo mezglu, nolasot iestatījumus no mezglu tabulas Zabbix datubāzē. Aktīvā mezgla adrese tiks izmantota kā Zabbix servera adrese.

Starpniekservera konfigurācija

HA klastera mezgliem (serveriem) jābūt norādītiem pasīvā vai aktīvā Zabbix starpniekservera konfigurācijā.

Pasīvam starpniekserverim mezglu nosaukumi jānorāda starpniekservera Server parametrā, atdalot tos ar **komatu**.

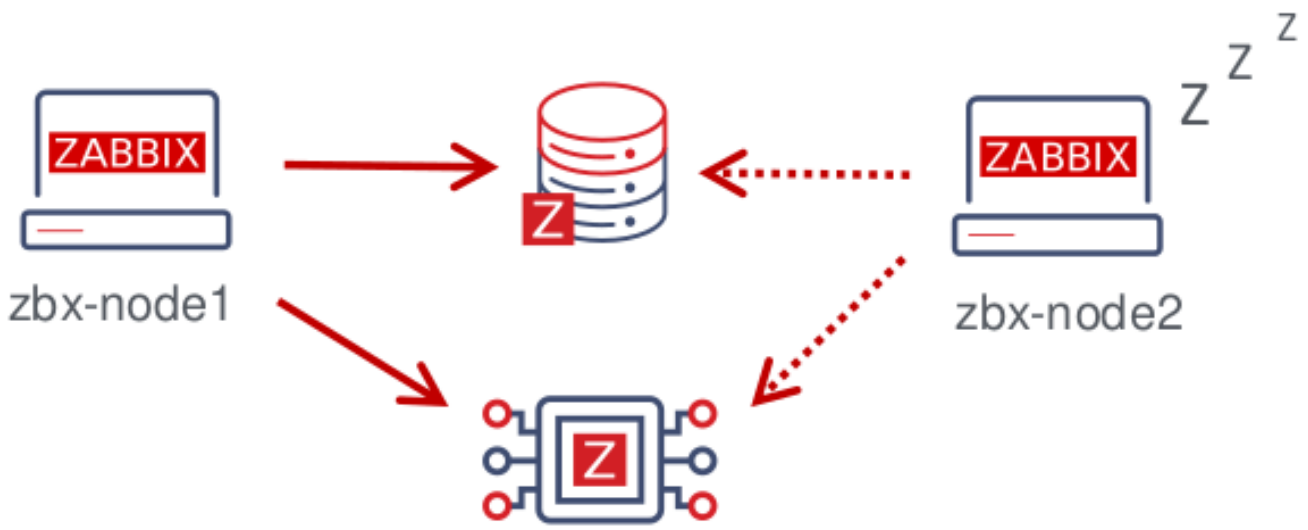
`Server=zabbix-node-01,zabbix-node-02`

Aktīvam starpniekserverim mezglu nosaukumi jānorāda starpniekservera Server parametrā, atdalot tos ar **semikolu**.

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

Aģenta konfigurācija

HA klastera mezgli (serveri) ir jānorāda Zabbix aģenta vai Zabbix agent 2 konfigurācijā.



Lai iespējotu pasīvās pārbaudes, mezglu nosaukumi jānorāda parametrā Server parameter, atdalot tos ar **komatu**.

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

Lai iespējotu aktīvās pārbaudes, mezglu nosaukumi jānorāda parametrā ServerActive parameter. Ņemiet vērā, ka aktīvajām pārbaudēm mezgli ir jāatdala ar komatu no jebkuriem citiem serveriem, savukārt paši mezgli ir jāatdala ar **semikolu**, piemēram:

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

Pārslēgšanās uz rezerves mezglu

Ja aktīvais mezgls apstājas, Zabbix automātiski pārslēgsies uz citu mezglu. Lai pārslēgšanās notiktu, gaidstāves statusā jābūt vismaz vienam mezglam.

Cik ātri notiks pārslēgšanās? Visi mezgli ik pēc 5 sekundēm atjaunina savu pēdējās piekļuves laiku (un statusu, ja tas ir mainījies). Tātad:

- Ja aktīvais mezgls izslēdzas un paspēj paziņot savu statusu kā "stopped", cits mezgls pārņems darbību **5 sekunžu** laikā.

- Ja aktīvais mezgls izslēdzas/kļūst nepieejams, nespējot atjaunināt savu statusu, gaidstāves mezgli gaidīs **pārslēgšanās aizkavi** + 5 sekundes, pirms pārņemt darbību.

Pārslēgšanās aizkave ir konfigurējama, un atbalstītais diapazons ir no 10 sekundēm līdz 15 minūtēm (pēc noklusējuma — viena minūte). Lai mainītu pārslēgšanās aizkavi, varat izpildīt:

```
zabbix_server -R ha_set_failover_delay=5m
```

HA klastera pārvaldība

Pašreizējo HA klastera statusu var pārvaldīt, izmantojot īpašās izpildlaika vadības opcijas:

- `ha_status` - reģistrēt HA klastera statusu Zabbix servera žurnālā (un izvadīt uz stdout)
- `ha_remove_node=target` - noņemt HA mezglu, kas identificēts pēc tā <target> - mezgla nosaukuma vai ID (nosaukumu/ID var iegūt no `ha_status` izpildes izvades), piemēram:

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

Ņemiet vērā, ka aktīvos/gaidstāves mezglus nevar noņemt.

- `ha_set_failover_delay=delay` - iestatīt HA pārslēgšanās aizkavi (no 10 sekundēm līdz 15 minūtēm; tiek atbalstīti laika sufiksi, piemēram, 10s, 1m)

Mezglu statusu var uzraudzīt:

- sadaļā *Pārskati* → *Sistēmas informācija*
- informācijas paneļa logrīkā *Sistēmas informācija*
- izmantojot servera `ha_status` izpildlaika vadības opciju (skatiet iepriekš).

Iekšējo vienumu `zabbix[cluster,discovery,nodes]` var izmantot mezglu atklāšanai, jo tas atgriež JSON ar augstas pieejamības mezglu informāciju.

HA klastera atspējošana

Lai atspējotu augstas pieejamības klasteri:

- izveidojiet konfigurācijas failu rezerves kopijas
- apturiet gaidstāves mezglus
- no aktīvā primārā servera noņemiet parametru HANodeName
- restartējiet primāro serveri (tas tiks palaists savrupā režīmā)

HA klastera jaunināšana

Lai veiktu HA mezglu galvenās versijas jaunināšanu:

- apturiet visus mezglus;
- izveidojiet pilnu datubāzes dublējumu;
- ja datubāze izmanto replikāciju, pārliecinieties, ka visi mezgli ir sinhronizēti un tiem nav problēmu. Neveiciet jaunināšanu, ja replikācija ir bojāta.
- atlasiet vienu mezglu, kas veiks datubāzes jaunināšanu, mainiet tā konfigurāciju uz savrupu režīmu, izkomentējot HANodeName, un jauniniet to;
- pārliecinieties, ka datubāzes jaunināšana ir pilnībā pabeigta (*Sistēmas informācijā* jābūt redzamam, ka Zabbix serveris darbojas);
- pārstartējiet mezglu HA režīmā;
- jauniniet un palaidiet pārējos mezglus (nav nepieciešams pārslēgt tos uz savrupu režīmu, jo šajā brīdī datubāze jau ir jaunināta).

Mazās versijas jaunināšanas gadījumā pietiek jaunināt pirmo mezglu, pārliecināties, ka tas ir jaunināts un darbojas, un pēc tam sākt nākamā mezgla jaunināšanu.

Ieviešanas detaļas

Augstas pieejamības (HA) klasteris ir izvēles risinājums, un tas tiek atbalstīts Zabbix serverim. Iebūvētais HA risinājums ir izstrādāts tā, lai būtu vienkārši lietojams, tas darbosies starp vietnēm un tam nav īpašu prasību attiecībā uz datubāzēm, ko Zabbix atpazīst. Lietotāji var brīvi izmantot iebūvēto Zabbix HA risinājumu vai trešās puses HA risinājumu atkarībā no tā, kas vislabāk atbilst augstas pieejamības prasībām viņu vidē.

Risinājums sastāv no vairākām zabbix_server instancēm jeb mezgliem. Katrs mezgls:

- tiek konfigurēts atsevišķi
- izmanto vienu un to pašu datubāzi
- var darboties vairākos režīmos: active, standby, unavailable, stopped

Vienlaikus aktīvs (darbojošs) var būt tikai viens mezgls. Gaidstāves mezglā darbojas tikai viens process — HA pārvaldnieks. Gaidstāves mezgls neveic datu vākšanu, apstrādi vai citas parastās servera darbības; tas neklausās portos; tam ir minimāls datubāzes savienojumu skaits.

Gan aktīvie, gan gaidstāves mezgli atjaunina savu pēdējās piekļuves laiku ik pēc 5 sekundēm. Katrs gaidstāves mezgls uzrauga aktīvā mezgla pēdējās piekļuves laiku. Ja aktīvā mezgla pēdējās piekļuves laiks pārsniedz 'failover delay' sekundes, gaidstāves mezgls pārslēdzas uz aktīvā mezgla lomu un piešķir iepriekš aktīvajam mezglam statusu 'unavailable'.

Aktīvais mezgls uzrauga savu savienojamību ar datubāzi — ja tā tiek zaudēta uz ilgāk nekā `failover delay`−5 sekundēm, tam jāpārtrauc visa apstrāde un jāpārslēdzas gaidstāves režīmā. Aktīvais mezgls arī uzrauga gaidstāves mezglu statusu — ja gaidstāves mezgla pēdējās piekļuves laiks pārsniedz 'failover delay' sekundes, gaidstāves mezglam tiek piešķirts statuss 'unavailable'.

Mezgli ir izstrādāti tā, lai būtu savietojami starp mazajām Zabbix versijām.

2 Aģents

Pārskats

Zabbix aģents tiek izvietots uz uzraudzības mērķa, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas (cietos diskus, atmiņu, procesora statistiku utt.).

Aģents lokāli apkopo darbības informāciju un nosūta datus uz Zabbix serveris turpmākai apstrādei. Kļūmju gadījumā (piemēram, ja cietais disks ir pilns vai pakalpojuma process ir avarējis), Zabbix serveris var aktīvi brīdināt tās konkrētās iekārtas administratorus, kura ziņoja par kļūmi.

Zabbix aģenti ir ļoti efektīvi, jo statistiskās informācijas apkopošanai tiek izmantoti vietējie sistēmas izsaukumi.

Pasīvās un aktīvās pārbaudes

Zabbix aģenti var veikt pasīvās un aktīvās pārbaudes:

- Pasīvās pārbaudes - Zabbix aģents atbild uz pieprasījumu no Zabbix servera (vai starpniekservera). Piemēram, serveris pieprasa datus (piem., CPU slodzi), un aģents atgriež rezultātu.
- Aktīvās pārbaudes - Zabbix aģents apkopo un nosūta datus, negaidot pieprasījumu no Zabbix servera (vai starpniekservera). Vispirms tas no servera iegūst uzraudzības vienumu sarakstu (CPU slodze, pieejamā atmiņa utt.), pēc tam apkopo nepieciešamos datus un periodiski nosūta tam atpakaļ jaunās vērtības.

Aģenta pārbaudes tips tiek konfigurēts, izvēloties atbilstošo uzraudzības vienuma tipu. Zabbix aģents apstrādā vienumus ar tipu "Zabbix agent" vai "Zabbix agent (active)".

Atbalstītās platformas

Informāciju par atbalstītajām platformām skatiet lapā Prasības.

Aģents UNIX līdzīgās sistēmās

Zabbix aģents UNIX līdzīgās sistēmās tiek palaists uz uzraugāmā hosta.

Instalēšana

Zabbix aģents var tikt instalēts Linux bāzētās sistēmās, izmantojot vienu no šādām metodēm:

- Zabbix pakotnes — atlasiet aģenta komponenti (pēc Zabbix versijas, OS distributīva un OS versijas atlasīšanas) un izpildiet sniegtos norādījumus.
- Zabbix pirmkods — lejupielādējiet pirmkoda failus un kompilējiet Zabbix aģentu, konfigurējot pirmkodu ar `--enable-agent` opciju.

Attention:
Kopumā 32 bitu Zabbix aģenti darbosies 64 bitu sistēmās, taču dažos gadījumos tie var nedarboties.

Note:
Iepriekš kompilēti Zabbix aģenta binārie faili ir pieejami lejupielādei operētājsistēmām macOS, IBM AIX, FreeBSD, OpenBSD un Solaris. Mantotie binārie faili, kas ir saderīgi ar pašreizējo Zabbix serveris/starpniekserveris versiju, ir pieejami NetBSD un HP-UX.

Ja instalēts kā pakotne

Zabbix aģents darbojas kā dēmona process. Aģentu var palaist, izpildot:

```
systemctl start zabbix-agent
```

Tas darbosies lielākajā daļā GNU/Linux sistēmu. Citās sistēmās var būt nepieciešams palaist:

```
/etc/init.d/zabbix-agent start
```

Lai apturētu, restartētu vai pārbaudītu Zabbix aģenta statusu, izmantojiet šādas komandas:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

Palaist manuāli

Jūs varat palaist Zabbix aģents, atrodot `zabbix_agentd` bināro failu un palaižot to tieši; piemēram:

```
zabbix_agentd
```

Aģents Windows sistēmās

Zabbix aģents Windows sistēmās darbojas kā Windows pakalpojums.

Instalēšana

Zabbix aģents var tikt instalēts Windows vidē, izmantojot vienu no šādām metodēm:

- Iepriekš kompilēti Zabbix aģenta binārie faili - lejupielādējiet Zabbix aģenta MSI instalācijas pakotni un izpildiet norādījumus lapā Windows aģenta instalēšana no MSI.
- Zabbix pirmkods - lejupielādējiet pirmkoda failus un izpildiet norādījumus lapā Zabbix aģenta būvēšana Windows vidē.

Papildinformāciju par Zabbix aģenta instalēšanu (no ZIP arhīva) kā Windows servisam skatiet lapā Zabbix aģents Microsoft Windows vidē.

Opcijas

Ir iespējams palaist vairākas aģenta instances vienā hostā. Viena instance var izmantot noklusējuma konfigurācijas failu vai komandrindā norādītu konfigurācijas failu. Ja tiek izmantotas vairākas instances, katrai aģenta instancei jābūt savam konfigurācijas failam (viena no instancēm var izmantot noklusējuma konfigurācijas failu).

Ar Zabbix aģentu var izmantot šādus komandrindas parametrus:

Parametrs	Apraksts
UNIX un Windows aģents	
-c --config <config-file>	Ceļš uz konfigurācijas failu. Šo opciju var izmantot, lai norādītu konfigurācijas failu, kas nav noklusējuma fails. UNIX sistēmās noklusējuma fails ir /usr/local/etc/zabbix_agentd.conf vai arī tas, kas iestatīts ar kompilēšanas laikā definētajiem mainīgajiem *--sysconfdir* vai *--prefix* Windows sistēmās noklusējuma fails ir C:\Program Files\Zabbix Agent\zabbix_agentd.conf
-f --foreground	Palaist Zabbix aģentu priekšplānā (noklusējums: true).
-p --print	Izdrukāt zināmos vienumus un iziet. *Piezīme*: lai atgrieztu arī lietotāja parametru rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma vietā).
-t --test <item key>	Pārbaudīt norādīto vienumu un iziet. *Piezīme*: lai atgrieztu arī lietotāja parametru rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma vietā).
-T --test-config	Validēt konfigurācijas failu un iziet.
-h --help	Parādīt palīdzības informāciju.
-V --version	Parādīt versijas numuru.
Tikai UNIX aģents	
-R --runtime-control <option>	Veikt administratīvās funkcijas. Skatiet izpildlaika vadību.
Tikai Windows aģents	
-m --multiple-agents	Izmantot vairākas aģenta instances (ar opcijām -i, -d, -s, -x). Lai atšķirtu instanču servisu nosaukumus, katrā servisa nosaukumā tiks iekļauta Hostname vērtība no norādītā konfigurācijas faila.

Parametrs	Apraksts
-S --startup-type <value>	Iestatīt Zabbix Windows aģenta servisa palaišanas tipu. Atļautās vērtības: `automatic` - *(noklusējums)* palaist servisu automātiski Windows startēšanas laikā; `delayed` - aizkavēt servisa palaišanu pēc tam, kad automātiski palaistie servisi ir pabeiguši startēšanu (pieejams Windows Server 2008/Vista un jaunākās versijās); `manual` - palaist servisu manuāli (lietotājs vai lietotne); `disabled` - atspējot servisu, lai to nevarētu palaist lietotājs vai lietotne. Šo opciju var izmantot kopā ar opciju -i vai arī atsevišķi, lai mainītu jau instalēta servisa palaišanas tipu.
-i --install	Instalēt Zabbix Windows aģentu kā servisu.
-d --uninstall	Atinstalēt Zabbix Windows aģenta servisu.
-s --start	Palaist Zabbix Windows aģenta servisu.
-x --stop	Apturēt Zabbix Windows aģenta servisu.

Konkrēti komandrindas parametru lietošanas **piemēri**:

- visu iebūvēto aģenta vienumu izdrukāšana ar vērtībām
- lietotāja parametra pārbaudei ar atslēgu "mysql.ping", kas definēta norādītajā konfigurācijas failā
- "Zabbix Agent" servisa instalēšana Windows vidē, izmantojot noklusējuma ceļu uz konfigurācijas failu C:\Program Files\Zabbix Agent\zabbix_agentd.conf
- "Zabbix Agent [Hostname]" servisa instalēšana Windows vidē, izmantojot konfigurācijas failu zabbix_agentd.conf, kas atrodas tajā pašā mapē, kur aģenta izpildfails, un padarot servisa nosaukumu unikālu, paplašinot to ar Hostname vērtību no konfigurācijas faila
- instalēta "Zabbix Agent" servisa palaišanas tipa mainīšana Windows vidē, izmantojot konfigurācijas failu zabbix_agentd.conf, kas atrodas tajā pašā mapē, kur aģenta izpildfails

```
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
```

Izpildlaika vadība

Ar izpildlaika vadības opcijām jūs varat mainīt aģents procesu žurnālošanas līmeni.

Option	Description	Target
log_level_increase[=<target>]	Palielināt žurnālošanas līmeni. Ja mērķis nav norādīts, tiek ietekmēti visi procesi.	Mērķi var norādīt kā: **procesa tips** - visi norādītā tipa procesi (piemēram, listener) Skatiet visus aģents procesu tipus. **procesa tips,N** - procesa tips un numurs (piemēram, listener,3) **pid** - procesa identifikators (no 1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process-type,N'.
log_level_decrease[=<target>]	Samazināt žurnālošanas līmeni. Ja mērķis nav norādīts, tiek ietekmēti visi procesi.	
userparameter_reload	Pārlādēt *UserParameter* un *Include* opciju vērtības no pašreizējā konfigurācijas faila.	

Piemēri:

- visu procesu žurnālošanas līmeņa palielināšana
- trešā listener procesa žurnālošanas līmeņa palielināšana
- procesa ar PID 1234 žurnālošanas līmeņa palielināšana
- visu aktīvo pārbaužu procesu žurnālošanas līmeņa samazināšana

```
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:
Izpildlaika vadība netiek atbalstīta OpenBSD, NetBSD un Windows vidē.

Aģenta procesu tipi

- `active checks` - process aktīvo pārbaužu veikšanai
- `collector` - process datu vākšanai
- `listener` - process pasīvo pārbaužu klausīšanai

Aģenta žurnālfailu var izmantot, lai novērotu šos procesu tipus.

Aģenta žurnālfails tiek izveidots ar lasīšanas-rakstīšanas atļaujām tikai faila īpašniekam. Papildus fails ir lasāms īpašnieka grupai. Visas pārējās atļaujas ir liegtas.

Procesa lietotājs

Zabbix aģents UNIX sistēmās ir paredzēts darbībai kā lietotājam bez root tiesībām. Tas darbosies ar to lietotāju bez root tiesībām, ar kuru tas ir palaists. Tāpēc jūs varat palaist aģentu kā jebkuru lietotāju bez root tiesībām bez jebkādām problēmām.

Ja mēģināsiet to palaist kā 'root', tas pārslēgsies uz stingri definētu 'zabbix' lietotāju, kuram jābūt pieejamam jūsu sistēmā. Jūs varat palaist aģentu kā 'root' tikai tad, ja attiecīgi modificējat parametru 'AllowRoot' aģenta konfigurācijas failā.

Konfigurācijas fails

Detalizētu informāciju par Zabbix aģents konfigurēšanu skatiet zabbix_agentd vai Windows aģents konfigurācijas faila opcijās.

Lokalizācija

Ņemiet vērā, ka aģents pieprasa UTF-8 lokalizāciju, lai daži teksta aģenta vienumi varētu atgriezt sagaidīto saturu. Lielākajā daļā mūsdienu Unix tipa sistēmu UTF-8 lokalizācija ir noklusējuma iestatījums, tomēr ir arī sistēmas, kurās tā ir jāiestata atsevišķi.

Iziešanas kods

Zabbix aģents atgriež 0 veiksmīgas iziešanas gadījumā un 1 kļūmes gadījumā.

3 Aģents 2

Pārskats

Zabbix aģents 2 ir jaunās paaudzes Zabbix aģents, kas rakstīts Go valodā (izmantojot arī daļu C koda no Zabbix aģenta). Tas ir izstrādāts, lai:

- Samazinātu TCP savienojumu skaitu.
- Nodrošinātu uzlabotu pārbaužu vienlaicīgu izpildi.
- Būtu viegli paplašināms ar spraudņiem, kas nodrošina vienkāršas pārbaudes ar minimālu koda apjomu un atbalsta sarežģītas pārbaudes, kas sastāv no ilgstoši izpildāmiem skriptiem un autonomas datu vākšanas ar periodisku ziņošanu.
- Darbotos kā Zabbix aģenta aizstājējs, atbalstot visas iepriekšējās iespējas.

Visi Zabbix aģenta 2 metriķi tiek vākti ar spraudņu palīdzību.

Šīs iespējas ir raksturīgas tieši spraudņiem:

- Plānoti un elastīgi intervāli gan pasīvajām, gan aktīvajām pārbaudēm
- Uzdevumu rindas pārvaldība, ņemot vērā grafiku un uzdevumu vienlaicīgu izpildi
- Spraudņa līmeņa noildzes
- Zabbix aģenta 2 un tā spraudņu saderības pārbaude palaišanas laikā

Pastāv divu veidu spraudņi:

- Iebūvētie spraudņi
- Ielādējamie spraudņi

Jūs varat arī izveidot savus spraudņus.

Pielāgoti izveidotie spraudņi nav obligāti jāiebūvē aģentā 2. Tos var pievienot kā ielādējamus spraudņus, tādējādi vienkāršojot spraudņu izveidi jauniem metriķiem.

Ielādējamos spraudņus var pievienot izpildes laikā, nepārkompilējot Zabbix aģentu 2.

Pasīvās un aktīvās pārbaudes

Zabbix aģents 2 atbalsta pasīvās un aktīvās pārbaudes, līdzīgi kā Zabbix aģents. Papildus tam Zabbix aģenta 2 aktīvās pārbaudes atbalsta elastīgus/plānošanas intervālus un pārbaužu vienlaicīgu izpildi viena aktīvā servera ietvaros.

Note:
Pēc noklusējuma pēc restartēšanas Zabbix aģents 2 ieplānos pirmo datu ievākšanu aktīvajām pārbaudēm nosacīti nejaušā laikā vienuma atjaunināšanas intervāla ietvaros, lai novērstu resursu izmantošanas pīķus. Lai aktīvās pārbaudes, kurām nav *Plānošanas* atjaunināšanas intervāla, tiktu veiktas uzreiz pēc aģenta restartēšanas, konfigurācijas failā iestatiet parametru `ForceActiveChecksOnStart` (globālajā līmenī) vai `Plugins.<Plugin name>.System.ForceActiveChecksOnStart` (attiecas tikai uz konkrētā spraudņa pārbaudēm). Ja ir iestatīts spraudņa līmeņa parametrs, tas ignorēs globālo parametru.

Vienlaicīgu pārbaužu skaits

Pārbaudes no dažādiem spraudņiem var tikt izpildītas vienlaicīgi. Vienlaicīgi izpildāmo pārbaužu skaitu viena spraudņa ietvaros ierobežo spraudņa kapacitātes iestatījums. Katram spraudnim var būt stingri noteikts kapacitātes iestatījums (1000 ir noklusējuma vērtība), ko var samazināt, izmantojot `Plugins.<PluginName>.System.Capacity=N` iestatījumu *Plugins* konfigurācijas parametrā.

Atbalstītās platformas

Informāciju par atbalstītajām platformām skatiet lapā Prasības.

Aģents 2 UNIX līdzīgās sistēmās

Zabbix aģents 2 UNIX līdzīgās sistēmās tiek palaists uz uzraugāmā hosta.

Instalēšana

Zabbix aģents 2 var tikt instalēts Linux sistēmās, izmantojot vienu no šādām metodēm:

- Zabbix pakotnes - atlasiet komponentu Agent 2 (pēc Zabbix versijas, OS distributīva un OS versijas atlasīšanas) un izpildiet norādījumus.
- Zabbix pirmkods - lejupielādējiet pirmkoda failus un kompilējiet aģentu, konfigurējot ar opciju `--enable-agent2`.

Note:
Zabbix aģenta 2 uzraudzības iespējas var paplašināt ar ielādējamiem spraudņiem, kas ir pieejami atsevišķi. Plašāku informāciju skatiet sadaļā Ielādējamie spraudņi.

Ja instalēts kā pakotne

Zabbix aģents 2 darbojas kā priekšplāna process un paļaujas uz ārēju servisu pārvaldnieku (piemēram, systemd), lai nodrošinātu izpildi fonā; Zabbix aģentam 2 nav iebūvēta demonizācijas atbalsta Linux vidē.

Aģentu var palaist, izpildot:

```
systemctl start zabbix-agent2
```

Lai apturētu, restartētu vai pārbaudītu Zabbix aģenta 2 statusu, izmantojiet šādas komandas:

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

Palaist manuāli

Jūs varat palaist Zabbix aģents, atrodot zabbix_agent2 bināro failu un palaižot to tieši; piemēram:

```
zabbix_agent2
```

Aģents 2 Windows sistēmās

Zabbix aģents 2 darbojas kā atsevišķs process; tomēr to var arī palaist kā Windows pakalpojumu.

Instalēšana

Zabbix aģents 2 var tikt instalēts operētājsistēmā Windows, izmantojot vienu no šādām metodēm:

- Iepriekš kompilēti Zabbix aģenta binārie faili - lejupielādējiet aģenta MSI instalēšanas pakotni un izpildiet norādījumus lapā Windows aģenta instalēšana no MSI.
- Zabbix pirmkods - lejupielādējiet pirmkoda failus un izpildiet norādījumus lapā Zabbix aģenta 2 būvēšana operētājsistēmā Windows.

Note:
Zabbix aģenta 2 uzraudzības iespējas var paplašināt ar ielādējamiem spraudņiem, kas ir pieejami atsevišķi. Plašāku informāciju skatiet sadaļā Ielādējamie spraudņi.

Papildinformāciju par Zabbix aģenta 2 instalēšanu (no ZIP arhīva) kā Windows pakalpojumu skatiet lapā Zabbix aģents Microsoft Windows vidē.

Opcijas

Ar Zabbix aģents 2 var izmantot šādus komandrindas parametrus:

Parametrs	Apraksts
UNIX un Windows aģents	
-c --config <config-file>	Ceļš uz konfigurācijas failu. Šo opciju var izmantot, lai norādītu konfigurācijas failu, kas nav noklusējuma fails. UNIX vidē noklusējuma fails ir /usr/local/etc/zabbix_agent2.conf vai kā noteikts ar kompilēšanas laikā iestatītajiem mainīgajiem *--sysconfdir* vai *--prefix* Windows vidē noklusējuma fails ir C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
-f --foreground	Palaist Zabbix aģentu priekšplānā (noklusējums: true).
-p --print	Izdrukāt zināmos vienumus un iziet. *Piezīme*: Lai atgrieztu arī lietotāja parametru rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma vietā).
-t --test <item key>	Pārbaudīt norādīto vienumu un iziet. *Piezīme*: Lai atgrieztu arī lietotāja parametru rezultātus, jānorāda konfigurācijas fails (ja tas neatrodas noklusējuma vietā).
-T --test-config	Validēt konfigurācijas failu un iziet.
-h --help	Izdrukāt palīdzības informāciju un iziet.
-v --verbose	Izdrukāt atkļūdošanas informāciju. Izmantojiet šo opciju kopā ar opcijām -p un -t.
-V --version	Izdrukāt aģenta versijas un licences informāciju.
-R --runtime-control <option>	Veikt administratīvās funkcijas. Skatiet izpildlaika vadību.
Tikai Windows aģents	
-m --multiple-agents	Izmantot vairākas aģenta instances (ar opcijām -i, -d, -s, -x). Lai atšķirtu instanču servisu nosaukumus, katrā servisa nosaukumā tiks iekļauta Hostname vērtība no norādītā konfigurācijas faila.
-S --startup-type <value>	Iestatīt Zabbix Windows aģenta servisa palaišanas tipu. Atļautās vērtības: `automatic` - *(noklusējums)* palaist servisu automātiski Windows startēšanas laikā; `delayed` - aizkavēt servisa palaišanu pēc tam, kad automātiski palaistie servisi ir pabeiguši startēšanu; `manual` - palaist servisu manuāli (lietotājam vai lietojumprogrammai); `disabled` - atspējot servisu, lai to nevarētu palaist lietotājs vai lietojumprogramma. Šo opciju var izmantot kopā ar opciju -i vai arī atsevišķi, lai mainītu jau instalēta servisa palaišanas tipu.
-i --install	Instalēt Zabbix Windows aģentu kā servisu.
-d --uninstall	Atinstalēt Zabbix Windows aģenta servisu.
-s --start	Palaist Zabbix Windows aģenta servisu.
-x --stop	Apturēt Zabbix Windows aģenta servisu.

Konkrēti komandrindas parametru izmantošanas **piemēri**:

- izdrukāt visus iebūvētos aģenta vienumus ar vērtībām
- pārbaudīt lietotāja parametru ar atslēgu "mysql.ping", kas definēta norādītajā konfigurācijas failā
- instalēt Windows servisu "Zabbix Agent", izmantojot noklusējuma ceļu uz konfigurācijas failu C:\Program Files\Zabbix Agent 2\zabbix_agent2.conf
- mainīt instalēta Windows servisa "Zabbix Agent" palaišanas tipu, izmantojot konfigurācijas failu zabbix_agent2.conf, kas atrodas tajā pašā mapē, kur aģenta izpildfails

```
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
```

Izpildlaika vadība

Izpildlaika vadība nodrošina dažas attālinātās vadības opcijas.

Option	Description
log_level_increase	Palielināt žurnālošanas līmeni.
log_level_decrease	Samazināt žurnālošanas līmeni.
metrics	Uzskaitīt pieejamos metriku datus.
version	Parādīt aģenta versiju.
userparameter_reload	Pārlādēt *UserParameter* un *Include* opciju vērtības no pašreizējā konfigurācijas faila.
help	Parādīt palīdzības informāciju par izpildlaika vadību.

Piemēri:

- palielināt žurnālošanas līmeni aģentam 2
- izdrukāt izpildlaika vadības opcijas

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

Konfigurācijas fails

Zabbix aģents 2 konfigurācijas parametri lielākoties atbilst Zabbix aģenta parametriem, ar dažiem izņēmumiem:

Parametri	Apraksts
Zabbix aģentam 2 specifiski konfigurācijas parametri	
ControlSocket	Izpildlaika vadības ligzdas ceļš. Zabbix aģents 2 izmanto vadības ligzdu izpildlaika komandām.
EnablePersistentBuffer, *PersistentBufferFile*, *PersistentBufferPeriod*	Šie parametri tiek izmantoti, lai konfigurētu pastāvīgo krātuvi Zabbix aģentā 2 aktīvajiem vienumiem.
ForceActiveChecksOnStart	Nosaka, vai aģentam aktīvās pārbaudes jāveic uzreiz pēc restartēšanas vai vienmērīgi jāsadala laika gaitā.
Plugins	Spraudņiem var būt savi parametri formātā `Plugins.<Plugin name>.<Parameter>=<value>`. Bieži izmantots spraudņa parametrs ir `System.Capacity`, kas nosaka vienlaikus izpildāmo pārbaužu ierobežojumu.
StatusPort	Ports, kurā Zabbix aģents 2 klausīsies HTTP statusa pieprasījumus un attēlos konfigurēto spraudņu sarakstu un dažus iekšējos parametrus.

Parametri	Apraksts
Zabbix aģen-tam speci-fiski konfig-urāci-jas parametri, kas netiek atbal-stīti Zab-bix aģentā 2	
AllowRoot, *User*	Netiek atbalstīti Zabbix aģentā 2, jo tas neatbalsta demonizāciju.
EnableRemoteCommands	Novecojis; tā vietā abos aģentos tiek izmantoti parametri `AllowKey` un `DenyKey`.
ListenBacklog	Tiek izmantots Zabbix aģentā, lai definētu maksimālo gaidošo savienojumu skaitu TCP rindā. Netiek atbalstīts Zabbix aģentā 2.
LoadModule, *LoadModulePath*	Ielādējamie moduļi netiek atbalstīti Zabbix aģentā 2.
MaxLinesPerSecond, *LogRemoteCommands*	Zabbix aģentā 2 to vietā tiek izmantoti `Plugins.Log.MaxLinesPerSecond` un `Plugins.SystemRun.LogRemoteCommands`.
StartAgents	Šis parametrs tiek izmantots Zabbix aģentā, lai palielinātu pasīvo pārbaužu vienlaicīgumu vai tās atspējotu. Zabbix aģentā 2 vienlaicīgums tiek konfigurēts spraudņa līmenī un to var ierobežot ar kapacitātes iestatījumu. Pasīvās pārbaudes Zabbix aģentā 2 tiek atspējotas, ja parametrs `Server` nav norādīts.

Papildinformāciju skatiet Zabbix agent2 konfigurācijas faila opcijās.

Iziešanas kodi

Zabbix aģents 2 var arī tikt kompilēts ar vecākām OpenSSL versijām (1.0.1, 1.0.2).

Šādā gadījumā Zabbix nodrošina mutexus bloķēšanai OpenSSL. Ja mutexa bloķēšana vai atbloķēšana neizdodas, tad standarta kļūdu plūsmā (STDERR) tiek izdrukāts kļūdas ziņojums, un aģents 2 beidz darbu ar atgriešanas kodu attiecīgi 2 vai 3.

Aģenta un aģenta 2 salīdzinājums

Šajā sadaļā ir aprakstītas atšķirības starp Zabbix aģentu un Zabbix aģentu 2.

Parametrs	Zabbix aģents	Zabbix aģents 2
Programmēšanas valoda	C	Go ar dažām daļām C valodā
Dēmonizācija	jā	tikai ar systemd (jā operētājsistēmā Windows)
Atbalstītie paplašinājumi	Pielāgoti ielādējami moduļi C valodā.	Pielāgoti spraudņi Go valodā.
Prasības		
Atbalstītās platformas	Linux, IBM AIX, FreeBSD, NetBSD, OpenBSD, HP-UX, Mac OS X, Solaris: 9, 10, 11, Windows: visas darbvirsmas un serveru versijas kopš XP	Linux, Windows: visas darbvirsmas un serveru versijas, kurās var instalēt atbalstītu Go versiju.

Parametrs	Zabbix aģents	Zabbix aģents 2
Atbalstītās kriptogrāfijas bibliotēkas	GnuTLS 3.1.18 un jaunāka OpenSSL 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x LibreSSL — testēta ar versijām 2.7.4, 2.8.2 (attiecas noteikti ierobežojumi; sīkāku informāciju skatiet lapā Šifrēšana).	Linux: OpenSSL 1.0.1 un jaunāka versija. MS Windows: OpenSSL 1.1.1 vai jaunāka versija. OpenSSL bibliotēkai jābūt iespējotam PSK atbalstam. LibreSSL netiek atbalstīts.
Procesu uzraudzība		
Procesi	Atsevišķs aktīvo pārbaužu process katram servera/starpniekservera ierakstam.	Viens process ar automātiski izveidotiem pavedieniem. Maksimālo pavedienu skaitu nosaka vides mainīgais GOMAXPROCS.
Metrikas	**UNIX**: skatiet atbalstīto vienumu sarakstu. **Windows**: skatiet papildu Windows specifisko vienumu sarakstu.	**UNIX**: visas Zabbix aģenta atbalstītās metrikas. Papildus aģents 2 nodrošina Zabbix vietējo uzraudzības risinājumu šādiem mērķiem: Docker, Memcached, MySQL, PostgreSQL, Redis, systemd un citiem uzraudzības mērķiem — pilnu aģentam 2 specifisko vienumu sarakstu skatiet šeit. **Windows**: visas Zabbix aģenta atbalstītās metrikas, kā arī net.tcp.service* HTTPS un LDAP pārbaudes. Papildus aģents 2 nodrošina Zabbix vietējo uzraudzības risinājumu: PostgreSQL, Redis.
Vienlaicīgums	Aktīvās pārbaudes vienam serverim tiek izpildītas secīgi.	Pārbaudes no dažādiem spraudņiem vai vairākas pārbaudes viena spraudņa ietvaros var tikt izpildītas vienlaicīgi.
Trešo pušu trapi	nē	jā
Papildu iespējas		
Pastāvīgā krātuve	nē	jā
Pastāvīgie faili log*[] metrikām	jā (tikai Unix)	nē
Žurnāla datu augšupielāde	Var tikt veikta žurnāla vākšanas laikā, lai atbrīvotu buferi.	Žurnāla vākšana tiek apturēta, kad buferis ir pilns, tādēļ parametram BufferSize jābūt vismaz MaxLinesPerSecond x 2.
Maina lietotāju izpildes laikā	jā (tikai Unix tipa sistēmās)	nē (kontrolē systemd)
Lietotāja konfigurējami šifru komplekti	jā	nē

Skatiet arī:

- *Zabbix procesu apraksts*: Zabbix aģents, Zabbix aģents 2
- *Konfigurācijas parametri*: Zabbix aģents UNIX / Windows, Zabbix aģents 2 UNIX / Windows

Iebūvētie spraudņi

Šie spraudņi ir iebūvēti Zabbix aģents 2 un ir pieejami uzreiz pēc instalēšanas. Noklikšķiniet uz spraudņa nosaukuma, lai pārietu uz spraudņa repozitoriju ar papildu informāciju.

Spraudņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
Agent	Izmantotā Zabbix aģenta metrikas.	agent.hostname, agent.ping, agent.version	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.

Sprauduņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
CPU	Sistēmas CPU uzraudzība (CPU/CPU kodolu skaits, atklātie CPU, izmantošanas procents).	system.cpu.discovery, system.cpu.num, system.cpu.util	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
Docker	Docker konteineru uzraudzība.	docker.container_info, docker.container_stats, docker.containers, docker.containers.discovery, docker.data_usage, docker.images, docker.images.discovery, docker.info, docker.ping	Skatiet arī: Konfigurācijas parametrus
File	Failu metriku vākšana.	vfs.file.cksum, vfs.file.contents, vfs.file.exists, vfs.file.md5sum, vfs.file.regexp, vfs.file.regmatch, vfs.file.size, vfs.file.time	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
Kernel	Kodola uzraudzība.	kernel.maxfiles, kernel.maxproc	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
Log	Žurnālfailu uzraudzība.	log, log.count, logrt, logrt.count	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām. Skatiet arī: Sprauduņa konfigurācijas parametri (Unix/Windows)
Memcached	Memcached servera uzraudzība.	memcached.ping, memcached.stats	
Modbus	Nolasa Modbus datus.	modbus.get	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
MQTT	Saņem publicētās MQTT tēmu vērtības.	mqtt.get	Lai konfigurētu šifrētu savienojumu ar MQTT brokeri, norādiet TLS parametrus aģenta konfigurācijas failā kā nosauktās sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus.
MySQL	MySQL un tā atvasinājumu uzraudzība.	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	Lai konfigurētu šifrētu savienojumu ar datubāzi, norādiet TLS parametrus aģenta konfigurācijas failā kā nosauktās sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus.
NetIf	Tīkla saskarņu uzraudzība.	net.if.collisions, net.if.discovery, net.if.in, net.if.out, net.if.total	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.

Sprauduņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
Oracle	Oracle Database uzraudzība.	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	Pirms sprauduņa izmantošanas instalējiet Oracle Instant Client.
Proc	Procesa CPU izmantošanas procents.	proc.cpu.util	Atbalstītajai atslēgai ir tie paši parametri kā Zabbix aģenta atslēgai.
Redis	Redis servera uzraudzība.	redis.config, redis.info, redis.ping, redis.slowlog.count	Lai konfigurētu šifrētu savienojumu ar Redis, norādiet TLS parametrus aģenta konfigurācijas failā kā nosauktās sesijas vai noklusējuma parametrus. TLS parametrus nevar nodot kā vienuma atslēgas parametrus. Ņemiet vērā, ka nepareiza vai citādi nederīga TLS konfigurācija var neļaut Zabbix aģents 2 startēties, tāpēc pirms TLS iespējošanas pārbaudiet sertifikātu failus, atļaujas un ceļus.

Smart	S.M.A.R.T. uzraudzība.	smart.attribute.discovery, smart.disk.discovery, smart.disk.get	Minimālā nepieciešamā smartctl versija ir 7.1.
			Lietotājam, kas izpilda Zabbix aģents 2, ir nepieciešamas sudo/root piekļuves tiesības uz smartctl. Spraudnis izmanto tikai šādas komandas:
			`/usr/sbin/smartctl -a *`
			`/usr/sbin/smartctl --scan *`
			`/usr/sbin/smartctl -j -V`
			
			Atbalstītās atslēgas var izmantot ar Zabbix aģents 2 tikai Linux/Windows vidē gan kā pasīvo, gan aktīvo pārbaudi.
			Skatiet arī:
			Konfigurācijas parametrus

SW	Instalēto pakotņu saraksts.	system.sw.packages, system.sw.packages.get	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgai.
Swap	Maiņvietas lielums baitos/procentos.	system.swap.size	Atbalstītajai atslēgai ir tie paši parametri kā Zabbix aģenta atslēgai.

Spraudņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
SystemRun	Izpilda norādīto komandu.	system.run	Atbalstītajai atslēgai ir tie paši parametri kā Zabbix aģenta atslēgai. Skatiet arī: Spraudņa konfigurācijas parametri (Unix/Windows)
Systemd	systemd servisu uzraudzība.	systemd.unit.discovery, systemd.unit.get, systemd.unit.info	
TCP	TCP savienojuma pieejamības pārbaude.	net.tcp.port	Atbalstītajai atslēgai ir tie paši parametri kā Zabbix aģenta atslēgai.
UDP	UDP servisu pieejamības un veiktspējas uzraudzība.	net.udp.service, net.udp.service.perf	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
Uname	Informācijas iegūšana par sistēmu.	system.hostname, system.sw.arch, system.uname	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
Uptime	Sistēmas darbības laika metriku vākšana.	system.uptime	Atbalstītajai atslēgai ir tie paši parametri kā Zabbix aģenta atslēgai.
VFSDev	VFS metriku vākšana.	vfs.dev.discovery, vfs.dev.read, vfs.dev.write	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
WebCertificate	TLS/SSL vietņu sertifikātu uzraudzība.	web.certificate.get	
WebPage	Tīmekļa lapu uzraudzība.	web.page.get, web.page.perf, web.page.regexp	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
ZabbixAsync	Asinhrona metriku vākšana.	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	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.
ZabbixStats	Zabbix servera/starpniekservera iekšējās metrikas vai aizkavēto vienumu skaits rindā.	zabbix.stats	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.

Spraudņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
ZabbixSync	Sinhrona metriku vākšana.	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.	Atbalstītajām atslēgām ir tie paši parametri kā Zabbix aģenta atslēgām.

Ielādējami spraudņi

Ielādējamie spraudņi nav pieejami uzreiz Zabbix aģents 2 un tie ir jāinstalē atsevišķi:

- Linux vidē varat izmantot pakotnes (piemēram, instalēt Ember+ Ubuntu vidē ar `apt install zabbix-agent2-plugin-ember-plu`
  vai veidot spraudņus
- Windows vidē varat instalēt no MSI vai veidot spraudņus

Attention:
Pirms spraudņa instalēšanas, lūdzu, pārbaudiet tā README failu. Tajā var būt ietvertas īpašas prasības un instalēšanas norādījumi.

Noklikšķiniet uz spraudņa nosaukuma, lai pārietu uz spraudņa repozitoriju, kurā ir spraudņa README fails ar papildu informāciju.

Spraudņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
Ceph	Ceph uzraudzība.	ceph.df.details, ceph.osd.stats, ceph.osd.discovery, ceph.osd.dump, ceph.ping, ceph.pool.discovery, ceph.status	Skatiet arī Ceph spraudņa konfigurācijas parametrus.
Ember+	Ember+ uzraudzība.	ember.get	Skatiet arī Ember+ spraudņa konfigurācijas parametrus.

Sprauduņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
MongoDB	MongoDB serveru un klasteru uzraudzība (uz dokumentiem balstīta, izkliedēta datubāze).	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	Lai konfigurētu šifrētus savienojumus ar datubāzi, norādiet TLS parametrus aģents konfigurācijas failā kā nosauktās sesijas parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus. Skatiet arī MongoDB sprauduņa konfigurācijas parametrus.
MSSQL	MSSQL datubāzes uzraudzība.	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	Lai konfigurētu šifrētu savienojumu ar datubāzi, norādiet TLS parametrus aģents konfigurācijas failā kā nosauktās sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus. Skatiet arī MSSQL sprauduņa konfigurācijas parametrus.
NVIDIA GPU	NVIDIA GPU uzraudzība.	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	Skatiet arī NVIDIA GPU sprauduņa konfigurācijas parametrus.

Sprauduņa nosaukums	Apraksts	Atbalstītās vienumu atslēgas	Komentāri
PostgreSQL	PostgreSQL un tā atvasinājumu uzraudzība.	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	Lai konfigurētu šifrētus savienojumus ar datubāzi, norādiet TLS parametrus aģents konfigurācijas failā kā nosauktās sesijas vai noklusējuma parametrus. Pašlaik TLS parametrus nevar nodot kā vienuma atslēgas parametrus. Skatiet arī PostgreSQL sprauduņa konfigurācijas parametrus.

Note:
Ielādējamie sprauduņi, palaižot ar:
 - `-V --version` - izvada sprauduņa versiju un licences informāciju;
 - `-h --help` - izvada palīdzības informāciju.
 - `-t, --test <item key>` — palaiž spraudni testēšanai (sprauduņa konfigurācija tiek ignorēta).

4 Starpniekserveris

Pārskats

Zabbix starpniekserveris ir process, kas var vākt uzraudzības datus no vienas vai vairākām uzraudzītām ierīcēm un nosūtīt šo informāciju uz Zabbix serveri, būtībā darbojoties servera vārdā. Visi savāktie dati tiek lokāli buferēti un pēc tam pārsūtīti uz Zabbix serveri, kuram starpniekserveris pieder.

Starpniekservera izvietošana nav obligāta, taču tā var būt ļoti noderīga, lai sadalītu viena Zabbix servera slodzi. Ja datus vāc tikai starpniekserveri, apstrāde serverī prasa mazāk CPU un diska I/O resursu.

Zabbix starpniekserveris ir ideāls risinājums attālinātu atrašanās vietu, filiāļu un tīklu centralizētai uzraudzībai bez lokāliem administratoriem.

Zabbix starpniekserverim ir nepieciešama atsevišķa datubāze.

Attention:
Ņemiet vērā, ka ar Zabbix starpniekserveri atbalstītās datubāzes ir SQLite, MySQL un PostgreSQL.

Skatiet arī: Starpniekserveru izmantošana izkliedētā vidē

Starpniekservera darbināšana

Ja instalēts kā pakotne

Zabbix starpniekserveris darbojas kā dēmona process. Starpniekserveri var palaist, izpildot:

```
systemctl start zabbix-proxy
```

Tas darbosies lielākajā daļā GNU/Linux sistēmu. Citās sistēmās var būt nepieciešams palaist:

```
/etc/init.d/zabbix-proxy start
```

Līdzīgi, lai apturētu/pārstartētu/apskatītu Zabbix starpniekservera statusu, izmantojiet šādas komandas:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

Palaist manuāli

Ja iepriekš minētais nedarbojas, tas ir jāpalaž manuāli. Atrodiet ceļu uz zabbix_proxy bināro failu un izpildiet:

```
zabbix_proxy
```

Ar Zabbix starpniekserveri var izmantot šādus komandrindas parametrus:

```
-c --config <file>              ceļš uz konfigurācijas failu
-f --foreground                 palaist Zabbix starpniekserveri priekšplānā
-R --runtime-control <option>   veikt administratīvās funkcijas
-T --test-config                pārbaudīt konfigurācijas failu un iziet
-h --help                       parādīt šo palīdzību
-V --version                    parādīt versijas numuru
```

Zabbix starpniekservera palaišanas piemēri ar komandrindas parametriem:

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

Izpildlaika vadība

Izpildlaika vadības opcijas:

Opcija	Apraksts	Mērķis
config_cache_reload	Pārlādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta. Aktīvs Zabbix starpniekserveris izveidos savienojumu ar Zabbix serveri un pieprasīs konfigurācijas datus. Pasīvs Zabbix starpniekserveris pieprasīs konfigurācijas datus no Zabbix servera nākamajā reizē, kad serveris izveidos savienojumu ar starpniekserveri.	
history_cache_clear[=<target>]	Notīrīt vēstures kešatmiņu vienumam, kas norādīts ar tā ID. Ietekmē visas vienuma vērtības, izņemot pirmo un pēdējo vērtību.	**target** - vienuma ID
diaginfo[=<**section**>]	Apkopot diagnostikas informāciju starpniekservera žurnālfailā.	**historycache** - vēstures kešatmiņas statistika **preprocessing** - priekšapstrādes pārvaldnieka statistika **locks** - muteksu saraksts (ir tukšs *BSD* sistēmās)
snmp_cache_reload	Pārlādēt SNMP kešatmiņu — notīrīt SNMP dzinēja rekvizītus (dzinēja laiks, dzinēja palaišanas reizes, dzinēja ID, akreditācijas dati) visiem hostiem. Izmantojiet, lai piespiestu globālu kešatmiņas notīrīšanu SNMP problēmu novēršanas laikā.	
housekeeper_execute	Sākt housekeeping procedūru. Tiek ignorēts, ja housekeeping procedūra pašlaik jau tiek izpildīta.	

Opcija	Apraksts	Mērķis
log_level_increase[=<**target**>]	Palielināt žurnālošanas līmeni; ietekmē visus procesus, ja mērķis nav norādīts. Netiek atbalstīts *BSD* sistēmās.	**process type** - Visi norādītā tipa procesi (piemēram, poller) Skatiet visus starpniekservera procesu tipus. **process type,N** - Procesa tips un numurs (piemēram, poller,3) **pid** - Procesa identifikators (no 1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'.
log_level_decrease[=<**target**>]	Samazināt žurnālošanas līmeni; ietekmē visus procesus, ja mērķis nav norādīts. Netiek atbalstīts *BSD* sistēmās.	
prof_enable[=<**target**>]	Ieslēgt profilēšanu. Ietekmē visus procesus, ja mērķis nav norādīts. Iespējota profilēšana sniedz detalizētu informāciju par visiem rwlocks/mutexes pēc funkcijas nosaukuma.	**process type** - Visi norādītā tipa procesi (piemēram, history syncer) Skatiet visus starpniekservera procesu tipus. **process type,N** - Procesa tips un numurs (piemēram, history syncer,1) **pid** - Procesa identifikators (no 1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'. **scope** - `rwlock`, `mutex`, `processing` var izmantot kopā ar procesa tipu un numuru (piemēram, history syncer,1,processing) vai visiem noteikta tipa procesiem (piemēram, history syncer,rwlock)
prof_disable[=<**target**>]	Atspējot profilēšanu. Ietekmē visus procesus, ja mērķis nav norādīts.	**process type** - Visi norādītā tipa procesi (piemēram, history syncer) Skatiet visus starpniekservera procesu tipus. **process type,N** - Procesa tips un numurs (piemēram, history syncer,1) **pid** - Procesa identifikators (no 1 līdz 65535). Lielākām vērtībām norādiet mērķi kā 'process type,N'.

Piemērs, kā izmantot izpildlaika vadību, lai pārlādētu starpniekservera konfigurācijas kešatmiņu:

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

Piemērs, kā izmantot izpildlaika vadību, lai notīrītu vēstures kešatmiņu vienumam:

```
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R history_cache_clear=42243
```

Piemēri, kā izmantot izpildlaika vadību, lai apkopotu diagnostikas informāciju:

```
### Apkopot visu pieejamo diagnostikas informāciju starpniekservera žurnālfailā:
zabbix_proxy -R diaginfo

### Apkopot vēstures kešatmiņas statistiku starpniekservera žurnālfailā:
zabbix_proxy -R diaginfo=historycache
```

Piemērs, kā izmantot izpildlaika vadību, lai pārlādētu SNMP kešatmiņu:

```
zabbix_proxy -R snmp_cache_reload
```

Attention:
Kad SNMPv3 saskarne tiek atjaunināta, izmantojot Zabbix lietotāja saskarni, Zabbix vairumā gadījumu automātiski pārlādēs jaunās SNMPv3 akreditācijas datus šai saskarnei; izmantojiet `-R snmp_cache_reload` tikai tad, ja aptauja joprojām neizdodas pēc akreditācijas datu izmaiņām (piemēram, engineBoots/engineID neatbilstību vai RFC neatbilstošu ierīču dēļ), vai ja problēmu novēršanas nolūkā nepieciešams piespiest globālu SNMP kešatmiņas notīrīšanu.

Piemērs, kā izmantot izpildlaika vadību, lai aktivizētu housekeeper izpildi:

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

Piemēri, kā izmantot izpildlaika vadību, lai mainītu žurnālošanas līmeni:

```
### Palielināt žurnālošanas līmeni visiem procesiem:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase

### Palielināt žurnālošanas līmeni otrajam poller procesam:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2

### Palielināt žurnālošanas līmeni procesam ar PID 1234:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234

### Samazināt žurnālošanas līmeni visiem http poller procesiem:
zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"
```

Procesa lietotājs

Zabbix starpniekserveris ir paredzēts darbībai kā lietotājam bez root tiesībām. Tas darbosies ar to lietotāju bez root tiesībām, ar kuru tas ir palaists. Tāpēc starpniekserveri varat palaist kā jebkuru lietotāju bez root tiesībām bez jebkādām problēmām.

Ja mēģināsiet to palaist kā 'root', tas pārslēgsies uz stingri definētu 'zabbix' lietotāju, kuram jābūt pieejamam jūsu sistēmā. Starpniekserveri var palaist kā 'root' tikai tad, ja attiecīgi maināt parametru 'AllowRoot' starpniekservera konfigurācijas failā.

Konfigurācijas fails

Detalizētu informāciju par zabbix_proxy konfigurēšanu skatiet konfigurācijas faila opcijās.

Starpniekservera procesu tipi un pavedieni

- `agent poller` - asinhrons aptaujātāja process pasīvajām pārbaudēm ar darba pavedienu
- `availability manager` - process hosta pieejamības atjauninājumiem
- `browser poller` - aptaujātājs pārlūka vienumu pārbaudēm
- `configuration syncer` - process konfigurācijas datu atmiņā esošās kešatmiņas pārvaldībai
- `data sender` - starpniekservera datu sūtītājs
- `discovery manager` - pārvaldnieka process ierīču atklāšanai
- `discovery worker` - process atklāšanas uzdevumu apstrādei no atklāšanas pārvaldnieka
- `history syncer` - vēstures DB rakstītājs
- `housekeeper` - process novecojušās vienumu vēstures noņemšanai
- `http agent poller` - asinhrons aptaujātāja process HTTP pārbaudēm ar darba pavedienu
- `http poller` - tīmekļa uzraudzības aptaujātājs
- `icmp pinger` - aptaujātājs icmpping pārbaudēm
- `internal poller` - aptaujātājs iekšējām pārbaudēm
- `ipmi manager` - IPMI aptaujātāju pārvaldnieks
- `ipmi poller` - aptaujātājs IPMI pārbaudēm
- `java poller` - aptaujātājs Java pārbaudēm
- `odbc poller` - aptaujātājs ODBC pārbaudēm
- `poller` - parasts aptaujātājs pasīvajām pārbaudēm
- `preprocessing manager` - priekšapstrādes uzdevumu pārvaldnieks ar priekšapstrādes darba pavedieniem
- `preprocessing worker` - pavediens datu priekšapstrādei
- `self-monitoring` - process iekšējās servera statistikas apkopošanai
- `snmp poller` - asinhrons aptaujātāja process SNMP pārbaudēm ar darba pavedienu (tikai `walk[OID]` un `get[OID]` vienumi)
- `snmp trapper` - uztvērējs SNMP slazdiem
- `task manager` - process citu komponentu pieprasīto uzdevumu attālinātai izpildei (piemēram, problēmas aizvēršana, problēmas apstiprināšana, vienuma vērtības pārbaude tagad, attālināto komandu funkcionalitāte)
- `trapper` - uztvērējs aktīvajām pārbaudēm, slazdiem, starpniekservera saziņai
- `unreachable poller` - aptaujātājs nesasniedzamām ierīcēm
- `vmware collector` - VMware datu savācējs, kas atbild par datu iegūšanu no VMware pakalpojumiem

Starpniekservera žurnālfailu var izmantot, lai novērotu šos procesu tipus.

Starpniekservera žurnālfails tiek izveidots ar lasīšanas-rakstīšanas atļaujām tikai faila īpašniekam. Papildus tam fails ir lasāms īpašnieka grupai. Visas pārējās atļaujas ir liegtas.

Dažādus Zabbix starpniekservera procesu tipus var uzraudzīt, izmantojot iekšējo **zabbix[process,<type>,<mode>,<state>]** vienumu.

Vēstures sinhronizētāja transakciju statistika

Vēstures sinhronizētāja procesa nosaukums attēlo detalizētu statistiku par vēstures sinhronizētāja transakcijām.

```
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
```

Laiki izteiksmē "processed...in N (<timings>) sec" ir šādi:

- Laiks, kas pavadīts, ierakstot vienumu vērtības datubāzē;
- Laiks, kas pavadīts, atjauninot vienumu datus (stāvokli, kļūdas).

Tīrīšanas procedūra

Zabbix starpniekserverim ir `housekeeper` process, kas noņem novecojušu vienumu vēsturi un tendences. Tas darbojas ciklos, kuru biežumu nosaka `HousekeepingFrequency`, bet vienā ciklā dzēšamo datu ierobežojumu nosaka `ProxyLocalBuffer` un `ProxyOfflineBuffer`. Atšķirībā no Zabbix servera tīrīšanas procedūras, starpniekservera `housekeeper` process neizmanto `housekeeper` tabulu — tas dzēš visus novecojušos datus vienu reizi katrā tīrīšanas ciklā.

Atbalstītās platformas

Zabbix starpniekserveris darbojas tajā pašā atbalstīto platformu sarakstā kā Zabbix serveris.

Atmiņas buferis

Atmiņas buferis ļauj saglabāt jaunus datus (vienumu vērtības, tīkla atklāšanu, hostu automātisko reģistrāciju) buferī un augšupielādēt tos uz Zabbix serveris, nepiekļūstot datubāzei. Atmiņas buferis starpniekserverim ir pieejams kopš Zabbix 7.0.

Instalācijās pirms Zabbix 7.0 savāktie dati pirms augšupielādes uz Zabbix serveris tika saglabāti datubāzē. Šādām instalācijām pēc jaunināšanas uz Zabbix 7.0 tā joprojām ir noklusējuma darbība.

Lai nodrošinātu optimālu veiktspēju, ieteicams starpniekserverī konfigurēt atmiņas bufera izmantošanu. To var izdarīt, mainot ProxyBufferMode vērtību no "disk" (stingri noteikta noklusējuma vērtība esošajām instalācijām) uz "hybrid" (ieteicams) vai "memory". Nepieciešams arī iestatīt atmiņas bufera lielumu (parametrs ProxyMemoryBufferSize).

Hibrīda režīmā buferis ir aizsargāts pret datu zudumu, ierakstot nenosūtītos datus datubāzē, ja starpniekserveris tiek apturēts, buferis ir pilns vai dati ir pārāk veci. Kad visas vērtības ir ierakstītas datubāzē, starpniekserveris atgriežas pie atmiņas bufera izmantošanas.

Atmiņas režīmā tiks izmantots atmiņas buferis, taču aizsardzība pret datu zudumu netiek nodrošināta. Ja starpniekserveris tiek apturēts vai atmiņa tiek pārpildīta, nenosūtītie dati tiks atmesti.

Hibrīda režīms (ProxyBufferMode=hybrid) tiek izmantots visās jaunajās instalācijās kopš Zabbix 7.0.

Papildu parametri, piemēram, ProxyMemoryBufferSize un ProxyMemoryBufferAge, attiecīgi nosaka atmiņas bufera lielumu un maksimālo datu glabāšanas ilgumu buferī.

Ņemiet vērā, ka konfliktējošas konfigurācijas gadījumā starpniekserveris izdrukās kļūdu un neizdosies palaist, piemēram, ja:

- ProxyBufferMode ir iestatīts uz "hybrid" vai "memory" un ProxyMemoryBufferSize ir "0";
- ProxyBufferMode ir iestatīts uz "hybrid" vai "memory" un ProxyLocalBuffer nav "0".

Lokalizācija

Ņemiet vērā, ka starpniekserverim ir nepieciešama UTF-8 lokalizācija, lai dažus teksta vienumus varētu interpretēt pareizi. Lielākajā daļā mūsdienu Unix tipa sistēmu UTF-8 lokalizācija ir iestatīta pēc noklusējuma, tomēr ir arī sistēmas, kurās tā ir jāiestata atsevišķi.

Rindu aprēķināšana apkopes laikā

Attention:
Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet sadaļā Rindu aprēķināšana apkopes laikā.

5 Java gateway

Pārskats

Zabbix Java gateway var instalēt no pirmkoda vai pakotnēm.

Vietējais atbalsts JMX lietotņu uzraudzībai ir pieejams Zabbix dēmona veidā ar nosaukumu "Zabbix Java gateway". Zabbix Java gateway ir Java valodā rakstīts dēmons. Lai noteiktu konkrēta JMX skaitītāja vērtību hostā, Zabbix serveris vaicā Zabbix Java gateway, kas izmanto JMX pārvaldības API attāli, lai vaicātu interesējošo lietotni. Lietotnei nav nepieciešama papildu programmatūras instalēšana; tā vienkārši jāpalaiž ar komandrindas opciju `-Dcom.sun.management.jmxremote`.

Java gateway pieņem ienākošos savienojumus no Zabbix servera vai starpniekservera un to var izmantot tikai kā "pasīvo starpniekserveri". Atšķirībā no Zabbix starpniekservera, to var izmantot arī no Zabbix starpniekservera (Zabbix starpniekserverus nevar ķēdēt). Piekļuve katram Java gateway tiek konfigurēta tieši Zabbix servera vai starpniekservera konfigurācijas failā, tādēļ katram Zabbix serverim vai Zabbix starpniekserverim var konfigurēt tikai vienu Java gateway. Ja hostam būs **JMX aģents** tipa vienumi un arī cita tipa vienumi, tikai **JMX aģents** vienumi tiks nodoti Java gateway iegūšanai.

Ja vienums ir jāatjaunina, izmantojot Java gateway, Zabbix serveris vai starpniekserveris izveidos savienojumu ar Java gateway un pieprasīs vērtību, kuru Java gateway savukārt iegūs un nodos atpakaļ serverim vai starpniekserverim. Tādējādi Java gateway nevienas vērtības neglabā kešatmiņā.

Zabbix serverim vai starpniekserverim ir īpaša tipa procesi, kas veido savienojumu ar Java gateway; to darbību kontrolē opcija **StartJavaPollers**. Iekšēji Java gateway palaiž vairākus pavedienus, ko kontrolē **START_POLLERS** opcija. Servera pusē, ja savienojuma izveide aizņem vairāk nekā **Timeout** sekundes, tā tiks pārtraukta, taču Java gateway joprojām var būt aizņemts, iegūstot vērtību no JMX skaitītāja. Lai to atrisinātu, Java gateway ir pieejama opcija **TIMEOUT**, kas ļauj iestatīt noildzi JMX tīkla operācijām.

Zabbix serveris vai starpniekserveris centīsies pēc iespējas vairāk apvienot pieprasījumus vienam JMX mērķim kopā (to ietekmē vienumu intervāli) un nosūtīt tos uz Java gateway vienā savienojumā, lai uzlabotu veiktspēju.

Ieteicams, lai **StartJavaPollers** būtu mazāks vai vienāds ar **START_POLLERS**, pretējā gadījumā var rasties situācijas, kad Java gateway nav pieejamu pavedienu ienākošo pieprasījumu apstrādei; šādā gadījumā Java gateway izmanto ThreadPoolExecutor.CallerRunsPolicy, kas nozīmē, ka galvenais pavediens apstrādās ienākošo pieprasījumu un īslaicīgi nepieņems nevienu jaunu pieprasījumu.

Ja mēģināt uzraudzīt uz Wildfly balstītas Java lietojumprogrammas ar Zabbix Java gateway, lūdzu, instalējiet jaunāko jboss-client.jar, kas pieejams Wildfly lejupielāžu lapā.

2 Iestatīšana no RHEL pakotnēm

Pārskats

Ja instalēts no pakotnēm, tālāk sniegtā informācija palīdzēs jums iestatīt Zabbix Java gateway.

Java gateway konfigurēšana un palaišana

Zabbix Java gateway konfigurācijas parametrus var pielāgot failā:

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

Papildinformāciju skatiet Zabbix Java gateway konfigurācijas parametros.

Lai palaistu Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Lai Zabbix Java gateway tiktu automātiski palaists sistēmas startēšanas laikā:

RHEL 7 un jaunākās versijās:

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

RHEL versijās pirms 7:

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

Servera konfigurēšana darbam ar Java gateway

Kad Java gateway ir palaists un darbojas, jums jānorāda Zabbix serverim, kur atrast Zabbix Java gateway. To izdara, norādot parametrus JavaGateway un JavaGatewayPort servera konfigurācijas failā. Ja hosts, kurā darbojas JMX lietotne, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametri jānorāda starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nevienu procesu, kas saistīts ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda iepriekš izveidoto Java aptaujātāju instanču skaits. To izdara tāpat, kā norādāt parastos aptaujātājus un uztvērējus.

```
StartJavaPollers=5
```

Neaizmirstiet restartēt serveri vai starpniekserveri, kad esat pabeidzis to konfigurēšanu.

Java gateway atkļūdošana

Zabbix Java gateway žurnālfails ir:

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

Ja vēlaties palielināt reģistrēšanas detalizācijas līmeni, rediģējiet failu:

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

un nomainiet `level="info"` uz ”debug” vai pat ”trace” (padziļinātai problēmu novēršanai):

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

</configuration>
```

JMX uzraudzība

Skatiet JMX uzraudzība lapu, lai iegūtu sīkāku informāciju.

3 Iestatīšana no Debian/Ubuntu pakotnēm

Pārskats

Ja instalēts no pakotnēm, tālāk sniegtā informācija palīdzēs jums iestatīt Zabbix Java gateway.

Java gateway konfigurēšana un palaišana

Java gateway konfigurāciju var pielāgot failā:

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

Plašāku informāciju skatiet Zabbix Java gateway konfigurācijas parametros.

Lai palaistu Zabbix Java gateway:

```
systemctl restart zabbix-java-gateway
```

Lai automātiski palaistu Zabbix Java gateway sistēmas startēšanas laikā:

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

Servera konfigurēšana darbam ar Java gateway

Kad Java gateway ir palaists un darbojas, jums jānorāda Zabbix serverim, kur atrast Zabbix Java gateway. To izdara, norādot parametrus JavaGateway un JavaGatewayPort servera konfigurācijas failā. Ja hosts, kurā darbojas JMX lietotne, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametri jānorāda starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nevienu procesu, kas saistīts ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda iepriekš izveidoto Java aptaujātāju instanču skaits. To izdara tāpat, kā norādāt parastos aptaujātājus un uztvērējus.

```
StartJavaPollers=5
```

Neaizmirstiet restartēt serveri vai starpniekserveri, kad esat pabeidzis to konfigurēšanu.

Java gateway atkļūdošana

Zabbix Java gateway žurnālfails ir:

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

Ja vēlaties palielināt žurnalēšanas detalizācijas līmeni, rediģējiet failu:

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

un nomainiet `level="info"` uz ”debug” vai pat ”trace” (padziļinātai problēmu novēršanai):

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

</configuration>
```

JMX uzraudzība

Skatiet JMX uzraudzība lapu, lai iegūtu sīkāku informāciju.

Iestatīšana no pirmkoda

Pārskats

Ja instalēts no pirmkoda, tālāk sniegtā informācija palīdzēs jums iestatīt Zabbix Java gateway.

Failu pārskats

Ja ieguvāt Java gateway no pirmkoda, jums vajadzētu būt iegūtai čaulas skriptu, JAR un konfigurācijas failu kopai mapē $PREFIX/sbin/zabbix_java. Šo failu loma ir apkopota zemāk.

```
bin/zabbix-java-gateway-$VERSION.jar
```

Pats Java gateway JAR fails.

```
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
```

Java gateway atkarības: Logback, SLF4J un Android JSON bibliotēka.

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

Logback konfigurācijas faili.

```
shutdown.sh
startup.sh
```

Palīgskripti Java gateway palaišanai un apturēšanai.

```
settings.sh
```

Konfigurācijas fails, ko izmanto iepriekš minētie startup un shutdown skripti.

Java gateway konfigurēšana un palaišana

Pēc noklusējuma Java gateway klausās 10052. portā. Ja plānojat palaist Java gateway citā portā, to varat norādīt skriptā settings.sh. Skatiet Java gateway konfigurācijas faila aprakstu, lai uzzinātu, kā norādīt šo un citas opcijas.

Warning:
10052. ports nav IANA reģistrēts.

Kad iestatījumi jūs apmierina, varat palaist Java gateway, izpildot palaišanas skriptu:

```
./startup.sh
```

Tāpat, kad Java gateway jums vairs nav nepieciešams, izpildiet apturēšanas skriptu, lai to apturētu:

```
./shutdown.sh
```

Ņemiet vērā, ka atšķirībā no serveris vai starpniekserveris, Java gateway ir viegls un tam nav nepieciešama datubāze.

Servera konfigurēšana darbam ar Java gateway

Kad Java gateway ir palaists un darbojas, jums jānorāda Zabbix serverim, kur atrast Zabbix Java gateway. To izdara, norādot parametrus JavaGateway un JavaGatewayPort servera konfigurācijas failā. Ja hosts, kurā darbojas JMX lietotne, tiek uzraudzīts ar Zabbix starpniekserveri, tad savienojuma parametri jānorāda starpniekservera konfigurācijas failā.

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

Pēc noklusējuma serveris nepalaiž nevienu procesu, kas saistīts ar JMX uzraudzību. Tomēr, ja vēlaties to izmantot, jums jānorāda iepriekš izveidoto Java aptaujātāju instanču skaits. To izdara tāpat, kā norādāt parastos aptaujātājus un uztvērējus.

```
StartJavaPollers=5
```

Neaizmirstiet restartēt serveri vai starpniekserveri, kad esat pabeidzis to konfigurēšanu.

Java gateway atkļūdošana

Ja rodas kādas problēmas ar Java gateway vai kļūdas ziņojums, ko redzat par vienumu lietotāja saskarnē, nav pietiekami aprakstošs, iespējams, vēlēsieties apskatīt Java gateway žurnālfailu.

Pēc noklusējuma Java gateway reģistrē savas darbības failā /tmp/zabbix_java.log ar žurnalēšanas līmeni "info". Dažreiz ar šo informāciju nepietiek un ir nepieciešama informācija ar žurnalēšanas līmeni "debug". Lai palielinātu žurnalēšanas līmeni, modificējiet failu lib/logback.xml un nomainiet atribūtu level tagam <root> uz "debug":

```
<root level="debug">
  <appender-ref ref="FILE" />
</root>
```

Ņemiet vērā, ka atšķirībā no Zabbix serveris vai Zabbix starpniekserveris nav nepieciešams pārstartēt Zabbix Java gateway pēc faila logback.xml maiņas — izmaiņas failā logback.xml tiks uztvertas automātiski. Kad atkļūdošana ir pabeigta, varat atjaunot žurnalēšanas līmeni uz "info".

Ja vēlaties reģistrēt citā failā vai pilnīgi citā vidē, piemēram, datubāzē, pielāgojiet failu logback.xml atbilstoši savām vajadzībām. Plašāku informāciju skatiet Logback rokasgrāmatā.

Dažreiz atkļūdošanas nolūkos ir lietderīgi palaist Java gateway kā konsoles lietotni, nevis dēmonu. Lai to izdarītu, izkomentējiet mainīgo PID_FILE failā settings.sh. Ja PID_FILE ir izlaists, skripts startup.sh palaiž Java gateway kā konsoles lietotni un liek Logback tā vietā izmantot failu lib/logback-console.xml, kas ne tikai reģistrē izvadi konsolē, bet arī izmanto žurnalēšanas līmeni "debug".

Visbeidzot, ņemiet vērā, ka, tā kā Java gateway žurnalēšanai izmanto SLF4J, varat aizstāt Logback ar sev vēlamo ietvaru, ievietojot atbilstošu JAR failu direktorijā lib. Plašāku informāciju skatiet SLF4J rokasgrāmatā.

JMX uzraudzība

Skatiet JMX uzraudzība lapu, lai iegūtu sīkāku informāciju.

6 Sūtītājs

Pārskats

Zabbix sender ir komandrindas utilīta, ko var izmantot, lai nosūtītu veiktspējas datus uz Zabbix serveris apstrādei.

Šo utilītu parasti izmanto ilgstoši darbojošos lietotāju skriptos pieejamības un veiktspējas datu periodiskai nosūtīšanai.

Lai nosūtītu rezultātus tieši uz Zabbix serveris vai starpniekserveris, ir jākonfigurē trapper vienums tips.

Skatiet arī Python bibliotēku Zabbix vajadzībām, kurai ir iebūvēta funkcionalitāte darboties kā Zabbix sender.

Zabbix sender palaišana

Zabbix UNIX sender palaišanas piemērs:

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

kur:

- z - Zabbix servera hosta nosaukums (var izmantot arī IP adresi)
- s - uzraudzītā hosta tehniskais nosaukums (kā reģistrēts Zabbix lietotāja saskarnē)
- k - vienuma atslēga
- o - nosūtāmā vērtība

Attention:
Opcijas, kas satur atstarpes, jāliek pēdiņās, izmantojot dubultās pēdiņas.

Zabbix sender var izmantot, lai nosūtītu vairākas vērtības no ievades faila. Plašāku informāciju skatiet Zabbix sender rokasgrāmatas lapā.

Ja ir norādīts konfigurācijas fails, Zabbix sender datu nosūtīšanai izmanto visas adreses, kas definētas aģenta ServerActive konfigurācijas parametrā. Ja nosūtīšana uz vienu adresi neizdodas, sender mēģina sūtīt uz pārējām adresēm. Ja pakešdatu nosūtīšana uz vienu adresi neizdodas, nākamās paketes uz šo adresi netiek sūtītas.

Zabbix sender pieņem virknes UTF-8 kodējumā (gan UNIX tipa sistēmās, gan Windows) bez baita secības atzīmes (BOM) faila sākumā.

Zabbix sender Windows vidē var palaist līdzīgi:

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

zabbix_sender reāllaika sūtīšanas scenārijos apkopos vairākas tam nodotās vērtības, kas tiek saņemtas īsā laika intervālā, un nosūtīs tās serverim vienā savienojumā. Vērtību, kas no iepriekšējās vērtības neatšķiras vairāk kā par 0,2 sekundēm, var ievietot tajā pašā kopā, taču maksimālais aptaujas laiks joprojām ir 1 sekunde.

> **Note:**
> Zabbix sender tiks pārtraukts, ja norādītajā konfigurācijas failā būs nederīgs parametra ieraksts (kas neatbilst *parameter=value* pierakstam).

Zabbix sender palaišana ar zema līmeņa atklāšanu

Piemērs, kā palaist Zabbix sender, lai nosūtītu JSON formātā noformētu vērtību zema līmeņa atklāšanai:

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

Lai tas darbotos, zema līmeņa atklāšanas kārtulai jābūt ar Zabbix trapper vienuma tipu (šajā piemērā ar atslēgu `trapper.discovery.item`).

7 Iegūt

Pārskats

Zabbix get ir komandrindas utilīta, ko var izmantot saziņai ar Zabbix aģents un nepieciešamās informācijas iegūšanai no aģents.

Šo utilītu parasti izmanto Zabbix aģenti problēmu novēršanai.

Skatiet arī Python bibliotēku Zabbix vajadzībām, kurai ir iebūvēta funkcionalitāte darboties kā Zabbix get.

Zabbix get palaišana

Piemērs, kā UNIX vidē palaist Zabbix get, lai no aģenta iegūtu procesora slodzes vērtību:

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

Vēl viens piemērs, kā palaist Zabbix get, lai iegūtu virkni no tīmekļa vietnes:

```
cd bin
./zabbix_get -s 192.168.1.1 -p 10050 -k "web.page.regexp[www.example.com,,,\"USA: ([a-zA-Z0-9.-]+)\",,\1]"
```

Ņemiet vērā, ka šeit vienuma atslēga satur atstarpi, tāpēc pēdiņas tiek izmantotas, lai čaula varētu atpazīt vienuma atslēgu. Pēdiņas nav daļa no vienuma atslēgas; čaula tās noņems, un tās netiks nodotas Zabbix aģentam.

Ja vienuma atslēga netiek atbalstīta, Zabbix get atgriezīs izejas kodu 1.

Zabbix get pieņem šādus komandrindas parametrus:

```
-s --host <host name or IP>             Norādiet hosta nosaukumu vai hosta IP adresi
-p --port <port number>                 Norādiet uz hosta darbojošā aģenta porta numuru (noklusējums: 1005
-I --source-address <IP address>        Norādiet avota IP adresi
-t --timeout <seconds>                  Norādiet noildzi. Derīgais diapazons: 1-600 sekundes (noklusējums:
-k --key <item key>                     Norādiet vienuma atslēgu, kurai iegūt vērtību
-P --protocol <value>                   Protokols, ko izmanto saziņai ar aģentu. Vērtības:
                                            auto - pieslēgties, izmantojot JSON protokolu, neveiksmes gadī
                                            json - pieslēgties, izmantojot JSON protokolu
                                            plaintext - pieslēgties, izmantojot vienkāršā teksta protokolu
-h --help                               Parādīt šo palīdzības ziņojumu
-V --version                            Parādīt versijas numuru

--tls-connect <value>                   Kā pieslēgties aģentam. Vērtības:
                                            unencrypted - pieslēgties bez šifrēšanas (noklusējums)
                                            psk - pieslēgties, izmantojot TLS un iepriekš koplietotu atslē
                                            cert - pieslēgties, izmantojot TLS un sertifikātu
--tls-ca-file <CA file>                 Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus otr
--tls-crl-file <CRL file>               Pilns ceļš uz failu, kas satur atsauktos sertifikātus
--tls-agent-cert-issuer <cert issuer>   Atļautais aģenta sertifikāta izdevējs
--tls-agent-cert-subject <cert subject> Atļautais aģenta sertifikāta subjekts
```

```
--tls-cert-file <cert file>              Pilns ceļš uz failu, kas satur sertifikātu vai sertifikātu ķēdi
--tls-key-file <key file>                Pilns ceļš uz failu, kas satur privāto atslēgu
--tls-psk-identity <PSK-identity>        Unikāla, reģistrjutīga virkne, ko izmanto iepriekš koplietotās ats
--tls-psk-file <PSK-file>                Pilns ceļš uz failu, kas satur iepriekš koplietoto atslēgu
--tls-cipher13 <cipher-string>           Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 vajadzībā
--tls-cipher <cipher-string>             GnuTLS prioritāšu virkne (TLS 1.2 un jaunākām versijām) vai OpenSS
```

Skatiet arī Zabbix get manpage, lai iegūtu vairāk informācijas.

Zabbix get sistēmā Windows var palaist līdzīgi:

```
zabbix_get.exe [options]
```

8 JS

Pārskats

zabbix_js ir komandrindas utilīta, ko var izmantot iegulto skriptu testēšanai.

Šī utilīta izpildīs lietotāja skriptu ar virknes parametru un izdrukās rezultātu. Skripti tiek izpildīti, izmantojot iegulto Zabbix skriptošanas dzini.

Kompilācijas vai izpildes kļūdu gadījumā zabbix_js izdrukās kļūdu stderr un beigs darbu ar kodu 1.

Lietošana

```
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 pieņem šādus komandrindas parametrus:

```
-s, --script script-file          Norādiet izpildāmā skripta faila nosaukumu. Ja kā faila nosaukums ir nor
-i, --input input-file            Norādiet ievades informācijas faila nosaukumu. Saturs tiks nodots skript
-p, --param input-param           Norādiet ievades parametru: mainīgo, kas tiks nodots skriptam kā vērtība
-l, --loglevel log-level          Norādiet žurnālošanas līmeni. Diapazons: 0-5.
-t, --timeout timeout             Norādiet noildzi sekundēs. Derīgais diapazons: 1-600 sekundes (noklusēju
-h, --help                        Parādīt palīdzības informāciju.
-V, --version                     Parādīt versijas numuru.
-w <webdriver url>                Iespējo pārlūka uzraudzību.
```

Piemēri:

1. piemērs: Skripta izpilde ar mainīgo, kas satur vērtību

Skripts (`script-file.js`):

```
return value;
```

Lietošana:

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

Izvade: `example_value`

2. piemērs: Skripta izpilde ar failu, kas satur ievades informāciju

Fails ar skriptu (`script-file.js`):

```
return value;
```

Fails ar ievades informāciju (`example.txt`):

```
Example of input information from the file
```

Lietošana:

```
zabbix_js -s script-file.js -i example.txt
```

Izvade: `Example of input information from the file`

3. piemērs: Skripta izpilde ar ievadi, kas nolasīta no stdin

Fails ar skriptu (`script-file.js`):

```
return value;
```

Lietošana:

```
zabbix_js -s script-file.js -i -
```

Ievade (stdin):

```
Example of input from stdin
```

Izvade: `Example of input from stdin`

9 Tīmekļa pakalpojums

Pārskats

Zabbix tīmekļa pakalpojums ir process, ko izmanto saziņai ar ārējiem tīmekļa pakalpojumiem. Pašlaik Zabbix tīmekļa pakalpojums tiek izmantots, lai ģenerētu un nosūtītu ieplānotos pārskatus, un nākotnē ir plānots pievienot papildu funkcionalitāti.

Zabbix serveris savienojas ar tīmekļa pakalpojumu, izmantojot HTTP(S). Zabbix tīmekļa pakalpojumam ir nepieciešams, lai tajā pašā hostā būtu instalēts Google Chrome; dažās distribūcijās pakalpojums var darboties arī ar Chromium (skatiet zināmās problēmas).

Instalēšana

Oficiālā zabbix-web-service pakotne ir pieejama Zabbix repozitorijā.

Lai kompilētu Zabbix tīmekļa servisu no pirmkoda, norādiet `--enable-webservice` konfigurēšanas opciju.

Lai konfigurētu Zabbix tīmekļa servisu, atjauniniet konfigurācijas faila `zabbix_web_service.conf` parametrus.

> **Attention:**
> Ir stingri ieteicams iestatīt šifrēšanu starp Zabbix serveri un Zabbix tīmekļa servisu, izmantojot sertifikātus. Pēc noklusējuma dati, kas tiek pārsūtīti starp Zabbix serveri un Zabbix tīmekļa servisu, netiek šifrēti, kas var izraisīt nesankcionētu piekļuvi.

5 Konfigurācija

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam sadaļā Konfigurācija.

1 Veidnes konfigurēšana

Pārskats

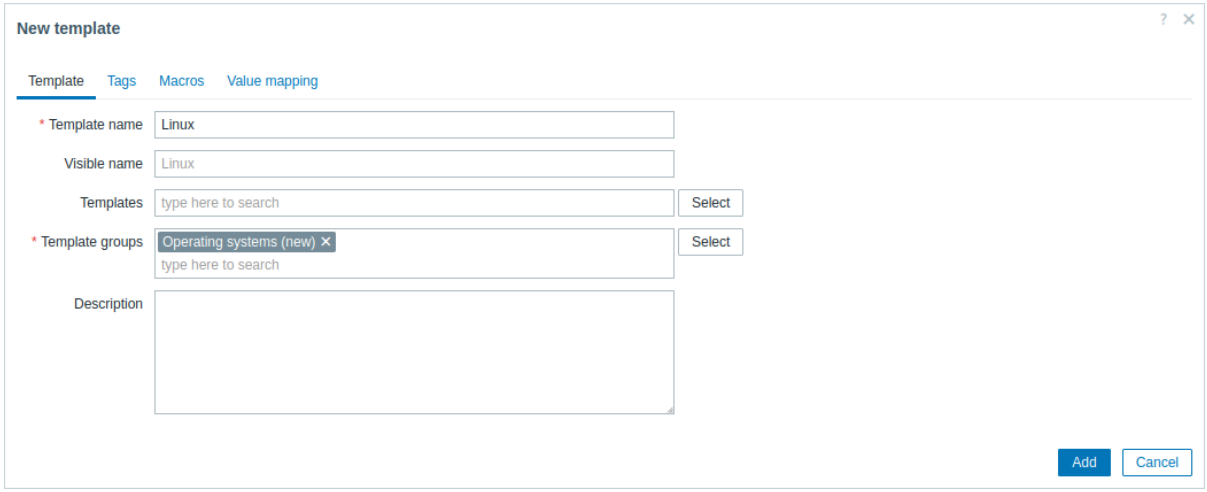
Veidnes konfigurēšanai vispirms ir jāizveido veidne, definējot tās vispārīgos parametrus, un pēc tam tai jāpievieno entītijas (vienumi, trigeri, grafiki u.c.).

Veidnes izveide

Lai izveidotu veidni, rīkojieties šādi:

1. Dodieties uz *Datu ievākšana* > *Veidnes*.
2. Noklikšķiniet uz *Izveidot veidni*.
3. Rediģējiet veidnes atribūtus.

Cilnē **Veidne** ir ietverti vispārīgie veidnes atribūti.

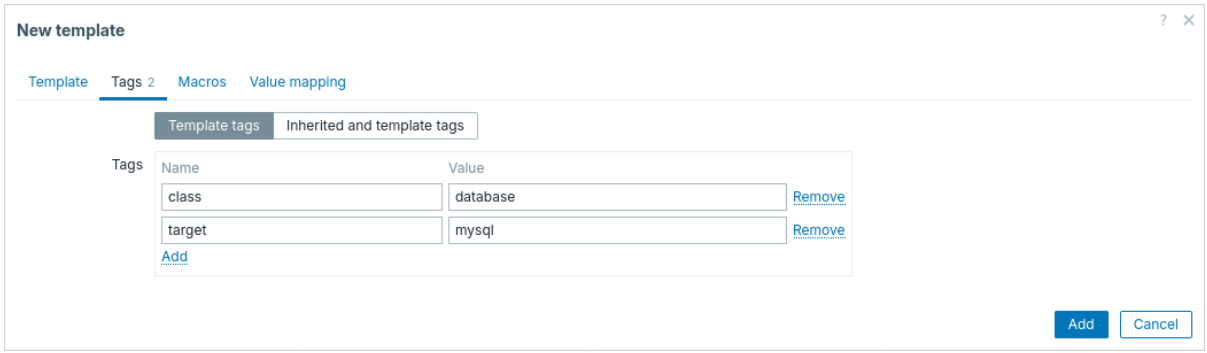


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

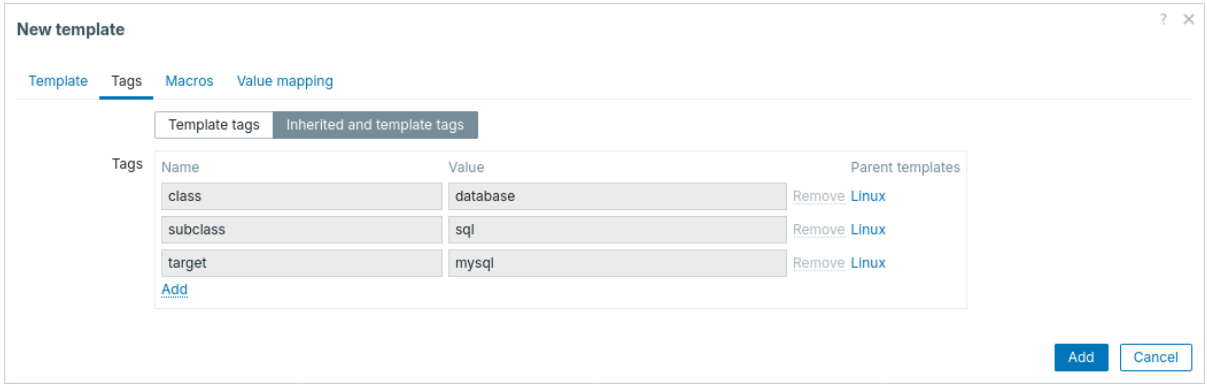
Veidnes atribūti:

Parametrs	Apraksts
Veidnes nosaukums	Unikāls veidnes nosaukums. Atļauti burti un cipari, atstarpes, punkti, defises un pasvītrojuma zīmes. Atstarpes sākumā un beigās nav atļautas.
Redzamais nosaukums	Ja iestatīsiet šo nosaukumu, tas būs redzams sarakstos, kartēs utt.
Veidnes	Saistiet ar šo veidni vienu vai vairākas veidnes. Visas entītijas (vienumi, trigeri utt.) tiks mantotas no saistītajām veidnēm. Lai saistītu jaunu veidni, ievadiet tās nosaukumu laukā *Veidnes* — tiks parādīts atbilstošo veidņu nolaižamais saraksts. Varat arī noklikšķināt uz *Atlasīt*, lai atvērtu uznirstošo logu *Veidnes*, kurā sākotnēji netiek rādīta neviena veidne. Lai aizpildītu sarakstu, vai nu ievadiet veidņu grupas nosaukumu laukā *Veidņu grupa* (un atlasiet to no piedāvātā saraksta), vai nospiediet pogu *Atlasīt* blakus laukam *Veidņu grupa*, lai atvērtu uznirstošo logu *Veidņu grupas*. Kad būsiet izvēlējies veidņu grupu, uznirstošais logs *Veidnes* tiks atjaunināts un rādīs šai grupai piederošās veidnes. Atlasītās veidnes tiks saistītas, kad konfigurācija tiks saglabāta vai atjaunināta. Lai atsaistītu veidni, izmantojiet vienu no divām opcijām blokā *Veidnes*: *Atsaistīt* - atsaistīt veidni, bet saglabāt tās entītijas (vienumus, trigerus utt.); *Atsaistīt un notīrīt* - atsaistīt veidni un noņemt visas tās entītijas (vienumus, trigerus utt.).
Veidņu grupas	Veidņu grupas, kurām pieder veidne.
Apraksts	Veidnes apraksts.
Piegādātājs un versija	Veidnes piegādātājs un versija; tiek rādīti tikai tad, kad tiek atjauninātas esošās veidnes (gatavās veidnes, ko nodrošina Zabbix, importētās veidnes vai veidnes, kas modificētas, izmantojot Template API), ja veidnes konfigurācijā ir šāda informācija. Nevar modificēt Zabbix lietotāja saskarne. Gatavām veidnēm versija tiek rādīta šādi: Zabbix galvenā versija, atdalītājs (”-”), revīzijas numurs (palielinās ar katru jaunu veidnes versiju un tiek atiestatīts ar katru jaunu Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-5, 8.0-0, 8.0-3.

Cilne **Birkas** ļauj definēt veidnes līmeņa birkas. Visas ar šo veidni saistīto hostu problēmas tiks marķētas ar šeit ievadītajām vērtībām.



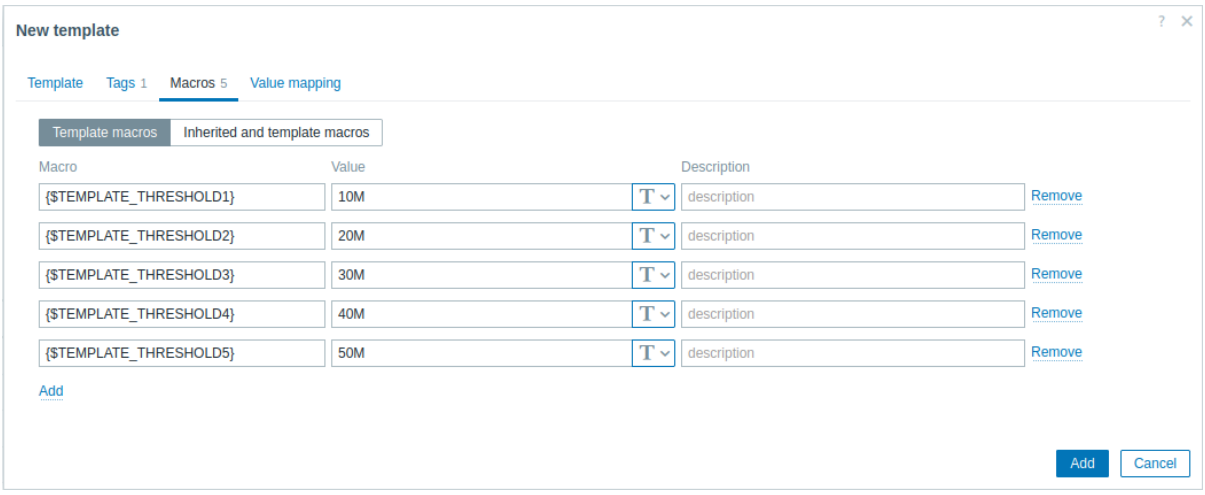
Ja atlasīsiet opciju *Mantotās un veidnes birkas*, varēsiet skatīt arī birkas no saistītajām veidnēm, kuras veidne mantos.



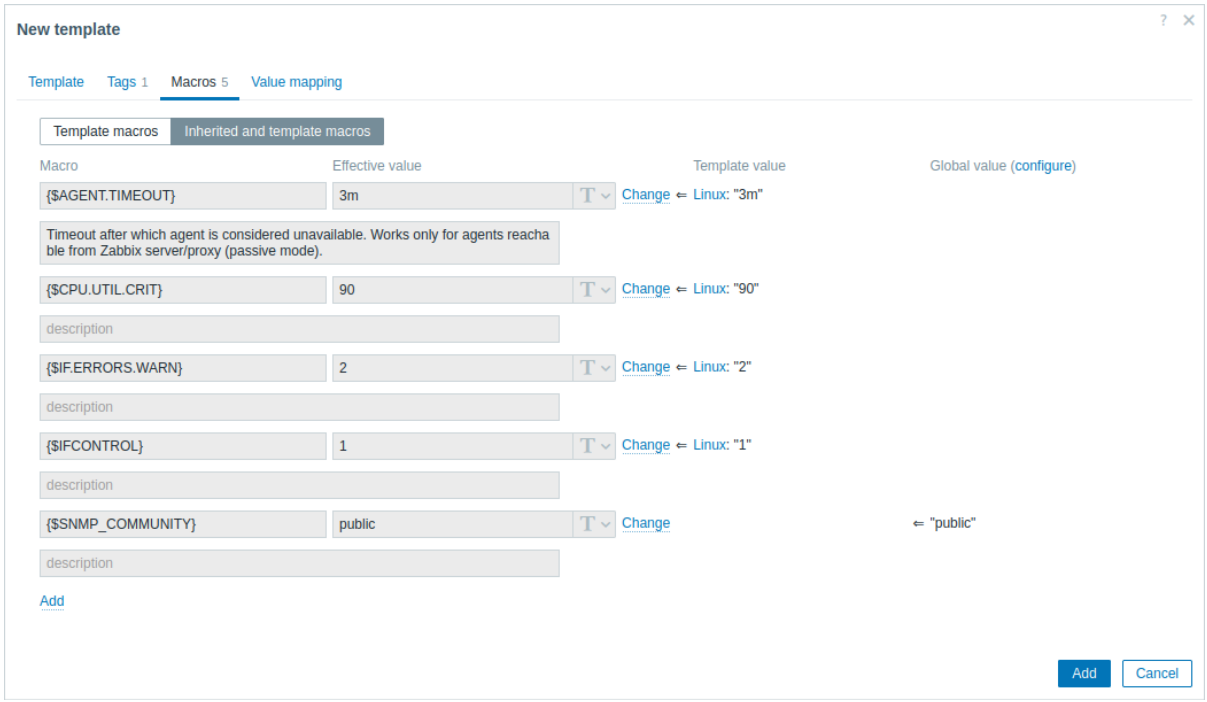
Ērtībai ir nodrošinātas saites uz attiecīgajām veidnēm.

Birkās tiek atbalstīti lietotāja makrosi, {INVENTORY.*} makrosi, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} makrosi.

Cilne **Makrosi** ļauj definēt veidnes līmeņa lietotāja makrosus kā nosaukuma-vērtības pārus. Ņemiet vērā, ka makrosu vērtības var tikt glabātas kā vienkāršs teksts, slepens teksts vai Vault noslēpums. Tiek atbalstīta arī apraksta pievienošana.



Ja atlasīsiet opciju *Mantotie un veidnes makrosi*, varēsiet skatīt arī makrosus no saistītajām veidnēm un globālos makrosus, kurus veidne mantos, kā arī vērtības, uz kurām makrosi tiks atrisināti.



Ērtībai ir nodrošinātas saites uz attiecīgajām veidnēm, kā arī saite uz globālo makrosu konfigurāciju. Ir iespējams arī rediģēt saistītās veidnes makrosu vai globālo makrosu veidnes līmenī, tādējādi faktiski izveidojot makrosa kopiju veidnē.

Cilne **Vērtību kartēšana** ļauj konfigurēt vienumu datu lietotājam draudzīgu attēlojumu vērtību kartēšanā.

Pogas:

Add	Pievienot veidni. Pievienotajai veidnei jāparādās sarakstā.
Update	Atjaunināt esošas veidnes rekvizītus.
Clone	Izveidot citu veidni, pamatojoties uz pašreizējās veidnes rekvizītiem. Tas **ietver** entītijas (vienumus, trigerus utt.), kas gan mantotas no saistītajām veidnēm, gan tieši pievienotas pašreizējai veidnei, bet **neietver** pašreizējās veidnes piegādātāju un versiju klonētajai veidnei, lai to varētu atšķirt no oriģināla.
Delete	Dzēst veidni; veidnes entītijas (vienumi, trigeri utt.) paliek saistītajos hostos.
Delete and clear	Dzēst veidni un visas tās entītijas no saistītajiem hostiem.
Cancel	Atcelt veidnes rekvizītu rediģēšanu.

Vienumu, trigeru un grafiku pievienošana

Attention:
Vispirms vienumi jāpievieno veidnei. Trigerus un grafikus nevar pievienot bez atbilstošā vienuma.

Ir divi veidi, kā pievienot vienumus veidnei:

1. Lai izveidotu jaunus vienumus, sekojiet norādījumiem sadaļā Vienuma izveide.
2. Lai pievienotu esošus vienumus veidnei:

- Dodieties uz *Datu ievākšana* > *Hosti* (vai *Veidnes*).
- Nepieciešamā hosta/veidnes rindā noklikšķiniet uz *Vienumi*.
- Atzīmējiet to vienumu izvēles rūtiņas, kurus vēlaties pievienot veidnei.
- Zem vienumu saraksta noklikšķiniet uz *Kopēt*.
- Atlasiet veidni (vai veidņu grupu), uz kuru vienumi jākopē, un noklikšķiniet uz *Kopēt*.
 Visi atlasītie vienumi ir jākopē uz veidni.

Trigeru un grafiku pievienošana tiek veikta līdzīgā veidā (attiecīgi no trigeru un grafiku saraksta), atkal paturot prātā, ka tos var pievienot tikai tad, ja vispirms ir pievienoti nepieciešamie vienumi.

Informācijas paneļu pievienošana

Lai pievienotu informācijas paneļus veidnei sadaļā *Datu ievākšana* > *Veidnes*, rīkojieties šādi:

1. Veidnes rindā noklikšķiniet uz *Informācijas paneļi*.
2. Konfigurējiet informācijas paneli, ievērojot informācijas paneļu konfigurēšanas vadlīnijas.

Attention:
Konfigurējot logrīkus veidnes informācijas panelī (nevis globālajā informācijas panelī), ar hostu saistītie parametri nav pieejami, un dažiem parametriem ir atšķirīgs apzīmējums. Tas ir tāpēc, ka veidnes informācijas paneļi attēlo datus tikai no hosta, ar kuru veidne ir saistīta. Piemēram, logrīkā *Problēmas* parametri *Hostu grupas*, *Izslēgt hostu grupas* un *Hosti* nav pieejami, logrīkā *Hostu pieejamība* parametrs *Hostu grupas* nav pieejams, un parametrs *Rādīt hostus uzturēšanā* ir pārdēvēts par *Rādīt datus uzturēšanā* utt. Plašāku informāciju par parametru pieejamību veidnes informācijas paneļu logrīkos skatiet katra informācijas paneļa logrīka konkrētajos parametros.

Note:
Lai iegūtu informāciju par piekļuvi hostu informācijas paneļiem, kas izveidoti no veidņu informācijas paneļiem, skatiet sadaļu hostu informācijas paneļi.

Zema līmeņa atklāšanas noteikumu konfigurēšana

Skatiet rokasgrāmatas sadaļu zema līmeņa atklāšana.

Web scenāriju pievienošana

Lai pievienotu web scenārijus veidnei sadaļā *Datu ievākšana* > *Veidnes*, rīkojieties šādi:

1. Veidnes rindā noklikšķiniet uz *Web*.
2. Konfigurējiet web scenāriju, izmantojot parasto web scenāriju konfigurēšanas metodi.

2 Veidņu grupas konfigurēšana

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.

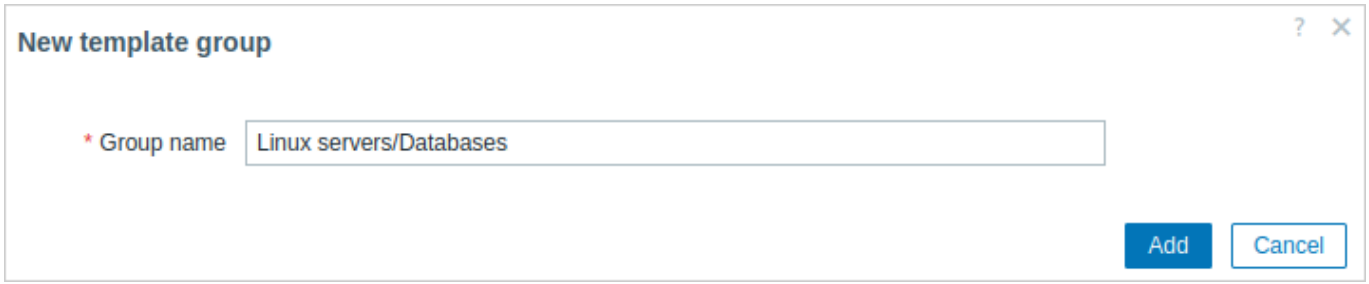
Konfigurācija

> **Attention:**
> Tikai Super admin lietotāji var izveidot veidņu grupas.

Ir divas iespējas, kā Zabbix lietotāja saskarnē izveidot veidņu grupu.

Pirmā iespēja:

- Dodieties uz: *Datu ievākšana* → *Veidņu grupas*
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot veidņu grupu*
- Veidlapā ievadiet grupas nosaukumu



Otrā iespēja: konfigurējot veidni, laukā *Veidņu grupas* ievadiet neeksistējošas grupas nosaukumu.

Kad veidņu grupa ir izveidota, varat noklikšķināt uz veidnes nosaukuma sarakstā sadaļā *Datu ievākšana* → *Veidņu grupas*, lai rediģētu grupas nosaukumu, klonētu grupu vai dzēstu grupu.

Dzēšot veidņu grupu, tiek dzēsta tikai loģiskā grupa, nevis grupā esošās veidnes. Nav iespējams dzēst veidņu grupu, kas ir vienīgā grupa kādai esošai veidnei.

Veidņu apakšgrupu izveide

Veidnes apakšgrupa (jeb ligzdota veidnes grupa) ir vecākgrupas veidnes grupas bērngrupa, kurā tā atrodas.

Apakšgrupa tiek izveidota, grupas nosaukuma ievades laukā izmantojot slīpsvītru '/' , lai norādītu tās saistību ar vecākgrupu(-ām). Piemēram:

- ievadot `Linux servers/Databases`, tiek izveidota apakšgrupa `Linux servers/Databases` vecākgrupā `Linux servers`.
- ievadot `Linux servers/Databases/MySQL/Tokyo`, tiek izveidota attiecīgā apakšgrupa ligzdotajās vecākgrupās `Linux servers`, `Linux servers/Databases`, `Linux servers/Databases/MySQL`.

Veidojot apakšgrupu, nav atļauts izmantot slīpsvītras sākumā vai beigās, kā arī vairākas slīpsvītras pēc kārtas. '/' ekrānēšana netiek atbalstīta.

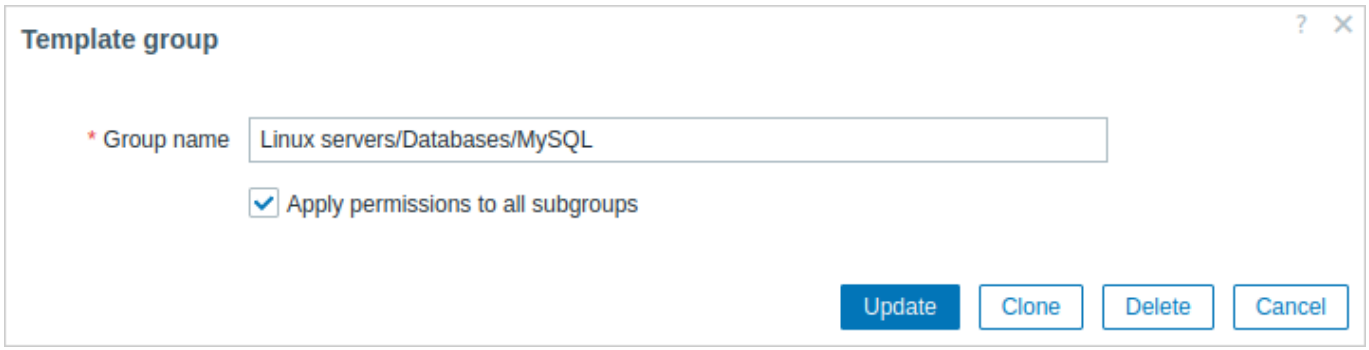
Pirms apakšgrupas izveides nav nepieciešams izveidot nevienu vecākgrupas veidnes grupu. Varat izvēlēties, vai sākt ar apakšgrupas izveidi (piemēram, `Linux servers/Databases`) vai ar jebkuras vecākgrupas veidnes grupas izveidi (mūsu piemērā — `Linux servers`). Ja sāksiet ar apakšgrupas izveidi, vecākgrupas veidnes grupa(-as) **netiks** izveidota(-as) automātiski.

Atļaujas veidņu grupām

- Veidojot apakšgrupu esošai vecākajai veidņu grupai (piemēram, izveidojot `Linux servers/Databases`, kad `Linux servers` jau pastāv), lietotāju grupas atļaujas apakšgrupai tiek mantotas no vecākās grupas.

- Veidojot vecāko veidņu grupu esošai apakšgrupai (piemēram, izveidojot `Linux servers`, kad `Linux servers/Databases` jau pastāv), vecākajai grupai netiek iestatītas nekādas atļaujas.

Rediģējot jebkuru veidņu grupu, varat iestatīt arī papildu opciju *Piemērot atļaujas visām apakšgrupām*.



Atzīmējot šo izvēles rūtiņu un noklikšķinot uz *Atjaunināt*, visām pašreizējām un turpmākajām rediģējamās veidņu grupas apakšgrupām tiks piemērots vienāds atļauju līmenis.

Tātad, ja kādām lietotāju grupām ir piešķirtas atšķirīgas atļaujas rediģējamās veidņu grupas apakšgrupām, atzīmējot šo izvēles rūtiņu, visām pašreizējām un turpmākajām apakšgrupām tiks piešķirtas tādas pašas lietotāju atļaujas kā rediģējamajai grupai.

Ņemiet vērā, ka šī opcija netiek saglabāta datubāzē un pārrakstīs esošās atļaujas. Visas ar šo opciju veiktās izmaiņas var atsaukt tikai manuāli.

3 Saišu izveide/noņemšana

Pārskats

Piesaistīšana ir process, kurā veidnes tiek piemērotas hostiem, savukārt atsaistīšana noņem hosta saistību ar veidni.

Veidnes piesaistīšana

Lai piesaistītu veidni hostam, rīkojieties šādi:

1. Atveriet *Datu ievākšana* → *Hosti*.
2. Noklikšķiniet uz vajadzīgā hosta.
3. Laukā *Veidnes* sāciet rakstīt veidnes nosaukumu. Tiks parādīts atbilstošo veidņu saraksts; ritiniet uz leju, lai atlasītu. Alternatīvi varat noklikšķināt uz *Atlasīt* blakus laukam un uznirstošajā logā atlasīt vienu vai vairākas veidnes no saraksta.
4. Hosta atribūtu formā noklikšķiniet uz *Pievienot/Atjaunināt*.

Hostam tagad būs visas veidnes entītijas. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas kārtulas, tīmekļa scenārijus, kā arī paneļus.

Attention:
Vairāku veidņu piesaistīšana vienam un tam pašam hostam neizdosies, ja šajās veidnēs ir vienumi ar vienādu vienuma atslēgu. Tā kā trigeri un grafiki izmanto vienumus, arī tos nevar piesaistīt vienam hostam no vairākām veidnēm, ja tiek izmantotas identiskas vienumu atslēgas.

Kad no veidnes tiek pievienotas entītijas (vienumi, trigeri utt.):

- iepriekš esošās identiskās entītijas hostā tiek atjauninātas kā veidnes entītijas, un **visi esošie hosta līmeņa entītijas pielāgojumi tiek zaudēti**;
- tiek pievienotas entītijas no veidnes;
- visas tieši piesaistītās entītijas, kas pirms veidnes piesaistīšanas eksistēja tikai hostā, paliek neskartas.

Sarakstos visām entītijām no veidnes tagad priekšā ir veidnes nosaukums, kas norāda, ka tās pieder konkrētajai veidnei. Pats veidnes nosaukums (pelēkā tekstā) ir saite, kas ļauj piekļūt šo entītiju sarakstam veidnes līmenī.

Note:
Dažiem vienumiem, piemēram, ārējām pārbaudēm, HTTP aģenta pārbaudēm, vienkāršajām pārbaudēm, SSH pārbaudēm un Telnet pārbaudēm, hosta saskarne nav obligāta. Ja veidnes piesaistīšanas brīdī hostam nav definēta saskarne, šie vienumi tiks pievienoti bez saskarnes. Ja vēlāk pievienosiet hosta saskarni, tā netiks automātiski piešķirta jau esošajiem vienumiem. Lai tikko pievienoto hosta saskarni piešķirtu visiem veidnes vienumiem uzreiz, atsaistiet veidni no hosta un pēc tam piesaistiet to vēlreiz. Lai saglabātu vienumu vēsturi, izmantojiet opciju *Atsaistīt*, nevis *Atsaistīt un notīrīt*.

Ja kādai entītijai priekšā nav veidnes nosaukuma, tas nozīmē, ka tā hostā eksistēja jau iepriekš un netika pievienota no veidnes.

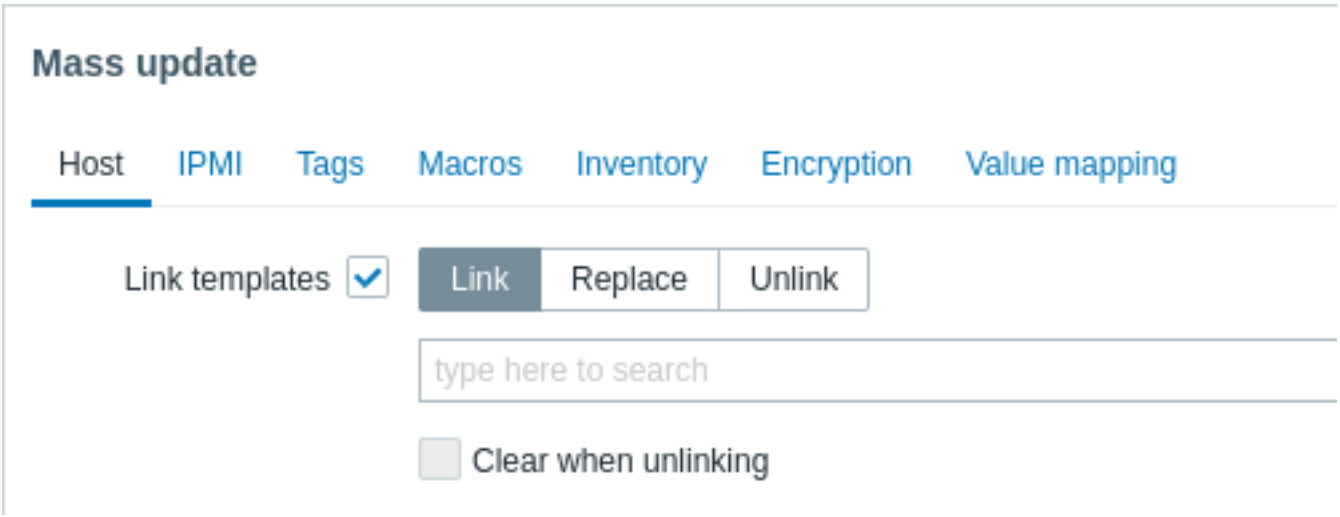
Entītiju unikalitātes kritēriji

Pievienojot entītijas (vienumus, trigerus u.c.) no veidnes, ir svarīgi zināt, kuras no šīm entītijām jau pastāv hostā un ir jāatjaunina, un kuras entītijas atšķiras. Unikalitātes kritēriji, pēc kuriem nosaka vienādību/atšķirību, ir šādi:

- vienumiem — vienuma atslēga;
- trigeriem — trigeru nosaukums un izteiksme;
- pielāgotajiem grafikiem — grafika nosaukums un tā vienumi.

Veidņu piesaiste vairākiem hostiem

Lai atjauninātu daudzu hostu veidņu piesaisti, sadaļā *Datu ievākšana* → *Hosti* atlasiet vairākus hostus, atzīmējot to izvēles rūtiņas, pēc tam zem saraksta noklikšķiniet uz **Masveida atjaunināšana** un tad atlasiet *Piesaistīt veidnes*:



Lai piesaistītu papildu veidnes, sāciet rakstīt veidnes nosaukumu automātiskās pabeigšanas laukā, līdz parādās nolaižamais saraksts ar atbilstošajām veidnēm. Vienkārši ritiniet uz leju, lai atlasītu piesaistāmo veidni.

Opcija *Aizstāt* ļaus piesaistīt jaunu veidni, vienlaikus atsaistot jebkuru veidni, kas iepriekš bija piesaistīta hostiem. Opcija *Atsaistīt* ļaus norādīt, kuras veidnes atsaistīt. Opcija *Notīrīt atsaistot* ļaus ne tikai atsaistīt jebkuras iepriekš piesaistītās veidnes, bet arī noņemt visas no tām mantotās entītijas (vienumus, trigerus utt.).

Note:
Zabbix piedāvā plašu iepriekš definētu veidņu kopu. Tās var izmantot kā atsauci, taču esiet piesardzīgi, izmantojot tās nemainītā veidā produkcijas vidē, jo tās var saturēt pārāk daudz vienumu un pārāk bieži pieprasīt datus. Ja tomēr vēlaties tās izmantot, pielāgojiet tās savām reālajām vajadzībām.

Saistīto entītiju rediģēšana

Ja mēģināsiet rediģēt vienumu vai trigeri, kas tika saistīts no veidnes, varat pamanīt, ka daudzas galvenās opcijas rediģēšanai nav pieejamas. Tas ir loģiski, jo veidņu ideja ir tāda, ka viss tiek rediģēts centralizēti veidnes līmenī. Tomēr jūs joprojām varat, piemēram, iespējot/atspējot vienumu atsevišķos hostos un iestatīt atjaunināšanas intervālu, vēstures glabāšanas ilgumu un dažus citus parametrus.

Ja vēlaties pilnībā rediģēt entītiju, tā ir jārediģē veidnes līmenī (veidnes līmeņa saīsne tiek parādīta formas nosaukumā), ņemot vērā, ka šīs izmaiņas ietekmēs visus hostus, kuriem šī veidne ir piesaistīta.

Attention:
Jebkādi entītiju pielāgojumi, kas veikti veidnes līmenī, pārrakstīs iepriekšējos entītiju pielāgojumus hosta līmenī.

Veidnes atsaistīšana

Lai atsaistītu veidni no hosta, rīkojieties šādi:

1. Dodieties uz *Datu ievākšana* → *Hosti*.
2. Noklikšķiniet uz vajadzīgā hosta un atrodiet lauku *Veidnes*.
3. Noklikšķiniet uz *Atsaistīt* vai *Atsaistīt un notīrīt* blakus veidnei, lai atsaistītu veidni.
4. Hosta atribūtu formā noklikšķiniet uz *Atjaunināt*.

Izvēloties opciju *Atsaistīt*, tiks vienkārši noņemta saistība ar veidni, vienlaikus atstājot visas tās entītijas pie hosta. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas kārtulas un tīmekļa scenārijus, bet neietver paneļus. Ņemiet vērā, ka tiks noņemtas arī no piesaistītajām veidnēm mantotās vērtību kartes un birkas.

Izvēloties opciju *Atsaistīt un notīrīt*, tiks noņemta gan saistība ar veidni, gan visas tās entītijas (vienumi, trigeri utt.).

4 Ligzdošana

Pārskats

Ligzdošana ir veids, kā viena veidne ietver vienu vai vairākas citas veidnes.

Tā kā ir lietderīgi nodalīt entītijas atsevišķās veidnēs dažādiem pakalpojumiem, lietotnēm utt., jums var izveidoties diezgan daudz veidņu, kuras visas var būt jāpiesaista diezgan daudziem hostiem. Lai vienkāršotu šo ainu, ir iespējams sasaistīt vairākas veidnes kopā vienā veidnē.

Ligzdošanas priekšrocība ir tāda, ka hostam jāpiesaista tikai viena veidne, un hosts automātiski mantos visas entītijas no veidnēm, kas ir piesaistītas šai vienai veidnei. Piemēram, ja mēs piesaistām *T1* un *T2* pie *T3*, mēs papildinām *T3* ar visām entītijām no *T1* un *T2*, bet ne otrādi. Ja mēs piesaistām *T1* pie *T2* un *T3*, mēs papildinām *T2* un *T3* ar entītijām no *T1*.

Ligzdotu veidņu konfigurēšana

Lai sasaistītu veidnes, jums jāizmanto esoša veidne (vai jāizveido jauna) un pēc tam:

1. Atveriet veidnes konfigurācijas formu.
2. Atrodiet lauku *Templates*.
3. Noklikšķiniet uz *Select*, lai atvērtu uznirstošo logu *Templates*.
4. Uznirstošajā logā izvēlieties nepieciešamās veidnes un pēc tam noklikšķiniet uz *Select*, lai pievienotu veidnes sarakstam.
5. Veidnes konfigurācijas formā noklikšķiniet uz *Add* vai *Update*.

Tādējādi visas konfigurētās veidnes entītijas, kā arī visas sasaistīto veidņu entītijas tagad būs redzamas veidnes konfigurācijā. Tas ietver vienumus, trigerus, grafikus, zema līmeņa atklāšanas kārtulas un tīmekļa scenārijus, bet neietver paneļus. Tomēr sasaistīto veidņu paneļi tik un tā tiks mantoti hostiem.

Lai atsaistītu jebkuru no sasaistītajām veidnēm, veidnes konfigurācijas formā noklikšķiniet uz *Unlink* vai *Unlink and clear* un pēc tam noklikšķiniet uz *Update*.

Opcija *Unlink* vienkārši noņems saistību ar sasaistīto veidni, bet nenoņems nevienu no tās entītijām (vienumiem, trigeriem utt.).

Opcija *Unlink and clear* noņems gan saistību ar sasaistīto veidni, gan visas tās entītijas (vienumus, trigerus utt.).

5 Masveida atjaunināšana

Pārskats

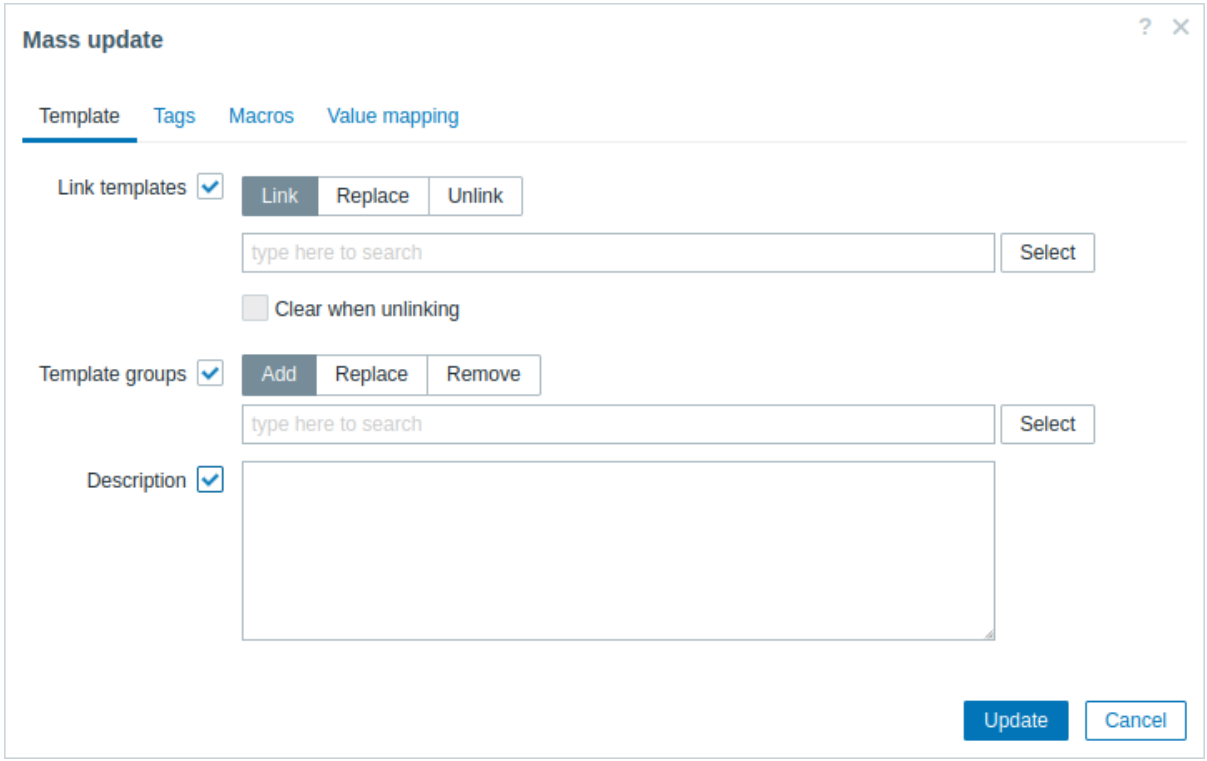
Dažreiz var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākām veidnēm. Tā vietā, lai atvērtu katru atsevišķo veidni rediģēšanai, šim nolūkam varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšanas izmantošana

Lai masveidā atjauninātu dažas veidnes, rīkojieties šādi:

1. Atzīmējiet izvēles rūtiņas pie veidnēm, kuras vēlaties atjaunināt veidņu sarakstā.
2. Zem saraksta noklikšķiniet uz *Masveida atjaunināšana*.
3. Pārejiet uz cilni ar nepieciešamajiem atribūtiem (*Veidne*, *Birkas*, *Makrosi* vai *Vērtību kartēšana*).
4. Atzīmējiet to atribūtu izvēles rūtiņas, kurus vēlaties atjaunināt, un ievadiet tiem jaunu vērtību.

Cilne **Veidne** ir ietvertas vispārīgas veidņu masveida atjaunināšanas opcijas.



Tālāk norādītās opcijas ir pieejamas, izvēloties attiecīgo pogu *Saistīt veidnes* atjauninājumam:

- *Saistīt* — norādiet, kuras papildu veidnes saistīt;
- *Aizstāt* — norādiet, kuras veidnes saistīt, vienlaikus atsaistot visas iepriekš saistītās veidnes;
- *Atsaistīt* — norādiet, kuras veidnes atsaistīt.

Lai norādītu saistāmās/atsaistāmās veidnes, sāciet rakstīt veidnes nosaukumu automātiskās pabeigšanas laukā, līdz tiek parādīts nolaižamais saraksts ar atbilstošajām veidnēm. Vienkārši ritiniet uz leju, lai atlasītu vajadzīgās veidnes.

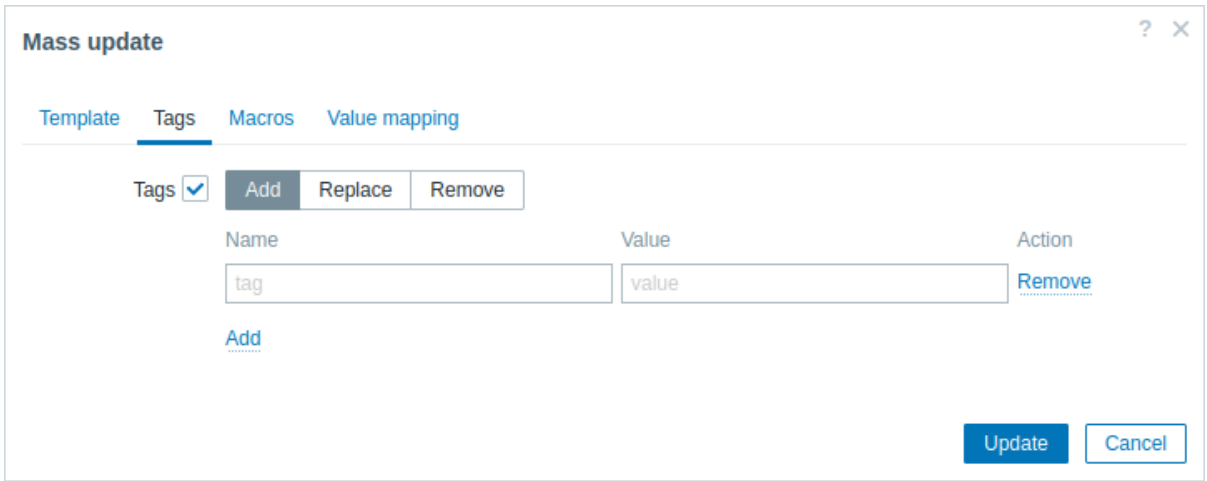
Opcija *Notīrīt atsaistot* ļaus atsaistīt visas iepriekš saistītās veidnes, kā arī noņemt visus no tām mantotos elementus (vienumus, trigerus, grafikus utt.).

Tālāk norādītās opcijas ir pieejamas, izvēloties attiecīgo pogu *Veidņu grupas* atjauninājumam:

- *Pievienot* — ļauj norādīt papildu veidņu grupas no esošajām vai ievadīt veidnēm pilnīgi jaunas veidņu grupas;
- *Aizstāt* — noņems veidni no visām esošajām veidņu grupām un aizstās tās ar šajā laukā norādīto(-ajām) grupu(-ām) (esošām vai jaunām veidņu grupām);
- *Noņemt* — noņems noteiktas veidņu grupas no veidnēm.

Šie lauki atbalsta automātisko pabeigšanu — sākot tajos rakstīt, tiek piedāvāts nolaižamais saraksts ar atbilstošajām veidņu grupām. Ja veidņu grupa ir jauna, tā arī parādās nolaižamajā sarakstā, un aiz virknes ir norāde *(new)*. Vienkārši ritiniet uz leju, lai atlasītu.

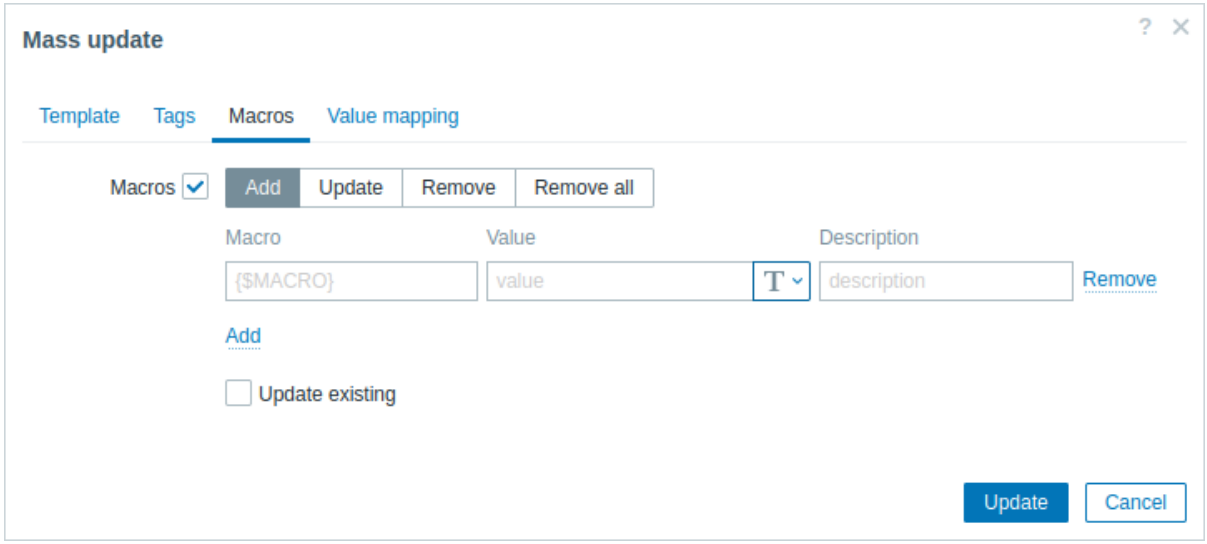
Cilne **Birkas** ļauj masveidā atjaunināt veidnes līmeņa birkas.



Lietotāja makrosi, {INVENTORY.*} makrosi, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT}

un {HOST.ID} makrosi tiek atbalstīti birkās. Ņemiet vērā, ka birkas ar vienādu nosaukumu, bet atšķirīgām vērtībām netiek uzskatītas par "dublikātiem" un tās var pievienot tai pašai veidnei.

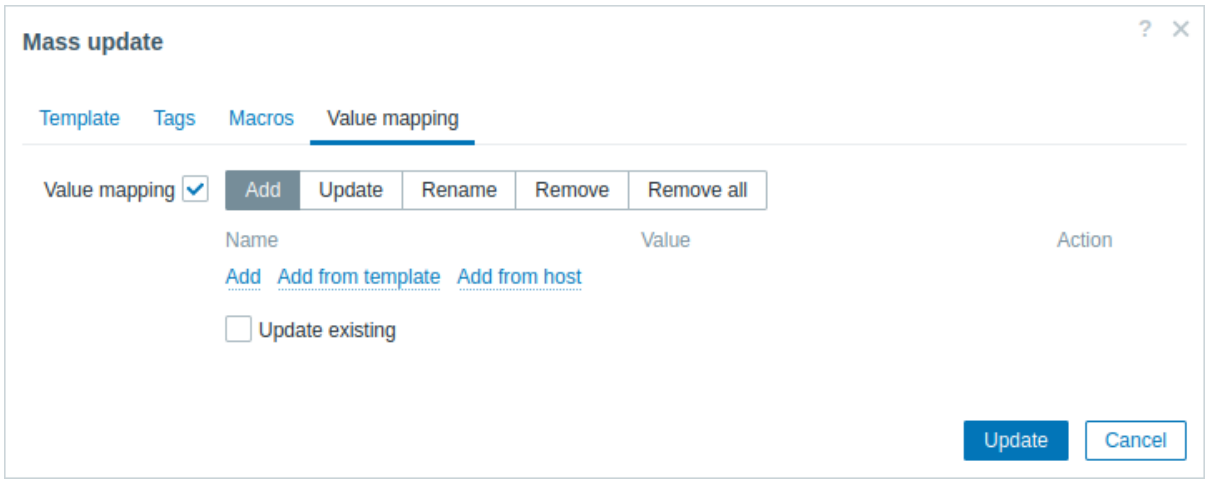
Cilne **Makrosi** ļauj masveidā atjaunināt veidņu līmeņa makrosus.



Izvēloties attiecīgo pogu makrosu atjaunināšanai, ir pieejamas šādas opcijas:

- *Pievienot* — ļauj norādīt papildu lietotāja makrosus veidnēm. Ja ir atzīmēta izvēles rūtiņa *Atjaunināt esošos*, tiks atjaunināta vērtība, tips un apraksts norādītajam makrosa nosaukumam. Ja tā nav atzīmēta un veidnē(-ēs) jau pastāv makross ar šādu nosaukumu, tas netiks atjaunināts.
- *Atjaunināt* — aizstās šajā sarakstā norādīto makrosu vērtības, tipus un aprakstus. Ja ir atzīmēta izvēles rūtiņa *Pievienot trūkstošos*, makross, kas iepriekš veidnē nepastāvēja, tiks pievienots kā jauns makross. Ja tā nav atzīmēta, tiks atjaunināti tikai tie makrosi, kas jau pastāv veidnē.
- *Noņemt* — noņems norādītos makrosus no veidnēm. Ja ir atzīmēta izvēles rūtiņa *Izņemot atlasītos*, tiks noņemti visi makrosi, izņemot sarakstā norādītos. Ja tā nav atzīmēta, tiks noņemti tikai sarakstā norādītie makrosi.
- *Noņemt visus* — noņems visus lietotāja makrosus no veidnēm. Ja izvēles rūtiņa *Es apstiprinu visu makrosu noņemšanu* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt visu makrosu noņemšanu.

Cilne **Vērtību kartēšana** varat veikt vērtību kartējumu masveida atjaunināšanu.



Vērtību kartējumu atjaunināšanai ir pieejamas pogas ar šādām opcijām:

- *Pievienot* — pievienot vērtību kartējumus veidnēm. Ja atzīmējat *Atjaunināt esošos*, tiks atjauninātas visas vērtību kartējuma ar šo nosaukumu īpašības. Pretējā gadījumā, ja vērtību kartējums ar šādu nosaukumu jau pastāv, tas netiks atjaunināts.
- *Atjaunināt* — atjaunināt esošos vērtību kartējumus. Ja atzīmējat *Pievienot trūkstošos*, vērtību kartējums, kas iepriekš veidnē nepastāvēja, tiks pievienots kā jauns vērtību kartējums. Pretējā gadījumā tiks atjaunināti tikai tie vērtību kartējumi, kas veidnē jau pastāv.
- *Pārdēvēt* — piešķirt jaunu nosaukumu esošam vērtību kartējumam.
- *Noņemt* — noņemt norādītos vērtību kartējumus no veidnēm. Ja atzīmējat *Izņemot atlasītos*, tiks noņemti visi vērtību kartējumi, **izņemot** norādītos.
- *Noņemt visus* — noņemt visus vērtību kartējumus no veidnēm. Ja izvēles rūtiņa *Es apstiprinu visu vērtību kartējumu noņemšanu* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt noņemšanu.

Vērtību kartējumu pievienošanas/atjaunināšanas darbībām ir pieejamas opcijas *Pievienot no veidnes* un *Pievienot no hosta*. Tās ļauj attiecīgi atlasīt vērtību kartējumus no veidnes vai hosta.

Kad visas nepieciešamās izmaiņas ir veiktas, noklikšķiniet uz *Atjaunināt*. Atribūti tiks attiecīgi atjaunināti visām atlasītajām veidnēm.

1 Hosti un hostu grupas

Kas ir ”hosts”?

Zabbix sistēmā ”hosts” attiecas uz jebkuru fizisku vai virtuālu ierīci, lietotni, pakalpojumu vai jebkuru citu loģiski saistītu uzraudzīto parametru kopu.

Hostu izveide ir viens no pirmajiem uzraudzības uzdevumiem Zabbix. Piemēram, ja vēlaties uzraudzīt dažus parametrus serverī ”x”, vispirms ir jāizveido hosts, piemēram, ar nosaukumu ”Server X”, un pēc tam tam var pievienot uzraudzības vienumus.

Hosti tiek organizēti hostu grupās.

Turpiniet ar:

- Hostu vednis — vadīta, soli pa solim izveidota lietotāja saskarne jauna vai esoša hosta izveidei un konfigurēšanai.
- Hosta konfigurēšana — klasiskā pieeja hosta izveidei un konfigurēšanai.

1 Hostu vednis

Pārskats

Host Wizard ir vadīta, soli pa solim veidota saskarne, kas paredzēta jūsu uzraudzības mērķa (ierīces, lietotnes, pakalpojuma u.c.) iestatīšanai Zabbix. Tā izved jūs cauri šādām darbībām:

- Veidnes izvēle
- Hosta izveide vai izvēle
- Zabbix aģenta vai agent 2 instalēšana
- Hosta saskarnes pievienošana
- Papildu konfigurācijas piemērošana jūsu uzraudzības mērķim vai Zabbix hostam (ja to pieprasa veidne)

Lai piekļūtu Host Wizard Zabbix lietotāja saskarnē, rīkojieties šādi:

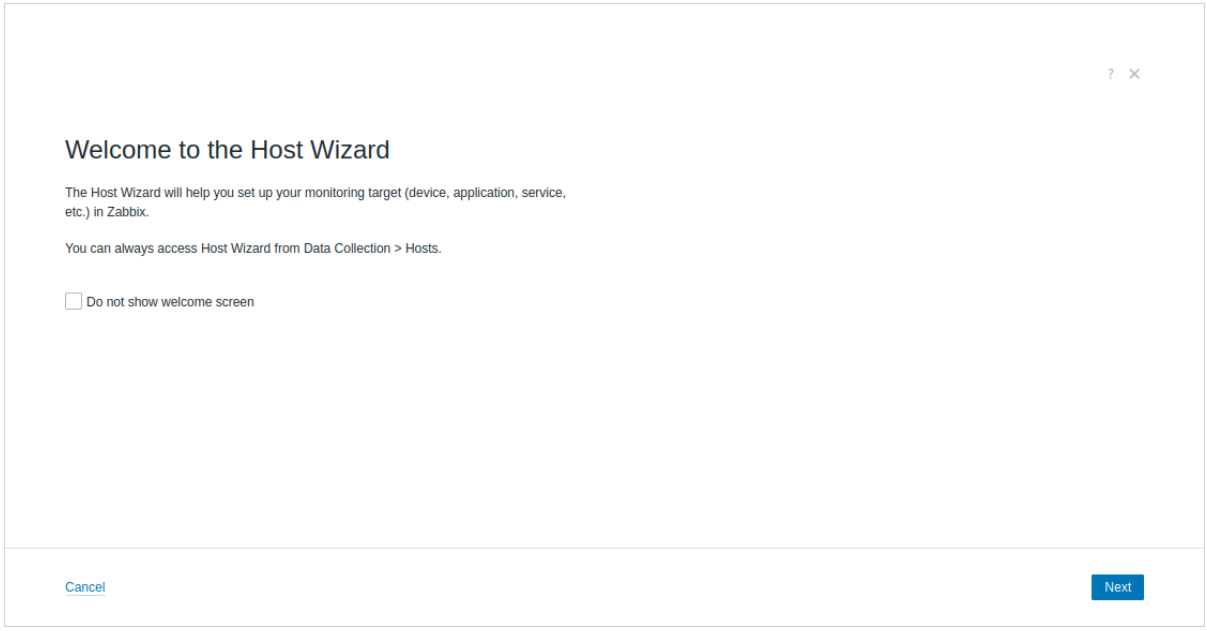
- Dodieties uz: *Data collection* > *Hosts*
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Host Wizard*
- Izpildiet vedņa norādījumus

Varat arī rediģēt esošu hostu, noklikšķinot uz tā nosaukuma (vai trīs punktu ikonas blakus tam) dažādās lietotāja saskarnes sadaļās, lai piekļūtu Host Wizard no hosta izvēlnes. Ņemiet vērā, ka, rediģējot esošu hostu, dažas darbības tiek izlaistas, ja tajā jau ir iekļauta izvēlētajai veidnei nepieciešamā konfigurācija.

Varat arī izmantot klasisko pieeju hosta izveidei un konfigurēšanai.

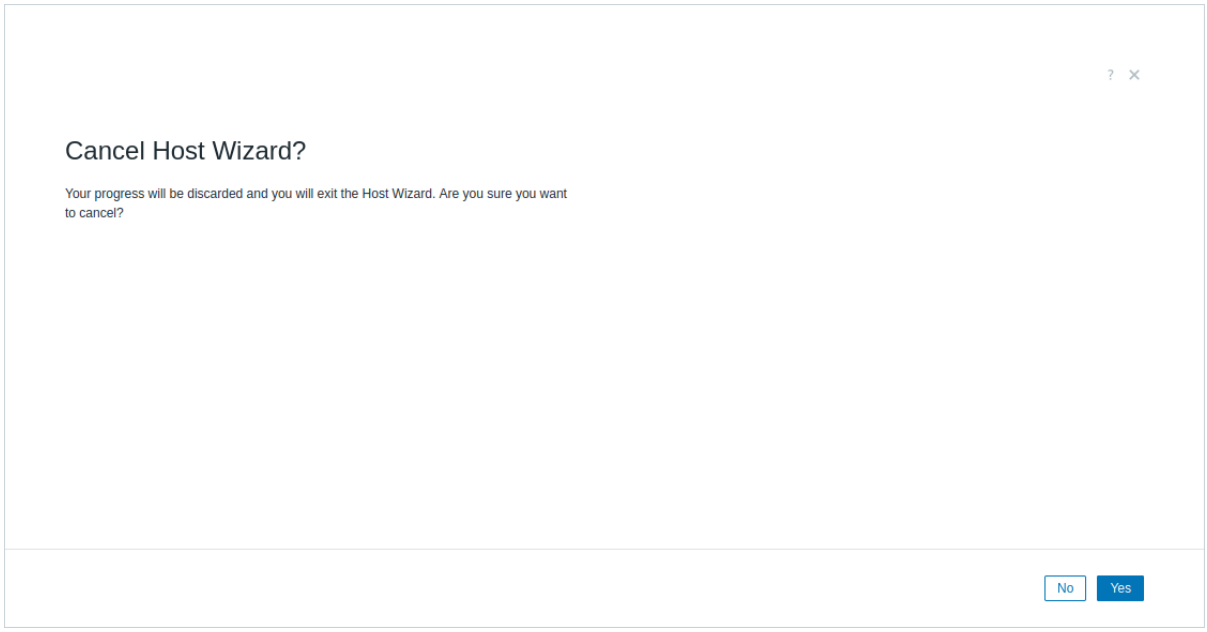
Laipni lūdzam hosta vednī

Kad palaižat hosta vedni, tiek parādīts sveiciena ekrāns:



Lai turpmākajās sesijās izlaistu šo ekrānu, varat atzīmēt izvēles rūtiņu *Nerādīt sveiciena ekrānu* un noklikšķināt uz *Tālāk*.

Ja nākamajās hosta vedņa darbībās sākat konfigurēt iestatījumus un mēģināt aizvērt vedni, noklikšķinot uz aizvēršanas ikonas vai nospiežot ESC, tiks parādīts apstiprinājuma ekrāns:



Noklikšķinot uz *Jā*, tiks iziets no hosta vedņa, nesaglabājot jūsu progresu.
Noklikšķinot uz *Nē*, nospiežot ESC vai aizvēršanas ikonu, tiks atgriezts uz pēdējo darbību.

Atlasiet veidni

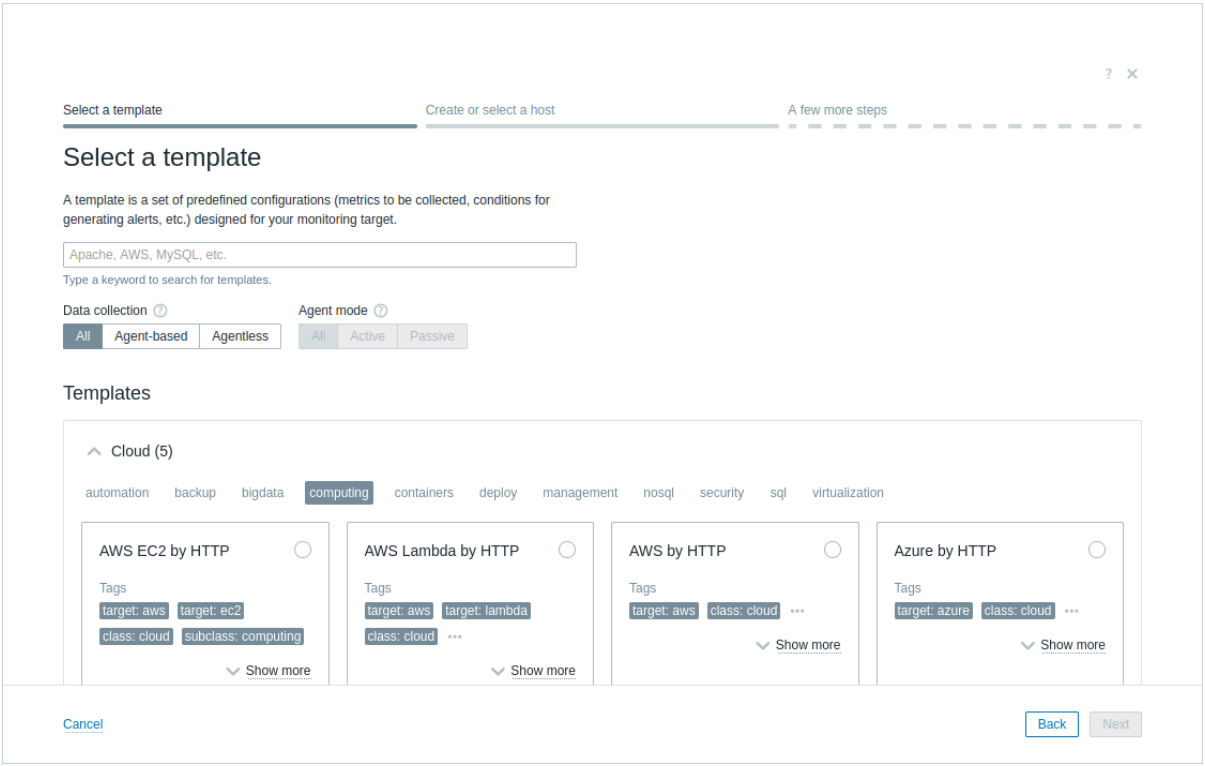
Pirmais solis monitorēšanas mērķa iestatīšanā ir atlasīt veidni — iepriekš definētu konfigurāciju kopu (vācamās metrikas, nosacījumi brīdinājumu ģenerēšanai u.c.), kas paredzēta jūsu monitorēšanas mērķim.

Zabbix gatavās veidnes piedāvā dažādas iepriekš definētas monitorēšanas konfigurācijas operētājsistēmām, lietotnēm, datubāzēm, tīkla ierīcēm, pakalpojumiem un citam.

Šajā solī varat:

- Pārlūkot veidnes pēc klases (Cloud, Database, Network u.c.) un apakšklases (piemēram, automation, backup), kuru pamatā ir veidņu birkas.
- Meklēt veidnes pēc atslēgvārdiem veidnes nosaukumā vai veidnes birku nosaukumos un vērtībās.
- Filtrēt veidnes pēc datu vākšanas metodes (uz aģentu balstītas vai bez aģenta; uz aģentu balstītas veidnes ietver vismaz vienu Zabbix aģents vienumu).
- Filtrēt veidnes pēc aģenta režīma (aktīvs vai pasīvs; aktīvās veidnes ietver vismaz vienu Zabbix aģents (active) vienumu, savukārt pasīvās veidnes ietver vismaz vienu pasīvu vienumu; sīkāku informāciju skatiet sadaļā Passive and active agent checks).

- Izmantojot Host Wizard esoša hosta konfigurēšanai, varat arī filtrēt veidnes, kas jau ir piesaistītas hostam.



Attention:
Atlasot veidni, var tikt parādīts šāds ziņojums: *Dažas veidnes (n) nav saderīgas ar Host Wizard. Skatiet, kā tās atjaunināt. Pielāgotas veidnes netiek atbalstītas.* Tas norāda, ka dažas gatavās veidnes vēl nav saderīgas ar Host Wizard un tās ir jājaunina; skatiet veidņu jaunināšanas norādījumus.

Jūsu atlasītā veidne noteiks nākamos soļus Host Wizard vednī. Piemēram, ja atlasāt veidni *MySQL by Zabbix agent*, kas datu vākšanai izmanto Zabbix aģents, vednis palīdzēs jums veikt aģenta instalēšanas procesu.

Izveidojiet vai atlasiet hostu

Veidnei jābūt piesaistītai hostam — entītijai Zabbix sistēmā, kas attēlo jūsu uzraudzības mērķi. Pēc piesaistīšanas hosts saņem visas veidnes entītijas, piemēram, vienumus (ievācamās metrikas) un trigerus (nosacījumus brīdinājumu ģenerēšanai), kas ir nepieciešami uzraudzībai.

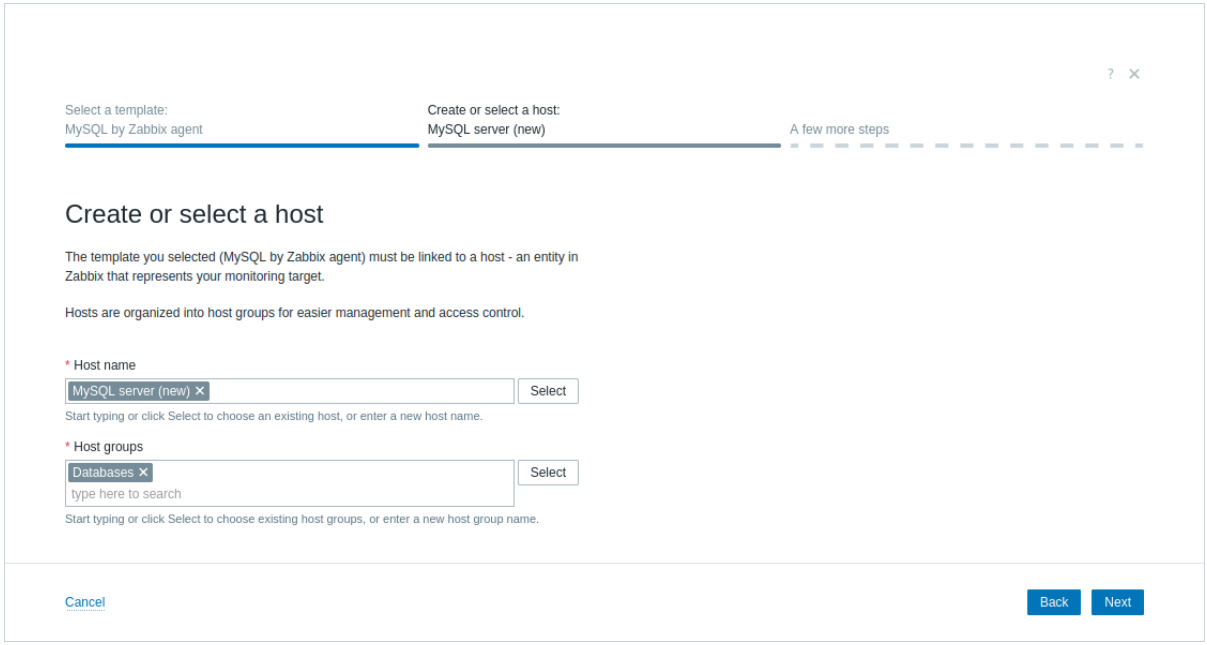
Katram hostam jāietilpst arī vismaz vienā hostu grupā, ko izmanto hostu organizēšanai un lietotāju atļauju piešķiršanai tiem.

Šajā solī varat:

- Izveidot jaunu hostu un hostu grupu, ievadot nosaukumus abiem.
- Izveidot jaunu hostu, ievadot tā nosaukumu, un atlasīt esošu hostu grupu, kurai jaunais hosts tiks piešķirts.
- Atlasīt esošu hostu; varat arī piešķirt šo hostu papildu hostu grupām, nenoņemot to no nevienas pašreizējās grupas.

Note:
Atlasot esošu hostu, tā pašreizējā konfigurācija netiek noņemta vai pārrakstīta, ja vien tas nav skaidri norādīts turpmākajās darbībās. Atklātie hosti nevar tikt atlasīti.

Piemēram, varat izveidot jaunu hostu ar nosaukumu *MySQL serveris*, lai attēlotu lokāli instalēto MySQL serveri, kuru vēlaties uzraudzīt. Iepriekš atlasītā veidne *MySQL by Zabbix agent* tiks piesaistīta šim hostam. Turklāt varat atlasīt esošo hostu grupu *Databases* (vai izveidot jaunu, piemēram, *MySQL serveri*), lai organizētu hostu kopā ar citiem ar datubāzēm saistītiem hostiem un vēlāk vienkāršotu atļauju pārvaldību.



Instalējiet Zabbix aģentu

Uz aģentu balstītām veidnēm (piemēram, *MySQL by Zabbix agent*) ir nepieciešams instalēt Zabbix aģentu vai aģentu 2 — procesu, kas tiek izvietots jūsu uzraudzības mērķī, lai aktīvi uzraudzītu lokālos resursus un lietojumprogrammas (ja to pieprasa veidne).

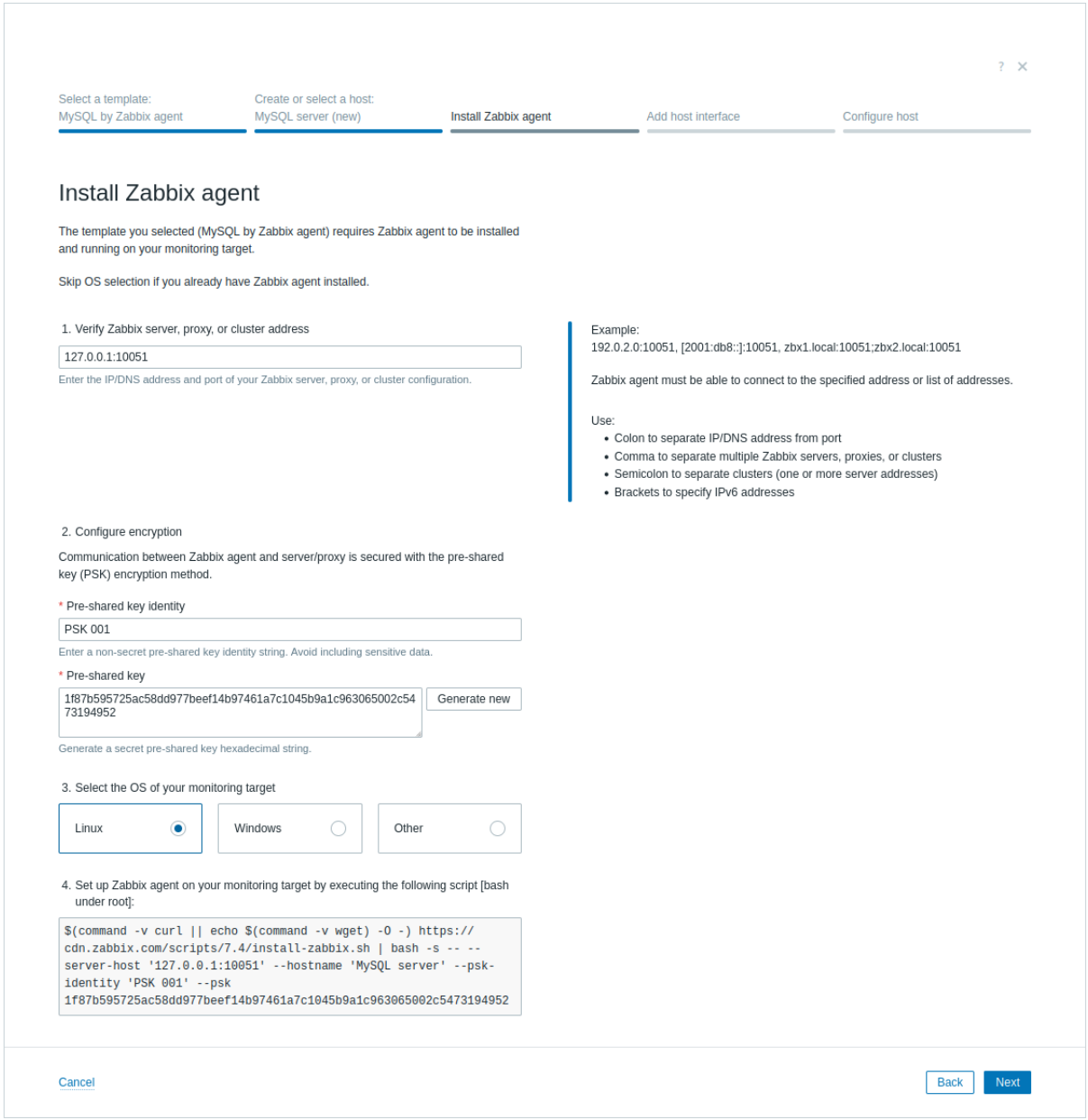
Šajā solī:

- Pārbaudiet Zabbix servera, starpniekservera vai klastera adresi (piemēram, 192.0.2.0:10051). Šī vērtība tiks izmantota, lai konfigurētu Server un ServerActive parametru Zabbix aģentam.
- Ievadiet jaunu, unikālu un ne slepenu iepriekš koplietotās atslēgas identitāti (piemēram, *PSK 001* vai *mysql-agent-psk1*) un ģenerējiet iepriekš koplietoto atslēgu. Šīs vērtības tiks izmantotas, lai konfigurētu iepriekš koplietotās atslēgas (PSK) šifrēšanu Zabbix hostam (piemēram, *MySQL server*) un Zabbix aģentam.

Attention:
Rediģējot esošu hostu, PSK konfigurācija pārrakstīs visus esošos šifrēšanas iestatījumus hostā.

- Atlasiet tās mašīnas operētājsistēmu, kurā atrodas jūsu uzraudzības mērķis.
- Instalējiet Zabbix aģentu šajā sistēmā, palaižot tajā nodrošināto skriptu vai sekojot instalēšanas norādījumiem.

Piemēram, jūs varat iestatīt *Pre-shared key identity* kā *PSK 001* un ģenerēt jaunu *Pre-shared key*. Pēc tam varat atlasīt *Linux* kā operētājsistēmu un palaist nodrošināto skriptu šajā sistēmā. Pēc Zabbix aģenta instalēšanas atgriezieties Host Wizard.

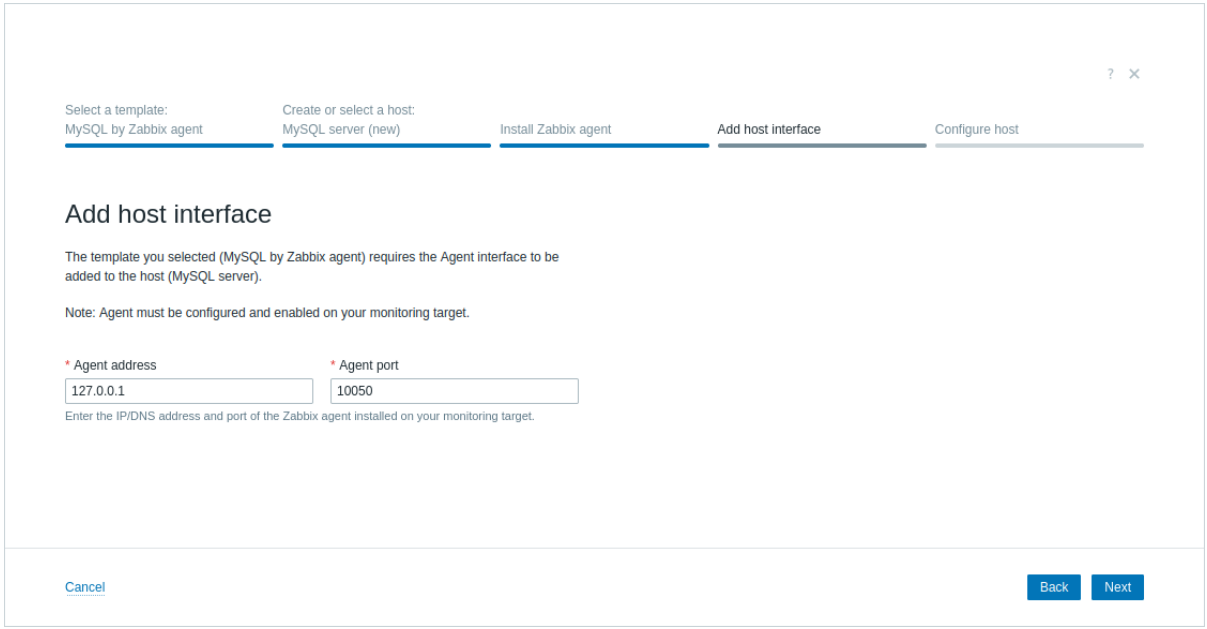


Pievienot hosta saskarni

Hosta saskarne nosaka, kā Zabbix serveris pa tīklu izveido savienojumu ar jūsu uzraudzības mērķi. Tā nosaka savienojuma parametrus, piemēram, IP adresi, DNS nosaukumu, portu un saskarnes tipu (aģents, SNMP, JMX vai IPMI) atkarībā no datu vākšanas metodes, kas nepieciešama atlasītajai veidnei.

Šajā solī ievadiet hosta saskarnes informāciju, kas nepieciešama atlasītajai veidnei.

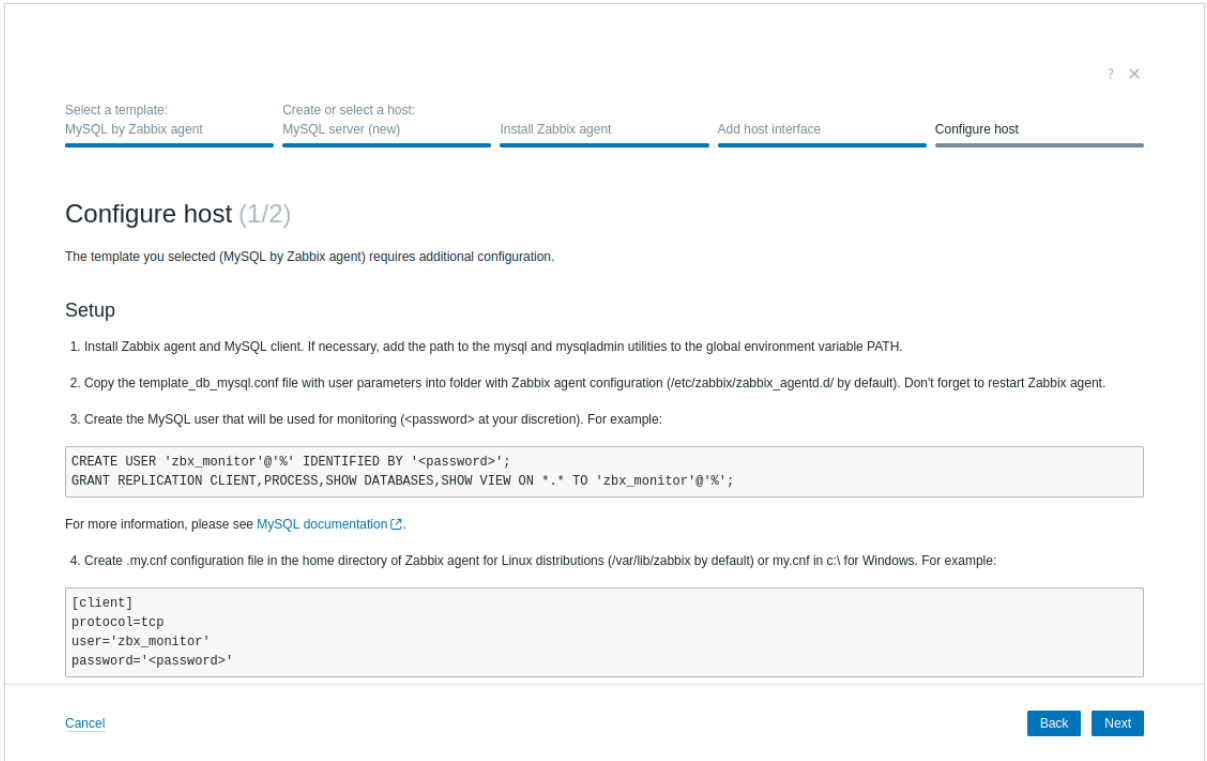
Piemēram, varat izmantot noklusējuma aģenta adresi (127.0.0.1) un portu (10050), ja Zabbix serveris, Zabbix aģents un MySQL serveris darbojas lokāli vienā un tajā pašā datorā.



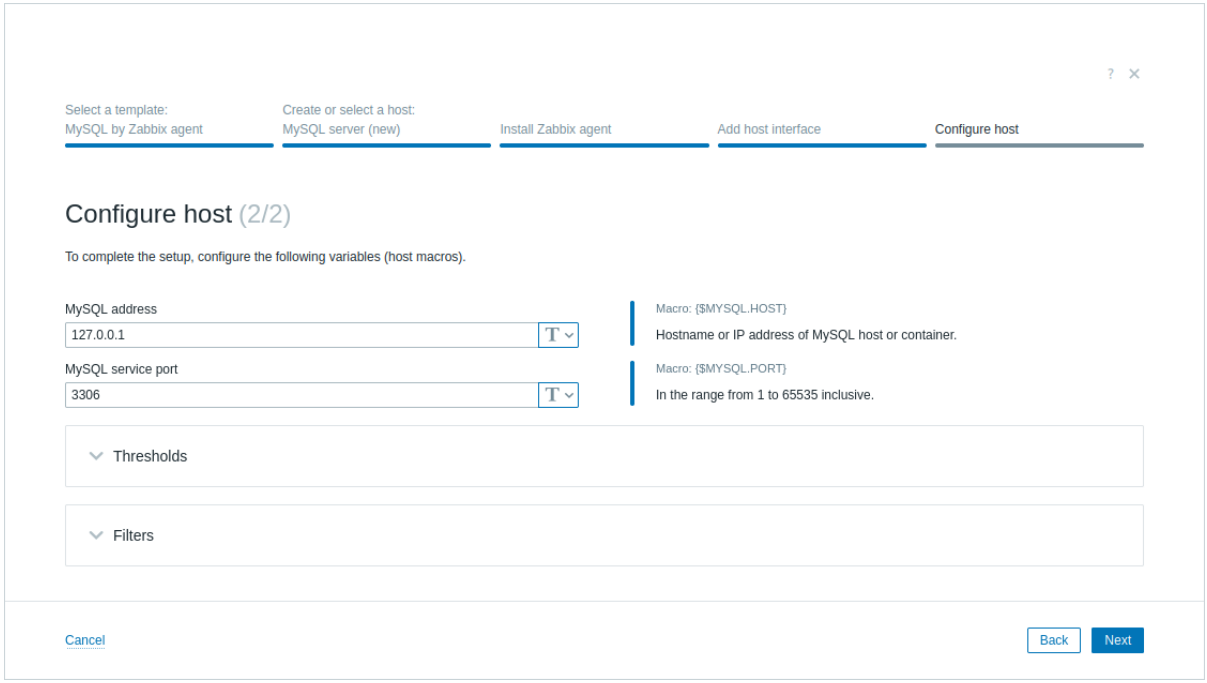
Konfigurēt hostu

Dažām veidnēm pirms hosta izveides ir nepieciešama papildu konfigurācija. Tā var ietvert:

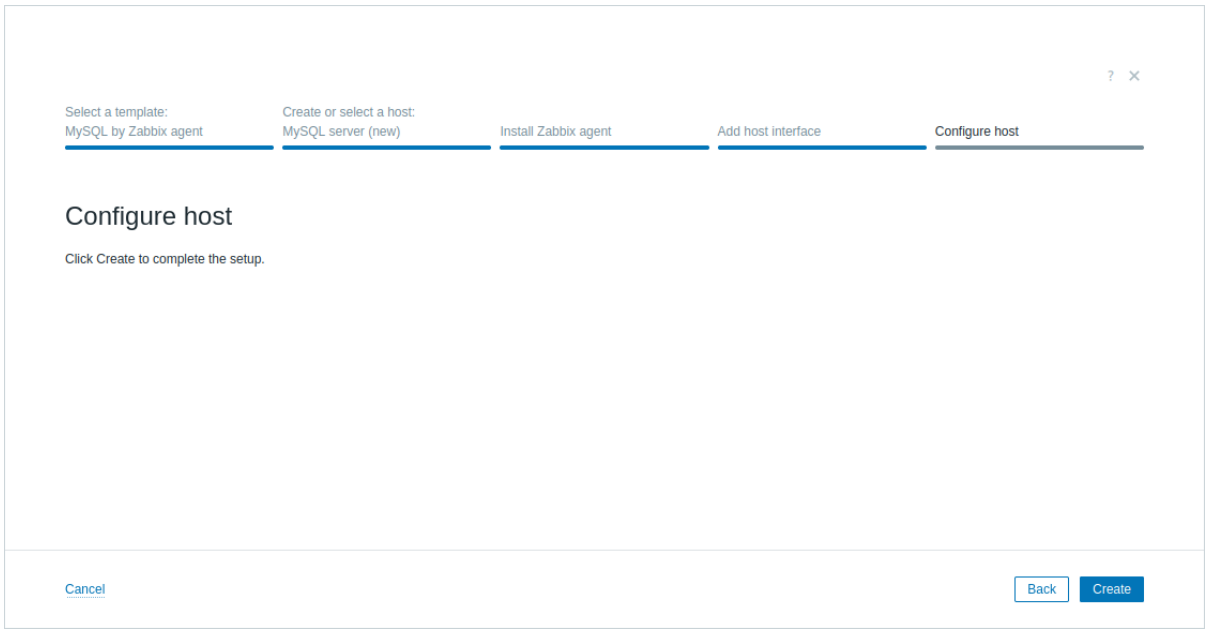
- Jūsu uzraudzības mērķa manuālu konfigurēšanu (piemēram, noteiktu pakalpojumu iespējošanu, pakalpojumu lietotāju izveidi vai atļauju piešķiršanu):



- Hosta makrosu definēšanu — mainīgos, kas kontrolē vienumu darbību, savienojuma iestatījumus, autentifikācijas akreditācijas datus un citus parametrus:



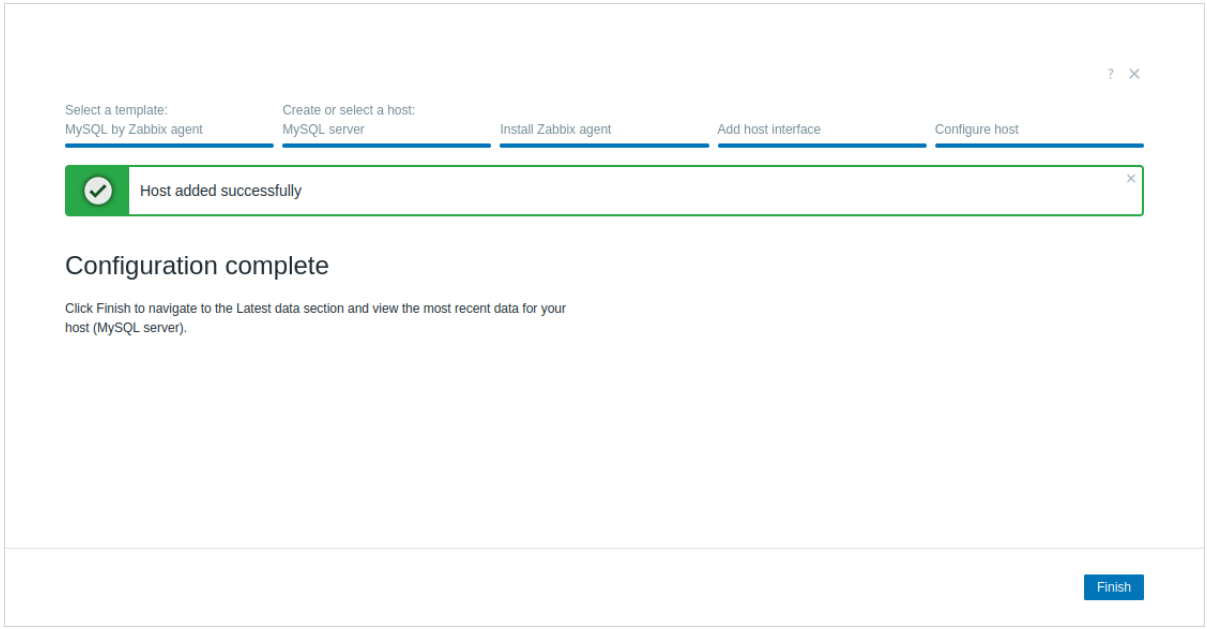
Pēc visu nepieciešamo konfigurācijas darbību pabeigšanas noklikšķiniet uz *Create*, lai pievienotu hostu Zabbix sistēmai (vai uz *Update*, lai atjauninātu esoša hosta konfigurāciju):



Konfigurēšana pabeigta

Šajā brīdī Zabbix jau uzrauga jūsu mērķi.

Jauniem hostiem noklikšķiniet uz *Pabeigt*, lai pārietu uz sadaļu Jaunākie dati un skatītu jaunākos datus par savu hostu. Esošajiem hostiem noklikšķiniet uz *Pabeigt*, lai aizvērtu hosta vedni un atgrieztos ekrānā, no kura tas tika atvērts.



2 Hosta konfigurēšana

Pārskats

Lai konfigurētu hostu Zabbix lietotāja saskarnē, rīkojieties šādi:

- Dodieties uz: *Datu ievākšana* > *Hosti* vai *Uzraudzība* > *Hosti*
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot hostu* (vai uz hosta nosaukuma, lai rediģētu esošu hostu)
- Veidlapā ievadiet hosta parametrus

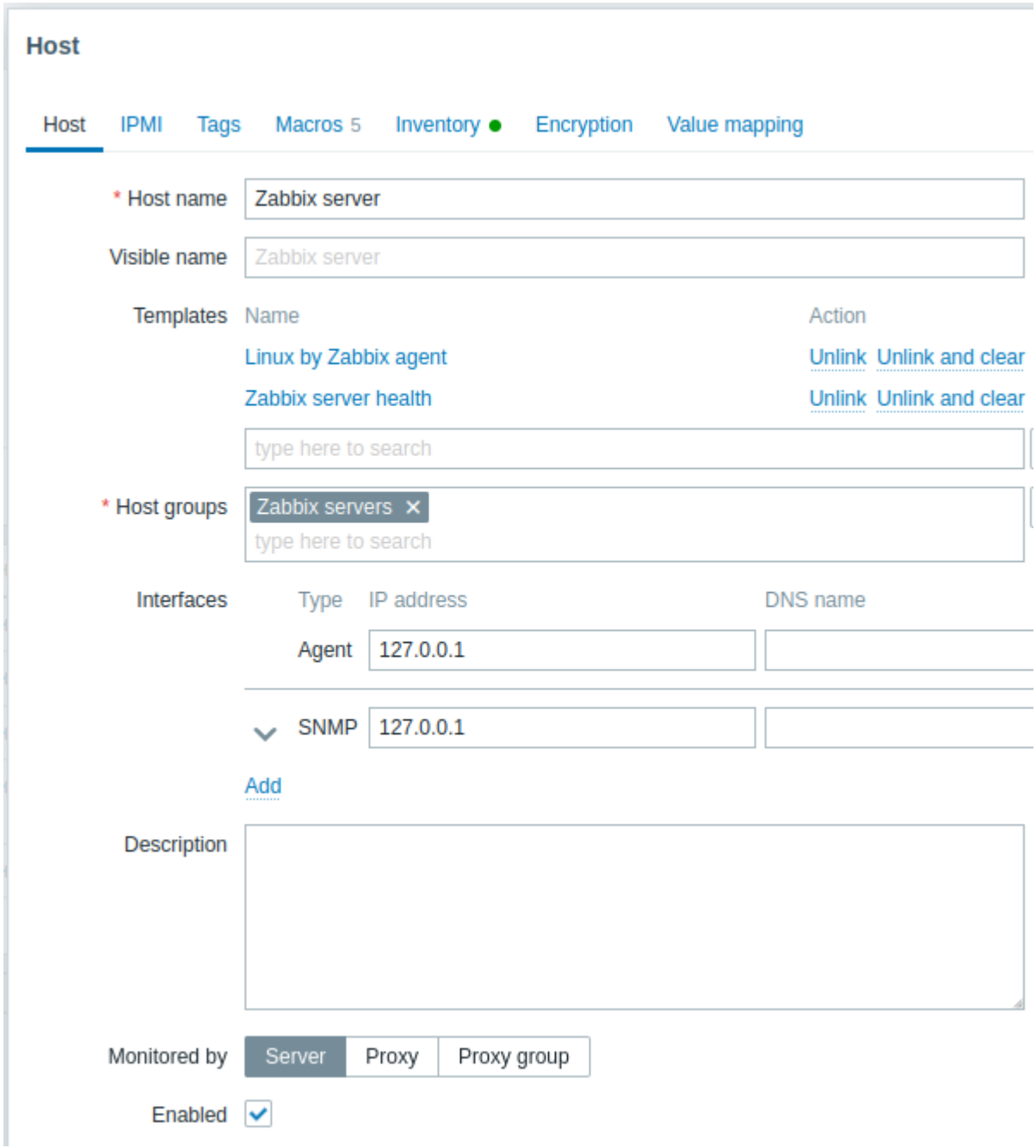
Varat arī izmantot pogu *Klonēt* esoša hosta konfigurācijas formā, lai izveidotu jaunu hostu. Šim hostam būs visas esošā hosta īpašības, tostarp piesaistītās veidnes, entītijas (vienumi, trigeri utt.) no šīm veidnēm, kā arī entītijas, kas tieši piesaistītas esošajam hostam.

Ņemiet vērā, ka, klonējot hostu, tas saglabās visas veidnes entītijas tieši tādas, kādas tās sākotnēji ir veidnē. Jebkādas izmaiņas šajās entītijās, kas veiktas esošā hosta līmenī (piemēram, mainīts vienuma intervāls, modificēta regulārā izteiksme vai zema līmeņa atklāšanas kārtulai pievienoti prototipi), netiks klonētas uz jauno hostu; tā vietā tās būs tādas, kādas tās ir veidnē.

Alternatīvi varat izmantot Host Wizard, lai konfigurētu hostu, izmantojot vadītu, soli pa solim izpildāmu saskarni.

Konfigurācija

Cilnē **Hosts** ir ietverti vispārīgie hosta atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

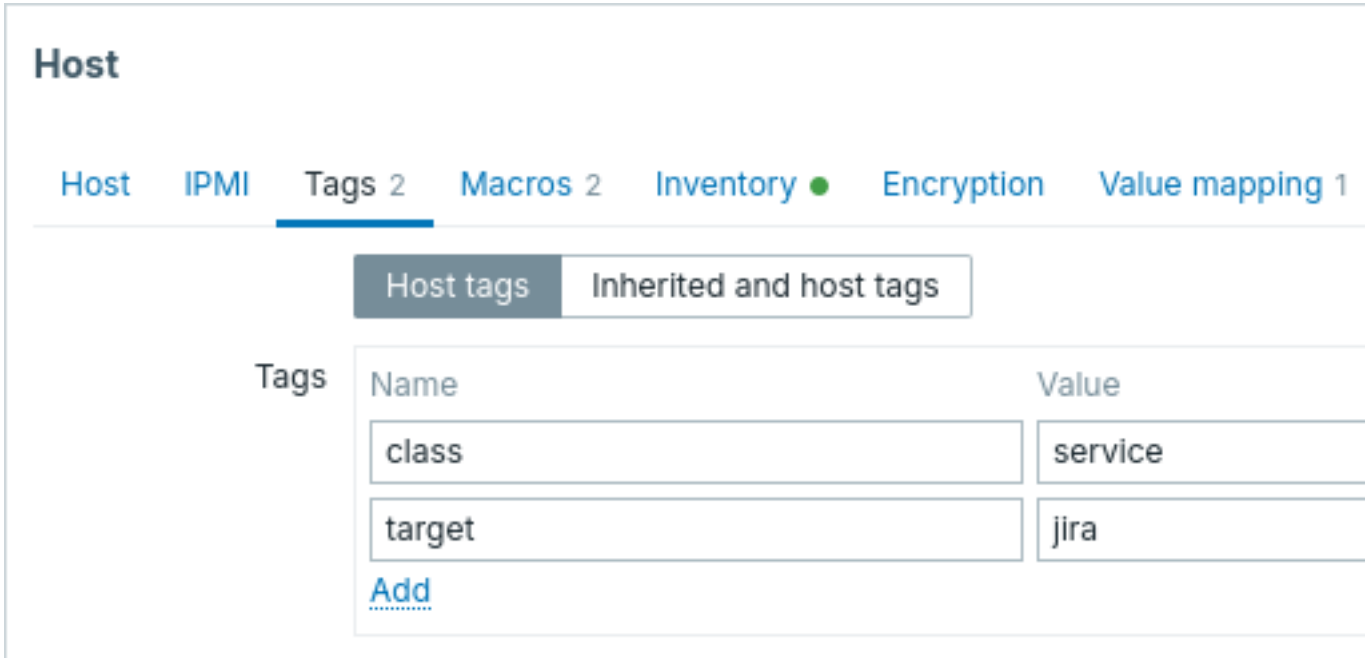
Parametrs	Apraksts
Hosta nosaukums	Ievadiet unikālu hosta nosaukumu. Ir atļauti burtciparu simboli, atstarpes, punkti, defises un pasvītrojuma zīmes. Tomēr atstarpes sākumā un beigās nav atļautas. *Piezīme:* Ja konfigurējamajā hostā darbojas Zabbix aģents, aģenta konfigurācijas faila parametrā *Hostname* jābūt tai pašai vērtībai, kas šeit ievadītajam hosta nosaukumam. Parametra nosaukums ir nepieciešams aktīvo pārbaužu apstrādē.
Redzamais nosaukums	Ievadiet unikālu hosta redzamo nosaukumu. Ja iestatīsiet šo nosaukumu, tas būs redzams sarakstos, kartēs utt. tehniskā hosta nosaukuma vietā. Šis atribūts atbalsta UTF-8.
Veidnes	Piesaistiet hostam veidnes. Visas vienības (vienumi, trigeri, utt.) tiks mantotas no veidnes. Lai piesaistītu jaunu veidni, sāciet rakstīt veidnes nosaukumu teksta ievades laukā. Tiks parādīts atbilstošo veidņu saraksts; ritiniet uz leju, lai atlasītu. Varat arī noklikšķināt uz *Atlasīt* blakus laukam un atlasīt veidnes no saraksta uznirstošajā logā. Visas atlasītās veidnes tiks piesaistītas hostam, kad hosta konfigurācijas forma tiks saglabāta vai atjaunināta. Lai atsaistītu veidni, izmantojiet vienu no divām opcijām blokā *Piesaistītās veidnes*: *Atsaistīt* - atsaistīt veidni, bet saglabāt tās entītijas (vienumus, trigerus, utt.); *Atsaistīt un notīrīt* - atsaistīt veidni un noņemt visas tās entītijas (vienumus, trigerus, utt.). Uzskaitītie veidņu nosaukumi ir klikšķināmas saites, kas ved uz veidnes konfigurācijas formu.

Parametrs	Apraksts
Hostu grupas	Atlasiet hostu grupas, kurām hosts pieder. Hostam jāpieder vismaz vienai hostu grupai. Jaunu grupu var izveidot un piesaistīt hostam, pievienojot neesošas grupas nosaukumu.
Saskarnes	Hostam tiek atbalstīti vairāki hosta saskarņu tipi: *Aģents*, *SNMP*, *JMX* un *IPMI*. Pēc noklusējuma nav definēta neviena saskarne. Lai pievienotu jaunu saskarni, blokā *Saskarnes* noklikšķiniet uz *Pievienot*, atlasiet saskarnes tipu un ievadiet *IP/DNS*, *Savienoties ar* un *Ports* informāciju. *Piezīme:* Saskarnes, kas tiek izmantotas jebkuros vienumos, nevar noņemt, un saite *Noņemt* tām ir pelēkota. "IP" vai "DNS" no SNMP saskarnes tiek izmantots arī SNMP slazdiem. Atbilstības noteikšanas laikā tiek izmantots tikai hosta saskarnē atlasītais "IP" vai "DNS". Papildu informāciju par SNMP saskarnes konfigurēšanu (v1, v2 un v3) skatiet sadaļā SNMP uzraudzības konfigurēšana.
IP adrese	Hosta IP adrese (neobligāti).
DNS nosaukums	Hosta DNS nosaukums (neobligāti).
Savienoties ar	Noklikšķinot uz attiecīgās pogas, jūs norādīsiet Zabbix serverim, ko izmantot datu iegūšanai no aģentiem: **IP** - savienoties ar hosta IP adresi (ieteicams) **DNS** - savienoties ar hosta DNS nosaukumu
Ports	TCP/UDP porta numurs. Noklusējuma vērtības ir: 10050 Zabbix aģentam, 161 SNMP aģentam, 12345 JMX un 623 IPMI.
Noklusējuma	Atzīmējiet radio pogu, lai iestatītu noklusējuma saskarni.
Apraksts	Ievadiet hosta aprakstu.
Uzrauga	Atlasiet, vai hostu uzrauga: **Serveris** - hostu uzrauga Zabbix serveris; **Starpniekserveris** - hostu uzrauga atsevišķs starpniekserveris; **Starpniekserveru grupa** - hostu uzrauga starpniekserveru grupa.
Starpniekserveris	Tiek parādīts piešķirtā starpniekservera nosaukums (tikai tad, ja Zabbix serveris ir piešķīris kādu no atlasītās starpniekserveru grupas). Šis lauks tiek rādīts tikai tad, ja hostu uzrauga starpniekserveru grupa.
Iespējots	Kad izvēles rūtiņa ir atzīmēta, hosts ir iespējots — gatavs uzraudzībai. Ja izvēles rūtiņa nav atzīmēta, hosts ir atspējots — netiek uzraudzīts: Pasīviem datu pieprasījumiem, ko iniciē Zabbix serveris/starpniekserveris (piemēram, Zabbix aģents, SNMP aģents, vienkāršās pārbaudes), uzraudzība tiek atspējota pēc konfigurācijas sinhronizācijas. Trigeri un darbības, kas saistītas ar hostu, arī tiek atspējotas tikai pēc konfigurācijas pārlādes. Zabbix aģenta aktīvajām pārbaudēm uzraudzība tiek pārtraukta laika intervālā (aptuveni 5 sekundes), kurā Zabbix aģents saņem informāciju, ka hosts ir atspējots. Šajā īsajā intervālā hosts turpinās lokāli vākt datus aktīvajām pārbaudēm un mēģinās tos nosūtīt serverim/starpniekserverim; tomēr, tā kā hosts ir atzīmēts kā *Atspējots*, serveris/starpniekserveris datus noraidīs. Atspējojot hostu, tā vienumi tiek nekavējoties noņemti no vēstures kešatmiņas (izņemot to pēdējās vērtības, kas tiek saglabātas žurnāliem).

Cilnē **IPMI** ir ietverti IPMI pārvaldības atribūti.

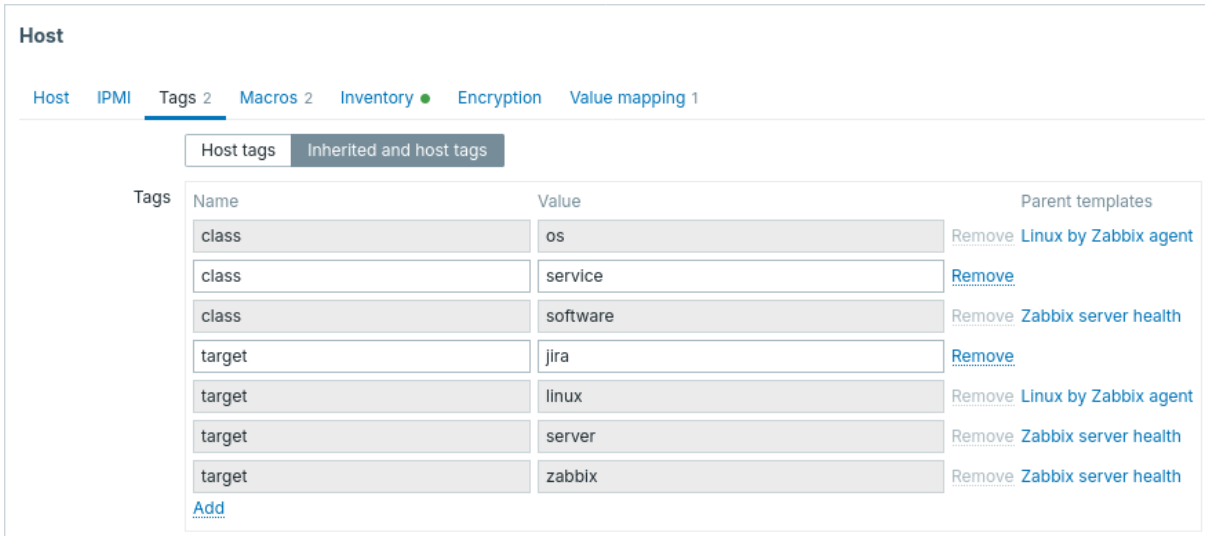
Parametrs	Apraksts
Autentifikācijas algoritms	Atlasiet autentifikācijas algoritmu.
Privilēģiju līmenis	Atlasiet privilēģiju līmeni.
Lietotājvārds	Lietotājvārds autentifikācijai. Var izmantot lietotāja makrosus.
Parole	Parole autentifikācijai. Var izmantot lietotāja makrosus.

Cilne **Birkas** ļauj definēt hosta līmeņa birkas. Visas šī hosta problēmas tiks marķētas ar šeit ievadītajām vērtībām.



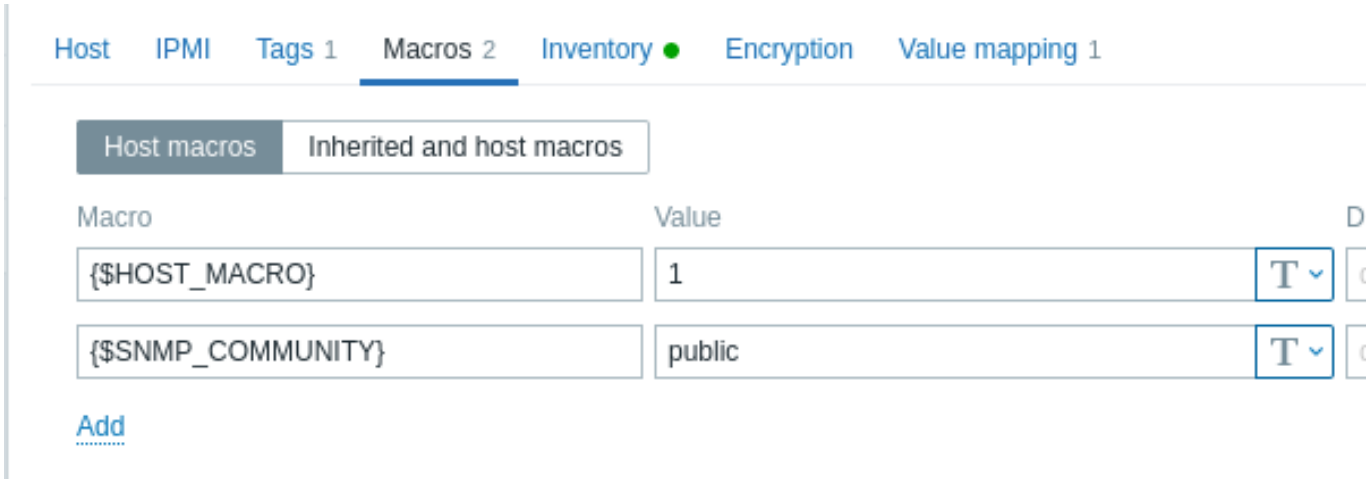
Birkās tiek atbalstīti lietotāja makrosi, {INVENTORY.*} makrosi, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} makrosi.

Šeit varat arī skatīt veidnes līmeņa birkas, ja atlasāt opciju *Mantotās un hosta birkas*. Tur tiek parādītas visas hostam definētās lietotāja birkas, kā arī to izcelsme.

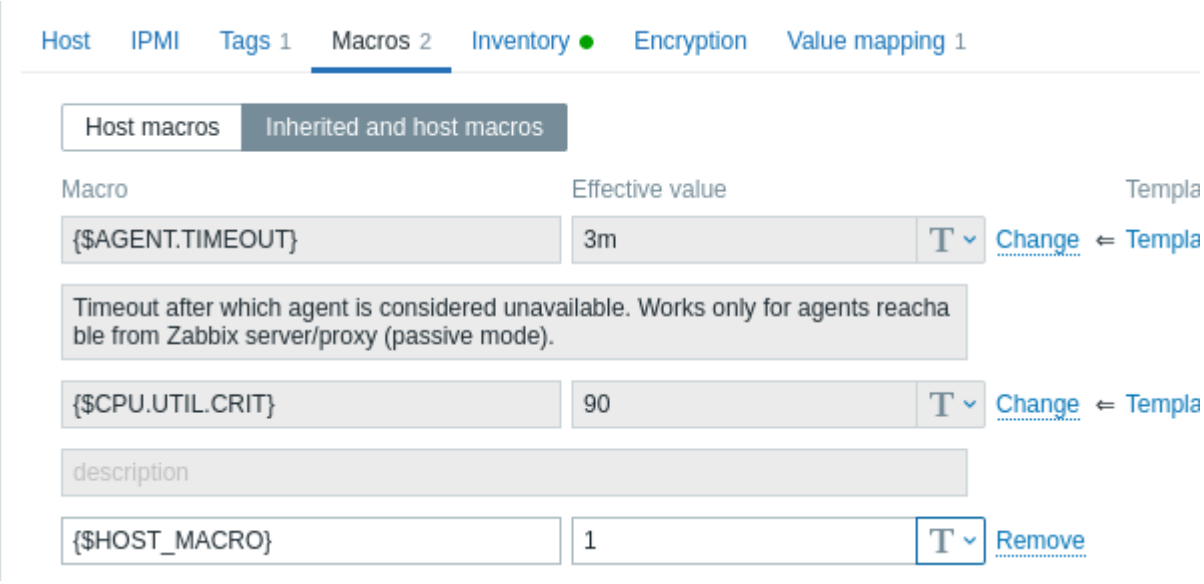


Ērtībai ir nodrošinātas saites uz attiecīgajām veidnēm.

Cilne **Makrosi** ļauj definēt hosta līmeņa lietotāja makrosus kā nosaukuma-vērtības pārus. Ņemiet vērā, ka makrosu vērtības var glabāt kā vienkāršu tekstu, slepenu tekstu vai Vault noslēpumu. Tiek atbalstīta arī apraksta pievienošana.

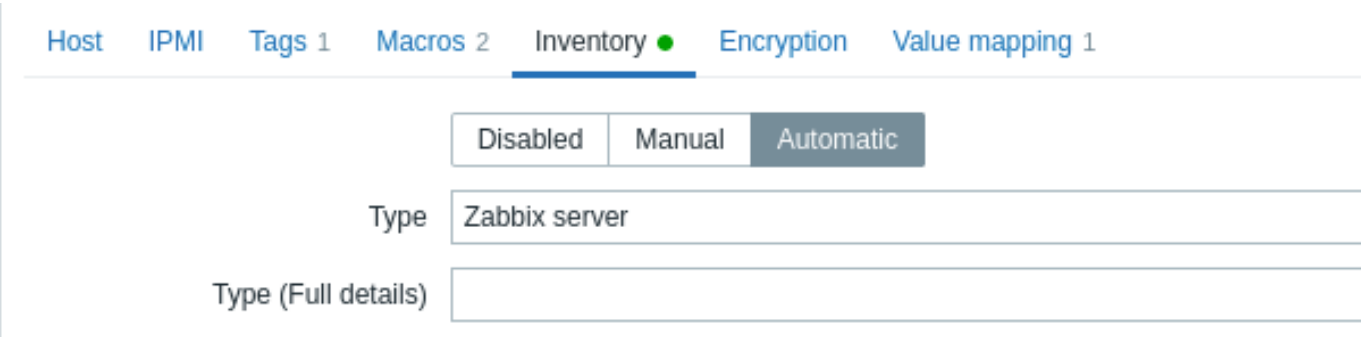


Šeit varat arī skatīt veidnes līmeņa un globālos lietotāja makrosus, ja atlasāt opciju *Mantotie un hosta makrosi*. Tur tiek parādīti visi hostam definētie lietotāja makrosi ar vērtību, uz kuru tie tiek atrisināti, kā arī to izcelsmi.



Ērtībai ir nodrošinātas saites uz attiecīgajām veidnēm un globālo makrosu konfigurāciju. Ir iespējams arī rediģēt veidnes/globālo makrosu hosta līmenī, faktiski izveidojot makrosa kopiju hostā.

Cilne **Inventārs** ļauj manuāli ievadīt hosta inventāra informāciju. Varat arī izvēlēties iespējot *Automātisku* inventāra aizpildīšanu vai atspējot inventāra aizpildīšanu šim hostam.



Ja inventārs ir iespējots (manuāli vai automātiski), pie cilnes nosaukuma tiek parādīts zaļš punkts.

Šifrēšana

Cilne **Šifrēšana** ļauj pieprasīt šifrētus savienojumus ar hostu.

Parametrs	Apraksts
Savienojumi ar hostu	Kā Zabbix serveris vai starpniekserveris savienojas ar Zabbix aģentu hostā: bez šifrēšanas (noklusējums), izmantojot PSK (iepriekš koplietotu atslēgu) vai sertifikātu.
Savienojumi no hosta	Izvēlieties, kāda veida savienojumi ir atļauti no hosta (t.i., no Zabbix aģenta un Zabbix sender). Vienlaikus var atlasīt vairākus savienojumu veidus (tas ir noderīgi testēšanai un pārslēgšanai uz citu savienojuma veidu). Noklusējums ir "Bez šifrēšanas".
Izdevējs	Atļautais sertifikāta izdevējs. Sertifikāts vispirms tiek pārbaudīts ar CA (sertifikācijas iestādi). Ja tas ir derīgs un CA to ir parakstījusi, tad lauku *Izdevējs* var izmantot, lai papildus ierobežotu atļauto CA. Šis lauks ir paredzēts lietošanai, ja jūsu Zabbix instalācijā tiek izmantoti sertifikāti no vairākām CA. Ja šis lauks ir tukšs, tiek pieņemta jebkura CA.
Subjekts	Atļautais sertifikāta subjekts. Sertifikāts vispirms tiek pārbaudīts ar CA. Ja tas ir derīgs un CA to ir parakstījusi, tad lauku *Subjekts* var izmantot, lai atļautu tikai vienu *Subjekts* virknes vērtību. Ja šis lauks ir tukšs, tiek pieņemts jebkurš derīgs sertifikāts, ko parakstījusi konfigurētā CA.
PSK identitāte	Iepriekš koplietotās atslēgas identitātes virkne. Neiekļaujiet sensitīvu informāciju PSK identitātē, jo tā tiek pārsūtīta nešifrēta pa tīklu, lai informētu saņēmēju, kuru PSK izmantot.

Parametrs	Apraksts
PSK	Iepriekš koplietotā atslēga (hex-virkne). Maksimālais garums: 512 hex-cipari (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, 64 hex-cipari (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Vērtību kartēšana

Cilne **Vērtību kartēšana** ļauj konfigurēt lietotājam draudzīgu vienuma datu attēlojumu vērtību kartēšanā.

3 Hostu grupas konfigurēšana

Pārskats

Hostu grupas tiek izmantotas hostu loģiskai grupēšanai un lietotāju atļauju piešķiršanai tām.

Katram hostam jābūt piešķirtai vismaz vienai hostu grupai. Hosts var piederēt vairākām hostu grupām, un katra hostu grupa var ietvert vairākus hostus.

Ņemiet vērā, ka Zabbix visas atļaujas ir balstītas uz grupām: lietotāju grupām, hostu grupām un veidņu grupām. Tāpēc pat tad, ja vienam lietotājam ir nepieciešama piekļuve vienam hostam, tā tiek piešķirta, pievienojot lietotāju lietotāju grupai, kurai ir atļauja piekļūt hostu grupai, kas satur šo hostu.

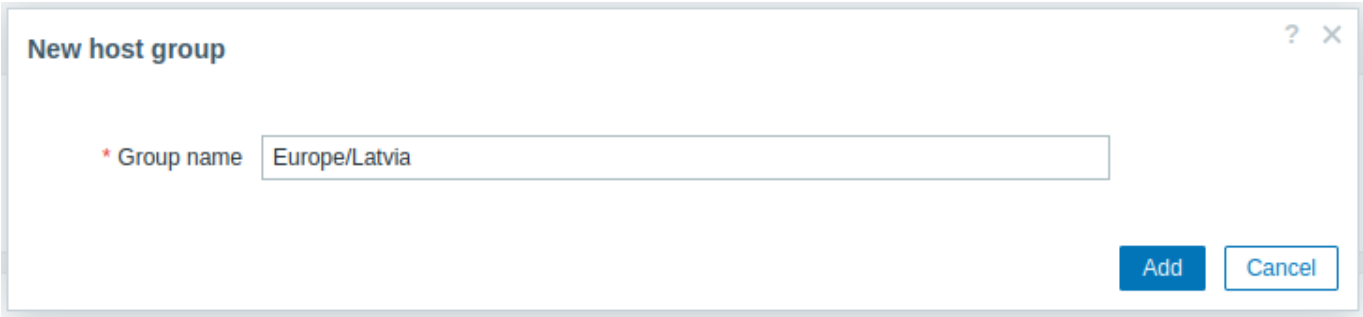
Konfigurācija

Attention:
Tikai Super admin lietotāji var izveidot hostu grupas.

Zabbix lietotāja saskarnē ir divas hostu grupas izveides iespējas.

Pirmā iespēja:

- Dodieties uz: *Datu ievākšana → Hostu grupas*
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot hostu grupu*
- Formā ievadiet grupas nosaukumu



Otrā iespēja: konfigurējot hostu, ievadiet neesošas grupas nosaukumu ievades laukā *Hostu grupas*.

Kad hostu grupa ir izveidota, varat noklikšķināt uz grupas nosaukuma sarakstā sadaļā *Datu ievākšana → Hostu grupas*, lai rediģētu grupas nosaukumu, klonētu grupu vai dzēstu grupu.

Dzēšot hostu grupu, tiek dzēsta tikai loģiskā grupa, nevis grupā esošie hosti. Nav iespējams dzēst hostu grupu, kas ir vienīgā grupa kādam esošam hostam.

Hostu apakšgrupu izveide

Hostu apakšgrupa (vai ligzdota hostu grupa) ir to saturošās vecākgrupas bērngrupa.

Apakšgrupa tiek izveidota, grupas nosaukuma ievades laukā izmantojot slīpsvītru '/' , lai norādītu tās attiecību ar vecākgrupu(-ām). Piemēram:

- ievadot `Europe/Latvia`, tiek izveidota vecākgrupas `Europe` apakšgrupa `Europe/Latvia`.
- ievadot `Europe/Latvia/Riga/Zabbix servers`, tiek izveidota attiecīgā apakšgrupa ligzdotajās vecākgrupās `Europe`, `Europe/Latvia`, `Europe/Latvia/Riga`.

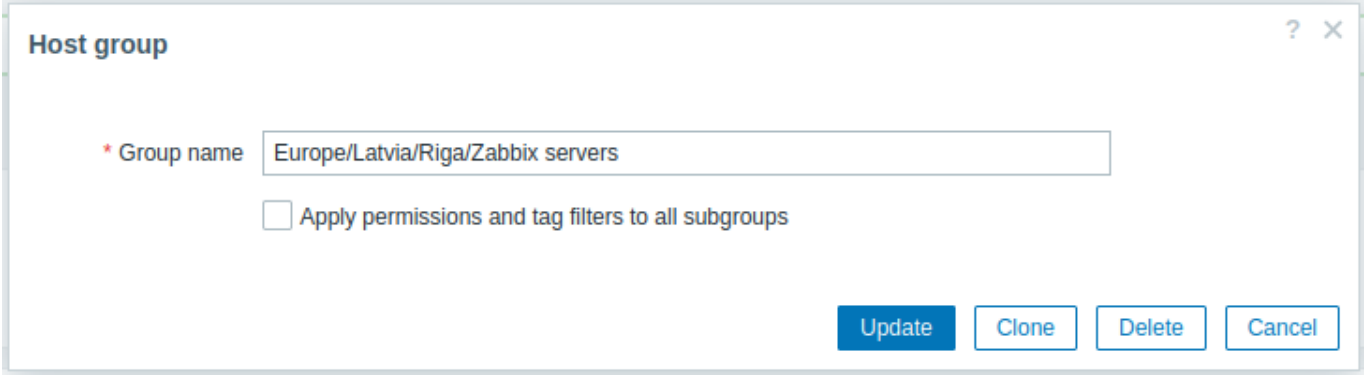
Veidojot apakšgrupu, nav atļauts izmantot slīpsvītras sākumā vai beigās, kā arī vairākas slīpsvītras pēc kārtas. '/' ekrānēšana netiek atbalstīta.

Pirms apakšgrupas izveides nav nepieciešams izveidot nevienu vecāku hostu grupu. Varat izvēlēties, vai sākt ar apakšgrupas izveidi (piemēram, `Europe/Latvia`) vai ar jebkuras vecākgrupas izveidi (mūsu piemērā — `Europe`). Ja sākat ar apakšgrupas izveidi, vecākās hostu grupas **netiks** izveidotas automātiski.

Atļaujas hostu grupām

- Veidojot apakšgrupu esošai vecākgrupai (piemēram, izveidojot `Europe/Latvia`, kad `Europe` jau pastāv), lietotāju grupas atļaujas apakšgrupai tiek mantotas no vecākgrupas.
- Veidojot vecākgrupu esošai apakšgrupai (piemēram, izveidojot `Europe`, kad `Europe/Latvia` jau pastāv), vecākgrupai netiek iestatītas nekādas atļaujas.

Rediģējot jebkuru hostu grupu, varat arī iestatīt papildu opciju *Piemērot atļaujas un birku filtrus visām apakšgrupām*.



Atzīmējot šo izvēles rūtiņu un noklikšķinot uz *Atjaunināt*, visām pašreizējām un nākotnes apakšgrupām rediģētajā hostu grupā tiks piemērots vienāds atļauju līmenis un birku filtri.

Tātad, ja kādām lietotāju grupām ir piešķirtas atšķirīgas atļaujas rediģētās hostu grupas apakšgrupām, atzīmējot šo izvēles rūtiņu, visām pašreizējām un nākotnes apakšgrupām tiks piešķirtas tādas pašas lietotāju atļaujas un uz birkām balstītas atļaujas kā rediģētajai grupai.

Ņemiet vērā, ka šī opcija netiek saglabāta datubāzē un pārrakstīs esošās atļaujas. Jebkādas izmaiņas, kas veiktas, izmantojot šo opciju, var atsaukt tikai manuāli.

4 Inventārs

Pārskats

Jūs varat uzturēt tīklā pievienoto ierīču inventāru Zabbix sistēmā.

Zabbix lietotāja saskarnē ir īpaša *Inventāra* izvēlne. Tomēr sākotnēji jūs tur neredzēsiet nekādus datus, un tā nav vieta, kur ievadīt datus. Inventāra datu veidošana tiek veikta manuāli, konfigurējot hostu, vai automātiski, izmantojot dažas automātiskas aizpildīšanas iespējas.

Inventāra veidošana

Manuālais režīms

Konfigurējot hostu, cilnē *Inventārs* varat ievadīt tādu informāciju kā ierīces tips, sērijas numurs, atrašanās vieta, atbildīgā persona, URL utt. — datus, kas aizpildīs inventāra informāciju.

Ja hosta inventāra informācijā ir iekļauts URL un tas sākas ar 'http' vai 'https', sadaļā *Inventārs* tas tiks attēlots kā noklikšķināma saite.

Automātiskais režīms

Hosta inventāru var arī aizpildīt automātiski. Lai tas darbotos, konfigurējot hostu, cilnē *Inventory* inventāra režīmam jābūt iestatītam uz *Automatic*.

Pēc tam varat konfigurēt hosta vienumus, lai aizpildītu jebkuru hosta inventāra lauku ar to vērtību, norādot mērķa lauku ar attiecīgo atribūtu (sauktu par *Item will populate host inventory field*) vienuma konfigurācijā.

Vienumi, kas ir īpaši noderīgi automatizētai inventāra datu apkopošanai:

- system.hw.chassis[full|type|vendor|model|serial] — noklusējums ir [full], nepieciešamas root tiesības
- system.hw.cpu[all|cpunum,full|maxfreq|vendor|model|curfreq] — noklusējums ir [all,full]
- system.hw.devices[pci|usb] — noklusējums ir [pci]
- system.hw.macaddr[interface,short|full] — noklusējums ir [all,full], interface ir regulārā izteiksme
- system.sw.arch

- system.sw.os[name|short|full] — noklusējums ir [name]
- system.sw.packages[regexp,manager,short|full] — noklusējums ir [all,all,full]

Inventarizācijas režīma izvēle

Inventarizācijas režīmu var izvēlēties hosta konfigurācijas formā.

Inventarizācijas režīms pēc noklusējuma jauniem hostiem tiek izvēlēts, pamatojoties uz iestatījumu *Noklusējuma hosta inventarizācijas režīms* sadaļā *Administrēšana* → *Vispārīgi* → *Cits*.

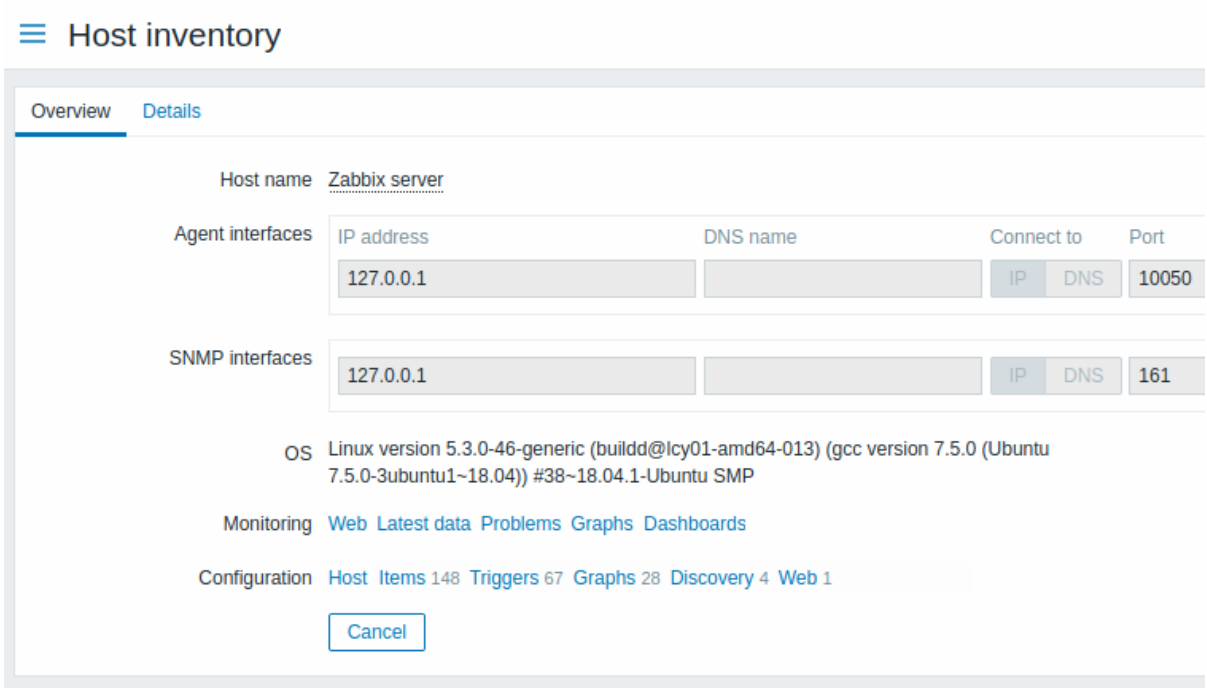
Hostiem, kas pievienoti ar tīkla atklāšanas vai automātiskās reģistrācijas darbībām, ir iespējams definēt operāciju *Iestatīt hosta inventarizācijas režīmu*, izvēloties manuālo vai automātisko režīmu. Šī operācija ignorē iestatījumu *Noklusējuma hosta inventarizācijas režīms*.

Inventāra pārskats

Informācija par visiem esošajiem inventāra datiem ir pieejama izvēlnē *Inventory*.

Sadaļā *Inventory* → *Overview* varat iegūt hostu skaitu pēc dažādiem inventāra laukiem.

Sadaļā *Inventory* → *Hosts* varat redzēt visus hostus, kuriem ir inventāra informācija. Noklikšķinot uz hosta nosaukuma, veidlapā tiks parādīta detalizēta inventāra informācija.



Cilnē **Overview** tiek rādīts:

Parameter	Description
Host name	Hosta nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta izvēlne ar hostam definētajiem skriptiem. Hosta nosaukums tiek attēlots ar oranžu ikonu, ja hosts atrodas uzturēšanas režīmā.
Visible name	Hosta redzamais nosaukums (ja definēts).
Host (Agent, SNMP, JMX, IPMI) interfaces	Šajā blokā ir sniegta informācija par hostam konfigurētajām saskarnēm.
OS	Hosta operētājsistēmas inventāra lauks (ja definēts).
Hardware	Hosta aparatūras inventāra lauks (ja definēts).
Software	Hosta programmatūras inventāra lauks (ja definēts).
Description	Hosta apraksts.
Monitoring	Saites uz uzraudzības sadaļām ar šī hosta datiem: *Web*, *Latest data*, *Problems*, *Graphs*, *Dashboards*.
Configuration	Saites uz konfigurācijas sadaļām šim hostam: *Host*, *Items*, *Triggers*, *Graphs*, *Discovery*, *Web*. Pēc katras saites ir norādīts konfigurēto vienību skaits.

Cilnē **Details** tiek rādīti visi aizpildītie inventāra lauki (nav tukši).

Inventāra makrosi

Ir pieejami hosta inventāra makrosi {INVENTORY.*}, kurus var izmantot paziņojumos, piemēram:

”Serverim vietā {INVENTORY.LOCATION1} ir problēma, atbildīgā persona ir {INVENTORY.CONTACT1}, tālruņa numurs {INVENTORY.POC.PRIMARY.PHONE.A1}.”

Lai iegūtu sīkāku informāciju, skatiet lapu Atbalstītie makrosi.

5 Masveida atjaunināšana

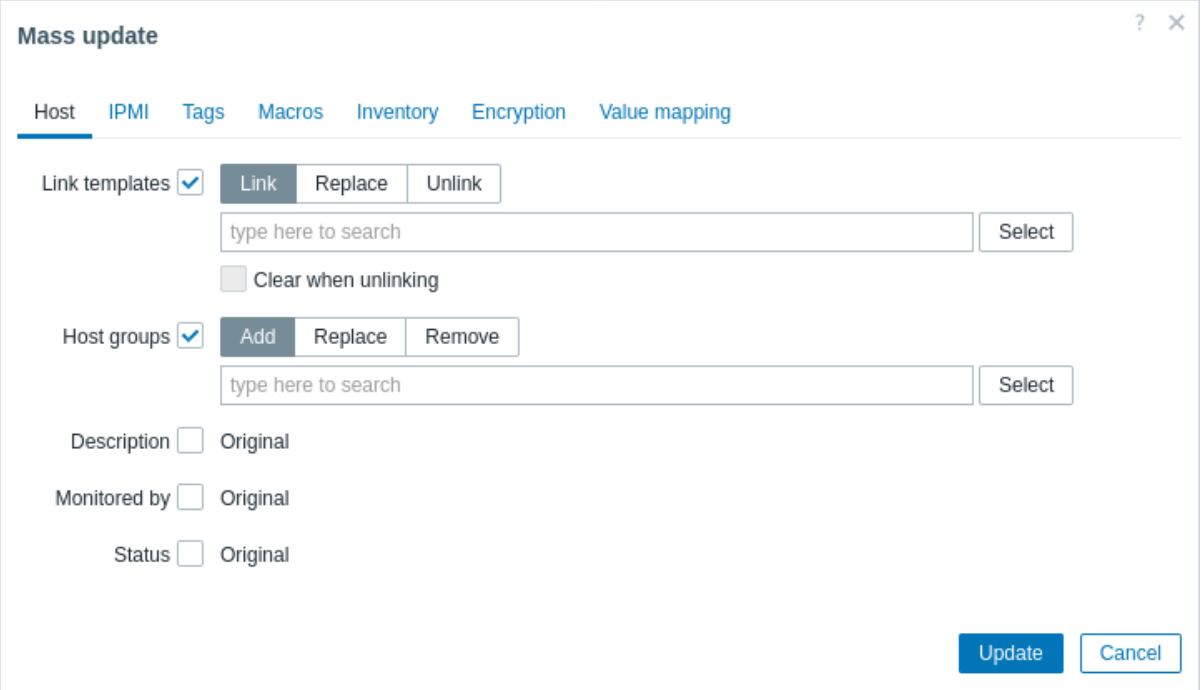
Pārskats

Dažkārt var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākiem hostiem. Tā vietā, lai atvērtu katru atsevišķo hostu rediģēšanai, varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšanas izmantošana

Lai masveidā atjauninātu dažus hostus, rīkojieties šādi:

- Atzīmējiet izvēles rūtiņas pie hostiem, kurus vēlaties atjaunināt, hostu sarakstā
- Zem saraksta noklikšķiniet uz *Mass update*
- Atveriet cilni ar nepieciešamajiem atribūtiem (*Host*, *IPMI*, *Tags*, *Macros*, *Inventory*, *Encryption* vai *Value mapping*)
- Atzīmējiet to atribūtu izvēles rūtiņas, kurus vēlaties atjaunināt, un ievadiet tiem jaunu vērtību



Tālāk norādītās opcijas ir pieejamas, izvēloties attiecīgo pogu **veidnes** piesaistes atjaunināšanai:

- *Link* - norādiet, kuras papildu veidnes piesaistīt
- *Replace* - norādiet, kuras veidnes piesaistīt, vienlaikus atsaistot jebkuru veidni, kas iepriekš bija piesaistīta hostiem
- *Unlink* - norādiet, kuras veidnes atsaistīt

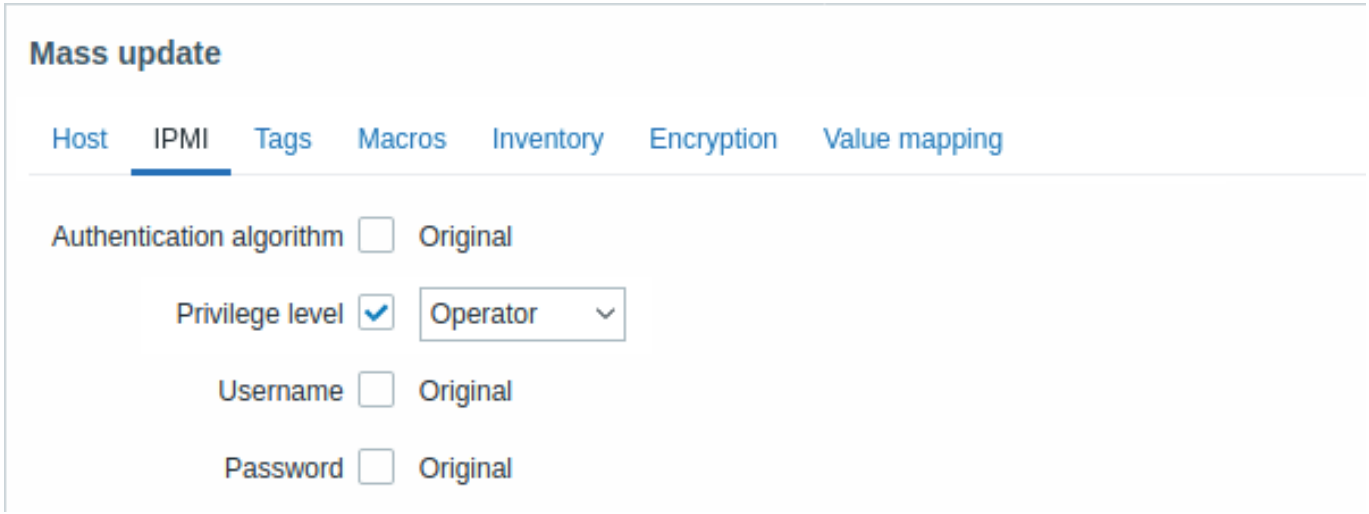
Lai norādītu piesaistāmās/atsaistāmās veidnes, sāciet rakstīt veidnes nosaukumu automātiskās pabeigšanas laukā, līdz parādās nolaižamais saraksts ar atbilstošajām veidnēm. Vienkārši ritiniet uz leju, lai atlasītu vajadzīgo veidni.

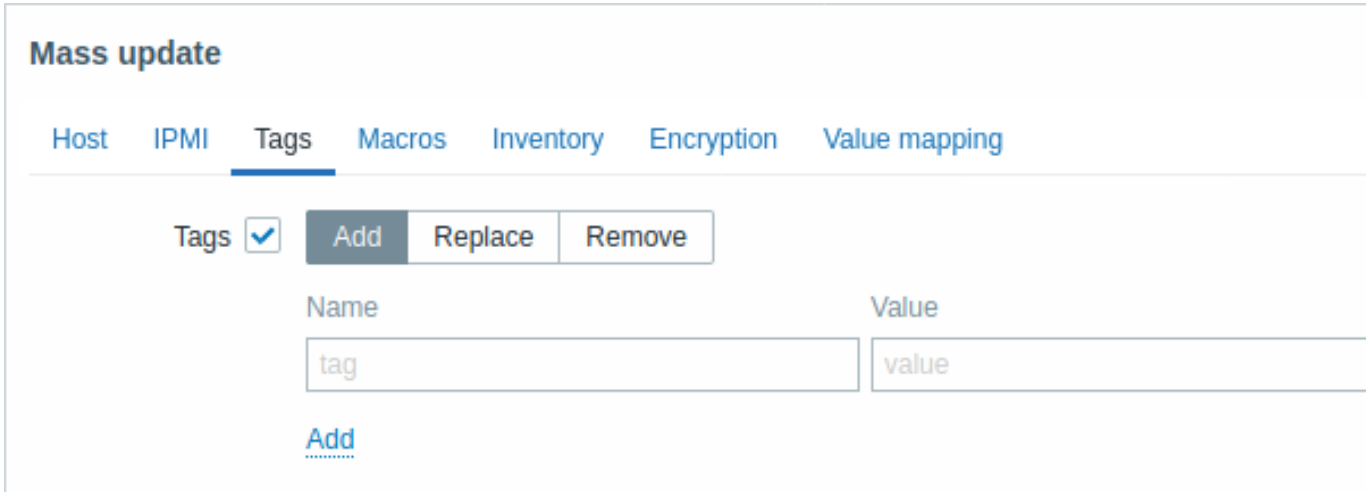
Opcija *Clear when unlinking* ļaus ne tikai atsaistīt iepriekš piesaistītās veidnes, bet arī noņemt visus no tām mantotos elementus (vienumus, trigerus utt.).

Tālāk norādītās opcijas ir pieejamas, izvēloties attiecīgo pogu **hostu grupu** atjaunināšanai:

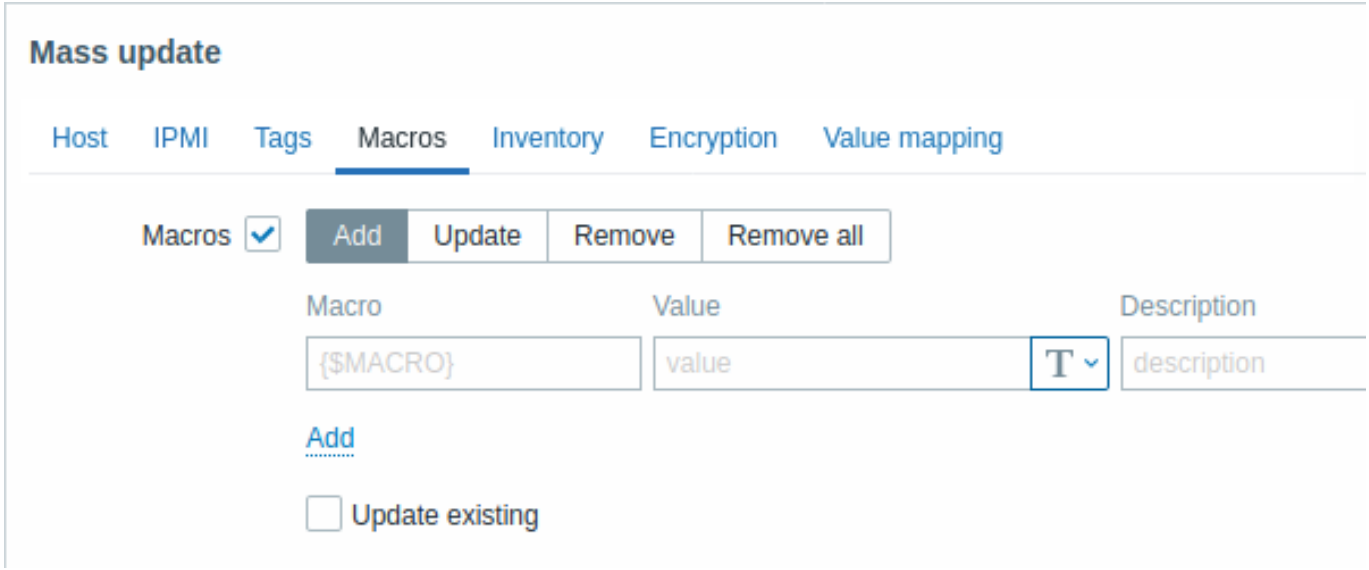
- *Add* - ļauj norādīt papildu hostu grupas no jau esošajām vai ievadīt hostiem pilnīgi jaunas hostu grupas
- *Replace* - noņems hostu no visām esošajām hostu grupām un aizstās tās ar šajā laukā norādīto(-ajām) grupu(-ām) (esošām vai jaunām hostu grupām)
- *Remove* - noņems no hostiem noteiktas hostu grupas

Šie lauki atbalsta automātisko pabeigšanu — sākot tajos rakstīt, tiek piedāvāts nolaižamais saraksts ar atbilstošām hostu grupām. Ja hostu grupa ir jauna, tā arī parādās nolaižamajā sarakstā, un aiz virknes ir norāde *(new)*. Vienkārši ritiniet uz leju, lai atlasītu.





Lietotāja makrosi, {INVENTORY.*} makrosi, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} makrosi tiek atbalstīti tagos. Ņemiet vērā, ka tagi ar vienādu nosaukumu, bet atšķirīgām vērtībām netiek uzskatīti par "dublikātiem" un tos var pievienot vienam un tam pašam hostam.

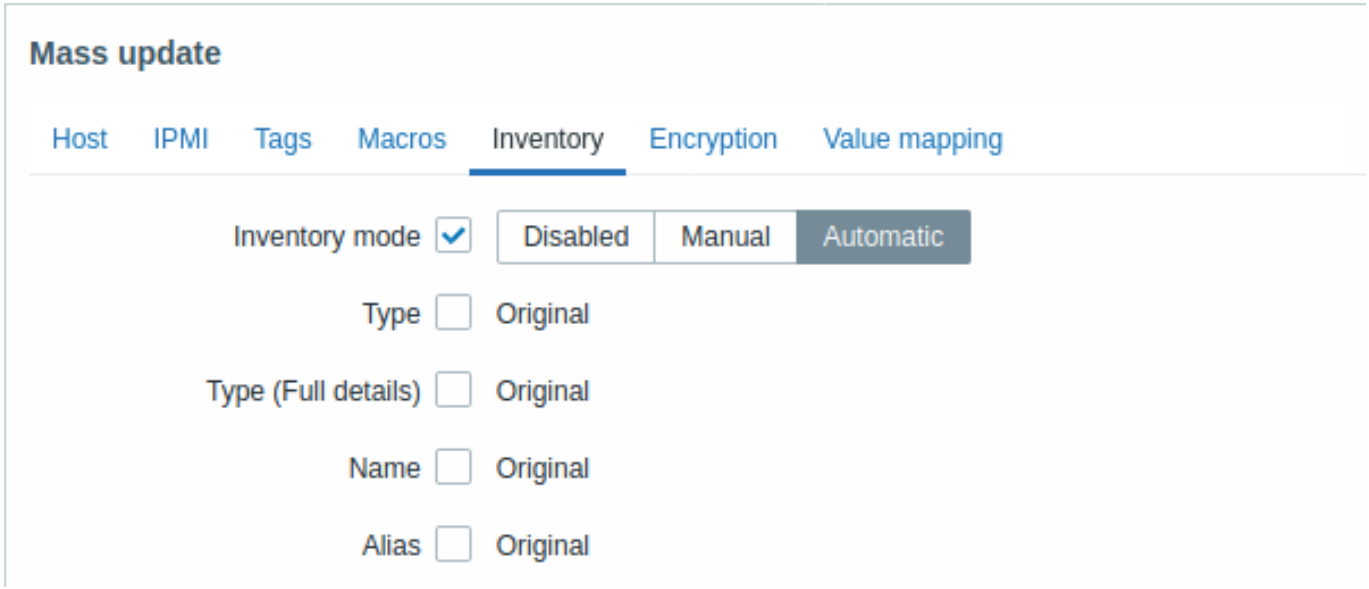


Tālāk norādītās opcijas ir pieejamas, izvēloties attiecīgo pogu makrosu atjaunināšanai:

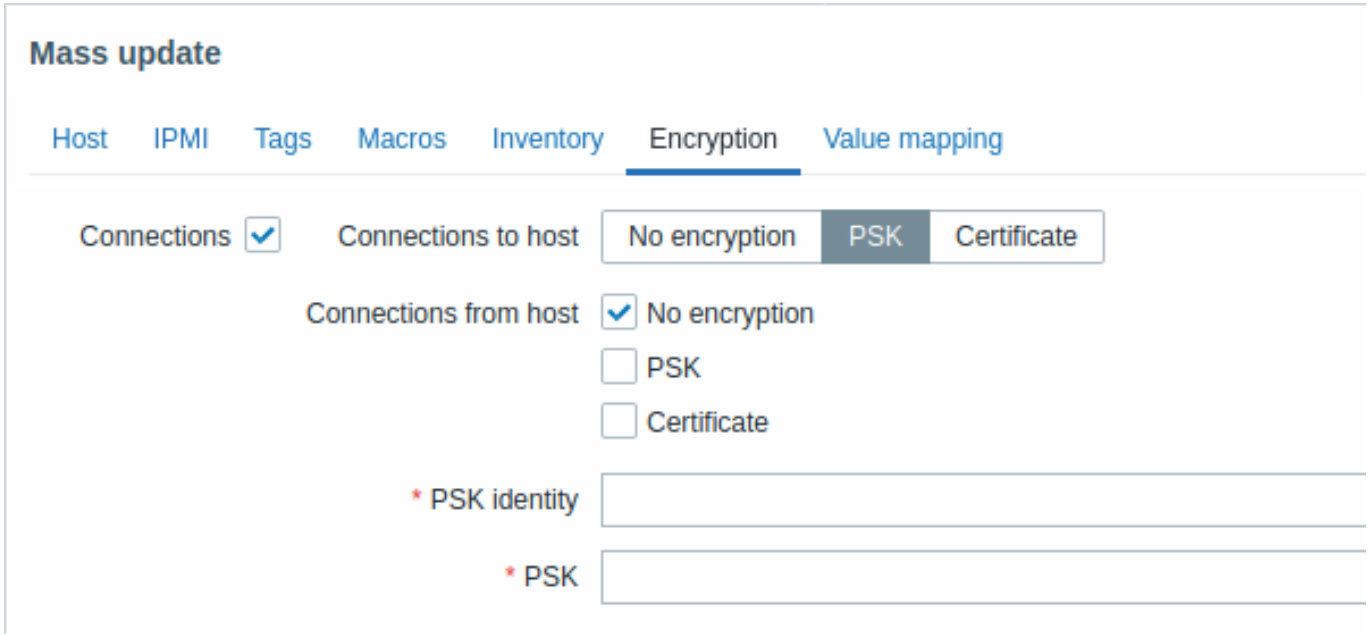
- *Add* - ļauj hostiem norādīt papildu lietotāja makrosus. Ja ir atzīmēta izvēles rūtiņa *Update existing*, tiks atjaunināta norādītā makrosa nosaukuma vērtība, tips un apraksts. Ja tā nav atzīmēta un hostā(-os) jau pastāv makross ar šādu nosaukumu, tas netiks atjaunināts.
- *Update* - aizstās šajā sarakstā norādīto makrosu vērtības, tipus un aprakstus. Ja ir atzīmēta izvēles rūtiņa *Add missing*, makross, kas iepriekš hostā nepastāvēja, tiks pievienots kā jauns makross. Ja tā nav atzīmēta, tiks atjaunināti tikai tie makrosi, kas hostā jau pastāv.
- *Remove* - noņems no hostiem norādītos makrosus. Ja ir atzīmēta izvēles rūtiņa *Except selected*, tiks noņemti visi makrosi,

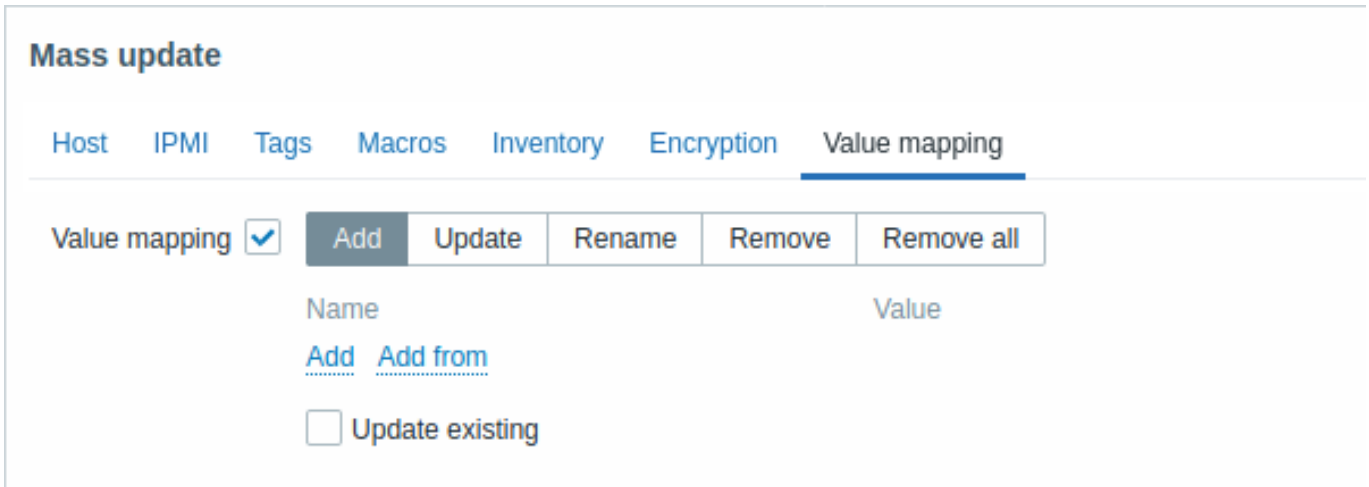
izņemot sarakstā norādītos. Ja tā nav atzīmēta, tiks noņemti tikai sarakstā norādītie makrosi.

- *Remove all* - noņems no hostiem visus lietotāja makrosus. Ja izvēles rūtiņa *I confirm to remove all macros* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt visu makrosu noņemšanu.



Lai varētu masveidā atjaunināt inventāra laukus, *Inventory mode* ir jābūt iestatītam uz “Manual” vai “Automatic”.





Vērtību kartēšanas atjaunināšanai ir pieejamas pogas ar šādām opcijām:

- *Add* - pievienot hostiem vērtību kartējumus. Ja atzīmējat *Update existing*, tiks atjauninātas visas vērtību kartējuma ar šo

nosaukumu īpašības. Pretējā gadījumā, ja vērtību kartējums ar šādu nosaukumu jau pastāv, tas netiks atjaunināts.
- *Update* - atjaunināt esošos vērtību kartējumus. Ja atzīmējat *Add missing*, vērtību kartējums, kas iepriekš hostā nepastāvēja, tiks pievienots kā jauns vērtību kartējums. Pretējā gadījumā tiks atjaunināti tikai tie vērtību kartējumi, kas hostā jau pastāv.
- *Rename* - piešķirt jaunu nosaukumu esošam vērtību kartējumam
- *Remove* - noņemt no hostiem norādītos vērtību kartējumus. Ja atzīmējat *Except selected*, tiks noņemti visi vērtību kartējumi **izņemot** norādītos.
- *Remove all* - noņemt no hostiem visus vērtību kartējumus. Ja izvēles rūtiņa *I confirm to remove all value maps* nav atzīmēta, tiks atvērts jauns uznirstošais logs ar lūgumu apstiprināt noņemšanu.

Kad visas nepieciešamās izmaiņas ir veiktas, noklikšķiniet uz *Update*. Atribūti tiks attiecīgi atjaunināti visiem atlasītajiem hostiem.

2 Vienumi

Pārskats

Vienums ir atsevišķa metrika.

Vienumi tiek izmantoti datu vākšanai. Kad esat konfigurējis hostu, jums jāpievieno vienumi, lai iegūtu faktiskos datus. Viens no veidiem, kā ātri pievienot daudz vienumu, ir piesaistīt hostam kādu no iepriekš definētajām veidnēm. Tomēr optimālai sistēmas veiktspējai var būt nepieciešams pielāgot veidnes, lai tajās būtu tieši tik daudz vienumu un tik bieža uzraudzība, cik nepieciešams.

Lai norādītu, kāda veida datus vākt no hosta, izmantojiet vienuma atslēgu. Piemēram, vienums ar atslēgas nosaukumu **system.cpu.load** vāks procesora slodzes datus, savukārt vienums ar atslēgas nosaukumu **net.if.in** vāks informāciju par ienākošo tīkla plūsmu.

Papildu parametrus var norādīt kvadrātiekavās aiz atslēgas nosaukuma. Piemēram, system.cpu.load**[avg5]** atgriezīs procesora vidējo slodzi pēdējās 5 minūtēs, savukārt net.if.in**[eth0]** parādīs ienākošo tīkla plūsmu saskarnē ”eth0”.

> **Note:**
> Visus atbalstītos vienumu tipus un vienumu atslēgas skatiet atsevišķajās vienumu tipu sadaļās.

Turpiniet ar vienuma izveidi un konfigurēšanu.

1 Vienuma izveide

Pārskats

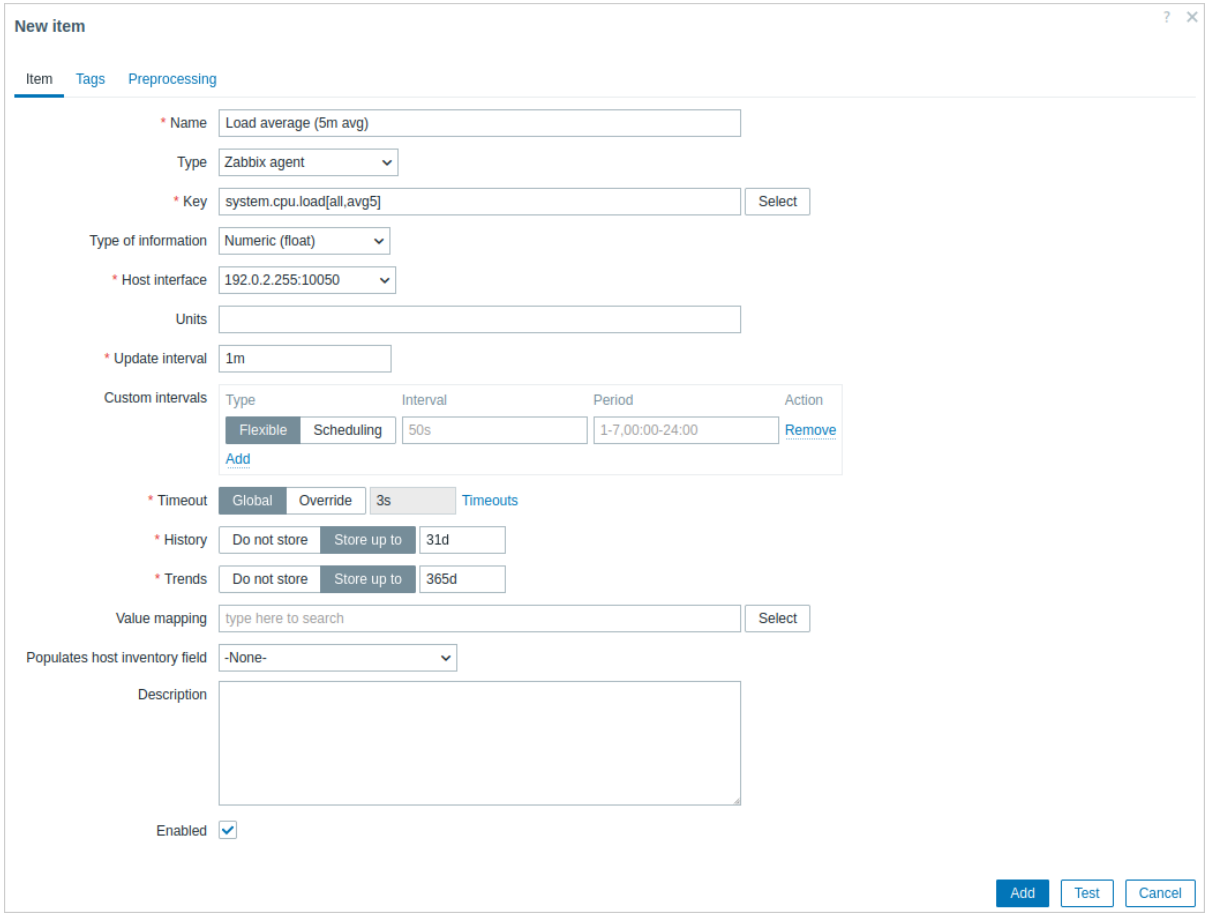
Lai izveidotu vienumu Zabbix lietotāja saskarnē, rīkojieties šādi:

- Dodieties uz: *Datu ievākšana* > *Hosts*
- Noklikšķiniet uz *Vienumi* attiecīgā hosta rindā
- Noklikšķiniet uz *Izveidot vienumu* ekrāna augšējā labajā stūrī
- Veidlapā ievadiet vienuma parametrus

Vienumu var izveidot arī, atverot esošu vienumu, nospiežot pogu *Klonēt* un pēc tam saglabājot to ar citu nosaukumu.

Konfigurācija

Cilne **Vienums** satur vispārīgos vienuma atribūtus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Vienuma nosaukums. Tiek atbalstīti lietotāja makrosi.
Type	Vienuma tips. Skatiet atsevišķās vienumu tipu sadaļas.
Key	Vienuma atslēga (līdz 2048 rakstzīmēm). Atbalstītās vienumu atslēgas var atrast atsevišķās vienumu tipu sadaļās. Atslēgai jābūt unikālai viena hosta ietvaros. Ja atslēgas tips ir 'Zabbix agent', 'Zabbix agent (active)' vai 'Simple check', atslēgas vērtībai jābūt atbalstītai Zabbix aģentā vai Zabbix serverī. Skatiet arī: pareizais atslēgas formāts.
Type of information	Datu tips, ko izmanto vienuma vērtības validēšanai un glabāšanai datubāzē, arī pēc jebkādām konversijām: **Numeric (unsigned)** - 64 bitu neparakstīts vesels skaitlis; **Numeric (float)** - 64 bitu peldošā komata skaitlis; **Character** - īss teksts; **Log** - garš teksts ar vai bez žurnāla īpašībām (laika zīmogs, avots, svarīgums, žurnāla notikuma ID); **Text** - garš teksts; **Binary** - binārs skaitlis (atbalstīts tikai atkarīgajiem vienumiem); **JSON** - strukturēti JSON dati, kas datubāzē tiek glabāti natīvi (nav atbalstīts aprēķinātajiem vienumiem). Vienumi, kas atgriež vērtības ar bināro vai JSON datu tipu, netiek atbalstīti aprēķināto vienumu formulās vai trigeru izteiksmēs. Skatiet vienuma datu ierobežojumus, lai uzzinātu glabāšanas ierobežojumus katram datu tipam un datubāzes aizmugursistēmai. Vienumiem, kas atgriež datus vienā konkrētā formātā, atbilstošais datu tips tiek izvēlēts automātiski — piemēram, `system.cpu.load` un `system.cpu.util` abi atgriež 64 bitu peldošā komata skaitli, tāpēc *Numeric (float)* tiek izvēlēts automātiski.
Host interface	Atlasiet hosta saskarni. Šis lauks ir pieejams, rediģējot vienumu hosta līmenī.

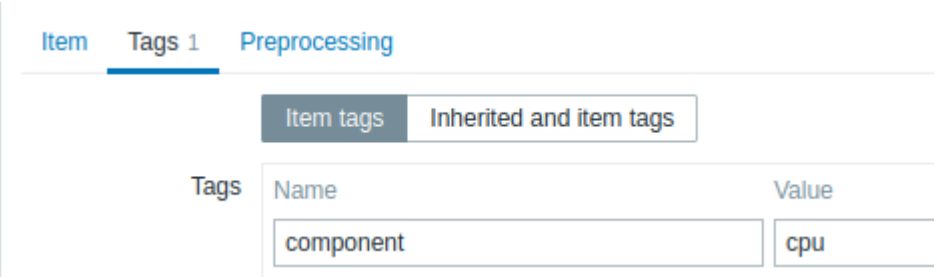
Parametrs	Apraksts
Units	Ja ir iestatīts mērvienības simbols, Zabbix piemēro pēcapstrādi saņemtajai vienuma vērtībai un attēlo to ar norādīto mērvienības sufiksu. Atbalstītie mērvienību simboli ar īpašu formatējumu (un saņemtās vienuma vērtības → attēlotās vērtības piemēriem): **B** - baiti (1024 → 1 KB) **Bps** - baiti sekundē (1024 → 1 KBps) **s** - sekundes, attēlotas, izmantojot līdz trim lielākajām nenulles laika vienībām (881764 → 10d 4h 56m) **uptime** - pagājušais laiks hh:mm:ss vai N days, hh:mm:ss formātā (881764 → 10 days, 04:56:04) **unixtime** - Unix laika zīmogs, formatēts kā yyyy.mm.dd hh:mm:ss (881764 → 1970-01-11 04:56:04 AM); pareizam formatējumam saņemtajai vienuma vērtībai jābūt *Numeric (unsigned)*. Citām mērvienībām (piemēram, Hz, W u.c.), ja saņemtā vērtība pārsniedz 1000, tā tiek dalīta ar 1000 un attēlota ar atbilstošu prefiksu (5000 → 5 KHz, 881764 → 881.76 KW). Ja mērvienības simbolam priekšā ir **!**, mērvienību konvertēšana un prefiksu pievienošana tiek atspējota (1024 !B → 1024 B, 61 !s → 61 s). Lai skatītu vairāk piemēru un detalizētu informāciju par mērvienību simboliem un mērvienību konvertēšanu, skatiet Vienuma vērtību sufiksi.
Update interval	Iegūt jaunu vērtību šim vienumam ik pēc N sekundēm. Maksimāli atļautais atjaunināšanas intervāls ir 86400 sekundes (1 diena). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. Vienam makrosam jāaizpilda viss lauks. Vairāki makrosi vienā laukā vai makrosi kopā ar tekstu netiek atbalstīti. *Piezīme*: atjaunināšanas intervālu var iestatīt uz '0' tikai tad, ja pastāv pielāgoti intervāli ar vērtību, kas nav nulle. Ja tas ir iestatīts uz '0' un pastāv pielāgots intervāls (elastīgs vai plānots) ar vērtību, kas nav nulle, vienums tiks aptaujāts pielāgotā intervāla darbības laikā. *Ņemiet vērā*, ka pirmā vienuma aptauja pēc tam, kad vienums kļūst aktīvs vai pēc atjaunināšanas intervāla maiņas, var notikt agrāk nekā konfigurētā vērtība. Jauni vienumi tiks pārbaudīti 60 sekunžu laikā pēc to izveides, ja vien tiem nav Scheduling vai Flexible atjaunināšanas intervāla un *Update interval* nav iestatīts uz 0. Esošu pasīvu vienumu var nekavējoties aptaujāt, nospiežot pogu *Execute now*.
Custom intervals	Varat izveidot pielāgotus noteikumus vienuma pārbaudei: **Flexible** - izveidot izņēmumu *Update interval* vērtībai (intervāls ar citu biežumu). **Scheduling** - izveidot pielāgotu aptaujas grafiku. Detalizētu informāciju skatiet sadaļā Pielāgotie intervāli. *Interval* laukā tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. Vienam makrosam jāaizpilda viss lauks. Vairāki makrosi vienā laukā vai makrosi kopā ar tekstu netiek atbalstīti.
Timeout	Iestatiet vienuma pārbaudes noildzi (pieejama atbalstītajiem vienumu tipiem). Atlasiet noildzes opciju: **Global** - tiek izmantota starpniekservera/globālā noildze (attēlota pelēkajā *Timeout* laukā). **Override** - tiek izmantota pielāgota noildze (iestatīta *Timeout* laukā; atļautais diapazons: 1 - 600s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makrosi. Noklikšķinot uz saites *Timeouts*, varat konfigurēt starpniekservera noildzes vai globālās noildzes (ja starpniekserveris netiek izmantots). Ņemiet vērā, ka saite *Timeouts* ir redzama tikai *Super admin* tipa lietotājiem ar atļaujām uz lietotāja saskarnes sadaļām *Administration* > *General* vai *Administration* > *Proxies*.

Parametrs	Apraksts
History	Atlasiet vienu no iespējām: **Do not store** - vienuma vēsture netiek glabāta. Noderīgi galvenajiem vienumiem, ja vēsture jāsaglabā tikai atkarīgajiem vienumiem. Šo iestatījumu nevar pārrakstīt ar globālajiem housekeeper iestatījumiem. **Store up to** - norādiet, cik ilgi datubāzē glabāt detalizēto vēsturi (no 1 stundas līdz 25 gadiem). Vecākie dati tiks noņemti ar housekeeper. Tiek glabāts sekundēs. Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Store up to* vērtību var globāli pārrakstīt sadaļā *Administration* > *Housekeeping*. Ja pastāv globāls pārrakstošs iestatījums, tiek attēlota oranža informācijas ikona. Novietojot peles kursoru virs tās, tiek parādīts brīdinājuma ziņojums, piemēram, *Overridden by global housekeeper settings (1d)*. Ieteicams saglabāt ierakstītās vērtības pēc iespējas īsāku laiku, lai samazinātu vērtību vēstures apjomu datubāzē. Tā vietā, lai glabātu ilgu vērtību vēsturi, varat ilgāk glabāt tendenču datus. Skatiet arī Vēsture un tendences.
Trends	Atlasiet vienu no iespējām: **Do not store** - tendences netiek glabātas. Šo iestatījumu nevar pārrakstīt ar globālajiem housekeeper iestatījumiem. **Store up to** - norādiet, cik ilgi datubāzē glabāt agregēto (stundas minimums, maksimums, vidējais, skaits) vēsturi (no 1 dienas līdz 25 gadiem). Vecākie dati tiks noņemti ar housekeeper. Tiek glabāts sekundēs. Tiek atbalstīti laika sufiksi, piemēram, 24h, 1d. Tiek atbalstīti lietotāja makrosi. *Store up to* vērtību var globāli pārrakstīt sadaļā *Administration* > *Housekeeping*. Ja pastāv globāls pārrakstošs iestatījums, tiek attēlota oranža informācijas ikona. Novietojot peles kursoru virs tās, tiek parādīts brīdinājuma ziņojums, piemēram, *Overridden by global housekeeper settings (7d)*. *Piezīme:* Tendenču glabāšana nav pieejama neskaitliskiem datiem - character, log un text. Skatiet arī Vēsture un tendences.
Value mapping	Piemērot vērtību kartēšanu šim vienumam. Vērtību kartēšana nemaina saņemtās vērtības, tā ir paredzēta tikai datu attēlošanai. Tā darbojas ar *Numeric(unsigned)*, *Numeric(float)* un *Character* vienumiem. Piemēram, ”Windows service states”.
Log time format	Pieejams tikai **Log** tipa vienumiem. Atbalstītie vietturi: **y**: *Gads (1970-2038)*. **M**: *Mēnesis (01-12)*. **d**: *Diena (01-31)*. **h**: *Stunda (00-23)*. **m**: *Minūte (00-59)*. **s**: *Sekunde (00-59)*. Ja lauks tiek atstāts tukšs, laika zīmogs Unix laikā tiks iestatīts uz 0, kas atbilst 1970. gada 1. janvārim. Piemēram, apsveriet šādu rindu no Zabbix aģenta žurnāla faila: ” 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211).” Tā sākas ar sešām rakstzīmju pozīcijām PID, kam seko datums, laiks un pārējais ziņojums. Šīs rindas žurnāla laika formāts būtu ”pppppp:yyyyMMdd:hhmmss”. Ņemiet vērā, ka rakstzīmes ”p” un ”:” ir vietturi un var būt jebkuras rakstzīmes, izņemot ”yMdhms”.
Populates host inventory field	Varat atlasīt hosta inventāra lauku, kuru aizpildīs vienuma vērtība. Tas darbosies, ja hostam ir iespējota automātiska inventāra aizpildīšana un tiek izmantoti *Numeric(unsigned)*, *Numeric(float)*, *Character* un *Text* vienumi.
Description	Ievadiet vienuma aprakstu. Tiek atbalstīti lietotāja makrosi.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu vienumu un tas tiktu apstrādāts. Atspējojot vienumu, tas nekavējoties tiek noņemts no vēstures kešatmiņas (izņemot tā pēdējo vērtību, kas tiek saglabāta žurnāliem).
Latest data	Noklikšķiniet uz saites, lai skatītu jaunākos vienuma datus. Šī saite ir pieejama tikai tad, kad rediģējat jau esošu vienumu.

Note:
Vienuma tipam specifiskie lauki ir aprakstīti atbilstošajās lapās.

Note:
Rediģējot esošu veidnes līmeņa vienumu hosta līmenī, vairāki lauki ir tikai lasāmi. Varat izmantot saiti formas galvenē, lai pārietu uz veidnes līmeni un rediģētu tos tur, paturot prātā, ka izmaiņas veidnes līmenī mainīs vienumu visiem hostiem, kuriem šī veidne ir piesaistīta.

Cilne **Tags** ļauj definēt vienuma līmeņa tagus.



Vienuma vērtības priekšapstrāde

Cilne **Priekšapstrāde** ļauj definēt saņemto vērtību transformācijas noteikumus.

Vienuma testēšana

Attention:
Lai veiktu vienuma testēšanu, pārliecinieties, ka sistēmas laiks serverī un starpniekserverī ir sinhronizēts. Ja servera laiks atpaliek, vienuma testēšana var atgriezt kļūdas ziņojumu "The task has been expired." Tomēr atšķirīgu laika joslu iestatīšana serverī un starpniekserverī neietekmēs testēšanas rezultātu.

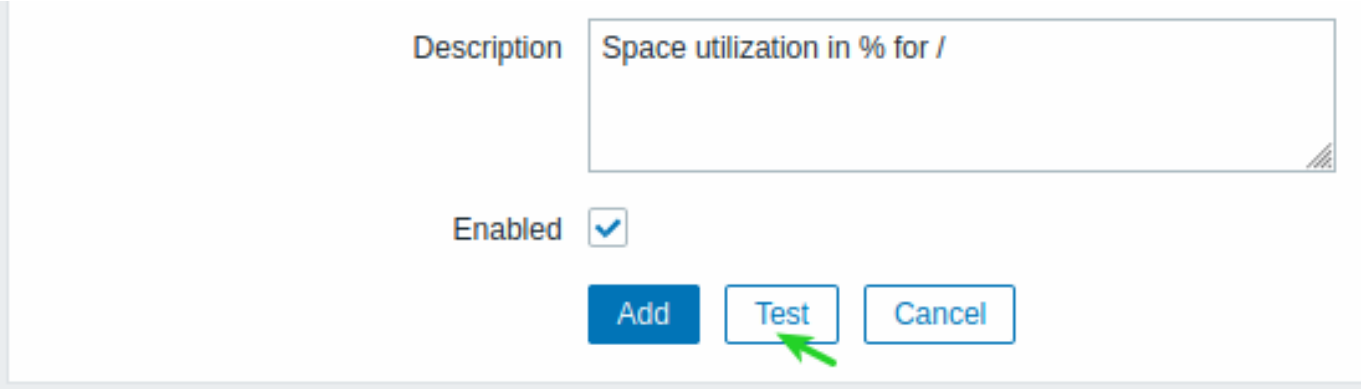
Ir iespējams testēt vienumu un, ja tas ir konfigurēts pareizi, saņemt reālu vērtību. Testēšanu var veikt pat pirms vienums ir saglabāts.

Testēšana ir pieejama hostu un veidņu vienumiem, vienumu prototipiem un zema līmeņa atklāšanas kārtulām. Testēšana nav pieejama aktīvajiem vienumiem.

Vienuma testēšana ir pieejama šādiem pasīvo vienumu tipiem:

- Zabbix aģents
- SNMP aģents (v1, v2, v3)
- IPMI aģents
- SSH pārbaudes
- Telnet pārbaudes
- JMX aģents
- Vienkāršās pārbaudes (izņemot `icmpping*`, `vmware.*` vienumus)
- Zabbix iekšējie
- Aprēķinātie vienumi
- Ārējās pārbaudes
- Datubāzes monitors
- HTTP aģents
- Skripts
- Pārlūks

Lai testētu vienumu, noklikšķiniet uz pogas *Test* vienuma konfigurācijas formas apakšdaļā. Ņemiet vērā, ka poga *Test* būs atspējota vienumiem, kurus nevar testēt (piemēram, aktīvās pārbaudes, izslēgtās vienkāršās pārbaudes).

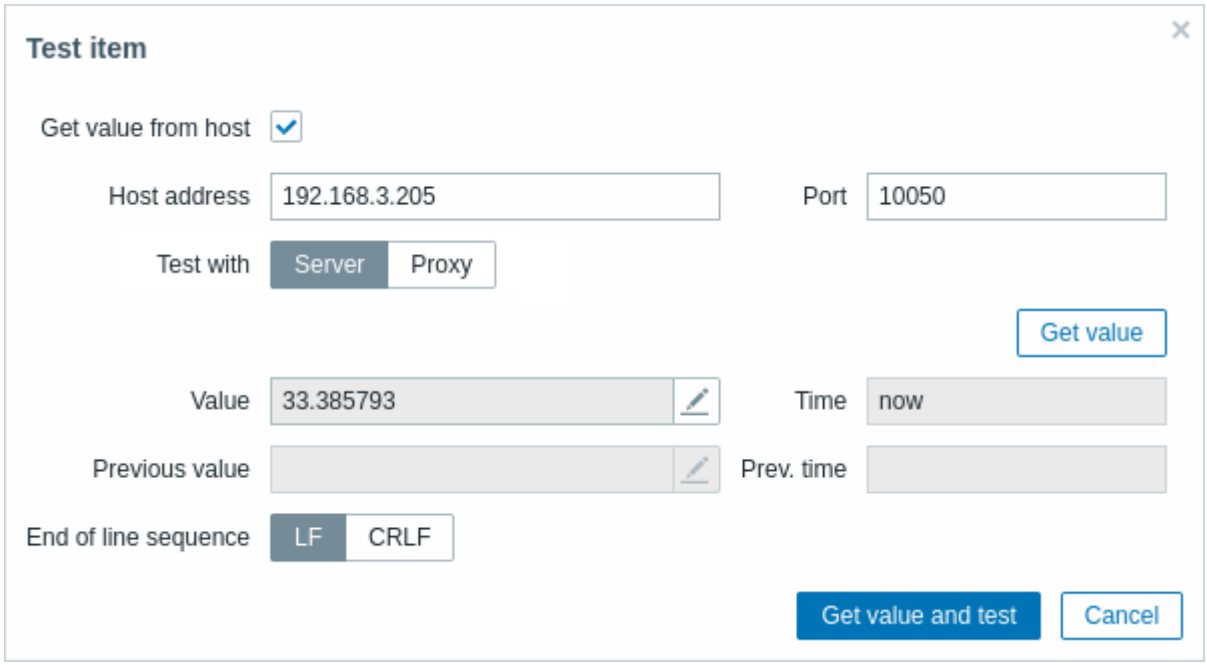


Vienuma testēšanas forma satur laukus nepieciešamajiem hosta parametriem (hosta adrese, ports, testēšana ar serveri/starpniekserveri (starpniekservera nosaukums)) un vienumam specifisku informāciju (piemēram, SNMPv2 kopienu vai SNMPv3 drošības akreditācijas datus). Šie lauki ir atkarīgi no konteksta:

- Vērtības tiek aizpildītas automātiski, kad tas ir iespējams, piemēram, vienumiem, kuriem nepieciešams aģents, izmantojot informāciju no hosta izvēlētās aģenta saskarnes.
- Veidņu vienumiem vērtības jāaizpilda manuāli.
- Vienkārša teksta makro vērtības tiek atrisinātas.
- Lauki, kuros vērtība (vai tās daļa) ir slepens vai Vault makro, ir tukši un jāaizpilda manuāli. Ja kāds vienuma parametrs satur slepena makro vērtību, tiek parādīts šāds brīdinājuma ziņojums: "Item contains user-defined macros with secret values. Values of these macros should be entered manually."
- Lauki tiek atspējoti, ja tie nav nepieciešami konkrētā vienuma tipa kontekstā (piemēram, hosta adreses lauks un starpniekservera lauks ir atspējoti aprēķinātajiem vienumiem)

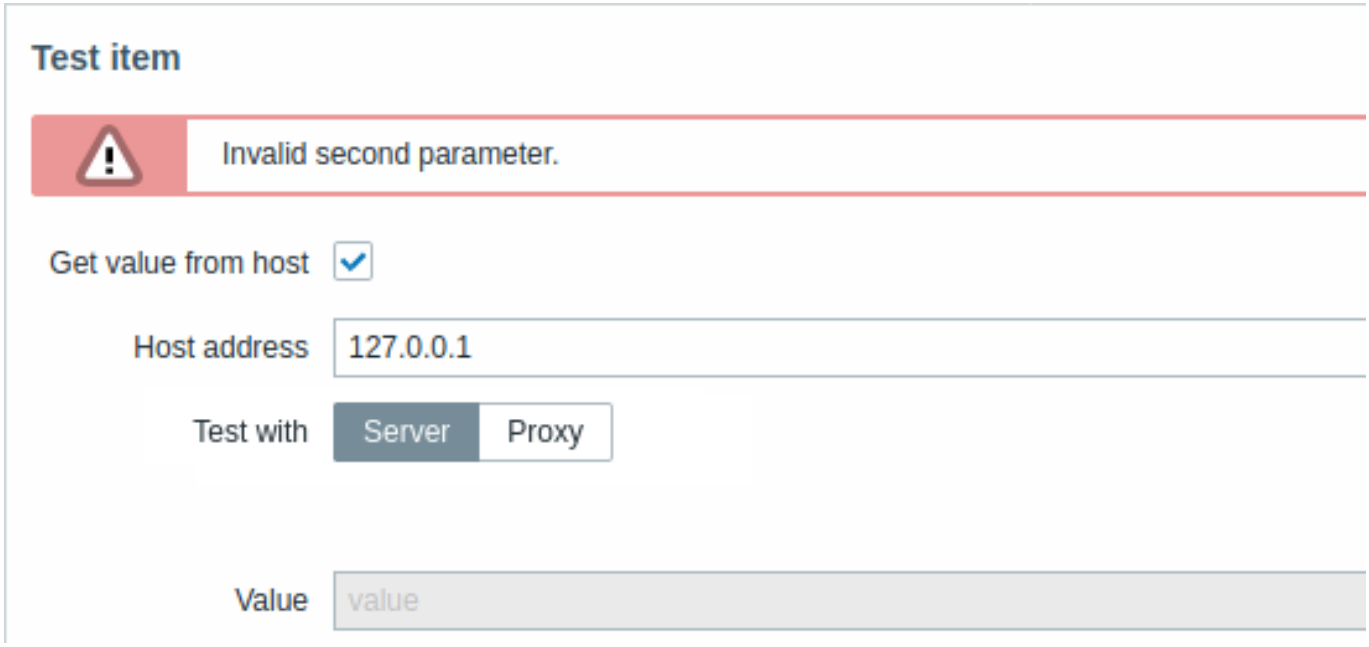
Lai testētu vienumu, noklikšķiniet uz *Get value*. Ja vērtība tiek veiksmīgi iegūta, tā aizpildīs lauku *Value*, pārvietojot pašreizējo vērtību (ja tāda ir) uz lauku *Previous value*, vienlaikus aprēķinot lauku *Prev. time*, t.i., laika starpību starp abām vērtībām (klikšķiem), kā arī mēģinot noteikt EOL secību un pārslēgties uz CRLF, ja iegūtajā vērtībā tiek konstatēts "\n\r".

Noklikšķiniet uz *Get value and test*, lai testētu priekšapstrādi.



No hosta iegūtās vērtības un testēšanas rezultāti tiek saīsināti līdz maksimālajam izmēram 512KB, kad tie tiek nosūtīti uz lietotāja saskarni. Ja testa rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona; novietojiet kursoru virs tās, lai skatītu detalizētu informāciju. Ja vērtība ir saīsināta, secīgos testos, kuros tiek salīdzinātas lielas vērtības (piemēram, JSON), tā var neizturēt datu tipa validāciju; šādos gadījumos tiek parādīts kļūdas ziņojums. Ņemiet vērā, ka dati, kas ir lielāki par 512KB, joprojām tiek pilnībā apstrādāti Zabbix serverī.

Ja konfigurācija ir nepareiza, tiek parādīts kļūdas ziņojums, kurā aprakstīts iespējamais cēlonis.



Veiksmīgi no hosta iegūtu vērtību var izmantot arī, lai testētu priekšapstrādes soļus.

Veidlapas pogas

Pogas veidlapas apakšdaļā ļauj veikt vairākas darbības.

Add	Pievienot vienumu. Šī poga ir pieejama tikai jauniem vienumiem.
Update	Atjaunināt vienuma rekvizītus.
Clone	Izveidot citu vienumu, pamatojoties uz pašreizējā vienuma rekvizītiem.
Execute now	Nekavējoties izpildīt pārbaudi jaunai vienuma vērtībai. Atbalstīts tikai **pasīvajām** pārbaudēm (skatiet sīkāku informāciju). *Ņemiet vērā*, ka, nekavējoties pārbaudot vērtību, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ vērtība neatspoguļos pavisam nesenās izmaiņas vienuma konfigurācijā.
Test	Pārbaudīt, vai vienuma konfigurācija ir pareiza, iegūstot vērtību.
Clear history and trends	Dzēst vienuma vēsturi un tendences.
Delete	Dzēst vienumu.
Cancel	Atcelt vienuma rekvizītu rediģēšanu.

Vienumu datu ierobežojumi

Vienumu datu ierobežojumi ir atkarīgi no vienuma datu tipa un datubāzes aizmugursistēmas.

Skaitliskās (bez zīmes) vērtības tiek saglabātas tādas, kādas tās ir. Ja tiek saņemta peldošā komata vērtība, tiek saglabāta tikai veselā daļa (piemēram, 1.23 → 1).

Skaitliskās (peldošā komata) vērtības tiek saglabātas tādas, kādas tās ir, un atbalsta aptuveni 15 līdz 17 precizitātes ciparus diapazonā no aptuveni -1.79E+308 līdz 1.79E+308. Tiek atbalstītas arī vērtības zinātniskajā pierakstā (piemēram, 1.23E+7, 1e308, 1.1E-4).

Teksta vērtības pirms saglabāšanas tiek saīsinātas, lai atbilstu datubāzes vērtības tipa ierobežojumam:

Datubāze	Informācijas tips		
	Rakstzīmes	**Žurnāls**	**Teksts**
MySQL	255 rakstzīmes	65536 baiti	65536 baiti

Datubāze	Informācijas tips		
PostgreSQL	255 rakstzīmes	65536 rakstzīmes	65536 rakstzīmes
SQLite (tikai Zabbix starpniekserveris)	255 rakstzīmes	65536 rakstzīmes	65536 rakstzīmes

Binārās vērtības ir ierobežotas līdz 16MiB (16777216 baitiem). Vērtības, kas pārsniedz šo ierobežojumu, tiek atmestas, un lietotāja saskarnē tiek parādīta atbilstoša kļūda.

Vērtības ar JSON datu tipu ir ierobežotas līdz 128MiB (134217728 baitiem). Vērtības, kas pārsniedz šo ierobežojumu, tiek atmestas, un lietotāja saskarnē tiek parādīta atbilstoša kļūda. MySQL/MariaDB gadījumā apsveriet iespēju iestatīt ikdienas particionēšanu tabulai `history_json` (ko izmanto JSON vērtību glabāšanai), jo tā var ātri kļūt ļoti liela un uz ilgāku laiku bloķēt housekeeper. Ja jūsu lietošanas scenārijs ietver lielas JSON vērtības (1MiB vai vairāk), skatiet, kā konfigurēt savu sistēmu lielu JSON vērtību atbalstam.

Kad Zabbix ieraksta jebkādus vienuma datus datubāzē, tas izmanto INSERT vaicājumus un reģistrē tos žurnālā (ja `DebugLevel` ir iestatīts uz 4 vai 5). Lieli žurnāla ieraksti tiek saīsināti līdz 64KB.

Pielāgota skripta ierobežojums

Pieejamais pielāgotā skripta garums ir atkarīgs no izmantotās datubāzes:

Datubāze	Ierobežojums rakstzīmēs	Ierobežojums baitos
MySQL	65535	65535
PostgreSQL	65535	nav ierobežots
SQLite (tikai Zabbix starpniekserveris)	65535	nav ierobežots

Vienuma noildze

Vienuma noildze nosaka, cik ilgi Zabbix jāgaida, pirms pārtraukt pārbaudi kā neveiksmīgu.

Kad tiek sasniegta noildze, pārbaude tiek pārtraukta pat tad, ja datu iegūšana vēl nav pabeigta. Ja dati tiek saņemti tikai daļēji, vienums kļūst neatbalstīts un kļūdas ziņojums tiek ierakstīts žurnālā (piemēram, ja SNMP pārbaudē dati tiek veiksmīgi savākti tikai vienam no vairākiem OID).

Daudziem vienumu tipiem var iestatīt **elastīgas** vienuma noildzes:

- katram atsevišķam vienumam
- katram vienuma tipam (starpniekserveris līmenī)
- katram vienuma tipam (globālā līmenī)

Pielāgota noildze *katram atsevišķam vienumam* ir noderīga, ja vēlaties noteikt garāku noildzi konkrētam vienumam, vienlaikus saglabājot zemas noildzes citiem vienumiem.

Noildzes prioritāte

1. Atsevišķa vienuma noildze ignorē jebkuras citas noildzes.
2. Starpniekservera līmeņa noildzes ignorē globālās noildzes.

Noildze no Zabbix serveris vai starpniekserveris konfigurācijas **neietekmē** pārbaudes, kurām ir iestatītas elastīgās noildzes.

Elastīga taimauta atbalsts

Elastīgi vienuma taimauti tiek atbalstīti šādiem vienumu tipiem:

- Zabbix aģents (gan pasīvās, gan aktīvās pārbaudes)
- SNMP aģents (izņemot mantotās SNMP pārbaudes[1])
- Vienkāršā pārbaude (izņemot `icmpping*` un VMware vienumus[2])
- SSH aģents
- Telnet aģents
- Ārējā pārbaude
- Datubāzes monitors
- Skripts
- HTTP aģents
- Pārlūks

[1] Mantotajām SNMP pārbaudēm (viena OID aptaujāšana) tiek piemēroti taimauta iestatījumi no serveris vai starpniekserveris.
[2] `icmpping*` vienumiem taimauta vērtība tiek norādīta tieši vienuma atslēgā; tiek piemērots arī stingri kodēts maksimālais iespējamais taimauts 600 sekundes. VMware uzraudzības vienumi izmanto savu VMwareTimeout parametru no serveris vai starpniekserveris konfigurācijas.

Neatbalstīti vienumi

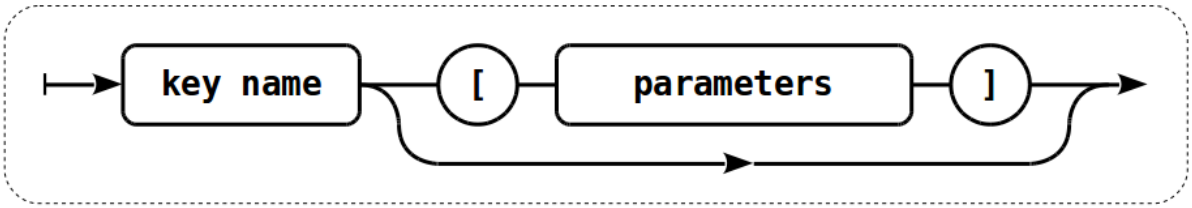
Par neatbalstītiem vienumiem tiek ziņots ar statusu *Not supported*, un tie joprojām tiek atkārtoti pārbaudīti to standarta *Update interval* laikā.

Vienums kļūst neatbalstīts, ja kāda iemesla dēļ nevar izgūt tā vērtību (piemēram, savienojuma kļūda, nav konfigurētu aptaujātāju, kas apstrādātu vienumu). Turklāt vienumi, kas nesaņem vēsturiskos datus, paliek statusā *Not supported*. Tā ir paredzēta darbība — vienums pāriet statusā *Normal* tikai tad, kad tas saņem jaunus, derīgus vēsturiskos datus.

Tāpat kā problēmas, neatbalstītie vienumi tiek pārvērtēti tikai tad, kad tiek saņemti jauni dati — pat ja šim vienumam vēsturiskie dati vairs nav pieejami. Citiem vārdiem sakot, vienumi un trigeri maina savu stāvokli tikai pēc jaunu datu saņemšanas. Tomēr, ja trigera izteiksme ietver date and time un/vai nodata() funkciju, trigeris tiks pārrēķināts ik pēc 30 sekundēm (sīkāku informāciju skatiet sadaļā trigeru aprēķināšanas laiks).

1 Vienuma atslēgas formāts

Vienuma atslēgas formātam, tostarp atslēgas parametriem, jāatbilst sintakses noteikumiem. Tālāk redzamajās ilustrācijās ir attēlota atbalstītā sintakse. Atļautos elementus un rakstzīmes katrā punktā var noteikt, sekojot bultiņām — ja kādu bloku var sasniegt pa līniju, tas ir atļauts, ja ne — tas nav atļauts.



Lai izveidotu derīgu vienuma atslēgu, vispirms jānorāda atslēgas nosaukums, pēc tam var izvēlēties, vai izmantot parametrus vai neizmantot tos — kā parādīts ar divām līnijām, kurām var sekot.

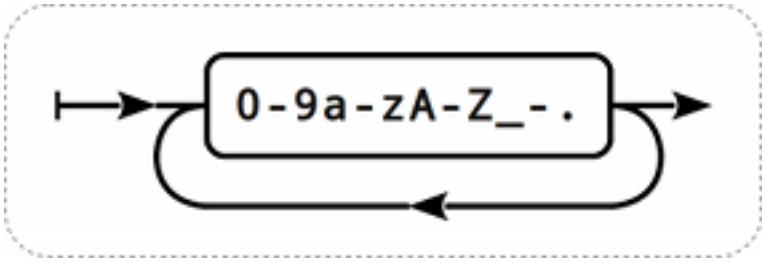
Atslēgas nosaukums

Pašam atslēgas nosaukumam ir ierobežots atļauto rakstzīmju diapazons, kuras vienkārši seko cita citai. Atļautās rakstzīmes ir:

```
0-9a-zA-Z_-.
```

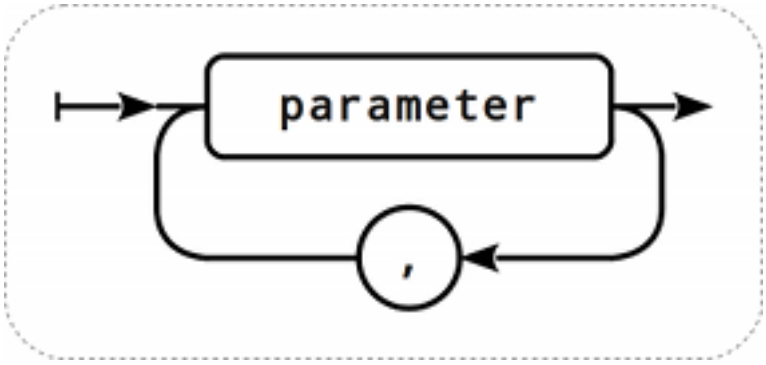
Tas nozīmē:

- visi cipari;
- visi mazie burti;
- visi lielie burti;
- pasvītrojuma zīme;
- defise;
- punkts.

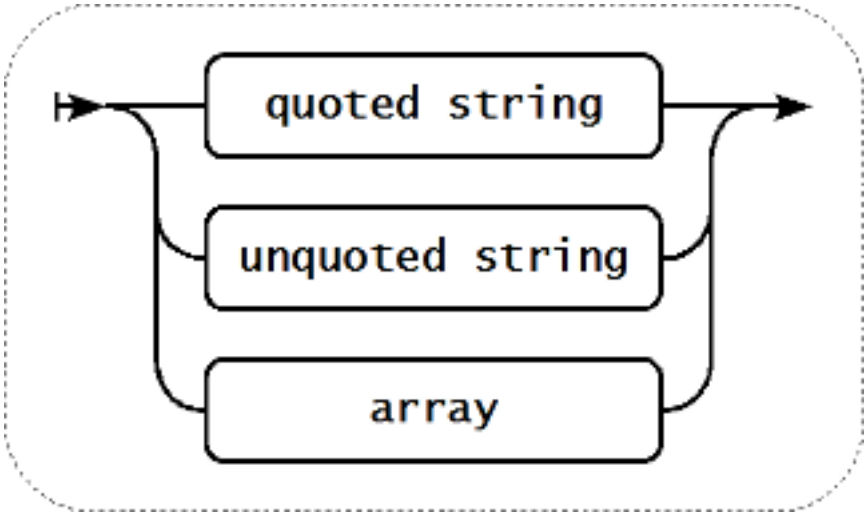


Atslēgas parametri

Vienuma atslēgai var būt vairāki parametri, kas ir atdalīti ar komatiem.



Katrs atslēgas parametrs var būt vai nu pēdiņās ievietota virkne, virkne bez pēdiņām vai masīvs.

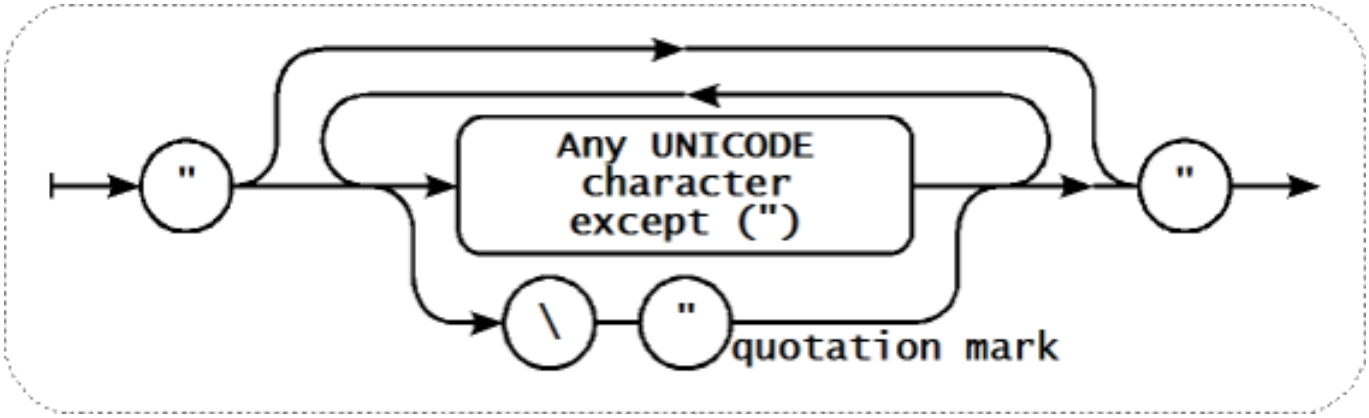


Parametru var arī atstāt tukšu, tādējādi izmantojot noklusējuma vērtību. Šādā gadījumā ir jāpievieno atbilstošs komatu skaits, ja ir norādīti vēl citi parametri. Piemēram, vienuma atslēga **icmpping[„200„500]** norāda, ka intervāls starp atsevišķiem ping pieprasījumiem ir 200 milisekundes, noildze — 500 milisekundes, bet visiem pārējiem parametriem tiek izmantotas to noklusējuma vērtības.

Parametros ir iespējams iekļaut makrosus. Tie var būt lietotāja makrosi vai daži no iebūvētajiem makrosiem. Lai redzētu, kuri konkrētie iebūvētie makrosi tiek atbalstīti vienuma atslēgas parametros, lapā Atbalstītie makrosi meklējiet ”item key parameters”.

Parametrs — pēdiņās ietverta virkne

Ja atslēgas parametrs ir pēdiņās ietverta virkne, ir atļauta jebkura Unicode rakstzīme. Ja atslēgas parametra virkne satur pēdiņu, šim parametram jābūt ietvertam pēdiņās, un katra pēdiņa, kas ir parametra virknes daļa, ir jāekranē ar slīpsvītras (\) rakstzīmi. Ja atslēgas parametra virkne satur komatu, šim parametram jābūt ietvertam pēdiņās.

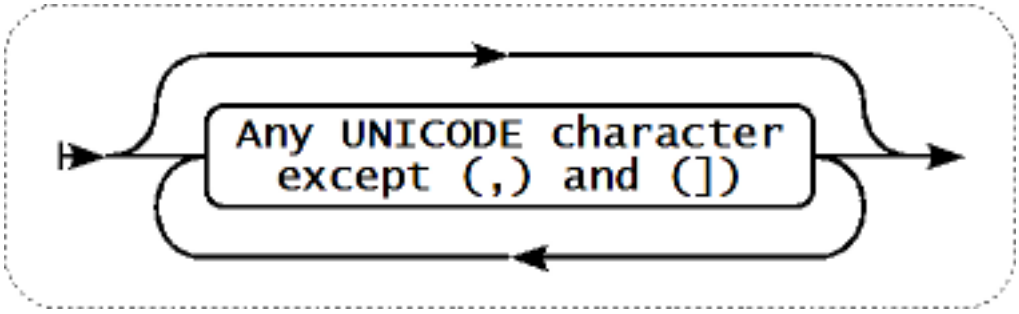


Warning:
Lai liktu vienums atslēgas parametrus pēdiņās, izmantojiet tikai dubultpēdiņas. Vienkāršās pēdiņas netiek atbalstītas.

Attention:
Daudzlīmeņu parametru masīvi, piemēram, `[a,[b,[c,d]],e]`, nav atļauti.

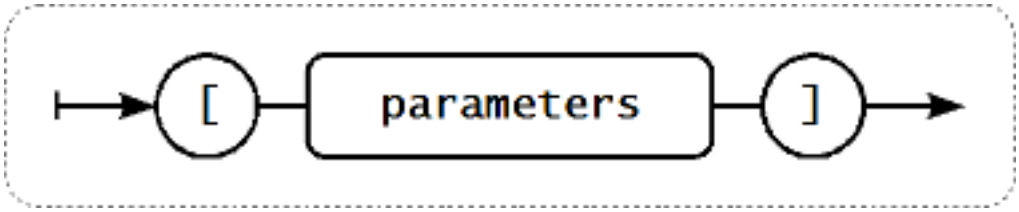
Parametrs — neievietota virkne pēdiņās

Ja atslēgas parametrs ir virkne bez pēdiņām, ir atļauta jebkura Unicode rakstzīme, izņemot komatu un labo kvadrātiekavu (]). Parametrs bez pēdiņām nedrīkst sākties ar kreiso kvadrātiekavu ([).



Parametrs - masīvs

Ja atslēgas parametrs ir masīvs, tas atkal tiek ietverts kvadrātiekavās, kur atsevišķi parametri tiek norādīti saskaņā ar vairāku parametru norādīšanas noteikumiem un sintaksi.



2 Pielāgoti intervāli

Pārskats

Ir iespējams izveidot pielāgotus noteikumus attiecībā uz laikiem, kad tiek pārbaudīts vienums. Tam ir paredzētas divas metodes: *Elastīgie intervāli*, kas ļauj no jauna definēt noklusējuma atjaunināšanas intervālu, un *Plānošana*, ar kuras palīdzību vienuma pārbaudi var izpildīt noteiktā laikā vai noteiktā laiku secībā.

Elastīgi intervāli

Elastīgi intervāli ļauj pārdefinēt noklusējuma atjaunināšanas intervālu noteiktiem laika periodiem. Elastīgs intervāls tiek definēts ar *Intervāls* un *Periods*, kur:

- *Intervāls* – atjaunināšanas intervāls norādītajam laika periodam. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d.
- *Periods* – laika periods, kad elastīgais intervāls ir aktīvs (detalizētu *Perioda* formāta aprakstu skatiet sadaļā laika periodi)

Ja vairāki elastīgie intervāli pārklājas, pārklāšanās periodam tiek izmantota mazākā *Intervāla* vērtība. Ņemiet vērā, ka, ja pārklājošos elastīgo intervālu mazākā vērtība ir ’0’, aptauja netiks veikta. Ārpus elastīgajiem intervāliem tiek izmantots noklusējuma atjaunināšanas intervāls.

Ņemiet vērā, ka, ja elastīgais intervāls ir vienāds ar perioda garumu, vienums tiks pārbaudīts tieši vienu reizi. Ja elastīgais intervāls ir lielāks par periodu, vienums var tikt pārbaudīts vienu reizi vai arī netikt pārbaudīts vispār (tādēļ šāda konfigurācija nav ieteicama). Ja elastīgais intervāls ir mazāks par periodu, vienums tiks pārbaudīts vismaz vienu reizi.

Ja elastīgais intervāls ir iestatīts uz ’0’, vienums elastīgā intervāla periodā netiek aptaujāts un pēc perioda beigām atsāk aptauju saskaņā ar noklusējuma *Atjaunināšanas intervālu*. Piemēri:

Interval	Period	Apraksts
10	1-5,09:00-18:00	Vienums tiks pārbaudīts ik pēc 10 sekundēm darba laikā.
0	1-7,00:00-7:00	Vienums netiks pārbaudīts naktī.
0	7-7,00:00-24:00	Vienums netiks pārbaudīts svētdienās.
60	1-7,12:00-12:01	Vienums tiks pārbaudīts katru dienu plkst. 12:00. Ņemiet vērā, ka tas tika izmantots kā pagaidu risinājums plānotajām pārbaudēm, un šādām pārbaudēm ieteicams izmantot plānošanas intervālus.

Plānošanas intervāli

Plānošanas intervāli tiek izmantoti, lai pārbaudītu vienumus noteiktos laikos. Kamēr elastīgie intervāli ir paredzēti, lai no jauna definētu vienuma noklusējuma atjaunināšanas intervālu, plānošanas intervāli tiek izmantoti, lai norādītu neatkarīgu pārbaudes grafiku, kas tiek izpildīts paralēli.

Plānošanas intervāls tiek definēts šādi: `md<filter>wd<filter>h<filter>m<filter>s<filter>`, kur:

- **md** - mēneša dienas
- **wd** - nedēļas dienas
- **h** - stundas
- **m** - minūtes
- **s** – sekundes

`<filter>` tiek izmantots, lai norādītu vērtības tā prefiksam (dienām, stundām, minūtēm, sekundēm), un tas tiek definēts šādi: `[<from>[-<to>]][/<step>][,<filter>]`, kur:

- `<from>` un `<to>` definē atbilstošo vērtību diapazonu (ieskaitot). Ja `<to>` ir izlaists, tad filtrs atbilst diapazonam `<from>` – `<from>`. Ja arī `<from>` ir izlaists, tad filtrs atbilst visām iespējamajām vērtībām.
- `<step>` definē skaitliskās vērtības izlaišanas soli diapazonā. Pēc noklusējuma `<step>` vērtība ir 1, kas nozīmē, ka atbilst visas definētā diapazona vērtības.

Lai gan filtru definīcijas nav obligātas, jāizmanto vismaz viens filtrs. Filtram jābūt definētam vai nu diapazonam, vai *<step>* vērtībai.

Tukšs filtrs atbilst vai nu '0', ja nav definēts zemāka līmeņa filtrs, vai visām iespējamajām vērtībām pretējā gadījumā. Piemēram, ja stundu filtrs ir izlaists, tad atbildīs tikai '0' stunda, ja arī minūšu un sekunžu filtri ir izlaisti; pretējā gadījumā tukšs stundu filtrs atbildīs visām stundu vērtībām.

Derīgās `<from>` un `<to>` vērtības attiecīgajam filtra prefiksam ir:

Prefix	Description	*<from>*	*<to>*
md	Mēneša dienas	1-31	1-31
wd	Nedēļas dienas	1-7	1-7
h	Stundas	0-23	0-23
m	Minūtes	0-59	0-59
s	Sekundes	0-59	0-59

`<from>` vērtībai jābūt mazākai vai vienādai ar `<to>` vērtību. `<step>` vērtībai jābūt lielākai vai vienādai ar 1 un mazākai vai vienādai ar `<to>` - `<from>`.

Viencipara mēneša dienu, stundu, minūšu un sekunžu vērtībām var pievienot prefiksu 0. Piemēram, `md01-31` un `h/02` ir derīgi intervāli, bet `md01-031` un `wd01-07` nav.

Zabbix lietotāja saskarnē vairāki plānošanas intervāli tiek ievadīti atsevišķās rindās. Zabbix API tie tiek apvienoti vienā virknē, izmantojot semikolu `;` kā atdalītāju.

Ja laikam atbilst vairāki intervāli, tas tiek izpildīts tikai vienu reizi. Piemēram, `wd1h9;h9` tiks izpildīts tikai vienu reizi pirmdien plkst. 9:00.

Piemēri:

Interval	Will be executed
m0-59	katru minūti
h9-17/2	ik pēc 2 stundām, sākot no 9:00 (9:00, 11:00 ...)
m0,30 or m/30	katru stundu plkst. hh:00 un hh:30
m0,5,10,15,20,25,30,35,40,45,50,55 or m/5	ik pēc piecām minūtēm
wd1-5h9	katru pirmdienu līdz piektdienai plkst. 9:00
wd1-5h9-18	katru pirmdienu līdz piektdienai plkst. 9:00,10:00,...,18:00
h9,10,11 or h9-11	katru dienu plkst. 9:00, 10:00 un 11:00
md1h9m30	katra mēneša 1. dienā plkst. 9:30
md1wd1h9m30	katra mēneša 1. dienā plkst. 9:30, ja tā ir pirmdiena
h9m/30	katru dienu plkst. 9:00, 9:30
h9m0-59/30	katru dienu plkst. 9:00, 9:30
h9,10m/30	katru dienu plkst. 9:00, 9:30, 10:00, 10:30
h9-10m30	katru dienu plkst. 9:30, 10:30
h9m10-40/30	katru dienu plkst. 9:10, 9:40
h9,10m10-40/30	katru dienu plkst. 9:10, 9:40, 10:10, 10:40
h9-10m10-40/30	katru dienu plkst. 9:10, 9:40, 10:10, 10:40
h9m10-40	katru dienu plkst. 9:10, 9:11, 9:12, ... 9:40
h9m10-40/1	katru dienu plkst. 9:10, 9:11, 9:12, ... 9:40

Interval	Will be executed
h9-12,15	katru dienu plkst. 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0	katru dienu plkst. 9:00, 10:00, 11:00, 12:00, 15:00
h9-12,15m0s30	katru dienu plkst. 9:00:30, 10:00:30, 11:00:30, 12:00:30, 15:00:30
h9-12s30	katru dienu plkst. 9:00:30, 9:01:30, 9:02:30 ... 12:58:30, 12:59:30
h9m/30;h10 (*API-specific syntax*)	katru dienu plkst. 9:00, 9:30, 10:00
h9m/30 h10 (*add this as another row in frontend*)	katru dienu plkst. 9:00, 9:30, 10:00

Laika joslu saskaņošana starpniekserveriem un aģentam

Ņemiet vērā, ka Zabbix starpniekserveri un aģents, apstrādājot plānošanas intervālus, izmanto savas lokālās laika joslas.

Tādēļ, ja plānošanas intervāli tiek piemēroti vienumiem, kurus uzrauga Zabbix starpniekserveris, vai aģenta aktīvajiem vienumiem, ieteicams attiecīgajiem starpniekserveriem vai aģentam iestatīt tādu pašu laika joslu kā Zabbix serverim, pretējā gadījumā rinda var nepareizi ziņot par vienumu aizturēm.

Zabbix starpniekservera vai aģenta laika joslu var iestatīt, izmantojot vides mainīgo TZ `systemd` vienības failā:

```
[Service]
...
Environment="TZ=Europe/Amsterdam"
```

2 Vienuma vērtības priekšapstrāde

Pārskats

Priekšapstrāde ļauj piemērot pārveidojumus saņemtajām vienuma vērtībām pirms to saglabāšanas datubāzē. Šie pārveidojumi/priekšapstrādes soļi tiek izpildīti Zabbix serverī vai starpniekserverī (ja vienumus uzrauga starpniekserveris).

Šī iespēja atbalsta dažādus lietošanas scenārijus, piemēram:

- baitu pārveidošanu bitos (piemēram, reizinot tīkla datplūsmas vērtības ar ”8”);
- sekundes statistikas aprēķināšanu pakāpeniski pieaugošām vērtībām;
- regulāro izteiksmju piemērošanu, lai iegūtu vai modificētu vērtības;
- pielāgotu skriptu izpildi vērtībām;
- nemainīgu vērtību atmešanu, lai optimizētu datubāzes glabātuves izmantošanu.

Vienumam var konfigurēt vienu vai vairākus priekšapstrādes soļus. Šie soļi tiek izpildīti to konfigurēšanas secībā.

Attention:
Ja priekšapstrādes solis neizdodas, vienums kļūst neatbalstīts. No tā var izvairīties, izmantojot kļūdu apstrādi *Custom on fail* (pieejama lielākajai daļai pārveidojumu), kas ļauj atmest vērtības vai iestatīt pielāgotas vērtības.

 Žurnāla vienumiem žurnāla metadati (bez vērtības) vienmēr atiestatīs vienuma neatbalstīto stāvokli, atkal padarot to atbalstītu. Tas notiek pat tad, ja sākotnējā kļūda radās pēc žurnāla vērtības saņemšanas no aģenta.

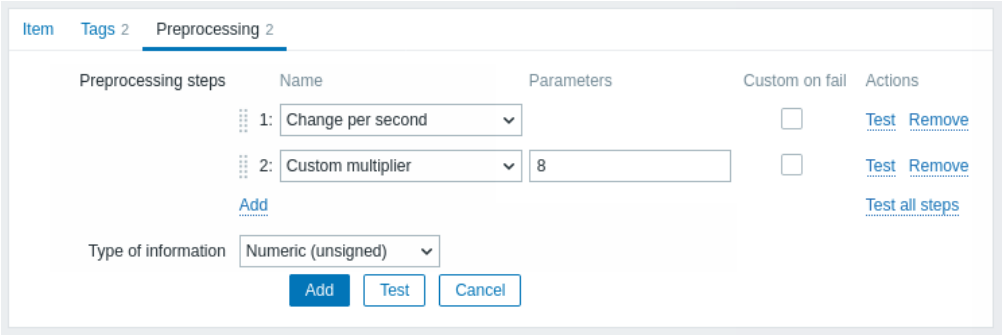
Visas vērtības, kas tiek nodotas priekšapstrādei, sākotnēji tiek apstrādātas kā virknes. Pārveidošana uz vēlamo vērtības tipu (kā definēts vienuma konfigurācijā) tiek veikta priekšapstrādes procesa beigās. Tomēr noteikti priekšapstrādes soļi vajadzības gadījumā var izraisīt agrāku pārveidošanu. Detalizētai tehniskai informācijai skatiet Priekšapstrādes informācija.

Lai pārliecinātos, ka jūsu priekšapstrādes konfigurācija darbojas, kā paredzēts, varat to pārbaudīt.

Skatiet arī: Priekšapstrādes piemēri

Konfigurācija

Priekšapstrādes soļi tiek definēti vienuma konfigurācijas formā cilnē **Priekšapstrāde**.



Noklikšķiniet uz *Pievienot*, lai atlasītu atbalstītu transformāciju.

Lauks *Informācijas tips* tiek parādīts cilnes apakšdaļā, kad ir definēts vismaz viens priekšapstrādes solis. Ja nepieciešams, informācijas tipu ir iespējams mainīt, neatstājot cilni *Priekšapstrāde*. Detalizētu parametru aprakstu skatiet sadaļā Vienuma izveide.

Atbalstītās transformācijas

Tālāk ir uzskaitītas visas atbalstītās transformācijas. Noklikšķiniet uz transformācijas nosaukuma, lai skatītu pilnu informāciju par to.

Nosaukums	Apraksts	Tips
Regulārā izteiksme	Salīdzināt vērtību ar regulāro izteiksmi un aizstāt to ar nepieciešamo izvadi.	Teksts
Aizstāt	Atrast meklējamo virkni un aizstāt to ar citu (vai neko).	
Apgriezt	Noņemt norādītās rakstzīmes no vērtības sākuma un beigām.	
Apgriezt pa labi	Noņemt norādītās rakstzīmes no vērtības beigām.	
Apgriezt pa kreisi	Noņemt norādītās rakstzīmes no vērtības sākuma.	
XML XPath	Izgūt vērtību vai fragmentu no XML datiem, izmantojot XPath funkcionalitāti.	Strukturēti dati
JSON Path	Izgūt vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti.	
CSV uz JSON	Pārveidot CSV faila datus JSON formātā.	
XML uz JSON	Pārveidot XML formāta datus uz JSON.	
SNMP walk vērtība	Izgūt vērtību pēc norādītā OID/MIB nosaukuma un piemērot formatēšanas opcijas.	SNMP
SNMP walk uz JSON	Pārveidot SNMP vērtības uz JSON.	
SNMP get vērtība	Piemērot formatēšanas opcijas SNMP get vērtībai.	
Pielāgots reizinātājs	Reizināt vērtību ar norādīto veselo skaitli vai peldošā komata vērtību.	Aritmētika
Vienkārša izmaiņa	Aprēķināt starpību starp pašreizējo un iepriekšējo vērtību.	Izmaiņa
Izmaiņa sekundē	Aprēķināt vērtības izmaiņas ātrumu sekundē (starpību starp pašreizējo un iepriekšējo vērtību).	
Būla vērtība uz decimālo	Pārveidot vērtību no būla formāta uz decimālo.	Skaitļu sistēmas
Oktālais uz decimālo	Pārveidot vērtību no oktālā formāta uz decimālo.	
Heksadecimālais uz decimālo	Pārveidot vērtību no heksadecimālā formāta uz decimālo.	
JavaScript	Ievadīt JavaScript kodu.	Pielāgoti skripti
Diapazonā	Definēt diapazonu, kurā vērtībai jāatrodas.	Validācija
Atbilst regulārajai izteiksmei	Norādīt regulāro izteiksmi, kurai vērtībai jāatbilst.	
Neatbilst regulārajai izteiksmei	Norādīt regulāro izteiksmi, kurai vērtība nedrīkst atbilst.	
Pārbaudīt kļūdu JSON	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath.	
Pārbaudīt kļūdu XML	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath.	
Pārbaudīt kļūdu, izmantojot regulāro izteiksmi	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, izmantojot regulāro izteiksmi.	
Pārbaudīt neatbalstītu vērtību	Pārbaudīt, vai neizdevās iegūt nevienu vienuma vērtību.	
Atmest nemainīgu	Atmest vērtību, ja tā nav mainījusies.	Datu plūsmas ierobežošana
Atmest nemainīgu ar sirdspukstu	Atmest vērtību, ja tā nav mainījusies noteiktajā laika periodā.	

Nosaukums	Apraksts	Tips
Prometheus modelis	Izmantot šādu vaicājumu, lai izgūtu nepieciešamos datus no Prometheus metrikām.	Prometheus
Prometheus uz JSON	Pārveidot nepieciešamās Prometheus metrikas uz JSON.	

Ņemiet vērā, ka *Izmaiņas* un *Datu plūsmas ierobežošanas* priekšapstrādes soļiem Zabbix ir jāatceras pēdējā vērtība, lai pēc nepieciešamības aprēķinātu/salīdzinātu jauno vērtību. Šīs iepriekšējās vērtības apstrādā priekšapstrādes pārvaldnieks. Ja Zabbix serveris vai starpniekserveris tiek restartēts vai priekšapstrādes soļos tiek veiktas jebkādas izmaiņas, atbilstošā vienuma pēdējā vērtība tiek atiestatīta, kā rezultātā:

- *Vienkāršas izmaiņas* un *Izmaiņas sekundē* soļiem — nākamā vērtība tiks ignorēta, jo nav iepriekšējās vērtības, no kuras aprēķināt izmaiņu;
- *Atmest nemainīgu* un *Atmest nemainīgu ar sirdspukstu* soļiem — nākamā vērtība nekad netiks atmesta, pat ja saskaņā ar atmešanas noteikumiem tai vajadzētu tikt atmestai.

Regulārā izteiksme

Salīdziniet vērtību ar regulāro izteiksmi un aizstājiet to ar nepieciešamo izvadi.

Parametri:

- **pattern** - regulārā izteiksme;
- **output** - izvades formatēšanas veidne. Atkāpju secība \N (kur N=1...9) tiek aizstāta ar N-to atrasto grupu. Atkāpju secība \0 tiek aizstāta ar atrasto tekstu.

Komentāri:

- Ja ievades vērtība neatbilst, vienums kļūs neatbalstīts;
- Regulārā izteiksme atbalsta ne vairāk kā 10 uztvertu grupu izvilkšanu ar \N secību;
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.
- Lūdzu, skatiet sadaļu regulārās izteiksmes, kur pieejami daži esoši piemēri.

Aizstāt

Atrodiet meklējamo virkni un aizstājiet to ar citu virkni (vai neko).

Parametri:

- **meklējamā virkne** - virkne, ko atrast un aizstāt, reģistrjutīga (obligāti);
- **aizstāšana** - virkne, ar ko aizstāt meklējamo virkni. Aizstāšanas virkne var būt arī tukša, tādējādi ļaujot dzēst meklējamo virkni, kad tā tiek atrasta.

Komentāri:

- Tiks aizstātas visas meklējamās virknes sakritības;
- Ir iespējams izmantot atsoļa secības, lai meklētu vai aizstātu rindas pārtraukumus, rindas atgriezi, tabulācijas un atstarpes ”\n \r \t \s”; atpakaļvērsto slīpsvītru var atsoļot kā ”\\” un atsoļa secības var atsoļot kā ”\\n”;
- Rindas pārtraukumu, rindas atgriezes un tabulāciju atsoļošana tiek veikta automātiski zema līmeņa atklāšanas laikā.

Apgriezt

Noņemt norādītās rakstzīmes no vērtības sākuma un beigām.

Labā apgriešana

Noņemt norādītās rakstzīmes no vērtības beigām.

Kreisā apgriešana

Noņem norādītās rakstzīmes no vērtības sākuma.

XML XPath

Izgūt vērtību vai fragmentu no XML datiem, izmantojot XPath funkcionalitāti.

Komentāri:

- Lai šī opcija darbotos, Zabbix serverim (vai Zabbix starpniekserverim) jābūt kompilētam ar libxml atbalstu;
- Vārdtelpas netiek atbalstītas;

- Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

Piemēri:

```
number(/document/item/value) #izgūs '10' no <document><item><value>10</value></item></document>
number(/document/item/@attribute) #izgūs '10' no <document><item attribute="10"></item></document>
/document/item #izgūs '<item><value>10</value></item>' no <document><item><value>10</value></item></docume
```

JSON Path

Izgūstiet vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti.

Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

CSV uz JSON

Pārveido CSV faila datus JSON formātā.

Papildinformāciju skatiet: CSV uz JSON priekšapstrāde.

XML uz JSON

Pārveido datus XML formātā uz JSON.

Papildinformāciju skatiet: Serializācijas noteikumi.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

SNMP walk vērtība

Izgūt vērtību pēc norādītā OID/MIB nosaukuma un piemērot formatēšanas opcijas:

- **Bez izmaiņām** - atgriezt heksadecimālo virkni kā neekranētu heksadecimālo virkni (*ņemiet vērā*, ka attēlošanas norādes joprojām tiek piemērotas);
- **UTF-8 no hex-STRING** - pārveidot heksadecimālo virkni par UTF-8 virkni;
- **MAC no hex-STRING** - validēt heksadecimālo virkni kā MAC adresi un atgriezt korektu MAC adreses virkni (kur ' ' tiek aizstātas ar ':');
- **Vesels skaitlis no BITS** - pārveidot pirmos 8 baitus bitu virknei, kas izteikta kā heksadecimālo rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu bezparaksta veselu skaitli. Bitu virknēs, kas ir garākas par 8 baitiem, nākamie baiti tiks ignorēti.

Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

SNMP walk uz JSON

Pārveido SNMP vērtības uz JSON.

Norādiet lauka nosaukumu JSON formātā un atbilstošo SNMP OID ceļu. Lauku vērtības tiks aizpildītas ar vērtībām no norādītā SNMP OID ceļa.

Komentāri:

- Ir pieejamas līdzīgas vērtību formatēšanas opcijas kā solī *SNMP walk value*;
- Jūs varat izmantot šo priekšapstrādes soli SNMP OID atklāšanai;
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

SNMP iegūtā vērtība

Pielietojiet formatēšanas opcijas SNMP iegūtajai vērtībai:

- **UTF-8 no Hex-STRING** - pārveidot hex-virkni par UTF-8 virkni;
- **MAC no Hex-STRING** - validēt hex-virkni kā MAC adresi un atgriezt korektu MAC adreses virkni (kur ' ' tiek aizstātas ar ':');
- **Vesels skaitlis no BITS** - pārveidot pirmos 8 baitus bitu virknei, kas izteikta kā heksadecimālu rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu bezparaksta veselu skaitli. Bitu virknēs, kas ir garākas par 8 baitiem, turpmākie baiti tiks ignorēti.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

Pielāgots reizinātājs

Reiziniet vērtību ar norādīto veselo skaitli vai peldošā komata skaitli.

Komentāri:

- Izmantojiet šo opciju, lai pārvērstu saņemtās vērtības KB, MBps u.c. vienībās uz B, Bps. Pretējā gadījumā Zabbix nevarēs pareizi iestatīt prefiksus (K, M, G u.c.).
- Ņemiet vērā, ka, ja vienuma informācijas tips ir *Numeric (unsigned)*, ienākošajām vērtībām ar daļskaitļa daļu tā tiks noņemta (t.i., '0.9' kļūs par '0') pirms pielāgotā reizinātāja piemērošanas;
- Ja izmantojat pielāgotu reizinātāju vai saglabājat vērtību kā *Change per second* vienumiem, kuriem informācijas tips ir iestatīts uz *Numeric (unsigned)*, un iegūtā aprēķinātā vērtība faktiski ir peldošā komata skaitlis, aprēķinātā vērtība joprojām tiek pieņemta kā korekta, noņemot decimāldaļu un saglabājot vērtību kā veselu skaitli;
- Tiek atbalstīts: zinātniskais pieraksts, piemēram, 1e+70; lietotāja makrosi un LLD makrosi; virknes, kas ietver makrosus, piemēram, `{#MACRO}e+10`, `{$MACRO1}e+{$MACRO2}`. Makrosiem ir jāatrisinās par veselu skaitli vai peldošā komata skaitli.
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Vienkārša izmaiņa

Aprēķina starpību starp pašreizējo un iepriekšējo vērtību.

Komentāri:

- Šis solis var būt noderīgs, lai mērītu pastāvīgi augošu vērtību;
- Tiek aprēķināts kā **value**-**prev_value**, kur *value* — pašreizējā vērtība; *prev_value* — iepriekš saņemtā vērtība;
- Katram vienumam ir atļauta tikai viena izmaiņas operācija ("Vienkārša izmaiņa" vai "Izmaiņa sekundē");
- Ja pašreizējā vērtība ir mazāka par iepriekšējo vērtību, Zabbix šo starpību atmet (neko nesaglabā) un gaida citu vērtību;
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Izmaiņa sekundē

Aprēķina vērtības izmaiņas (starpību starp pašreizējo un iepriekšējo vērtību) ātrumu sekundē.

Komentāri:

- Šis solis ir noderīgs, lai aprēķinātu pastāvīgi augošas vērtības pieauguma ātrumu sekundē;
- Tā kā šis aprēķins var radīt peldošā komata skaitļus, ieteicams iestatīt 'Type of information' uz *Numeric (float)*, pat ja ienākošās neapstrādātās vērtības ir veseli skaitļi. Tas ir īpaši svarīgi maziem skaitļiem, kur nozīme ir decimāldaļai. Ja peldošā komata vērtības ir lielas un var pārsniegt 'float' lauka garumu, kā rezultātā visa vērtība var tikt zaudēta, faktiski ieteicams izmantot *Numeric (unsigned)* un tādējādi nogriezt tikai decimāldaļu.
- Tiek novērtēts kā (**value**-**prev_value**)/(**time**-**prev_time**), kur *value* — pašreizējā vērtība; *prev_value* — iepriekš saņemtā vērtība; *time* — pašreizējā laika zīmoga vērtība; *prev_time* — iepriekšējās vērtības laika zīmogs;
- Katram vienums ir atļauta tikai viena izmaiņu operācija ("Simple change" vai "Change per second");
- Ja pašreizējā vērtība ir mazāka par iepriekšējo vērtību, Zabbix atmet šo starpību (neko nesaglabā) un gaida nākamo vērtību. Tas palīdz korekti strādāt, piemēram, ar 32 bitu SNMP skaitītāju pārgriešanos (pārpildi).
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Būla vērtības pārveidošana decimālskaitlī

Pārveido vērtību no būla formāta uz decimālskaitli.

Komentāri:

- Tekstuālais attēlojums tiek pārveidots par 0 vai 1. Tādējādi 'TRUE' tiek saglabāts kā 1 un 'FALSE' tiek saglabāts kā 0. Visas vērtības tiek salīdzinātas, neņemot vērā burtu reģistru. Pašlaik atpazītās vērtības ir šādas: *TRUE* — true, t, yes, y, on, up, running, enabled, available, ok, master; *FALSE* — false, f, no, n, off, down, unused, disabled, unavailable, err, slave. Papildus tam jebkura skaitliska vērtība, kas nav nulle, tiek uzskatīta par TRUE, bet nulle tiek uzskatīta par FALSE.
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Oktālais uz decimālo

Pārveido vērtību no oktālā formāta uz decimālo.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

No heksadecimālā uz decimālo

Pārveido vērtību no heksadecimālā formāta uz decimālo.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Ja priekšapstrādes solis neizdodas, vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

JavaScript

Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam.

> **Attention:**
> Neizmantojiet neizsludinātas piešķires priekšapstrādes JavaScript kodā. Izmantojiet `var`, lai deklarētu lokālos mainīgos.

Komentāri:

- Pieejamais JavaScript garums ir atkarīgs no izmantotās datubāzes;
- Papildinformāciju skatiet: Javascript priekšapstrāde.

Diapazonā

Definējiet diapazonu, kurā vērtībai jāatrodas, norādot minimālo/maksimālo vērtību (ieskaitot robežvērtības).

Komentāri:

- Tiek pieņemtas skaitliskas vērtības (ieskaitot jebkādu ciparu skaitu, neobligātu decimāldaļu un neobligātu eksponentdaļu, kā arī negatīvas vērtības);
- Minimālajai vērtībai jābūt mazākai par maksimālo;
- Jābūt norādītai vismaz vienai vērtībai;
- Var izmantot lietotāja makrosus un zema līmeņa atklāšanas makrosus;
- Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

Ja priekšapstrādes solis neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Atbilst regulārajai izteiksmei

Norādiet regulāro izteiksmi, kurai vērtībai ir jāatbilst.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Neatbilst regulārajai izteiksmei

Norādiet regulāro izteiksmi, kurai vērtība nedrīkst atbilst.

Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Neveiksmīga priekšapstrādes soļa gadījumā vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

Kļūdas pārbaude JSON formātā

Pārbauda lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath. Pārtrauc apstrādi, ja darbība ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpina apstrādi ar vērtību, kāda bija pirms šīs priekšapstrādes darbības.

Komentāri:

- Šīs ārējo pakalpojumu kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību;
- Kļūda netiks ziņota, ja neizdosies parsēt nederīgu JSON;
- Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādīto vērtību vai iestatīt norādīto kļūdas ziņojumu. Neveiksmīgas priekšapstrādes darbības gadījumā vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādīto vērtību.

Kļūdas pārbaude XML failā

Pārbauda lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath. Pārtrauc apstrādi, ja darbība ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpina apstrādi ar vērtību, kāda bija pirms šīs priekšapstrādes darbības.

Komentāri:

- Šīs ārējo pakalpojumu kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību;
- Kļūda netiks ziņota, ja neizdosies parsēt nederīgu XML;
- Ja atzīmējat izvēles rūtiņu *Pielāgots kļūmes gadījumā*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes darbība neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Kļūdas pārbaude, izmantojot regulāro izteiksmi

Pārbaudiet lietojumprogrammas līmeņa kļūdas ziņojumu, izmantojot regulāro izteiksmi. Pārtrauciet apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpiniet apstrādi ar vērtību, kāda bija pirms šīs priekšapstrādes darbības.

Parametri:

- **pattern** - regulārā izteiksme;
- **output** - izvades formatēšanas veidne. Aizvietošanas secība \N (kur N=1...9) tiek aizstāta ar N-to atrasto grupu. Aizvietošanas secība \0 tiek aizstāta ar atrasto tekstu.

Komentāri:

- Šīs ārējo pakalpojumu kļūdas lietotājam tiek parādītas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes darbību;
- Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Neveiksmīgas priekšapstrādes darbības gadījumā vienums nekļūs par neatbalstītu, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.

Nepārbaudītas vērtības pārbaude

Pārbaudiet, vai nevarēja iegūt nevienu vienuma vērtību. Norādiet, kā kļūme jāapstrādā, pamatojoties uz atgrieztā kļūdas ziņojuma pārbaudi.

Parametri:

- **scope** - atlasiet kļūdu apstrādes tvērumu:
jebkura kļūda - jebkura kļūda;
kļūda atbilst - tikai kļūda, kas atbilst regulārajai izteiksmei, kas norādīta laukā *pattern*;
kļūda neatbilst - tikai kļūda, kas neatbilst regulārajai izteiksmei, kas norādīta laukā *pattern*
- **pattern** - regulārā izteiksme, ar kuru salīdzināt kļūdu. Ja parametrā scope ir atlasīts *jebkura kļūda*, šis lauks netiek parādīts. Ja tas tiek parādīts, šis lauks ir obligāts.

Komentāri:

- Parasti vērtības neesamība/neizdošanās to iegūt noved pie tā, ka vienums kļūst neatbalstīts. Šī priekšapstrādes darbība ļauj mainīt šo uzvedību. Ja atzīmējat izvēles rūtiņu *Custom on fail* (šai priekšapstrādes darbībai tā vienmēr ir atzīmēta un pelēkota), ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu. Ja priekšapstrādes darbība neizdodas, vienums nekļūs neatbalstīts, ja ir atlasīta opcija atmest vērtību vai iestatīt norādītu vērtību.
- Šī priekšapstrādes darbība pārbauda tikai to, vai nevarēja iegūt nevienu vienuma vērtību. Tā nepārbauda, piemēram, vai iegūtās vērtības tips (piemēram, virkne) atbilst vienuma informācijas tipam (piemēram, skaitlisks); sīkāku informāciju skatiet sadaļā Priekšapstrādes piemēri. Ja ir tipu neatbilstība, vienums joprojām var kļūt neatbalstīts pēc visu priekšapstrādes darbību izpildes. Lai pārbaudītu tipu neatbilstību, varat izmantot, piemēram, priekšapstrādes darbību *Custom multiplier*; skatiet Priekšapstrādes piemēri.
- Regulāro izteiksmju grupu uztveršana tiek atbalstīta laukos *Set value to* vai *Set error to*. Izmantojiet \N (kur N=1...9), lai iegūtu N-to atbilstošo grupu; izmantojiet \0, lai iegūtu atbilstošo tekstu;
- Šīs darbības vienmēr tiek izpildītas kā pirmās priekšapstrādes darbības un pēc izmaiņu saglabāšanas vienumā tiek novietotas virs visām pārējām;
- Tiek atbalstītas vairākas darbības *Nepārbaudītas vērtības pārbaude* norādītajā secībā. Darbība *jebkura kļūda* tiks automātiski novietota kā pēdējā darbība šajā grupā.

Atmest nemainīto

Atmest vērtību, ja tā nav mainījusies.

Komentāri:

- Ja vērtība tiek atmesta, tā netiek saglabāta datubāzē, un Zabbix serveris nezina, ka šī vērtība tika saņemta. Trigeru izteiksmes netiks novērtētas, un rezultātā saistītajiem trigeriem netiks izveidotas/atrisinātas nekādas problēmas. Funkcijas darbosies, tikai pamatojoties uz datiem, kas faktiski ir saglabāti datubāzē. Tā kā tendences tiek veidotas, pamatojoties uz datiem datubāzē, ja stundas laikā netiek saglabāta neviena vērtība, tad arī par šo stundu nebūs tendenču datu.
- Katram vienumam var norādīt tikai vienu ierobežošanas opciju.

Atmest nemainīgu ar sirdspukstu

Atmest vērtību, ja tā nav mainījusies noteiktajā laika periodā (sekundēs).

Komentāri:

- Tiek atbalstītas pozitīvas veselu skaitļu vērtības, lai norādītu sekundes (minimums — 1 sekunde);
- Var izmantot laika sufiksus (piemēram, 30s, 1m, 2h, 1d);
- Var izmantot lietotāja makrosus un zema līmeņa atklāšanas makrosus;
- Ja vērtība tiek atmesta, tā netiek saglabāta datubāzē, un Zabbix serveris nezina, ka šī vērtība tika saņemta. Netiks novērtētas nekādas trigeru izteiksmes, un rezultātā saistītajiem trigeriem netiks izveidotas/atrisinātas nekādas problēmas. Funkcijas darbosies tikai, pamatojoties uz datiem, kas faktiski ir saglabāti datubāzē. Tā kā tendences tiek veidotas, pamatojoties uz datiem datubāzē, ja stundas laikā netiek saglabāta neviena vērtība, tad arī par šo stundu nebūs tendenču datu.
- Katram vienumam var norādīt tikai vienu droselēšanas opciju.

Prometheus raksts

Izmantojiet šo vaicājumu, lai no Prometheus metrikām iegūtu nepieciešamos datus.

Plašāku informāciju skatiet sadaļā Prometheus pārbaudes.

Prometheus uz JSON

Pārveidojiet nepieciešamos Prometheus metriku datus JSON formātā.

Plašāku informāciju skatiet sadaļā Prometheus checks.

Makrosu atbalsts

Tietotāja makrosi un lietotāja makrosi ar kontekstu tiek atbalstīti šeit:

- priekšapstrādes soļa parametros, tostarp JavaScript kodā;
- pielāgotās kļūdu apstrādes parametros (*Set value to* un *Set error to* laukos).

Note:
Makrosa konteksts tiek ignorēts, kad makross tiek aizstāts ar tā vērtību. Makrosa vērtība tiek ievietota kodā tāda, kāda tā ir; pirms vērtības ievietošanas JavaScript kodā nav iespējams pievienot papildu atsoļošanu. Lūdzu, ņemiet vērā, ka dažos gadījumos tas var izraisīt JavaScript kļūdas.

Testēšana

Skatiet priekšapstrādes testēšanu.

1 Priekšapstrādes testēšana

Testēšana

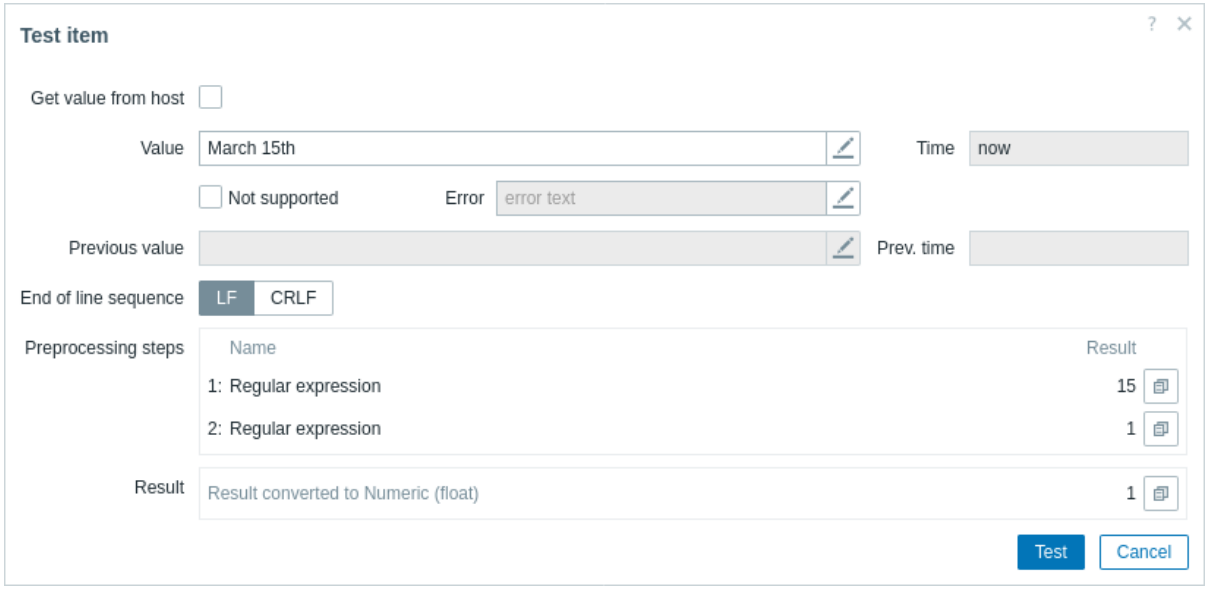
Priekšapstrādes soļu testēšana ir noderīga, lai pārliecinātos, ka sarežģītas priekšapstrādes plūsmas dod no tām sagaidāmos rezultātus, negaidot, līdz tiks saņemta un priekšapstrādāta vienuma vērtība.

Ir iespējams testēt:

- ar hipotētisku vērtību
- ar reālu vērtību no hosta

Katru priekšapstrādes soli var testēt atsevišķi, kā arī visus soļus var testēt kopā. Noklikšķinot attiecīgi uz pogas *Test* vai *Test all steps* darbību blokā, tiek atvērts testēšanas logs.

Hipotētiskas vērtības testēšana



Parametrs	Apraksts
Iegūt vērtību no hosta	Ja vēlaties testēt hipotētisku vērtību, atstājiet šo izvēles rūtiņu neatzīmētu. Skatiet arī: Reālas vērtības testēšana.
Vērtība	Ievadiet testējamo ievades vērtību. Noklikšķinot parametra laukā vai uz skatīšanas/rediģēšanas pogas, tiks atvērts teksta apgabala logs vērtības vai koda bloka ievadīšanai.
Nav atbalstīts	Atzīmējiet šo izvēles rūtiņu, lai testētu neatbalstītu vērtību. Šī opcija ir noderīga, lai testētu priekšapstrādes soli *Pārbaudīt neatbalstītu vērtību*.
Kļūda	Ievadiet kļūdas tekstu. Šis lauks ir iespējots, ja *Iegūt vērtību no hosta* nav atzīmēts, bet *Nav atbalstīts* ir atzīmēts. Ja *Iegūt vērtību no hosta* ir atzīmēts, šis lauks tiek aizpildīts ar faktisko kļūdas ziņojumu (tikai lasāms) no hosta.
Laiks	Tiek parādīts ievades vērtības laiks: `now` (tikai lasāms).
Iepriekšējā vērtība	Ievadiet iepriekšējo ievades vērtību salīdzināšanai. Tikai priekšapstrādes soļiem *Izmaiņa* un *Droselēšana*.
Iepriekšējais laiks	Ievadiet iepriekšējās ievades vērtības laiku salīdzināšanai. Tikai priekšapstrādes soļiem *Izmaiņa* un *Droselēšana*. Noklusējuma vērtība ir balstīta uz vienuma lauka 'Update interval' vērtību (ja '1m', tad šis lauks tiek aizpildīts ar `now-1m`). Ja nekas nav norādīts vai lietotājam nav piekļuves hostam, noklusējuma vērtība ir `now-30s`.
Makrosi	Ja tiek izmantoti makrosi, tie tiek uzskaitīti kopā ar to vērtībām. Vērtības var rediģēt testēšanas nolūkos, taču izmaiņas tiks saglabātas tikai testēšanas kontekstā.
Rindas beigu secība	Atlasiet rindas beigu secību vairākrindu ievades vērtībām: **LF** - LF (rindas pārneses) secība **CRLF** - CRLF (karietes atgriešanas-rindas pārneses) secība.
Priekšapstrādes soļi	Tiek uzskaitīti priekšapstrādes soļi; pēc pogas *Testēt* nospiešanas katram solim tiek parādīts testēšanas rezultāts. Testēšanas rezultāti tiek saīsināti līdz maksimālajam izmēram 512KB, kad tie tiek nosūtīti uz lietotāja saskarne. Testēšanas rezultātus var kopēt (ne vairāk kā saīsinātos 512KB). Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, novietojot peles kursoru virs ikonas. Ņemiet vērā, ka dati, kas ir lielāki par 512KB, joprojām tiek pilnībā apstrādāti Zabbix serveris. Ja solis testēšanas laikā neizdevās, tiek parādīta kļūdas ikona. Kļūdas apraksts tiek parādīts, novietojot peles kursoru virs ikonas. Ja solim ir norādīts "Custom on fail" un šī darbība tiek izpildīta, uzreiz pēc priekšapstrādes testēšanas soļa rindas parādās jauna rinda, kurā redzams, kāda darbība tika veikta un kāds bija tās rezultāts (kļūda vai vērtība).

Parametrs	Apraksts
Rezultāts	Priekšapstrādes soļu testēšanas gala rezultāts tiek parādīts visos gadījumos, kad visi soļi tiek testēti kopā (noklikšķinot uz pogas *Testēt visus soļus*). Tiek parādīts arī pārveidošanas veids uz vienuma vērtības tipu, piemēram, `Result converted to Numeric (unsigned)`. Testēšanas rezultāti tiek saīsināti līdz maksimālajam izmēram 512KB, kad tie tiek nosūtīti uz lietotāja saskarne. Gala rezultātu var kopēt (ne vairāk kā saīsinātos 512KB). Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, novietojot peles kursoru virs ikonas. Ņemiet vērā, ka dati, kas ir lielāki par 512KB, joprojām tiek pilnībā apstrādāti Zabbix serveris.

Noklikšķiniet uz *Testēt*, lai redzētu rezultātu pēc katra priekšapstrādes soļa.

Testa vērtības tiek saglabātas starp testēšanas sesijām gan atsevišķiem soļiem, gan visiem soļiem, ļaujot lietotājam mainīt priekšapstrādes soļus vai vienuma konfigurāciju un pēc tam atgriezties testēšanas logā, lai nebūtu atkārtoti jāievada informācija. Tomēr pēc lapas atsvaidzināšanas vērtības tiek zaudētas.

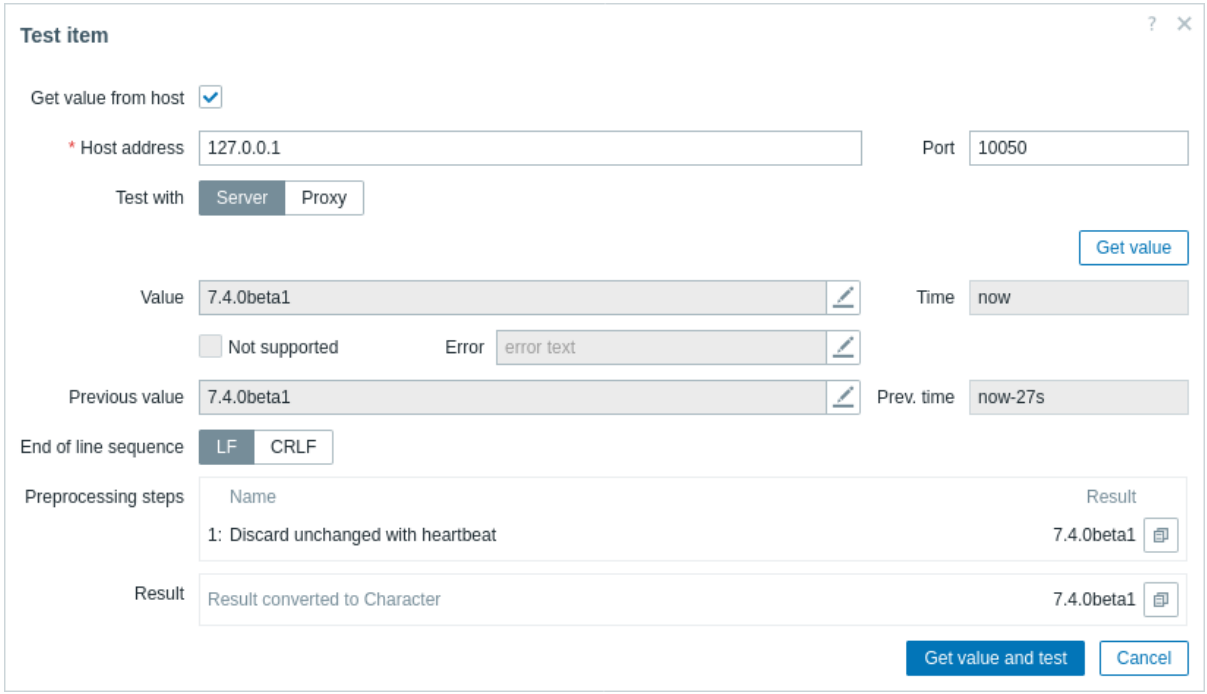
Testēšanu veic Zabbix serveris. Lietotāja saskarne nosūta atbilstošu pieprasījumu serveris un gaida rezultātu. Pieprasījumā ir ietverta ievades vērtība un priekšapstrādes soļi (ar izvērstiem lietotāja makrosiem). Soļiem *Izmaiņa* un *Droselēšana* var norādīt arī neobligātu iepriekšējo vērtību un laiku. Serveris atbild ar rezultātiem katram priekšapstrādes solim.

Visas tehniskās kļūdas vai ievades validācijas kļūdas tiek parādītas kļūdu lodziņā testēšanas loga augšdaļā.

Reālas vērtības testēšana

Lai testētu priekšapstrādi ar reālu vērtību:

- Atzīmējiet izvēles rūtiņu *Get value from host*
- Ievadiet vai pārbaudiet hosta parametrus (hosta adrese, ports, starpniekservera nosaukums/nav starpniekservera) un vienumam specifisko informāciju (piemēram, SNMPv2 kopienu vai SNMPv3 drošības akreditācijas datus). Šie lauki ir atkarīgi no konteksta:
  - Vērtības tiek aizpildītas automātiski, kad tas ir iespējams, t.i., vienumiem, kuriem nepieciešams aģents, izmantojot informāciju no hosta atlasītās aģenta saskarnes
  - Veidņu vienumiem vērtības jāaizpilda manuāli
  - Vienkārša teksta makro vērtības tiek atrisinātas
  - Ja lauka vērtība (vai daļa no vērtības) ir slepens vai Vault makro, lauks būs tukšs un tas būs jāaizpilda manuāli. Ja kāds vienuma parametrs satur slepena makro vērtību, tiek parādīts šāds brīdinājuma ziņojums: ”Vienums satur lietotāja definētus makro ar slepenām vērtībām. Šo makro vērtības jāievada manuāli.”
  - Lauki tiek atspējoti, ja tie nav nepieciešami attiecīgā vienuma tipa kontekstā (piemēram, hosta adreses un starpniekservera lauki ir atspējoti aprēķinātajiem vienumiem)
- Noklikšķiniet uz *Get value and test*, lai testētu priekšapstrādi



Ja vienuma konfigurācijas formā esat norādījis vērtību kartējumu (lauks ’Show value’), vienuma testēšanas dialoglodziņā pēc gala rezultāta tiks parādīta vēl viena rinda ar nosaukumu ’Result with value map applied’.

Parametri, kas ir specifiski reālas vērtības iegūšanai no hosta:

Parameter	Description
Get value from host	Atzīmējiet šo izvēles rūtiņu, lai iegūtu reālu vērtību no hosta.
Host address	Ievadiet hosta adresi. Šis lauks tiek automātiski aizpildīts ar vienuma hosta saskarnes adresi.
Port	Ievadiet hosta portu. Šis lauks tiek automātiski aizpildīts ar vienuma hosta saskarnes portu.
Additional fields for SNMP interfaces (SNMP version, SNMP community, Context name, etc.)	Papildinformāciju par SNMP saskarnes konfigurēšanu (v1, v2 un v3) skatiet sadaļā Configuring SNMP monitoring. Šie lauki tiek automātiski aizpildīti no vienuma hosta saskarnes.
Proxy	Norādiet starpniekserveri, ja hosts tiek uzraudzīts ar starpniekservera palīdzību. Šis lauks tiek automātiski aizpildīts ar hosta starpniekserveri (ja tāds ir).
Value	No hosta iegūtā vērtība. Noklikšķinot parametra laukā vai uz skatīšanas/rediģēšanas pogas, tiks atvērts vērtības vai koda bloka teksta apgabala logs. Vērtības tiek saīsinātas līdz maksimālajam izmēram 512KB un tikai lietotāja saskarnē. Ja rezultāts ir saīsināts, tiek parādīta brīdinājuma ikona. Brīdinājuma apraksts tiek parādīts, novietojot peles kursoru virs tās. Ņemiet vērā, ka dati, kas ir lielāki par 512KB, joprojām tiek pilnībā apstrādāti Zabbix serverī.

Par pārējiem parametriem skatiet iepriekš sadaļu Testing hypothetical value.

2 Priekšapstrādes detaļas

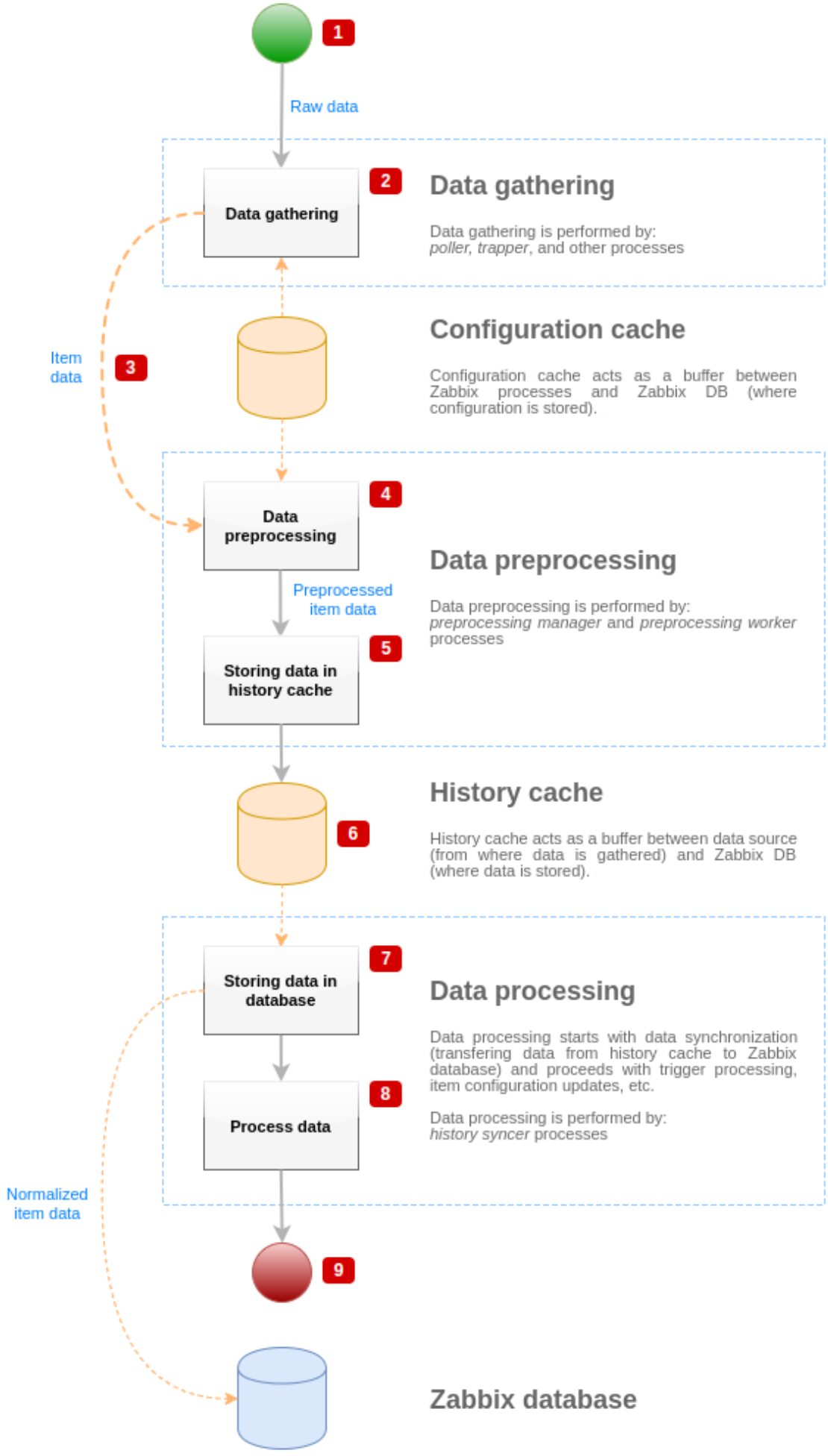
Pārskats

Šajā sadaļā ir sniegta informācija par vienumu vērtību priekšapstrādi. Vienumu vērtību priekšapstrāde ļauj definēt un izpildīt saņemtajām vienumu vērtībām transformācijas noteikumus.

Priekšapstrādi pārvalda priekšapstrādes pārvaldnieka process kopā ar priekšapstrādes darbiniekiem, kas izpilda priekšapstrādes soļus. Visas vērtības ar priekšapstrādi, kas saņemtas no dažādiem datu ievācējiem, pirms to pievienošanas vēstures kešatmiņai iziet caur priekšapstrādes pārvaldnieku. Saziņai, kas balstīta uz ligzdām, tiek izmantota IPC starp datu ievācējiem (aptaujātājiem, uztvērējiem u.c.) un priekšapstrādes procesu. Priekšapstrādes soļus izpilda vai nu Zabbix serveris, vai Zabbix starpniekserveris (vienumiem, kurus uzrauga starpniekserveris).

Vienuma vērtības apstrāde

Lai vizualizētu datu plūsmu no datu avota līdz Zabbix datubāzei, varam izmantot šādu vienkāršotu diagrammu:


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

Iepriekš redzamajā diagrammā **vienkāršotā** veidā ir parādīti tikai tie procesi, objekti un darbības, kas saistīti ar vienuma vērtības apstrādi. Diagrammā nav attēlotas nosacītas virziena maiņas, kļūdu apstrāde vai cikli. Tāpat nav parādīta priekšapstrādes pārvaldnieka lokālā datu kešatmiņa, jo tā tieši neietekmē datu plūsmu. Šīs diagrammas mērķis ir parādīt procesus, kas iesaistīti vienuma vērtības apstrādē, un to savstarpējo mijiedarbību.

- Datu iegūšana sākas ar neapstrādātiem datiem no datu avota. Šajā brīdī dati satur tikai ID, laika zīmogu un vērtību (iespējamas arī vairākas vērtības).
- Neatkarīgi no tā, kāds datu ieguvēja tips tiek izmantots, pamatideja ir vienāda gan aktīvajām, gan pasīvajām pārbaudēm, trapper vienumiem u.c., jo mainās tikai datu formāts un komunikācijas iniciators (vai nu datu ieguvējs gaida savienojumu un datus, vai arī datu ieguvējs uzsāk komunikāciju un pieprasa datus). Neapstrādātie dati tiek validēti, un vienuma konfigurācija tiek iegūta no konfigurācijas kešatmiņas (dati tiek papildināti ar konfigurācijas datiem).
- Lai nodotu datus no datu ieguvējiem priekšapstrādes pārvaldniekam, tiek izmantots uz ligzdām balstīts IPC mehānisms. Šajā brīdī datu ieguvējs turpina iegūt datus, negaidot atbildi no priekšapstrādes pārvaldnieka.
- Tiek veikta datu priekšapstrāde. Tas ietver priekšapstrādes soļu izpildi un atkarīgo vienumu apstrādi.

Note:
Vienums var mainīt savu stāvokli uz NOT SUPPORTED priekšapstrādes laikā, ja kāds no priekšapstrādes soļiem neizdodas.

- Vēstures dati no priekšapstrādes pārvaldnieka lokālās datu kešatmiņas tiek pārvietoti uz vēstures kešatmiņu.
- Šajā brīdī datu plūsma apstājas līdz nākamajai vēstures kešatmiņas sinhronizācijai (kad history syncer process veic datu sinhronizāciju).
- Sinhronizācijas process sākas ar datu normalizēšanu pirms datu saglabāšanas Zabbix datubāzē. Datu normalizēšana veic pārveidošanu uz vēlamo vienuma tipu (tipu, kas definēts vienuma konfigurācijā), tostarp teksta datu saīsināšanu atbilstoši iepriekš noteiktajiem izmēriem, kas ir atļauti šiem tipiem (HISTORY_STR_VALUE_LEN virknei, HISTORY_TEXT_VALUE_LEN tekstam un HISTORY_LOG_VALUE_LEN žurnāla vērtībām). Pēc normalizēšanas pabeigšanas dati tiek nosūtīti uz Zabbix datubāzi.

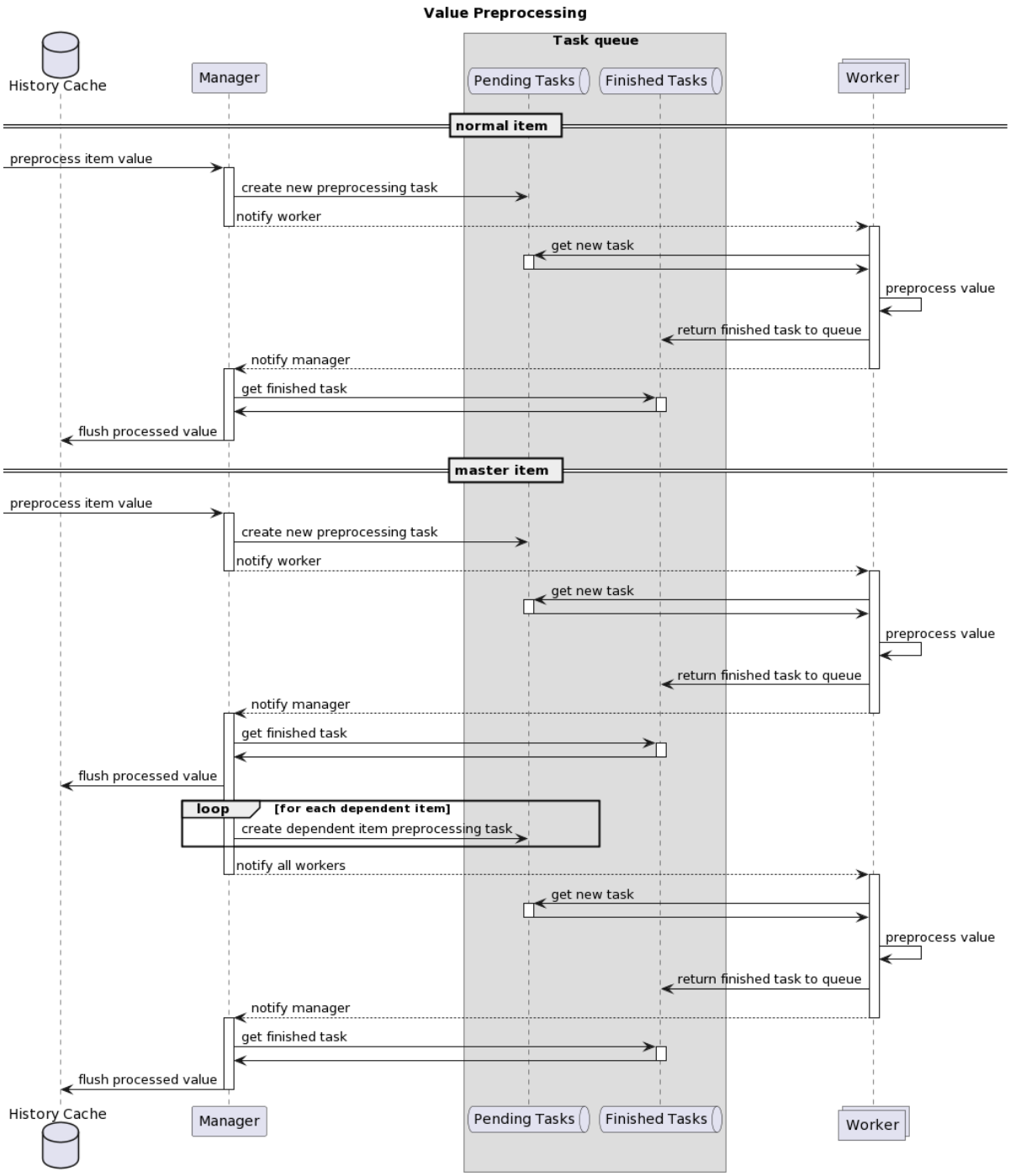
Note:
Vienums var mainīt savu stāvokli uz NOT SUPPORTED, ja datu normalizēšana neizdodas (piemēram, ja teksta vērtību nevar pārveidot par skaitli).

- Iegūtie dati tiek apstrādāti — tiek pārbaudīti trigeri, atjaunināta vienuma konfigurācija, ja vienums kļūst par NOT SUPPORTED, utt.
- No vienuma vērtības apstrādes viedokļa tas tiek uzskatīts par datu plūsmas beigu punktu.

Vienuma vērtības priekšapstrāde

Datu priekšapstrāde tiek veikta šādos soļos:

- Ja vienumam nav ne priekšapstrādes, ne atkarīgo vienumu, tā vērtība tiek vai nu pievienota vēstures kešatmiņai, vai nosūtīta LLD pārvaldniekam. Pretējā gadījumā vienuma vērtība tiek nodota priekšapstrādes pārvaldniekam, izmantojot uz UNIX ligzdām balstītu IPC mehānismu.
- Tiek izveidots priekšapstrādes uzdevums, pievienots rindai, un priekšapstrādes darbinieki tiek informēti par jauno uzdevumu.
- Šajā brīdī datu plūsma apstājas, līdz ir pieejams vismaz viens neaizņemts (t.i., neizpilda nevienu uzdevumu) priekšapstrādes darbinieks.
- Kad priekšapstrādes darbinieks ir pieejams, tas paņem nākamo uzdevumu no rindas.
- Pēc priekšapstrādes pabeigšanas (gan neveiksmīgas, gan veiksmīgas priekšapstrādes soļu izpildes gadījumā) priekšapstrādātā vērtība tiek pievienota pabeigto uzdevumu rindai, un pārvaldnieks tiek informēts par jaunu pabeigtu uzdevumu.
- Priekšapstrādes pārvaldnieks pārveido rezultātu vajadzīgajā formātā (to nosaka vienuma vērtības tips) un vai nu pievieno to vēstures kešatmiņai, vai nosūta LLD pārvaldniekam.
- Ja apstrādātajam vienumam ir atkarīgie vienumi, tad atkarīgie vienumi tiek pievienoti priekšapstrādes rindai ar priekšapstrādāto galvenā vienuma vērtību. Atkarīgie vienumi tiek ievietoti rindā, apejot parastos vērtību priekšapstrādes pieprasījumus, bet tikai galvenajiem vienumiem ar iestatītu vērtību, kas nav stāvoklī NOT SUPPORTED.



Ņemiet vērā, ka diagrammā galvenā vienuma priekšapstrāde ir nedaudz vienkāršota, izlaižot priekšapstrādes kešošanu.

Priekšapstrādes rinda

Priekšapstrādes rinda ir organizēta šādi:

- gaidošo uzdevumu saraksts:
  - uzdevumi, kas izveidoti tieši no vērtību priekšapstrādes pieprasījumiem to saņemšanas secībā
- tūlītējo uzdevumu saraksts (tiek apstrādāti pirms gaidošajiem uzdevumiem):
  - testēšanas uzdevumi (izveidoti, atbildot uz vienuma/priekšapstrādes testēšanas pieprasījumiem no lietotāja saskarnes)
  - atkarīgo vienumu uzdevumi
  - secības uzdevumi (uzdevumi, kas jāizpilda stingrā secībā):
    - ar priekšapstrādes soļiem, kuros tiek izmantota pēdējā vērtība:
      - izmaiņa
      - ierobežošana
      - JavaScript (baitkoda kešošana)
    - atkarīgo vienumu priekšapstrādes kešošana
- pabeigto uzdevumu saraksts

Priekšapstrādes kešošana

Priekšapstrādes kešošana tika ieviesta, lai uzlabotu priekšapstrādes veiktspēju vairākiem atkarīgajiem vienumiem ar līdzīgām priekšapstrādes darbībām (tas ir bieži sastopams LLD rezultāts).

Kešošana tiek veikta, priekšapstrādājot vienu atkarīgo vienumu un atkārtoti izmantojot daļu no iekšējiem priekšapstrādes datiem pārējiem atkarīgajiem vienumiem. Priekšapstrādes kešatmiņa tiek atbalstīta tikai pirmajā šādu tipu priekšapstrādes solī:

- Prometheus pattern (indeksē ievadi pēc metrikām)
- JSONPath (parsē datus objektu kokā un indeksē pirmo izteiksmi `[?(@.path == "value")]`)

Priekšapstrādes darbinieki

Zabbix servera konfigurācijas fails ļauj lietotājiem iestatīt priekšapstrādes darbinieku pavedienu skaitu. Konfigurācijas parametrs StartPreprocessors jāizmanto, lai iestatītu iepriekš palaisto priekšapstrādes darbinieku instanču skaitu, kam vismaz jāatbilst pieejamo CPU kodolu skaitam.

Ja priekšapstrādes uzdevumi nav saistīti ar CPU noslodzi un ietver biežus tīkla pieprasījumus, ieteicams konfigurēt papildu darbiniekus. Optimālo priekšapstrādes darbinieku skaitu var noteikt daudzi faktori, tostarp ”priekšapstrādājamo” vienumu skaits (vienumi, kuriem jāizpilda jebkādi priekšapstrādes soļi), datu ievākšanas procesu skaits, vidējais vienuma priekšapstrādes soļu skaits utt. Nepietiekams darbinieku skaits var izraisīt lielu atmiņas patēriņu. Lai novērstu pārmērīga atmiņas patēriņa problēmas jūsu Zabbix instalācijā, skatiet Pārmērīga atmiņas patēriņa profilēšana ar tcmalloc.

Tomēr, pieņemot, ka nav smagu priekšapstrādes darbību, piemēram, lielu XML/JSON fragmentu parsēšanas, priekšapstrādes darbinieku skaits var atbilst kopējam datu ievācēju skaitam. Tādējādi lielākoties (izņemot gadījumus, kad dati no ievācēja tiek saņemti paketēs) savāktajiem datiem būs pieejams vismaz viens neizmantots priekšapstrādes darbinieks.

Warning:
Pārāk liels datu ievākšanas procesu skaits (aptaujātāji, nesasniedzamu resursu aptaujātāji, ODBC aptaujātāji, HTTP aptaujātāji, Java aptaujātāji, pingotāji, trapperi, starpniekserveru aptaujātāji) kopā ar IPMI pārvaldnieku, SNMP trapperi un priekšapstrādes darbiniekiem var izsmelt priekšapstrādes pārvaldniekam pieejamo failu deskriptoru limita apjomu uz procesu.

 Ja tiek izsmelts failu deskriptoru limits uz procesu, Zabbix serveris apstāsies, parasti neilgi pēc palaišanas, taču dažkārt tas var notikt vēlāk. Lai izvairītos no šādām problēmām, pārskatiet Zabbix servera konfigurācijas failu, lai optimizētu vienlaicīgo pārbaužu un procesu skaitu. Papildus tam, ja nepieciešams, pārliecinieties, ka failu deskriptoru limits ir iestatīts pietiekami augsts, pārbaudot un pielāgojot sistēmas limitus.

Vērtību apstrādes konveijers

Vienuma vērtību apstrāde tiek izpildīta vairākos soļos (vai fāzēs) ar vairāku procesu palīdzību. Tas var izraisīt:

- Atkarīgais vienums var saņemt vērtības, kamēr GALVENĀ vērtība to nevar. To var panākt, izmantojot šādu lietošanas gadījumu:
  - Galvenajam vienumam ir vērtības tips `UINT` (var izmantot trapper vienumu), atkarīgajam vienumam ir vērtības tips `TEXT`.
  - Ne galvenajam, ne atkarīgajam vienumam nav nepieciešami priekšapstrādes soļi.
  - Teksta vērtība (piemēram, ”abc”) ir jānodod galvenajam vienumam.
  - Tā kā nav izpildāmu priekšapstrādes soļu, priekšapstrādes pārvaldnieks pārbauda, vai galvenais vienums nav stāvoklī NOT SUPPORTED un vai vērtība ir iestatīta (abi nosacījumi ir patiesi), un ievieto rindā atkarīgo vienumu ar tādu pašu vērtību kā galvenajam vienumam (jo nav priekšapstrādes soļu).
  - Kad gan galvenais, gan atkarīgais vienums sasniedz vēstures sinhronizācijas fāzi, galvenais vienums kļūst par NOT SUPPORTED vērtības konvertēšanas kļūdas dēļ (teksta datus nevar konvertēt par unsigned integer).

Rezultātā atkarīgais vienums saņem vērtību, kamēr galvenais vienums maina savu stāvokli uz NOT SUPPORTED.

- Atkarīgais vienums saņem vērtību, kas nav atrodama galvenā vienuma vēsturē. Lietošanas gadījums ir ļoti līdzīgs iepriekšējam, izņemot galvenā vienuma tipu. Piemēram, ja galvenajam vienumam tiek izmantots tips `CHAR`, tad galvenā vienuma vērtība vēstures sinhronizācijas fāzē tiks saīsināta, kamēr atkarīgie vienumi saņems savas vērtības no galvenā vienuma sākotnējās (nesaīsinātās) vērtības.

3 Priekšapstrādes piemēri

Pārskats

Šajā sadaļā ir sniegti piemēri, kā izmantot priekšapstrādes soļus, lai veiktu dažus praktiskus uzdevumus.

VMware notikumu žurnāla ierakstu filtrēšana

Šajā piemērā tiek izmantots priekšapstrādes solis Atbilst regulārajai izteiksmei, lai no VMware notikumu žurnāla atfiltrētu nevajadzīgos notikumus.

1. Darbojošā VMware Hypervisor hostā pārbaudiet, ka vienums vmware.eventlog ir pieejams un darbojas pareizi. Ņemiet vērā, ka notikumu žurnāla vienums hipervizorā jau var būt pieejams, ja hosta izveides laikā tika piesaistīta veidne VMware.

2. VMware Hypervisor hostā izveidojiet atkarīgo vienumu ar tipu *Log* un iestatiet notikumu žurnāla vienumu kā tā galveno vienumu.

3. Atkarīgā vienuma cilnē *Preprocessing* noklikšķiniet uz *Add*, lai izveidotu priekšapstrādes soli, un nolaižamajā sarakstā atlasiet *Matches regular expression*. Pēc tam norādiet vienu no šiem paraugiem:

- Lai filtrētu visus žurnāla notikumus:

```
.* logged in .*
```

- Lai filtrētu rindas, kurās pēc "User" ir lietotājvārdi:

```
\bUser\s+\K\S+
```

Attention:
Ja regulārā izteiksme neatbilst, atkarīgais vienums kļūst neatbalstīts un tiek parādīts atbilstošs kļūdas ziņojums. Lai no tā izvairītos, atzīmējiet izvēles rūtiņu *Custom on fail* un atlasiet opciju, piemēram, vērtības atmešanu vai pielāgotas vērtības iestatīšanu. Ņemiet vērā, ka atmestās vērtības datubāzē netiek saglabātas; tādēļ trigeri netiek novērtēti un tendenču dati netiek ģenerēti.

Alternatīvi varat izmantot priekšapstrādes soli Regular expression, lai izvilktu atbilstošās grupas un kontrolētu izvadi:

- Lai izvilktu un izvadītu visu žurnāla notikumu, kas satur "logged in", norādiet šādus parametrus:

```
Pattern: .*logged in.*
Output: \0
```

- Lai izvilktu un izvadītu lietotājvārdus pēc "User":

```
Pattern: User (.*?)(?=\ )
Output: \1
```

Iegūtās vērtības tipa pārbaude

Šajā piemērā tiek izmantots priekšapstrādes solis Pielāgots reizinātājs, lai pārbaudītu, vai iegūtā vienuma vērtības tips ir skaitlisks.

Vienuma cilnē *Priekšapstrāde* atlasiet priekšapstrādes soli *Pielāgots reizinātājs* un norādiet šādu parametru (reizina iegūto vērtību ar 1):

```
1
```

Attention:
Ja priekšapstrāde neizdodas (piemēram, ievade nav skaitliska), tad vienums kļūst neatbalstīts ar atbilstošu kļūdas ziņojumu. Lai no tā izvairītos, atzīmējiet izvēles rūtiņu *Pielāgots kļūmes gadījumā* un atlasiet opciju, piemēram, vērtības atmešanu vai pielāgotas vērtības iestatīšanu. Lūdzu, ņemiet vērā, ka atmestās vērtības netiek saglabātas datubāzē; tādēļ trigeri netiek novērtēti un tendenču dati netiek ģenerēti.

Neatbalstītas vērtības pārbaude

Šajā piemērā tiek izmantots priekšapstrādes solis Check for not supported value, lai pārbaudītu, vai vienuma vērtību neizdevās iegūt.

Kad Zabbix servera/starpniekservera aptaujātāja process mēģina savākt vienuma vērtību, tas var:

- Atgriezt derīgu rezultātu.
- Atgriezt rezultātu, kas sākotnēji šķiet derīgs, bet vēlāk var kļūt neatbalstīts (piemēram, vērtības tipa neatbilstības dēļ pēc priekšapstrādes).
- Atgriezt vērtības iegūšanas kļūdu, kā rezultātā vienums kļūst neatbalstīts. Biežākie iemesli:
  - Nezināma vienuma atslēga (Zabbix aģentam, vienkāršajai pārbaudei vai Zabbix iekšējiem vienumiem)
  - Nezināms OID (SNMP aģents), nezināms sensors (IPMI aģents) vai nav JMX metrikas (JMX aģents)
  - Nevar nolasīt slazdu failu (SNMP trap)
  - Skripts nav atrasts (ārējā pārbaude)
  - Šāds URL nepastāv (HTTP aģents, pārlūks)
  - Pieteikšanās neizdevās (SSH aģents, TELNET aģents)
  - Nederīga formulas sintakse (Calculated), JavaScript sintakses kļūda (Script) vai nederīgs SQL (Database monitor)

Lai noteiktu un apstrādātu vienumu vērtību iegūšanas kļūdas, varat izmantot priekšapstrādes soli *Check for not supported value*. Ņemiet vērā, ka šis solis vienmēr tiek izpildīts pirmais un nosaka tikai tās kļūdas, kas rodas pirms priekšapstrādes sākuma.

Vienuma cilnē *Preprocessing* atlasiet priekšapstrādes soli *Check for not supported value* un norādiet vienu no šiem parametriem:

- Jebkurām kļūdām:

```
Parameter: any error
```

- Kļūdām, kas satur ”cannot connect”:

```
Parameter: error matches
Pattern: (?i)cannot connect
```

Pēc tam izmantojiet opciju *Custom on fail*, lai atmestu vērtību (šajā gadījumā — kļūdu), iestatītu pielāgotu vērtību vai atgrieztu pielāgotu kļūdas ziņojumu. Lūdzu, ņemiet vērā, ka discarded vērtības netiek saglabātas datubāzē; tādēļ trigeri netiek novērtēti un tendenču dati netiek ģenerēti.

4 JSONPath funkcionalitāte

Pārskats

Šajā sadaļā ir aprakstīta atbalstītā JSONPath funkcionalitāte vienuma vērtības priekšapstrādes soļos.

JSONPath sastāv no segmentiem, kas atdalīti ar punktiem. Segments var būt vienkāršs vārds, kas apzīmē JSON vērtības nosaukumu, aizstājējzīme (`*`) vai sarežģītāka konstrukcija kvadrātiekavās. Punkts pirms segmenta kvadrātiekavās nav obligāts, un to var izlaist.

JSONPath example	Apraksts
`$.object.name`	Atgriež `object.name` saturu.
`$.object['name']`	Atgriež `object.name` saturu.
`$.object.['name']`	Atgriež `object.name` saturu.
`$["object"]['name']`	Atgriež `object.name` saturu.
`$.['object'].["name"]`	Atgriež `object.name` saturu.
`$.object.history.length()`	Atgriež `object.history` masīva elementu skaitu.
`$[?(@.name == 'Object')].price.first()`	Atgriež rekvizīta `price` vērtību no pirmā objekta ar nosaukumu ”Object”.
`$[?(@.name == 'Object')].history.first().length()`	Atgriež vēstures masīva elementu skaitu no pirmā objekta ar nosaukumu ”Object”.
`$[?(@.price > 10)].length()`	Atgriež objektu skaitu, kuru cena ir lielāka par 10.

Skatiet arī: Speciālo rakstzīmju atsoļošana no LLD makro vērtībām JSONPath izteiksmēs.

Atbalstītie segmenti

Segments	Apraksts
`<name>`	Atbilst objekta īpašībai pēc nosaukuma.
`*`	Atbilst visām objekta īpašībām.
`['<name>']`	Atbilst objekta īpašībai pēc nosaukuma.
`['<name>', '<name>', ...]`	Atbilst objekta īpašībai pēc jebkura no uzskaitītajiem nosaukumiem.
`[<index>]`	Atbilst masīva elementam pēc indeksa.
`[<number>, <number>, ...]`	Atbilst masīva elementam pēc jebkura no uzskaitītajiem indeksiem.
`[*]`	Atbilst visām objekta īpašībām vai masīva elementiem.
`[<start>:<end>]`	Atbilst masīva elementiem pēc norādītā diapazona: **<start>** - pirmais indekss, kam jāatbilst (ieskaitot); ja nav norādīts, atbilst visiem masīva elementiem no sākuma; ja ir negatīvs, norāda sākuma nobīdi no masīva beigām; **<end>** - pēdējais indekss, kam jāatbilst (neieskaitot); ja nav norādīts, atbilst visiem masīva elementiem līdz beigām; ja ir negatīvs, norāda sākuma nobīdi no masīva beigām.
`[?(<expression>)]`	Atbilst objektiem/masīva elementiem, piemērojot filtra izteiksmi.

Lai atrastu atbilstošu segmentu, ignorējot tā izcelsmi (atdalīts segments), tam jāpievieno prefikss ar diviem punktiem (`..`). Piemēram, `$..name` vai `$..['name']` atgriež visu `name` īpašību vērtības.

Atbilstošo elementu nosaukumus var iegūt, JSONPath pievienojot sufiksu ar tildi (`~`). Tas atgriež atbilstošā objekta nosaukumu vai atbilstošā masīva vienuma indeksu virknes formātā. Izvades formāts ievēro tos pašus noteikumus kā citiem JSONPath vaicājumiem

— noteikta ceļa rezultāti tiek atgriezti "kā ir", bet nenoteikta ceļa rezultāti tiek atgriezti masīvā. Tomēr elementa nosaukuma iegūšanai, kas atbilst noteiktam ceļam, ir maza praktiska vērtība, jo tas jau ir zināms.

Filtra izteiksme

Filtra izteiksme ir aritmētiska izteiksme infiksa pierakstā.

Atbalstītie operandi:

Operands	Apraksts
"<text>" '<text>'	Teksta konstante. Piemērs: 'value: \\'1\\'' "value: '1'"
<number>	Skaitliska konstante ar zinātniskā pieraksta atbalstu. Piemērs: 123
<jsonpath starting with $>	Vērtība, uz kuru norāda JSONPath no ievades dokumenta saknes mezgla; tiek atbalstīti tikai noteiktie ceļi. Piemērs: $.object.name
<jsonpath starting with @>	Vērtība, uz kuru norāda JSONPath no pašreizējā objekta/elementa; tiek atbalstīti tikai noteiktie ceļi. Piemērs: @.name

Atbalstītie operatori:

Operators	Tips	Apraksts	Rezultāts
-	Binārs	Atņemšana	Skaitlis
+	Binārs	Saskaitīšana	Skaitlis
/	Binārs	Dalīšana	Skaitlis
*	Binārs	Reizināšana	Skaitlis
==	Binārs	Vienādība	Būla vērtība (1/0)
!=	Binārs	Nevienādība	Būla vērtība (1/0)
	Binārs	Mazāks par	Būla vērtība (1/0)
<=	Binārs	Mazāks par vai vienāds ar	Būla vērtība (1/0)
>	Binārs	Lielāks par	Būla vērtība (1/0)
>=	Binārs	Lielāks par vai vienāds ar	Būla vērtība (1/0)
=~	Binārs	Atbilst regulārajai izteiksmei	Būla vērtība (1/0)
!	Unārs	Būla NOT	Būla vērtība (1/0)
||	Binārs	Būla OR	Būla vērtība (1/0)
&&	Binārs	Būla AND	Būla vērtība (1/0)

Funkcijas

Funkcijas var izmantot JSONPath beigās. Var sasaistīt vairākas funkcijas, ja iepriekšējā funkcija atgriež vērtību, ko pieņem nākamā funkcija.

Atbalstītās funkcijas:

Function	Description	Input	Output
avg	Skaitļu vidējā vērtība ievades masīvā	Skaitļu masīvs	Skaitlis
min	Skaitļu minimālā vērtība ievades masīvā	Skaitļu masīvs	Skaitlis
max	Skaitļu maksimālā vērtība ievades masīvā	Skaitļu masīvs	Skaitlis
sum	Skaitļu summa ievades masīvā	Skaitļu masīvs	Skaitlis
length	Elementu skaits ievades masīvā	Masīvs	Skaitlis
first	Masīva pirmais elements	Masīvs	JSON konstrukcija (objekts, masīvs, vērtība) atkarībā no ievades masīva satura

JSONPath agregācijas funkcijas pieņem skaitliskas vērtības pēdiņās. Šīs vērtības tiek automātiski pārveidotas no virknēm uz skaitliskiem tipiem, kad nepieciešama agregācija. Nesaderīga ievade izraisīs funkcijas kļūdu.

Izvades vērtība

JSONPath var iedalīt noteiktos un nenoteiktos ceļos. Noteikts ceļš var atgriezt tikai null vai vienu atbilstību. Nenoteikts ceļš var atgriezt vairākas atbilstības: JSONPath ar atdalītiem, vairāku nosaukumu/indeksu sarakstiem, masīva izgriezumiem vai izteiksmju segmentiem. Tomēr, ja tiek izmantota funkcija, JSONPath kļūst noteikts, jo funkcijas vienmēr izvada vienu vērtību.

Noteikts ceļš atgriež objektu/masīvu/vērtību, uz kuru tas atsaucas. Savukārt nenoteikts ceļš atgriež masīvu ar atbilstošajiem objektiem/masīviem/vērtībām.

> **Attention:**
> Īpašību secība JSONPath vaicājuma rezultātos var nesakrist ar sākotnējo JSON īpašību secību iekšējo optimizācijas metožu dēļ. Piemēram, JSONPath `$.books[1]["author", "title"]` var atgriezt `["title", "author"]`. Ja ir būtiski saglabāt sākotnējo īpašību secību, jāapsver alternatīvas pēcapstrādes metodes pēc vaicājuma izpildes.

Ceļa formatēšanas noteikumi

Atstarpes (atstarpes, tabulācijas rakstzīmes) var izmantot iekavu notācijas segmentos un izteiksmēs, piemēram: `$[ 'a' ][ 0 ][ ?( $.b == 'c' ) ][ : -1 ].first( )`.

Virknes jāiekļauj vienpēdiņās (') vai dubultpēdiņās ("). Virkņu iekšpusē vienpēdiņas vai dubultpēdiņas (atkarībā no tā, kuras tiek izmantotas to ietveršanai) un atpakaļvērstās slīpsvītras (\) tiek atdalītas ar atpakaļvērstās slīpsvītras (\) rakstzīmi.

Piemērs


```
{
  "books": [
    {
      "category": "uzziņu literatūra",
      "author": "Nigel Rees",
      "title": "Gadsimta teicieni",
      "price": 8.95,
      "id": 1
    },
    {
      "category": "daiļliteratūra",
      "author": "Evelyn Waugh",
      "title": "Goda zobens",
      "price": 12.99,
      "id": 2
    },
    {
      "category": "daiļliteratūra",
      "author": "Herman Melville",
      "title": "Mobijs Diks",
      "isbn": "0-553-21311-3",
      "price": 8.99,
      "id": 3
    },
    {
      "category": "daiļliteratūra",
      "author": "J. R. R. Tolkien",
      "title": "Gredzenu pavēlnieks",
      "isbn": "0-395-19395-8",
      "price": 22.99,
      "id": 4
    }
  ],
  "services": {
    "delivery": {
      "servicegroup": 1000,
      "description": "Piegāde nākamajā dienā vietējā pilsētā",
      "active": true,
      "price": 5
```

```
    },
    "bookbinding": {
      "servicegroup": 1001,
      "description": "Grāmatas drukāšana un salikšana A5 formātā",
      "active": true,
      "price": 154.99
    },
    "restoration": {
      "servicegroup": 1002,
      "description": "Dažādas restaurācijas metodes",
      "active": false,
      "methods": [
        {
          "description": "Ķīmiskā tīrīšana",
          "price": 46
        },
        {
          "description": "Mitruma bojātu lapu presēšana",
          "price": 24.5
        },
        {
          "description": "Saplēstas grāmatas pārsiešana",
          "price": 99.49
        }
      ]
    }
  },
  "filters": {
    "price": 10,
    "category": "daiļliteratūra",
    "no filters": "nav \"filtru\""
  },
  "closed message": "Veikals ir slēgts",
  "tags": [
    "a",
    "b",
    "c",
    "d",
    "e"
  ]
}
```

JSONPath	Type	Result
`$.filters.price`	definite	10
`$.filters.category`	definite	daiļliteratūra
`$.filters['no filters']`	definite	nav ”filtru”
`$.filters`	definite	{ ”price”: 10, ”category”: ”daiļliteratūra”, ”no filters”: ”nav \”filtru\”” }
`$.books[1].title`	definite	Goda zobens
`$.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	[”Mobijs Diks”, ”Gadsimta teicieni”]

JSONPath	Type	Result
`$.books[1]['author', "title"]`	indefinite	[”Goda zobens”, ”Evelyn Waugh”]
`$..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	[”Goda zobens”] Piezīme: Šis vaicājums parāda, ka vaicājumos var izmantot aritmētiskās darbības; to var vienkāršot līdz `$.books[?(@.id == 2)].title`
`$.books[?(@.id == 2 || @.id == 4)].title`	indefinite	[”Goda zobens”, ”Gredzenu pavēlnieks”]
`$.books[?(!(@.id == 2))].title`	indefinite	[”Gadsimta teicieni”, ”Mobijs Diks”, ”Gredzenu pavēlnieks”]
`$.books[?(@.id != 2)].title`	indefinite	[”Gadsimta teicieni”, ”Mobijs Diks”, ”Gredzenu pavēlnieks”]
`$.books[?(@.title =~ " of ")].title`	indefinite	[”Gadsimta teicieni”, ”Goda zobens”, ”Gredzenu pavēlnieks”]
`$.books[?(@.price > 12.99)].title`	indefinite	[”Gredzenu pavēlnieks”]
`$.books[?(@.author > "Herman Melville")].title`	indefinite	[”Gadsimta teicieni”, ”Gredzenu pavēlnieks”]
`$.books[?(@.price > $.filters.price)].title`	indefinite	[”Goda zobens”, ”Gredzenu pavēlnieks”]
`$.books[?(@.category == $.filters.category)].title`	indefinite	[”Goda zobens”,”Mobijs Diks”,”Gredzenu pavēlnieks”]
`$.books[?(@.category == "fiction" && @.price < 10)].title`	indefinite	[”Mobijs Diks”]

`$..[?(@.id)]`	indefinite	[
		{
		”price”: 8.95,
		”id”: 1,
		”category”: ”uzziņu literatūra”,
		”author”: ”Nigel Rees”,
		”title”: ”Gadsimta teicieni”
		},
		{
		”price”: 12.99,
		”id”: 2,
		”category”: ”daiļliteratūra”,
		”author”: ”Evelyn Waugh”,
		”title”: ”Goda zobens”
		},
		{
		”price”: 8.99,
		”id”: 3,
		”category”: ”daiļliteratūra”,
		”author”: ”Herman Melville”,
		”title”: ”Mobijs Diks”,
		”isbn”: ”0-553-21311-3”
		},
		{
		”price”: 22.99,
		”id”: 4,
		”category”: ”daiļliteratūra”,
		”author”: ”J. R. R. Tolkien”,
		”title”: ”Gredzenu pavēlnieks”,
		”isbn”: ”0-395-19395-8”
		}
		]

`$.services..[?(@.price > 50)].description`	indefinite	[”Grāmatas drukāšana un salikšana A5 formātā”, ”Saplēstas grāmatas pārsiešana”]

JSONPath	Type	Result
`$..id.length()`	definite	4
`$.books[?(@.id == 2)].title.first()`	definite	Goda zobens
`$..tags.first().length()`	definite	5 Piezīme: `$..tags` ir nenoteikts ceļš, tāpēc tas atgriež atbilstošo elementu masīvu, t. i., `[["a", "b", "c", "d", "e" ]]`; `first()` atgriež pirmo elementu, t. i., `["a", "b", "c", "d", "e"]`; `length()` aprēķina elementa garumu, t. i., 5.
`$.books[*].price.min()`	definite	8.95
`$..price.max()`	definite	154.99
`$.books[?(@.category == "fiction")].price.avg()`	definite	14.99
`$.books[?(@.category == $.filters.xyz)].title`	indefinite	Piezīme: Vaicājums bez atbilstības atgriež NULL noteiktiem un nenoteiktiem ceļiem.
`$.services[?(@.active=="true")].servicegroup`	indefinite	[1001,1000] Piezīme: Salīdzinājumos ar Būla vērtībām jāizmanto teksta konstantes.
`$.services[?(@.active=="false")].servicegroup`	indefinite	[1002] Piezīme: Salīdzinājumos ar Būla vērtībām jāizmanto teksta konstantes.
`$.services[?(@.servicegroup=="1002")]~.first()`	definite	true

1 Speciālo rakstzīmju atsoļošana no LLD makro vērtībām JSONPath

Kad JSONPath priekšapstrādē tiek izmantoti zema līmeņa atklāšanas makro un to vērtības tiek aizstātas, tiek piemēroti šādi speciālo rakstzīmju atsoļošanas noteikumi:

- atsoļošanai tiek ņemtas vērā tikai atpakaļējā slīpsvītra (\) un dubultpēdiņa (") rakstzīmes;
- ja aizstātā makro vērtība satur šīs rakstzīmes, katra no tām tiek atsoļota ar atpakaļējo slīpsvītru;
- ja tās jau ir atsoļotas ar atpakaļējo slīpsvītru, tas netiek uzskatīts par atsoļošanu, un gan atpakaļējā slīpsvītra, gan tai sekojošā speciālā rakstzīme tiek atsoļota vēlreiz.

Piemēram:

JSONPath	LLD makro vērtība	Pēc aizstāšanas
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`
	c:\temp	`$.[?(@.value == "c:\\temp")]`
	a\\b	`$.[?(@.value == "a\\\\b")]`

Ja makro tiek izmantots izteiksmē, makro, kurā var būt speciālās rakstzīmes, jāieliek dubultpēdiņās:

JSONPath	LLD makro vērtība	Pēc aizstāšanas	Rezultāts
`$.[?(@.value == "{#MACRO}")]`	special "value"	`$.[?(@.value == "special \"value\"")]`	OK
`$.[?(@.value == {#MACRO})]`		`$.[?(@.value == special \"value\")]`	**Nederīga JSONPath izteiksme**

Ja makro tiek izmantots ceļā, makro, kurā var būt speciālās rakstzīmes, jāieliek kvadrātiekavās **un** dubultpēdiņās:

JSONPath	LLD makro vērtība	Pēc aizstāšanas	Rezultāts
`$.["{#MACRO}"].value`	c:\temp	`$.["c:\\temp"].value`	OK
`$.{#MACRO}.value`		`$.c:\\temp.value`	**Nederīga JSONPath izteiksme**

5 JavaScript priekšapstrāde

Pārskats

Šajā sadaļā ir sniegta informācija par priekšapstrādi, izmantojot JavaScript.

> **Attention:**
> Neizmantojiet nedeklarētus piešķīrumus priekšapstrādes JavaScript kodā. Izmantojiet `var`, lai deklarētu lokālos mainīgos.

JavaScript priekšapstrāde

JavaScript priekšapstrāde tiek veikta, izsaucot JavaScript funkciju ar vienu parametru 'value' un lietotāja norādītu funkcijas pamatdaļu. Priekšapstrādes soļa rezultāts ir šīs funkcijas atgrieztā vērtība, piemēram, lai veiktu konvertēšanu no Fārenheita uz Celsiju, JavaScript priekšapstrādes parametros ievadiet:

```
return (value - 32)  * 5 / 9
```

serveris to ietvers JavaScript funkcijā:

```
function (value)
{
   return (value - 32) * 5 / 9
}
```

Ievades parametrs 'value' vienmēr tiek nodots kā virkne. Atgrieztā vērtība tiek automātiski pārveidota par virkni, izmantojot metodi toString() (ja tas neizdodas, kļūda tiek atgriezta kā virknes vērtība), ar dažiem izņēmumiem:

- atgriežot undefined vērtību, tiks radīta kļūda;
- atgriežot null vērtību, ievades vērtība tiks atmesta, līdzīgi kā priekšapstrāde 'Discard value' darbībā 'Custom on fail'.

Kļūdas var atgriezt, izmetot vērtības/objektus (parasti tās ir vai nu virknes, vai Error objekti).

Piemēram:

```
if (value == 0)
    throw "Zero input value"
return 1/value
```

Katram skriptam ir 10 sekunžu izpildes noildze (atkarībā no skripta var paiet ilgāks laiks, līdz noildze nostrādā); tās pārsniegšana atgriezīs kļūdu. Tiek piemērots 512 megabaitu kaudzes atmiņas ierobežojums.

JavaScript priekšapstrādes soļa baitkods tiek kešots un izmantots atkārtoti, kad šis solis tiek lietots nākamreiz. Jebkādas izmaiņas vienuma priekšapstrādes soļos izraisīs kešotā skripta atiestatīšanu un vēlāku pārkompilēšanu.

Secīgas izpildlaika kļūmes (3 pēc kārtas) izraisīs dzinēja atkārtotu inicializēšanu, lai mazinātu iespēju, ka viens skripts sabojā izpildes vidi nākamajiem skriptiem (šī darbība tiek reģistrēta žurnālā ar DebugLevel 4 un augstāku).

JavaScript priekšapstrāde ir ieviesta, izmantojot Duktape JavaScript dzinēju.

Skatiet arī: Papildu JavaScript objekti un globālās funkcijas

Makro izmantošana skriptos

Ir iespējams izmantot lietotāja makro (kā arī LLD makro zema līmeņa atklāšanas kontekstā) JavaScript kodā. Ja skripts satur lietotāja makro, šie makro tiek atrisināti serverī/starpniekserverī pirms konkrētu priekšapstrādes soļu izpildes. Ņemiet vērā, ka, testējot priekšapstrādes soļus lietotāja saskarnē, makro vērtības netiks ielasītas un tās būs jāievada manuāli.

> **Note:**
> Konteksts tiek ignorēts, kad makro tiek aizstāts ar tā vērtību. Makro vērtība tiek ievietota kodā tāda, kāda tā ir; pirms vērtības ievietošanas JavaScript kodā nav iespējams pievienot papildu atsoļošanu. Lūdzu, ņemiet vērā, ka dažos gadījumos tas var izraisīt JavaScript kļūdas.

Tālāk redzamajā piemērā, ja saņemtā vērtība pārsniedz {$THRESHOLD} makro vērtību, tā vietā tiks atgriezta sliekšņa vērtība (ja tā ir norādīta):

```
var threshold = '{$THRESHOLD}';
return (!isNaN(threshold) && value > threshold) ? threshold : value;
```

Piemēri

Tālāk sniegtie piemēri ilustrē, kā var izmantot JavaScript priekšapstrādi.

Katrs piemērs satur īsu aprakstu, funkcijas pamattekstu JavaScript priekšapstrādes parametriem un priekšapstrādes soļa rezultātu — funkcijas atgriezto vērtību.

Piemērs 1: Pārveidot skaitli (zinātniskā pieraksta formu uz veselu skaitli)

Pārveidojiet skaitli "2.62128e+07" no zinātniskā pieraksta formas uz veselu skaitli.

```
return (Number(value))
```

Funkcijas atgrieztā vērtība: 26212800.

Piemērs 2: Pārveidot skaitli (no binārās uz decimālo sistēmu)

Pārveidojiet bināro skaitli "11010010" par decimālo skaitli.

```
return(parseInt(value,2))
```

Funkcijas atgrieztā vērtība: 210.

Piemērs 3: Noapaļot skaitli

Noapaļojiet skaitli "18.2345" līdz 2 cipariem aiz komata.

```
return(Math.round(value* 100) / 100)
```

Funkcijas atgrieztā vērtība: 18.23.

Piemērs 4: Saskaitīt burtus virknē

Saskaitiet burtu skaitu virknē "Zabbix".

```
return (value.length)
```

Funkcijas atgrieztā vērtība: 6.

Piemērs 5: Iegūt atlikušo laiku

Iegūstiet atlikušo laiku (sekundēs) līdz sertifikāta derīguma termiņa beigām (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);
```

Funkcijas atgrieztā vērtība: 44380233.

Piemērs 6: Noņemt JSON rekvizītus

Modificējiet JSON datu struktūru, noņemot visus rekvizītus ar atslēgu `"data_size"` vai `"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)
```

Funkcijas pieņemtā vērtība:

```
[
    {
        "table_name":"history",
        "data_size":"326.05",
        "index_size":"174.34"
    },
    {
        "table_name":"history_log",
        "data_size":"6.02",
        "index_size":"3.45"
```

```
    }
]
```

Funkcijas atgrieztā vērtība:

```
[
    {
        "table_name":"history"
    },
    {
        "table_name":"history_log"
    }
]
```

Piemērs 7: Apache statusa pārveidošana par JSON

Pārveidojiet vērtību, kas saņemta no web.page.get Zabbix aģenta vienuma (piemēram, web.page.get[http://127.0.0.1:80/server-status?auto]), par JSON objektu.

```
// Pārveidot Apache statusu par JSON

// Sadalīt vērtību apakšvirknēs un ievietot šīs apakšvirknes masīvā
var lines = value.split('\n');

// Izveidot tukšu objektu "output"
var output = {};

// Izveidot objektu "workers" ar iepriekš definētām īpašībām
var workers = {
    '_': 0, 'S': 0, 'R': 0, 'W': 0,
    'K': 0, 'D': 0, 'C': 0, 'L': 0,
    'G': 0, 'I': 0, '.': 0
};

// Pievienot apakšvirknes no masīva "lines" objektam "output" kā īpašības (atslēgas-vērtības pārus)
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]);
    }
}

// Daudzversiju metrikas
output.ServerUptimeSeconds = output.ServerUptimeSeconds || output.Uptime;
output.ServerVersion = output.ServerVersion || output.Server;

// Parsēt īpašību "Scoreboard", lai iegūtu darba procesu skaitu
if (typeof output.Scoreboard === 'string') {
    for (var i = 0; i < output.Scoreboard.length; i++) {
        var char = output.Scoreboard[i];

        workers[char]++;
    }
}

// Pievienot darba procesu datus objektam "output"
output.Workers = {
    waiting: workers['_'], starting: workers['S'], reading: workers['R'],
    sending: workers['W'], keepalive: workers['K'], dnslookup: workers['D'],
    closing: workers['C'], logging: workers['L'], finishing: workers['G'],
    cleanup: workers['I'], slot: workers['.']
};

// Atgriezt JSON virkni
```

```
return JSON.stringify(output);
```

Funkcijas pieņemtā vērtība:

```
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__.....................................................................................
```

Funkcijas atgrieztā vērtība:


```
{
    "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 Papildu JavaScript objekti

Pārskats

Šajā sadaļā ir aprakstīti Zabbix papildinājumi JavaScript valodai, kas ieviesti ar Duktape, un atbalstītās globālās JavaScript funkcijas.

Attention:
Neizmantojiet nedeklarētus piešķīrumus priekšapstrādes JavaScript kodā. Izmantojiet `var`, lai deklarētu lokālos mainīgos.

Iebūvētie objekti

Zabbix

Zabbix objekts nodrošina mijiedarbību ar Zabbix iekšējo funkcionalitāti.

Method	Description
`log(loglevel, message)`	Ieraksta <message> Zabbix žurnālā, izmantojot <loglevel> žurnāla līmeni (skatiet konfigurācijas faila parametru DebugLevel).

Piemērs:

```
Zabbix.log(3, "this is a log entry written with 'Warning' log level")
```

Varat izmantot šādus aizstājvārdus:

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:
Visu reģistrēto ziņojumu kopējais lielums ir ierobežots līdz 8 MB vienai skripta izpildei.

Method	Description
`sleep(delay)`	Aizkavē JavaScript izpildi par `delay` milisekundēm.

Piemērs (aizkavē izpildi par 15 sekundēm):

```
Zabbix.sleep(15000)
```

HttpRequest

Šis objekts ietver cURL turi, kas ļauj veikt vienkāršus HTTP pieprasījumus. Kļūdas tiek izmestas kā izņēmumi.

Attention:
Vairāku `HttpRequest` objektu inicializācija ir ierobežota līdz 10 vienam skripta izpildes gadījumam.

Method	Description
`addHeader(value)`	Pievieno HTTP galvenes lauku. Šis lauks tiek izmantots visiem turpmākajiem pieprasījumiem, līdz tas tiek notīrīts ar `clearHeader()` metodi. Kopējais galvenes lauku garums, ko var pievienot vienam `HttpRequest` objektam, ir ierobežots līdz 128 Kbaitiem (ieskaitot speciālās rakstzīmes un galveņu nosaukumus).
`clearHeader()`	Notīra HTTP galveni. Ja nav iestatīts neviens galvenes lauks, `HttpRequest` iestatīs Content-Type uz application/json, ja nosūtāmie dati ir JSON formātā; pretējā gadījumā — text/plain.
`connect(url)`	Nosūta HTTP CONNECT pieprasījumu uz URL un atgriež atbildi.
`customRequest(method, url, data)`	Ļauj pirmajā parametrā norādīt jebkuru HTTP metodi. Nosūta šīs metodes pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`delete(url, data)`	Nosūta HTTP DELETE pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`getHeaders(<asArray>)`	Atgriež saņemto HTTP galvenes lauku objektu. Parametram `asArray` var iestatīt vērtību "true" (piemēram, `getHeaders(true)`), "false" vai tas var nebūt definēts. Ja iestatīts uz "true", saņemto HTTP galvenes lauku vērtības tiks atgrieztas kā masīvi; tas jāizmanto, lai iegūtu vairāku vienāda nosaukuma galveņu lauku vērtības. Ja parametrs nav iestatīts vai ir iestatīts uz "false", saņemto HTTP galvenes lauku vērtības tiks atgrieztas kā virknes.
`get(url, data)`	Nosūta HTTP GET pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`head(url)`	Nosūta HTTP HEAD pieprasījumu uz URL un atgriež atbildi.
`options(url)`	Nosūta HTTP OPTIONS pieprasījumu uz URL un atgriež atbildi.
`patch(url, data)`	Nosūta HTTP PATCH pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`put(url, data)`	Nosūta HTTP PUT pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`post(url, data)`	Nosūta HTTP POST pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.
`getStatus()`	Atgriež pēdējā HTTP pieprasījuma statusa kodu.
`setProxy(proxy)`	Iestata HTTP starpniekserveri uz "proxy" vērtību. Ja šis parametrs ir tukšs, starpniekserveris netiek izmantots.

`setHttpAuth(bitmask, username, password)`	Iestata iespējotās HTTP autentifikācijas metodes (HTTPAUTH_BASIC, HTTPAUTH_DIGEST, HTTPAUTH_NEGOTIATE, HTTPAUTH_NTLM, HTTPAUTH_NONE) parametrā 'bitmask'.
	Karogs HTTPAUTH_NONE ļauj atspējot HTTP autentifikāciju.
	Piemēri:
	`request.setHttpAuth(HTTPAUTH_NTLM | HTTPAUTH_BASIC, username, password)`
	`request.setHttpAuth(HTTPAUTH_NONE)`

`trace(url, data)`	Nosūta HTTP TRACE pieprasījumu uz URL ar neobligātu *data* slodzi un atgriež atbildi.

Piemērs:

```
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

XML objekts ļauj apstrādāt XML datus vienuma un zema līmeņa atklāšanas priekšapstrādē, kā arī webhook.

Attention:
Lai izmantotu XML objektu, serverim/starpniekserverim jābūt kompilētam ar libxml2 atbalstu.

Method	Description
`XML.query(data, expression)`	Iegūst mezgla saturu, izmantojot XPath. Atgriež null, ja mezgls netiek atrasts. **expression** - XPath izteiksme; **data** - XML dati virknes formā.
`XML.toJson(data)`	Pārveido datus XML formātā uz JSON.
`XML.fromJson(object)`	Pārveido datus JSON formātā uz XML.

Piemērs:

Ievade:

```
<menu>
    <food type = "breakfast">
        <name>Chocolate</name>
        <price>$5.95</price>
        <description></description>
        <calories>650</calories>
    </food>
</menu>
```

Izvade:

```
{
    "menu": {
        "food": {
            "@type": "breakfast",
            "name": "Chocolate",
            "price": "$5.95",
            "description": null,
            "calories": "650"
        }
    }
}
```

Serializācijas noteikumi

XML pārveidošana uz JSON tiks apstrādāta saskaņā ar šādiem noteikumiem (JSON pārveidošanai uz XML tiek piemēroti apgriezti noteikumi):

1. XML atribūti tiks pārveidoti par atslēgām, kuru nosaukumiem priekšā ir pievienots '@'.

Piemērs:

Ievade:

```
<xml foo="FOO">
  <bar>
    <baz>BAZ</baz>
  </bar>
</xml>
```

Izvade:

```
{
  "xml": {
    "@foo": "FOO",
    "bar": {
      "baz": "BAZ"
    }
  }
}
```

2. Pašaizverošie elementi (<foo/>) tiks pārveidoti kā tādi, kuriem ir 'null' vērtība.

Piemērs:

Ievade:

```
<xml>
  <foo/>
</xml>
```

Izvade:

```
{
  "xml": {
    "foo": null
  }
}
```

3. Tukši atribūti (ar "" vērtību) tiks pārveidoti kā tādi, kuriem ir tukšas virknes (") vērtība.

Piemērs:

Ievade:

```
<xml>
  <foo bar="" />
</xml>
```

Izvade:

```
{
  "xml": {
```

```
    "foo": {
      "@bar": ""
    }
  }
}
```

4. Vairāki bērnelementi ar vienādu elementa nosaukumu tiks pārveidoti par vienu atslēgu, kuras vērtība ir vērtību masīvs.

Piemērs:

Ievade:

```
<xml>
  <foo>BAR</foo>
  <foo>BAZ</foo>
  <foo>QUX</foo>
</xml>
```

Izvade:

```
{
  "xml": {
    "foo": ["BAR", "BAZ", "QUX"]
  }
}
```

5. Ja teksta elementam nav atribūtu un nav bērnelementu, tas tiks pārveidots par virkni.

Piemērs:

Ievade:

```
<xml>
    <foo>BAZ</foo>
</xml>
```

Izvade:

```
{
  "xml": {
    "foo": "BAZ"
   }
}
```

6. Ja teksta elementam nav bērnelementu, bet tam ir atribūti, teksta saturs tiks pārveidots par elementu ar atslēgu '#text' un saturu kā vērtību; atribūti tiks pārveidoti, kā aprakstīts 1. serializācijas noteikumā.

Piemērs:

Ievade:

```
<xml>
  <foo bar="BAR">
    BAZ
  </foo>
</xml>
```

Izvade:

```
{
  "xml": {
    "foo": {
      "@bar": "BAR",
      "#text": "BAZ"
    }
  }
}
```

Globālās JavaScript funkcijas

Ar Duktape ir ieviestas papildu globālās JavaScript funkcijas:

- `btoa(data)` — kodē datus base64 virknē.
- `atob(base64_string)` — dekodē base64 virkni kā Uint8Array buferi.

```
try {
    b64 = btoa("test string");
    buffer = atob(b64);

    // Ņemiet vērā, ka dekodēšanas loģika ir atkarīga no bufera datu formāta.
    decoded = String.fromCharCode.apply(this, [].slice.call(buffer));
}
catch (error) {
    return {'error.name' : error.name, 'error.message' : error.message};
}
```

- `md5(data)` — aprēķina datu MD5 jaucējvērtību.
- `sha256(data)` — aprēķina datu SHA256 jaucējvērtību.
- `hmac('<hash type>',key,data)` — atgriež HMAC jaucējvērtību kā heksadecimāli formatētu virkni; kā `hash type` tiek atbalstīti `md5` un `sha256`; parametri `key` un `data` atbalsta binārus datus.

  Piemēri:

  - `hmac('md5',key,data)`
  - `hmac('sha256',key,data)`

- `sign(hash,key,data)` — atgriež aprēķināto parakstu (RSA paraksts ar SHA-256) kā virkni, kur:
 hash — ir atļauts tikai `sha256`, pretējā gadījumā tiks izraisīta kļūda.
 key — privātā atslēga. Tai jāatbilst PKCS#1 vai PKCS#8 standartam. Atslēgu var norādīt dažādās formās:

  - ar atstarpēm jaunu rindu vietā
  - ar ekrānētu vai neekrānētu `\n` jaunu rindu vietā
  - bez jebkādām jaunām rindām kā vienas rindas virkni
  - kā JSON formatētu virkni

  Atslēgu var arī ielādēt no lietotāja makrosa/secret macro/vault.

  data — dati, kas tiks parakstīti. Tā var būt virkne (tiek atbalstīti arī binārie dati) vai buferis (Uint8Array/ArrayBuffer).

  Piemērs:

  - `sign('sha256',key,data)`

  Parakstu aprēķināšanai tiek izmantots OpenSSL vai GnuTLS. Ja Zabbix ir būvēts bez kādas no šīm šifrēšanas bibliotēkām, tiks izraisīta kļūda (’missing OpenSSL or GnuTLS library’).

2 Pārlūka vienuma JavaScript objekti

Pārskats

Šajā sadaļā ir aprakstīti Zabbix papildinājumi JavaScript valodai, kas ieviesti ar Duktape izmantošanai Pārlūka vienuma skriptā. Šie papildinājumi papildina JavaScript objektus, kas aprakstīti lapā Papildu JavaScript objekti.

Browser

Objekts `Browser` pārvalda WebDriver sesijas, inicializējot sesiju izveides brīdī un pārtraucot to iznīcināšanas brīdī. Viens skripts var atbalstīt līdz četriem `Browser` objektiem.

Lai izveidotu `Browser` objektu, izmantojiet sintaksi `new Browser(options)`. Parametrs `options` (*JSON object*) norāda pārlūka opcijas, parasti WebDriver opciju metodes rezultātu (piemēram, `Browser.chromeOptions()`).

Ar `Browser` objektu tiek atbalstītas šādas metodes.

Method	Description
`navigate(url)`	Pāriet uz norādīto URL. Parametri: `url` - (string) URL, uz kuru pāriet.
`getUrl()`	Atgriež atvērtās lapas URL kā virkni.
`getPageSource()`	Atgriež atvērtās lapas pirmkodu kā virkni.

Method	Description
`findElement(strategy, selector)`	Atgriež `Element` objektu ar vienu elementu atvērtajā lapā (vai atgriež `null`, ja neviens elements neatbilst `strategy` un `selector`). Parametri: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Atrašanās vietas stratēģija; `selector` - (string) Elementa selektors, izmantojot norādīto atrašanās vietas stratēģiju.
`findElements(strategy, target)`	Atgriež `Element` objektu masīvu ar vairākiem elementiem atvērtajā lapā (vai atgriež tukšu masīvu, ja neviens elements neatbilst atrašanās vietas stratēģijai un mērķim). Parametri: `strategy` - (string, css selector/link text/partial link text/tag name/xpath) Atrašanās vietas stratēģija; `target` - (string) Elementa selektors, izmantojot norādīto atrašanās vietas stratēģiju.
`getCookies()`	Atgriež `Cookie` objektu masīvu.
`addCookie(cookie)`	Iestata sīkdatnes. Parametri: `cookie` - (`Cookie` object) Iestatāmā sīkdatne.
`getScreenshot()`	Atgriež base64 kodētu virkni, kas attēlo pārlūka skata apgabala attēlu.
`setScreenSize(x,y)`	Iestata pārlūka skata apgabala izmēru. Parametri: `x` - (string) Platums pikseļos; `y` - (string) Augstums pikseļos.
`setScriptTimeout(timeout)`	Iestata skripta ielādes noildzi. Parametri: `timeout` - (integer) Noildzes vērtība milisekundēs.
`setSessionTimeout(timeout)`	Iestata sesijas (lapas ielādes) noildzi. Parametri: `timeout` - (integer) Noildzes vērtība milisekundēs.
`setElementWaitTimeout(timeout)`	Iestata elementa atrašanās vietas stratēģijas (netiešo) noildzi. Parametri: `timeout` - (integer) Noildzes vērtība milisekundēs.
`collectPerfEntries(mark)`	Savāc veiktspējas ierakstus, ko izgūt ar metodi `getResult()`. Parametri: `mark` - (string, optional) veiktspējas momentuzņēmuma atzīme.
`getRawPerfEntries()`	Atgriež veiktspējas ierakstu objektu masīvu.
`getResult()`	Atgriež `Result` objektu ar pārlūka sesijas statistiku (kļūdu informāciju, veiktspējas momentuzņēmumiem u.c.).
`getError()`	Atgriež `BrowserError` objektu ar pārlūka kļūdām (vai atgriež `null`, ja pārlūka kļūdu nav).
`setError(message)`	Iestata pielāgotu kļūdas ziņojumu, kas tiks iekļauts `Result` objektā. Parametri: `message` - (string) Kļūdas ziņojums.
`discardError()`	Atmet kļūdu, kas tiks atgriezta `Result` objektā.
`getAlert()`	Atgriež `Alert` objektu ar pārlūka brīdinājumiem (vai atgriež `null`, ja pārlūka brīdinājumu nav).
`chromeOptions()`	Atgriež `chromeOptions` objektu ar iepriekš definētām Chrome pārlūka opcijām.
`firefoxOptions()`	Atgriež `firefoxOptions` objektu ar iepriekš definētām Firefox pārlūka opcijām.
`safariOptions()`	Atgriež `safariOptions` objektu ar iepriekš definētām Safari pārlūka opcijām.
`edgeOptions()`	Atgriež `edgeOptions` objektu ar iepriekš definētām Edge pārlūka opcijām.

Method	Description
`switchFrame(target)`	Pārslēdzas uz norādīto rāmi. Parametri: `target` - (browser element or integer, optional) Mērķa rāmis. Lai atlasītu rāmi pēc elementa, nododiet elementu. Lai atlasītu rāmi pēc indeksa, nododiet skaitli. Ja atstāts tukšs, tiks pārslēgts uz augstākā līmeņa pārlūkošanas kontekstu.

Visas Browser metodes var izraisīt šādas kļūdas:

- `BrowserError` - atvasināts no objekta `Error`, kas tiek izraisīts, ja `Browser` konstruktors neizdodas; satur papildu rekvizītu `browser` ar `Browser` objektu, kas izraisīja šo `BrowserError`.
- `WebdriverError` - atvasināts no `BrowserError`; satur tās pašas īpašības kā objekts `BrowserError`, kas norāda, vai kļūda tika ģenerēta, reaģējot uz kļūdu WebDriver atbildē.

Elements

`Element` objekts tiek atgriezts ar `Browser` objekta `findElement()`/`findElements()` metodēm, un to nevar izveidot tieši.

`Element` objekts attēlo elementu tīmekļa lapā un nodrošina metodes mijiedarbībai ar to.

Ar `Element` objektu tiek atbalstītas šādas metodes.

Method	Description
`getAttribute(name)`	Atgriež elementa atribūta vērtības virkni (vai atgriež `null`, ja norādītais atribūts netika atrasts). Parametri: `name` - (string) Atribūta nosaukums.
`getProperty(name)`	Atgriež elementa īpašības vērtības virkni (vai atgriež `null`, ja norādītā īpašība netika atrasta). Parametri: `name` - (string) Īpašības nosaukums.
`getText()`	Atgriež elementa teksta vērtības virkni.
`click()`	Noklikšķina uz elementa.
`clear()`	Notīra rediģējama elementa saturu.
`sendKeys(keys)`	Nosūta taustiņus. Parametri: `keys` - (string) Nosūtāmie taustiņi.

Cookie

Objektu `Cookie` atgriež objekta `Browser` metode `getCookies()`, un tas tiek nodots metodei `addCookie()`.

Lai gan objektam `Cookie` nav metožu, tas var saturēt šādas īpašības:

Īpašība	Type	Apraksts
name	string	Sīkdatnes nosaukums.
value	string	Sīkdatnes vērtība.
path	string	Ceļš, kuram sīkdatne ir derīga. Ja, pievienojot sīkdatni, tas nav norādīts, pēc noklusējuma tiek izmantots `"/"`.
domain	string	Domēns, kurā sīkdatne ir redzama. Ja, pievienojot sīkdatni, tas nav norādīts, pēc noklusējuma tiek izmantots sesijas pašreizējā pārlūkošanas konteksta aktīvā dokumenta URL domēns.
secure	boolean	Norāda, vai sīkdatne ir droša. Ja, pievienojot sīkdatni, tas nav norādīts, pēc noklusējuma tiek izmantots `false`.
httpOnly	boolean	Norāda, vai sīkdatne ir tikai HTTP. Ja, pievienojot sīkdatni, tas nav norādīts, pēc noklusējuma tiek izmantots `false`.

Īpašība	Type	Apraksts
expiry	integer	Sīkdatnes derīguma termiņš (sekundēs kopš Unix laikmeta sākuma). To nedrīkst iestatīt, ja, pievienojot sīkdatni, tas nav norādīts.
sameSite	string	Sīkdatnes atribūts `sameSite`, kas nosaka, vai sīkdatnei jābūt ierobežotai līdz pirmās puses vai tās pašas vietnes kontekstam. Var iestatīt uz `"Lax"` vai `"Strict"`. Ja, pievienojot sīkdatni, tas nav norādīts, pēc noklusējuma tiek izmantots `"None"`.

Brīdinājums

Objekts `Alert` attēlo tīmekļa lapas brīdinājumu, to atgriež objekta `Browser` metode `getAlert()`, un to nevar izveidot tieši.

Objekts `Alert` satur rekvizītu `text` ar brīdinājuma tekstu (vai `null`, ja brīdinājumu nav).

Ar objektu `Alert` tiek atbalstītas šādas metodes.

Method	Description
`accept()`	Apstiprināt brīdinājumu.
`dismiss()`	Noraidīt brīdinājumu.

Rezultāts

Objekts `Result` satur sesijas statistiku un tiek atgriezts ar objekta `Browser` metodi `getResult()`.

Parasti objekts `Result` tiek pārveidots par virkni un atgriezts no skripta, bet pēc tam, izmantojot priekšapstrādi, parsēts atkarīgo vienumu vērtībās.

Lai gan objektam `Result` nav metožu, tas var saturēt tālāk norādītās īpašības.

Īpašība	Type	Apraksts
duration	string	Sesijas ilgums no sesijas izveides līdz rezultāta iegūšanai.
error	object	Kļūdas informācija.
http_status	integer	HTTP statuss, ko atgrieza WebDriver (vai 0, ja nav WebDriver kļūdu).
error_code	string	Kļūda, ko atgrieza WebDriver (vai tukša virkne, ja nav WebDriver kļūdu).
message	string	WebDriver kļūdas ziņojums (vai tukša virkne, ja nav WebDriver kļūdu).
performance_data	object	Veiktspējas statistika.
summary	object	Veiktspējas kopsavilkums.
navigation	object	Navigācijas kopsavilkums.
resource	object	Resursu kopsavilkums.
details	array of objects	Veiktspējas statistika pēc katras darbības, kas varēja izraisīt navigāciju.
mark	string	*(neobligāti)* Veiktspējas momentuzņēmuma atzīme, kas norādīta ar metodi `collectPerfEntries()`.
navigation	object	Navigācijas statistika.
resource	object	Šīs darbības resursu kopsavilkums.
user	array of objects	Masīvs ar mark/measure tipa statistiku.
marks	array of objects	Atzīmēto veiktspējas momentuzņēmumu indeksu masīvs.
name	string	Veiktspējas momentuzņēmuma atzīmes nosaukums.
index	integer	Veiktspējas momentuzņēmuma indekss masīvā details.

6 CSV uz JSON priekšapstrāde

Pārskats

Šajā priekšapstrādes solī ir iespējams pārveidot CSV faila datus JSON formātā. Tas tiek atbalstīts:

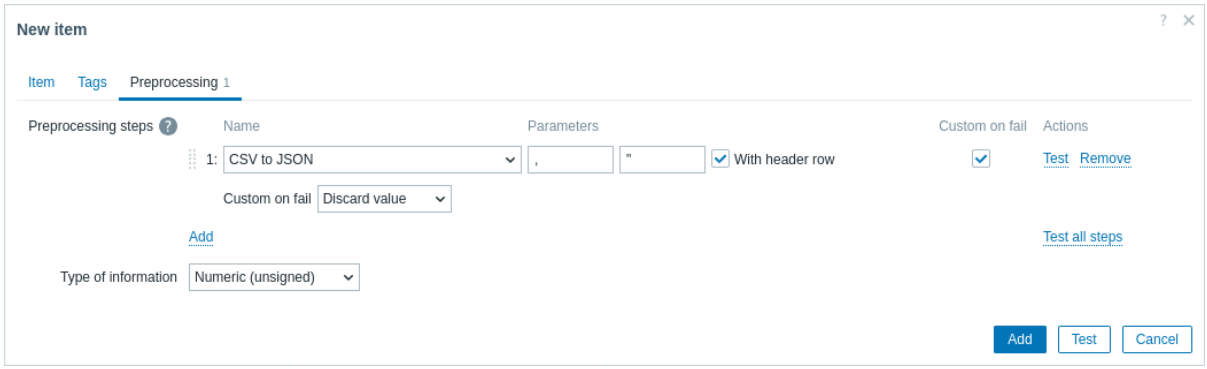
- vienumos (vienumu prototipos)
- zema līmeņa atklāšanas kārtulās

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 Vienumu tipi

Pārskats

Vienumu tipi aptver dažādas datu iegūšanas metodes no jūsu sistēmas. Katram vienuma tipam ir savs atbalstīto vienumu atslēgu kopums un nepieciešamie parametri.

Pašlaik Zabbix piedāvā šādus vienumu tipus:

- Zabbix aģenta pārbaudes
- SNMP aģenta pārbaudes
- SNMP slazdi
- IPMI pārbaudes
- Vienkāršās pārbaudes
    - VMware uzraudzība
- Žurnālfailu uzraudzība
- Aprēķinātie vienumi
    - Apkopotie aprēķini
- Zabbix iekšējās pārbaudes
- SSH pārbaudes
- Telnet pārbaudes
- Ārējās pārbaudes
- Trapper vienumi
- JMX uzraudzība
- ODBC pārbaudes
- Atkarīgie vienumi
- HTTP pārbaudes
- Prometheus pārbaudes
- Skripta vienumi
- Pārlūka vienumi

Detalizēta informācija par visiem vienumu tipiem ir iekļauta šīs sadaļas apakšlapās. Lai gan vienumu tipi piedāvā daudz iespēju datu iegūšanai, pastāv arī papildu iespējas, izmantojot lietotāja parametrus vai ielādējamus moduļus.

Dažas pārbaudes veic tikai Zabbix serveris (kā bezaģenta uzraudzību), savukārt citām ir nepieciešams Zabbix aģents vai pat Zabbix Java gateway (JMX uzraudzībai).

Attention:
Ja konkrētam vienuma tipam ir nepieciešama noteikta saskarne (piemēram, IPMI pārbaudei hostā ir nepieciešama IPMI saskarne), šai saskarnei jābūt definētai hosta konfigurācijā.

Hosta definīcijā var iestatīt vairākas saskarnes: Zabbix aģenta, SNMP aģenta, JMX un IPMI. Ja vienums var izmantot vairāk nekā vienu saskarni, tas meklēs pieejamās hosta saskarnes (šādā secībā: Aģents→SNMP→JMX→IPMI), lai piesaistītu pirmo atbilstošo.

Visi vienumi, kas atgriež tekstu (rakstzīmju, žurnāla, teksta informācijas tipi), var arī atgriezt tikai atstarpes (ja piemērojams), iestatot atgriezto vērtību kā tukšu virkni (atbalstīts kopš 2.0).

1 Zabbix aģents

Pārskats

Šajā sadaļā ir sniegta informācija par vienumu atslēgām, kas datu iegūšanai izmanto saziņu ar Zabbix aģents.

Ir pieejamas pasīvās un aktīvās aģents pārbaudes. Konfigurējot vienumu, varat atlasīt nepieciešamo tipu:

- *Zabbix aģents* - pasīvajām pārbaudēm
- *Zabbix aģents (active)* - aktīvajām pārbaudēm

Ņemiet vērā, ka visas vienumu atslēgas, ko atbalsta Zabbix aģents, atbalsta arī jaunās paaudzes Zabbix aģents 2. Skatiet papildu vienumu atslēgas, kuras var izmantot tikai ar aģents 2.

Atbalstītās vienumu atslēgas

Tālāk ir uzskaitītas vienumu atslēgas, kuras varat izmantot ar Zabbix aģents.

Vienumu atslēgas ir uzskaitītas bez parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts	Vienumu grupa
kernel.maxfiles	Maksimālais operētājsistēmas atbalstītais atvērto failu skaits.	Kodols
kernel.maxproc	Maksimālais operētājsistēmas atbalstītais procesu skaits.	
kernel.openfiles	Pašlaik atvērto failu deskriptoru skaits.	
log	Žurnālfaila uzraudzība.	Žurnālu uzraudzība
log.count	Atbilstošo rindu skaits uzraudzītā žurnālfailā.	
logrt	Rotēta žurnālfaila uzraudzība.	
logrt.count	Atbilstošo rindu skaits uzraudzītā rotētā žurnālfailā.	
modbus.get	Nolasa Modbus datus.	Modbus
net.dns	Pārbauda DNS servisa statusu.	Tīkls
net.dns.perf	Pārbauda DNS servisa veiktspēju.	
net.dns.record	Izpilda DNS vaicājumu.	
net.if.collisions	Ārpus loga kolīziju skaits.	
net.if.discovery	Tīkla saskarņu saraksts.	
net.if.in	Ienākošās datplūsmas statistika tīkla saskarnē.	
net.if.out	Izejošās datplūsmas statistika tīkla saskarnē.	
net.if.total	Ienākošās un izejošās datplūsmas statistikas summa tīkla saskarnē.	
net.tcp.listen	Pārbauda, vai šis TCP ports ir LISTEN stāvoklī.	
net.tcp.port	Pārbauda, vai ir iespējams izveidot TCP savienojumu ar norādīto portu.	
net.tcp.service	Pārbauda, vai serviss darbojas un pieņem TCP savienojumus.	
net.tcp.service.perf	Pārbauda TCP servisa veiktspēju.	
net.tcp.socket.count	Atgriež TCP ligzdu skaitu, kas atbilst parametriem.	
net.udp.listen	Pārbauda, vai šis UDP ports ir LISTEN stāvoklī.	
net.udp.service	Pārbauda, vai serviss darbojas un atbild uz UDP pieprasījumiem.	
net.udp.service.perf	Pārbauda UDP servisa veiktspēju.	
net.udp.socket.count	Atgriež UDP ligzdu skaitu, kas atbilst parametriem.	
proc.cpu.util	Procesa CPU izmantojuma procentuālā vērtība.	Procesi

Vienuma atslēga	Apraksts	Vienumu grupa
proc.get	Operētājsistēmas procesu un to parametru saraksts.	
proc.mem	Procesa izmantotā atmiņa baitos.	
proc.num	Procesu skaits.	
sensor	Aparatūras sensora rādījums.	Sensori
system.boottime	Sistēmas sāknēšanas laiks.	Sistēma
system.cpu.discovery	Noteikto CPU/CPU kodolu saraksts.	
system.cpu.intr	Ierīču pārtraukumi.	
system.cpu.load	CPU slodze.	
system.cpu.num	CPU skaits.	
system.cpu.switches	Konteksta pārslēgšanu skaits.	
system.cpu.util	CPU izmantojuma procentuālā vērtība.	
system.hostname	Sistēmas hosta nosaukums.	
system.hw.chassis	Šasijas informācija.	
system.hw.cpu	CPU informācija.	
system.hw.devices	PCI vai USB ierīču saraksts.	
system.hw.macaddr	MAC adrešu saraksts.	
system.localtime	Sistēmas laiks.	
system.run	Izpilda norādīto komandu hostā.	
system.stat	Sistēmas statistika.	
system.sw.arch	Programmatūras arhitektūras informācija.	
system.sw.os	Operētājsistēmas informācija.	
system.sw.os.get	Detalizēta informācija par operētājsistēmu (versija, tips, distributīva nosaukums, mazākā un lielākā versija u.c.).	
system.sw.packages	Instalēto pakotņu saraksts.	
system.sw.packages.get	Detalizēts instalēto pakotņu saraksts.	
system.swap.in	Swap-in (no ierīces uz atmiņu) statistika.	
system.swap.out	Swap-out (no atmiņas uz ierīci) statistika.	
system.swap.size	Mijmaiņas vietas lielums baitos vai procentos no kopējā apjoma.	
system.uname	Sistēmas identifikācija.	
system.uptime	Sistēmas darbības laiks sekundēs.	
system.users.num	Pieteikušos lietotāju skaits.	
vfs.dev.discovery	Blokierīču un to tipu saraksts.	Virtuālās failu sistēmas
vfs.dev.read	Diska lasīšanas statistika.	
vfs.dev.write	Diska rakstīšanas statistika.	
vfs.dir.count	Direktorija ierakstu skaits.	
vfs.dir.get	Direktorija ierakstu saraksts.	
vfs.dir.size	Direktorija lielums.	
vfs.file.cksum	Faila kontrolsumma, kas aprēķināta ar UNIX cksum algoritmu.	
vfs.file.contents	Faila satura iegūšana.	
vfs.file.exists	Pārbauda, vai fails eksistē.	
vfs.file.get	Atgriež informāciju par failu.	
vfs.file.md5sum	Faila MD5 kontrolsumma.	
vfs.file.owner	Iegūst faila īpašnieku.	
vfs.file.permissions	Atgriež 4 ciparu virkni, kas satur oktālo skaitli ar UNIX atļaujām.	
vfs.file.regexp	Iegūst virkni failā.	
vfs.file.regmatch	Atrod virkni failā.	
vfs.file.size	Faila lielums.	
vfs.file.time	Faila laika informācija.	
vfs.fs.discovery	Piemontēto failu sistēmu saraksts ar to tipiem un montēšanas opcijām.	
vfs.fs.get	Piemontēto failu sistēmu saraksts ar to tipiem, pieejamo diska vietu, inode statistiku un montēšanas opcijām.	
vfs.fs.inode	Inode skaits vai procentuālā vērtība.	
vfs.fs.size	Diska vieta baitos vai procentos no kopējā apjoma.	
vm.memory.size	Atmiņas lielums baitos vai procentos no kopējā apjoma.	Virtuālā atmiņa
web.page.get	Iegūst tīmekļa lapas saturu.	Tīmekļa uzraudzība
web.page.perf	Pilnas tīmekļa lapas ielādes laiks.	
web.page.regexp	Atrod virkni tīmekļa lapā.	

Vienuma atslēga	Apraksts	Vienumu grupa
agent.hostmetadata	Aģenta hosta metadati.	Zabbix
agent.hostname	Aģenta hosta nosaukums.	
agent.ping	Aģenta pieejamības pārbaude.	
agent.variant	Zabbix aģents variants (Zabbix aģents vai Zabbix aģents 2).	
agent.version	Zabbix aģents versija.	
zabbix.stats	Atgriež Zabbix serveris vai starpniekserveris iekšējo metriku kopu. Ja ir norādīti `ip` un `port`, metrikas tiek iegūtas no attālās instances; pretējā gadījumā — no lokālās instances.	
zabbix.stats	Atgriež uzraudzīto vienumu skaitu rindā, kuru apstrāde Zabbix serveris vai starpniekserveris ir aizkavēta. Ja ir norādīti `ip` un `port`, metrikas tiek iegūtas no attālās instances; pretējā gadījumā — no lokālās instances.	

Atbalstītās platformas

Ja vien vienuma detaļās nav norādīts citādi, aģenta vienumi (un visi parametri) tiek atbalstīti šādās platformās:

- **Linux**
- **FreeBSD**
- **Solaris**
- **HP-UX**
- **AIX**
- **MacOS X**
- **OpenBSD**
- **NetBSD**

Daudzi aģenta vienumi tiek atbalstīti arī **Windows** vidē. Sīkāku informāciju skatiet lapā Windows aģenta vienums.

Vienuma atslēgas detaļas

Parametri bez leņķiekavām ir obligāti. Parametri, kas atzīmēti ar leņķiekavām **< >**, ir neobligāti.

kernel.maxfiles

 Maksimālais atvērto failu skaits, ko atbalsta operētājsistēma.
 Atgrieztā vērtība: *vesels skaitlis*.
 Atbalstītās platformas: Linux, FreeBSD, MacOS X, OpenBSD, NetBSD.

kernel.maxproc

 Maksimālais procesu skaits, ko atbalsta operētājsistēma.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux 2.6 un jaunākas versijas, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

kernel.openfiles

 Pašlaik atvērto failu deskriptoru skaits.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux (vienums var darboties arī citās UNIX tipa platformās).

log[file,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Žurnālfaila uzraudzība.
 Atgrieztā vērtība: *Log*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - žurnālfaila pilns ceļš un nosaukums;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu;
- **encoding** - kodlapas identifikators;
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveris vai starpniekserveris. Šis parametrs pārraksta vērtību 'MaxLinesPerSecond' failā zabbix_agentd.conf;
- **mode** - iespējamās vērtības: *all* (noklusējums) vai *skip* - izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem);
- **output** - neobligāta izvades formatēšanas veidne. Aizvietošanas secība **\0** tiek aizstāta ar atbilstošo teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt aizvietošanas secība **\N** (kur N=1...9) tiek aizstāta ar N-to atbilstošo grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu);
- **maxdelay** - maksimālā aizkave sekundēs. Tips: float. Vērtības: 0 - (noklusējums) nekad neignorēt žurnālfaila rindas; > 0.0 - ignorēt vecākas rindas, lai nodrošinātu jaunāko rindu analīzi "maxdelay" sekunžu laikā. Pirms lietošanas izlasiet piezīmes par maxdelay!
- **options** - papildu opcijas:
mtime-noreread - neunikāli ieraksti, pārlasīt tikai tad, ja mainās faila izmērs (ignorēt modifikācijas laika izmaiņas). (Šis parametrs ir novecojis kopš 5.0.2, jo tagad mtime tiek ignorēts.)

- **persistent dir** (tikai zabbix_agentd Unix sistēmās; netiek atbalstīts Zabbix aģents 2) - direktorijas absolūtais ceļš, kur glabāt persistentos failus. Skatiet arī papildu piezīmes par persistentajiem failiem.

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei.
- Ja fails nav pieejams vai piekļuves tiesības neļauj tam piekļūt, vienums kļūst neatbalstīts.
- Ja `output` ir atstāts tukšs, tiek atgriezta visa rinda, kas satur atbilstošo tekstu. Ņemiet vērā, ka visi globālo regulāro izteiksmju tipi, izņemot 'Result is TRUE', vienmēr atgriež visu atbilstošo rindu, un parametrs `output` tiek ignorēts.
- Satura izvilkšana, izmantojot parametru `output`, notiek aģentā.

Piemēri:

```
log[/var/log/syslog]
log[/var/log/syslog,error]
log[/home/zabbix/logs/logfile,,,100]
```

Piemērs, kā izmantot parametru `output`, lai izvilktu skaitli no žurnāla ieraksta:

```
log[/app1/app.log,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #šis vienums at
```

Piemērs, kā izmantot parametru `output`, lai pārrakstītu žurnāla ierakstu pirms nosūtīšanas uz serveris:

```
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>]

 Atbilstošo rindu skaits uzraudzītā žurnālfailā.
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** — pilns žurnālfaila ceļš un nosaukums;
- **regexp** — regulāra izteiksme, kas apraksta nepieciešamo paraugu;
- **encoding** — kodlapas identifikators;
- **maxproclines** — maksimālais jauno rindu skaits sekundē, ko aģents analizēs (nedrīkst pārsniegt 10000). Noklusējuma vērtība ir 10*'MaxLinesPerSecond' failā zabbix_agentd.conf.
- **mode** — iespējamās vērtības: *all* (noklusējums) vai *skip* — izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem).
- **maxdelay** — maksimālā aizkave sekundēs. Tips: float. Vērtības: 0 — (noklusējums) nekad neignorēt žurnālfaila rindas; > 0.0 — ignorēt vecākas rindas, lai nodrošinātu, ka jaunākās rindas tiek analizētas "maxdelay" sekunžu laikā. Pirms lietošanas izlasiet piezīmes par maxdelay!
- **options** — papildu opcijas:
mtime-noreread — neunikāli ieraksti, pārlasīt tikai tad, ja mainās faila izmērs (ignorēt modifikācijas laika izmaiņas). (Šis parametrs ir novecojis kopš 5.0.2, jo tagad mtime tiek ignorēts.)
- **persistent dir** (tikai zabbix_agentd Unix sistēmās; netiek atbalstīts Zabbix aģentā 2) — direktorijas absolūtais ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei.
- Atbilstošās rindas tiek skaitītas starp jaunajām rindām kopš pēdējās žurnāla pārbaudes, ko veicis aģents, tādēļ tas ir atkarīgs no vienuma atjaunināšanas intervāla.
- Ja fails nav pieejams vai piekļuves tiesības neļauj tam piekļūt, vienums kļūst neatbalstīts.

logrt[file regexp,<regexp>,<encoding>,<maxlines>,<mode>,<output>,<maxdelay>,<options>,<persistent dir>]

 Rotēta žurnālfaila uzraudzība.
 Atgrieztā vērtība: *Log*.
 Skatiet atbalstītās platformas.

Parametri:

- **file regexp** - absolūtais ceļš uz failu, kur faila nosaukums ir norādīts, izmantojot regulāru izteiksmi. Ņemiet vērā, ka regulārā izteiksme attiecas tikai uz faila nosaukumu un tai nav jāsakrīt ar visu nosaukumu pilnībā (piemēram, /path/to/agent atbildīs zabbix_agentd.log).
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo satura šablonu.
- **encoding** - kodlapas identifikators.
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveris vai starpniekserveris. Šis parametrs ignorē vērtību 'MaxLinesPerSecond' failā zabbix_agentd.conf.
- **mode** - iespējamās vērtības: *all* (noklusējums) vai *skip* - izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem).
- **output** - neobligāta izvades formatēšanas veidne. Aizvietošanas secība **\0** tiek aizstāta ar atbilstošo teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt aizvietošanas secība **\N** (kur N=1...9) tiek aizstāta ar N-to atrasto grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu).

- **maxdelay** - maksimālā aizkave sekundēs. Tips: float. Vērtības: 0 - (noklusējums) nekad neignorēt žurnālfaila rindas; > 0.0 - ignorēt vecākas rindas, lai nodrošinātu, ka jaunākās rindas tiek analizētas ”maxdelay” sekunžu laikā. Pirms lietošanas izlasiet piezīmes par maxdelay!
- **options** - žurnālfaila rotācijas tips un citas opcijas. Iespējamās vērtības:
rotate (noklusējums),
copytruncate - ņemiet vērā, ka *copytruncate* nevar izmantot kopā ar *maxdelay*. Šajā gadījumā *maxdelay* jābūt 0 vai arī tas nav jānorāda; skatiet piezīmes par copytruncate,
mtime-reread - neunikāli ieraksti, pārlasīt, ja mainās modifikācijas laiks vai izmērs (tiek izmantots pēc noklusējuma, ja neviena `mtime-*` opcija nav skaidri iestatīta),
mtime-noreread - neunikāli ieraksti, pārlasīt tikai tad, ja mainās izmērs (ignorēt modifikācijas laika izmaiņas).
- **persistent dir** (tikai zabbix_agentd Unix sistēmās; netiek atbalstīts Zabbix aģents 2) - absolūtais direktorija ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei.
- Žurnālfailu rotācija ir balstīta uz failu pēdējās modifikācijas laiku.
- Ņemiet vērā, ka logrt ir paredzēts darbam ar vienu pašlaik aktīvu žurnālfailu un vairākiem citiem atbilstošiem neaktīviem rotētiem failiem. Ja, piemēram, direktorijā ir daudz aktīvu žurnālfailu, katram no tiem jāizveido atsevišķs logrt vienums. Pretējā gadījumā, ja viens logrt vienums aptver pārāk daudz failu, tas var izraisīt atmiņas izsīkumu un uzraudzības avāriju.
- Ja `output` ir atstāts tukšs, tiek atgriezta visa rinda, kas satur atrasto tekstu. Ņemiet vērā, ka visi globālo regulāro izteiksmju tipi, izņemot ’Result is TRUE’, vienmēr atgriež visu atbilstošo rindu, un parametrs `output` tiek ignorēts.
- Satura izvilkšana, izmantojot parametru `output`, notiek aģentā.
- Parametrā `file regexp` žurnālu direktorija ceļam un žurnālfaila regulārajai izteiksmei jābūt atdalītiem ar pareizo direktoriju atdalītāju:
  - Windows sistēmā atdalītājam jābūt atpakaļvērstajai slīpsvītrai (\); parastās slīpsvītras var tikt pieļautas citās pozīcijās, izņemot to, kas atdala žurnālu direktorija ceļu un žurnālfaila regulāro izteiksmi (skatiet piemērus zemāk).
  - Citās sistēmās tam jābūt parastajai slīpsvītrai (/).

Piemēri Windows sistēmām:

```
logrt["c:/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #šis vienums vāks datus no žurnālfailiem dir
logrt["//example.com/share/dir1/dir2/dir3\filename.*\.log","pattern_to_match"] #šis vienums vāks datus no
```

Piemēri citām sistēmām:

```
logrt["/home/zabbix/logs/^logfile[0-9]{1,3}$",,,100] #šis vienums atbildīs failam, piemēram, "logfile1" (n
logrt["/home/user/^logfile_.*_[0-9]{1,3}$","pattern_to_match","UTF-8",100] #šis vienums vāks datus no fail
```

Piemērs, kā izmantot parametru `output`, lai izvilktu skaitli no žurnāla ieraksta:

```
logrt[/app1/^test.*log$,"task run [0-9.]+ sec, processed ([0-9]+) records, [0-9]+ errors",,,,\1] #šis vien
```

Piemērs, kā izmantot parametru `output`, lai pārrakstītu žurnāla ierakstu pirms nosūtīšanas uz serveris:

```
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>]

 Atbilstošo rindu skaits uzraudzītā žurnālfailā, kuram tiek veikta rotācija.
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **file regexp** — absolūtais ceļš uz failu, kur faila nosaukums ir norādīts, izmantojot regulāru izteiksmi. Ņemiet vērā, ka regulārā izteiksme attiecas tikai uz faila nosaukumu un tai nav jāsakrīt ar visu nosaukumu pilnībā (piemēram, /path/to/agent atbildīs zabbix_agentd.log).
- **regexp** — regulāra izteiksme, kas apraksta nepieciešamo paraugu.
- **encoding** — kodlapas identifikators.
- **maxproclines** — maksimālais jauno rindu skaits sekundē, ko aģents analizēs (nedrīkst pārsniegt 10000). Noklusējuma vērtība ir 10*’MaxLinesPerSecond’ failā zabbix_agentd.conf.
- **mode** — iespējamās vērtības: *all* (noklusējums) vai *skip* — izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem).
- **maxdelay** — maksimālā aizkave sekundēs. Tips: float. Vērtības: 0 — (noklusējums) nekad neignorēt žurnālfaila rindas; > 0.0 — ignorēt vecākas rindas, lai nodrošinātu, ka jaunākās rindas tiek analizētas ”maxdelay” sekunžu laikā. Pirms lietošanas izlasiet piezīmes par maxdelay!
- **options** — žurnālfaila rotācijas veids un citas opcijas. Iespējamās vērtības:
rotate (noklusējums),
copytruncate — ņemiet vērā, ka *copytruncate* nevar izmantot kopā ar *maxdelay*. Šajā gadījumā *maxdelay* jābūt 0 vai arī tas nav jānorāda; skatiet piezīmes par copytruncate,
mtime-reread — neunikāli ieraksti, pārlasīt, ja mainās modifikācijas laiks vai izmērs (noklusējums),
mtime-noreread — neunikāli ieraksti, pārlasīt tikai tad, ja mainās izmērs (ignorēt modifikācijas laika izmaiņas).

- **persistent dir** (tikai zabbix_agentd Unix sistēmās; netiek atbalstīts Zabbix aģentā 2) — absolūtais direktorija ceļš, kur glabāt pastāvīgos failus. Skatiet arī papildu piezīmes par pastāvīgajiem failiem.

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvai pārbaudei.
- Atbilstošās rindas tiek skaitītas starp jaunajām rindām kopš pēdējās žurnāla pārbaudes, ko veicis aģents, un tādēļ ir atkarīgas no vienuma atjaunināšanas intervāla.
- Žurnāla rotācija ir balstīta uz failu pēdējās modifikācijas laiku.
- Parametrā `file regexp` žurnāla direktorija ceļš un žurnālfaila regulārā izteiksme jāatdala ar pareizo direktoriju atdalītāju:
  - Windows sistēmā atdalītājam jābūt apgrieztajai slīpsvītrai (\); parastās slīpsvītras citās pozīcijās var tikt pieļautas, izņemot to, kas atdala žurnāla direktorija ceļu un žurnālfaila regulāro izteiksmi (skatiet piemērus zemāk).
  - Citās sistēmās tam jābūt parastajai slīpsvītrai (/).

Piemēri Windows sistēmai:

```
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
```

Piemēri citām sistēmām:

```
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>]

 Nolasa Modbus datus.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, Windows.

Parametri:

- **endpoint** - galapunkts, kas definēts kā `protocol://connection_string`;
- **slave id** - pakārtotās ierīces ID;
- **function** - Modbus funkcija;
- **address** - pirmā reģistra, spoles vai ievades adrese;
- **count** - nolasāmo ierakstu skaits;
- **type** - datu tips;
- **endianness** - baitu secības konfigurācija;
- **offset** - reģistru skaits, sākot no 'address', kuru rezultāti tiks atmesti.

Sistēmā Windows šis vienums seko UNC ceļiem parametrā **endpoint**, kas var radīt drošības risku.

Skatiet parametru detalizētu aprakstu.

net.dns[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Pārbauda DNS pakalpojuma statusu.
 Atgrieztās vērtības: 0 - DNS atrisināšana neizdevās (DNS serveris neatbildēja vai atgrieza kļūdu); 1 - DNS atrisināšana izdevās.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** - DNS nosaukums, kuram veikt vaicājumu;
- **type** - ieraksta tips, kuram tiks veikts vaicājums (noklusējums ir *SOA*);
- **timeout** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma mēģinājumu skaits (noklusējums ir 2);
- **protocol** - protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējums) vai *tcp*.

Komentāri:

- Iespējamās `type` vērtības ir: *ANY*, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (netiek atbalstīts Zabbix aģentam Windows vidē, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Reversajiem DNS vaicājumiem (ja `type` ir iestatīts uz *PTR*) DNS nosaukumu var norādīt gan apgrieztā, gan neapgrieztā formātā (skatiet tālāk esošos piemērus). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums patiesībā ir IP adrese.
- Internacionalizētie domēnu nosaukumi netiek atbalstīti; lūdzu, izmantojiet ar IDNA kodētus nosaukumus.

Piemēri:

```
net.dns[198.51.100.1,example.com,MX,2,1]
```

```
net.dns[,198.51.100.1,PTR]
net.dns[,1.100.51.198.in-addr.arpa,PTR]

net.dns[,2a00:1450:400f:800::200e,PTR]
net.dns[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.perf[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Pārbauda DNS servisa veiktspēju.
 Atgrieztā vērtība: *Float* (0 - serviss nav pieejams; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no servisa).
 Skatiet atbalstītās platformas.

Parametri:

- **ip** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** - DNS nosaukums, kuram veikt vaicājumu;
- **type** - ieraksta tips, kuram tiks veikts vaicājums (noklusējums ir *SOA*);
- **timeout** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) - pieprasījuma mēģinājumu skaits (noklusējums ir 2);
- **protocol** - protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējums) vai *tcp*.

Komentāri:

- Iespējamās `type` vērtības ir:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (netiek atbalstīts Zabbix aģentam Windows vidē, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Reversajiem DNS uzmeklējumiem (ja `type` ir iestatīts uz *PTR*) DNS nosaukumu var norādīt gan apgrieztā, gan neapgrieztā formātā (skatiet piemērus zemāk). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums faktiski ir IP adrese.
- Internacionalizēti domēnu nosaukumi netiek atbalstīti; tā vietā, lūdzu, izmantojiet IDNA kodētus nosaukumus.
- Vienums atgriež atbildes laiku 0 vietā, ja DNS serveris atbild ar kļūdas kodu (piemēram, `NXDOMAIN` vai `SERVFAIL`).

Piemēri:

```
net.dns.perf[198.51.100.1,example.com,MX,2,1]

net.dns.perf[,198.51.100.1,PTR]
net.dns.perf[,1.100.51.198.in-addr.arpa,PTR]

net.dns.perf[,2a00:1450:400f:800::200e,PTR]
net.dns.perf[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.dns.record[<ip>,name,<type>,<timeout>,<count>,<protocol>]

 Veic DNS vaicājumu.

Zabbix aģents 2 nodrošina arī net.dns.get, kas piedāvā papildu iespējas, piemēram, vairāk ierakstu tipu un plašāku DNS uzraudzības kontroli. Atgrieztā vērtība: rakstzīmju virkne ar nepieciešamā tipa informāciju.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) — DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** — DNS nosaukums, kuram veikt vaicājumu;
- **type** — vaicājamā ieraksta tips (noklusējums ir *SOA*);
- **timeout** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) — pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** (tiek ignorēts Windows vidē, ja vien netiek izmantots Zabbix aģents 2) — pieprasījuma mēģinājumu skaits (noklusējums ir 2);
- **protocol** — protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējums) vai *tcp*.

Komentāri:

- Iespējamās `type` vērtības ir:
ANY, *A*, *NS*, *CNAME*, *MB*, *MG*, *MR*, *PTR*, *MD*, *MF*, *MX*, *SOA*, *NULL*, *WKS* (netiek atbalstīts Zabbix aģentam Windows vidē, Zabbix aģentam 2 visās OS), *HINFO*, *MINFO*, *TXT*, *SRV*.
- Reversajiem DNS vaicājumiem (ja `type` ir iestatīts uz *PTR*) DNS nosaukumu var norādīt apgrieztā vai neapgrieztā formātā (skatiet piemērus zemāk). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums faktiski ir IP adrese.
- Internacionalizētie domēnu nosaukumi netiek atbalstīti; tā vietā izmantojiet IDNA kodētus nosaukumus.

Piemēri:

```
net.dns.record[198.51.100.1,example.com,MX,2,1]

net.dns.record[,198.51.100.1,PTR]
net.dns.record[,1.100.51.198.in-addr.arpa,PTR]

net.dns.record[,2a00:1450:400f:800::200e,PTR]
net.dns.record[,e.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.0.f.0.0.4.0.5.4.1.0.0.a.2.ip6.arpa,PTR]
```

net.if.collisions[if]

 Ārpus loga sadursmju skaits.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD. NetBSD ir nepieciešamas root privilēģijas.

Parametrs:

- **if** - tīkla saskarnes nosaukums

net.if.discovery

 Tīkla saskarņu saraksts. Tiek izmantots zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

net.if.in[if,<mode>]

 Ienākošās datplūsmas statistika tīkla saskarnē.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD sistēmā ir nepieciešamas root privilēģijas.

Parametri:

- **if** — tīkla saskarnes nosaukums (Unix); pilns tīkla saskarnes apraksts vai IPv4 adrese; vai, ja iekavās, tīkla saskarnes GUID (Windows);
- **mode** — iespējamās vērtības:
bytes — baitu skaits (noklusējums)
packets — pakešu skaits
errors — kļūdu skaits
dropped — nomesto pakešu skaits
overruns (fifo) — FIFO bufera kļūdu skaits
frame — pakešu kadrēšanas kļūdu skaits
compressed — ierīces draivera saņemto saspiesto pakešu skaits
multicast — ierīces draivera saņemto multiraides kadru skaits

Komentāri:

- Jūs varat izmantot šo atslēgu kopā ar *Change per second* priekšapstrādes soli, lai iegūtu statistiku baitos sekundē.
- Režīms *dropped* tiek atbalstīts tikai Linux, FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Režīmi *overruns*, *frame*, *compressed*, *multicast* tiek atbalstīti tikai Linux.
- HP-UX sistēmā šis vienums nesniedz detalizētu informāciju par loopback saskarnēm (piemēram, lo0).

Piemēri:

```
net.if.in[eth0]
net.if.in[eth0,errors]
```

net.if.out[if,<mode>]

 Izejošās datplūsmas statistika tīkla saskarnē.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD sistēmā ir nepieciešamas root privilēģijas.

Parametri:

- **if** - tīkla saskarnes nosaukums (Unix); tīkla saskarnes pilns apraksts vai IPv4 adrese; vai, ja iekavās, tīkla saskarnes GUID (Windows);
- **mode** - iespējamās vērtības:
bytes - baitu skaits (noklusējums)
packets - pakešu skaits
errors - kļūdu skaits
dropped - nomesto pakešu skaits
overruns (fifo) - FIFO bufera kļūdu skaits
collisions (colls) - saskarnē konstatēto sadursmju skaits
carrier - ierīces draivera konstatēto nesēja zudumu skaits
compressed - ierīces draivera nosūtīto saspiesto pakešu skaits

Komentāri:

- Jūs varat izmantot šo atslēgu ar *Change per second* priekšapstrādes soli, lai iegūtu statistiku baitos sekundē.
- Režīms *dropped* tiek atbalstīts tikai Linux un HP-UX.
- Režīmi *overruns*, *collision*, *carrier*, *compressed* tiek atbalstīti tikai Linux.
- HP-UX sistēmā šis vienums nesniedz detalizētu informāciju par loopback saskarnēm (piemēram, lo0).

Piemēri:

```
net.if.out[eth0]
net.if.out[eth0,errors]
```

net.if.total[if,<mode>]

 Ienākošās un izejošās datplūsmas statistikas summa tīkla saskarnē.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[5], HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows. NetBSD ir nepieciešamas root privilēģijas.

Parametri:

- **if** - tīkla saskarnes nosaukums (Unix); pilns tīkla saskarnes apraksts vai IPv4 adrese; vai, ja iekavās, tīkla saskarnes GUID (Windows);
- **mode** - iespējamās vērtības:
bytes - baitu skaits (noklusējums)
packets - pakešu skaits
errors - kļūdu skaits
dropped - nomesto pakešu skaits
overruns (fifo) - FIFO bufera kļūdu skaits
collisions (colls) - saskarnē konstatēto sadursmju skaits
compressed - ierīces draivera nosūtīto vai saņemto saspiesto pakešu skaits

Komentāri:

- Jūs varat izmantot šo atslēgu ar priekšapstrādes soli *Change per second*, lai iegūtu baitu sekundē statistiku.
- Režīms *dropped* tiek atbalstīts tikai Linux, HP-UX. Nomestās paketes tiek atbalstītas tikai tad, ja jūsu platformā gan `net.if.in`, gan `net.if.out` darbojas nomestajām paketēm.
- Režīmi *overruns*, *collision*, *compressed* tiek atbalstīti tikai Linux.
- HP-UX šis vienums nesniedz detalizētu informāciju par loopback saskarnēm (piemēram, lo0).

Piemēri:

```
net.if.total[eth0]
net.if.total[eth0,errors]
```

net.tcp.listen[port]

 Pārbauda, vai šis TCP ports ir LISTEN stāvoklī.
 Atgrieztās vērtības: 0 - tas nav LISTEN stāvoklī; 1 - tas ir LISTEN stāvoklī.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, MacOS X, Windows.

Parametrs:

- **port** - TCP porta numurs

Linux kodolos 2.6.14 un jaunākās versijās informācija par TCP ligzdām LISTEN stāvoklī tiek iegūta no kodola NETLINK saskarnes, ja iespējams. Pretējā gadījumā informācija tiek iegūta no failiem /proc/net/tcp un /roc/net/tcp6.

Piemērs:

```
net.tcp.listen[80]
```

net.tcp.port[<ip>,port]

 Pārbauda, vai ir iespējams izveidot TCP savienojumu ar norādīto portu.
 Atgrieztās vērtības: 0 - nevar izveidot savienojumu; 1 - var izveidot savienojumu.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** - IP adrese vai DNS nosaukums (noklusējuma vērtība ir 127.0.0.1);
- **port** - porta numurs.

Komentāri:

- Vienkāršai TCP veiktspējas testēšanai izmantojiet `net.tcp.service.perf[tcp,<ip>,<port>]`.
- Šīs pārbaudes var radīt papildu ziņojumus sistēmas dēmonu žurnālfailos (parasti tiek reģistrētas SMTP un SSH sesijas).

Piemērs:

```
net.tcp.port[,80] #šo vienumu var izmantot, lai pārbaudītu tīmekļa servera pieejamību, kas darbojas 80. po
```

net.tcp.service[service,<ip>,<port>]

 Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus.
 Atgrieztās vērtības: 0 - pakalpojums nedarbojas; 1 - pakalpojums darbojas.
 Skatiet atbalstītās platformas.

Parametri:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* vai *telnet* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (noklusējuma vērtība ir 127.0.0.1);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Komentāri:

- Šo pārbaužu rezultātā sistēmas dēmona žurnālfailos var parādīties papildu ziņojumi (parasti tiek reģistrētas SMTP un SSH sesijas).

- Šobrīd šifrētu protokolu pārbaude (piemēram, IMAP portā 993 vai POP portā 995) netiek atbalstīta. Kā pagaidu risinājumu šādām pārbaudēm izmantojiet `net.tcp.port[]`.
- LDAP un HTTPS pārbaude operētājsistēmā Windows tiek atbalstīta tikai ar Zabbix aģents 2.
- Telnet pārbaude meklē pieteikšanās uzvedni (':' beigās).

Piemērs:

```
net.tcp.service[ftp,,45] #šo vienums var izmantot, lai pārbaudītu FTP servera pieejamību TCP portā 45
```

net.tcp.service.perf[service,<ip>,<port>]

 Pārbauda TCP servisa veiktspēju.
 Atgrieztās vērtības: *Float* (0 - serviss nav pieejams; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no servisa).
 Skatiet atbalstītās platformas.

Parametri:

- **service** - *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https* vai *telnet* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (noklusējuma vērtība ir 127.0.0.1);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta servisa porta numurs).

Komentāri:

- Šifrētu protokolu (piemēram, IMAP portā 993 vai POP portā 995) pārbaude pašlaik netiek atbalstīta. Kā risinājumu, lūdzu, izmantojiet `net.tcp.service.perf[tcp,<ip>,<port>]` šādām pārbaudēm.
- Telnet pārbaude meklē pieteikšanās uzvedni (':' beigās).

Piemērs:

```
net.tcp.service.perf[ssh] #šo vienumu var izmantot, lai pārbaudītu sākotnējās atbildes ātrumu no SSH serve
```

net.tcp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Atgriež TCP ligzdu skaitu, kas atbilst parametriem.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, Windows.

Parametri:

- **laddr** — lokālā IPv4/6 adrese vai CIDR apakštīkls;
- **lport** — lokālā porta numurs vai servisa nosaukums;
- **raddr** — attālā IPv4/6 adrese vai CIDR apakštīkls;
- **rport** — attālā porta numurs vai servisa nosaukums;
- **state** — savienojuma stāvoklis (*established*, *syn_sent*, *syn_recv*, *fin_wait1*, *fin_wait2*, *time_wait*, *close*, *close_wait*, *last_ack*, *listen*, *closing*).

Piemērs:

```
net.tcp.socket.count[,80,,,established] #savienojumu skaits ar lokālo TCP portu 80 established stāvoklī
```

net.udp.listen[port]

 Pārbauda, vai šis UDP ports ir LISTEN stāvoklī.
 Atgrieztās vērtības: 0 - tas nav LISTEN stāvoklī; 1 - tas ir LISTEN stāvoklī.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, MacOS X.

Parametrs:

- **port** - UDP porta numurs

Piemērs:

```
net.udp.listen[68]
```

net.udp.service[service,<ip>,<port>]

 Pārbauda, vai pakalpojums darbojas un atbild uz UDP pieprasījumiem.
 Atgrieztās vērtības: 0 - pakalpojums nav pieejams; 1 - pakalpojums darbojas.
 Skatiet atbalstītās platformas.

Parametri:

- **service** - *ntp* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (noklusējuma vērtība ir 127.0.0.1);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Piemērs:

```
net.udp.service[ntp,,45] #šo vienumu var izmantot, lai pārbaudītu NTP pakalpojuma pieejamību UDP 45. portā
```

net.udp.service.perf[service,<ip>,<port>]

 Pārbauda UDP pakalpojuma veiktspēju.
 Atgrieztās vērtības: *Float* (0 - pakalpojums nav pieejams; sekundes - sekunžu skaits, kas pavadīts, gaidot atbildi no pakalpojuma).
 Skatiet atbalstītās platformas.

Parametri:

- **service** - *ntp* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (noklusējuma vērtība ir 127.0.0.1);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Piemērs:

```
net.udp.service.perf[ntp] #šo vienumu var izmantot, lai pārbaudītu atbildes laiku no NTP pakalpojuma
```

net.udp.socket.count[<laddr>,<lport>,<raddr>,<rport>,<state>]

 Atgriež UDP ligzdu skaitu, kas atbilst parametriem.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, Windows.

Parametri:

- **laddr** — lokālā IPv4/6 adrese vai CIDR apakštīkls;
- **lport** — lokālā porta numurs vai servisa nosaukums;
- **raddr** — attālā IPv4/6 adrese vai CIDR apakštīkls;
- **rport** — attālā porta numurs vai servisa nosaukums;
- **state** — savienojuma stāvoklis (*established*, *unconn*).

Piemērs:

```
net.udp.socket.count[,,,,established] #atgriež UDP ligzdu skaitu savienotā stāvoklī
```

proc.cpu.util[<name>,<user>,<type>,<cmdline>,<mode>,<zone>]

 Procesa CPU izmantošanas procents.
 Atgrieztā vērtība: *Float*.
 Atbalstītās platformas: Linux, Solaris[6].

Parametri:

- **name** - procesa nosaukums (noklusējums ir *visi procesi*);
- **user** - lietotāja vārds (noklusējums ir *visi lietotāji*);
- **type** - CPU izmantošanas tips: *total* (noklusējums), *user* vai *system*;
- **cmdline** - filtrēšana pēc komandrindas (tā ir regulāra izteiksme);
- **mode** - datu iegūšanas režīms: *avg1* (noklusējums), *avg5* vai *avg15*;
- **zone** - mērķa zona: *current* (noklusējums) vai *all*. Šis parametrs tiek atbalstīts tikai Solaris.

Komentāri:

- Atgrieztā vērtība ir balstīta uz viena CPU kodola izmantošanas procentu. Piemēram, procesa CPU izmantošana, kas pilnībā izmanto divus kodolus, ir 200%.
- Procesa CPU izmantošanas dati tiek iegūti ar kolektoru, kas atbalsta ne vairāk kā 1024 unikālus vaicājumus (pēc nosaukuma, lietotāja un komandrindas). Vaicājumi, kuriem nav piekļūts pēdējo 24 stundu laikā, tiek noņemti no kolektora.
- Ja parametrs zone ir iestatīts uz *current* (vai pēc noklusējuma) gadījumā, ja aģents ir kompilēts Solaris vidē bez zonu atbalsta, bet darbojas jaunākā Solaris versijā, kur zonas tiek atbalstītas, tad aģents atgriezīs NOTSUPPORTED (aģents nevar ierobežot rezultātus tikai ar pašreizējo zonu). Tomēr šajā gadījumā tiek atbalstīts *all*.

Piemēri:

```
proc.cpu.util[,root] #Visu procesu CPU izmantošana, kas darbojas zem "root" lietotāja
proc.cpu.util[zabbix_server,zabbix] #Visu zabbix_server procesu CPU izmantošana, kas darbojas zem zabbix l
```

proc.get[<name>,<user>,<cmdline>,<mode>]

 OS procesu un to parametru saraksts. Var izmantot zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, FreeBSD, Windows, OpenBSD, NetBSD.

Parametri:

- **name** - procesa nosaukums (noklusējums *visi procesi*);
- **user** - lietotāja vārds (noklusējums *visi lietotāji*);
- **cmdline** - filtrēšana pēc komandrindas (tā ir regulāra izteiksme). Šis parametrs netiek atbalstīts Windows; citās platformās tas netiek atbalstīts, ja režīmam ir iestatīta vērtība 'summary'.
- **mode** - iespējamās vērtības:
process (noklusējums), *thread* (NetBSD netiek atbalstīts), *summary*. Skatiet procesa parametru sarakstu, kas tiek atgriezti katram režīmam un OS.

Komentāri:

- Ja vērtību nevar iegūt, piemēram, kļūdas dēļ (process jau ir beidzies, trūkst atļauju, sistēmas izsaukuma kļūme), tiks atgriezts -1.
- Skatiet piezīmes par procesu atlasīšanu ar parametriem `name` un `cmdline` (attiecas tikai uz Linux).

Piemēri:

```
proc.get[zabbix_server,zabbix,,process] #visu zabbix_server procesu saraksts, kas darbojas zabbix lietotāj
proc.get[java,,,thread] #visu Java procesu saraksts; tiek atgriezts viens ieraksts katram pavedienam
proc.get[,zabbix,,summary] #apvienotie dati par katra tipa procesiem, kas darbojas zabbix lietotāja vārdā;
```

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

 Procesa izmantotā atmiņa baitos.
 Atgrieztā vērtība: *Integer* - ja `mode` ir *max*, *min*, *sum*; *Float* - ja `mode` ir *avg*
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

Parametri:

- **name** - procesa nosaukums (noklusējums ir *visi procesi*);
- **user** - lietotāja vārds (noklusējums ir *visi lietotāji*);
- **mode** - iespējamās vērtības: *avg*, *max*, *min* vai *sum* (noklusējums);
- **cmdline** - filtrēšana pēc komandrindas (tā ir regulāra izteiksme);
- **memtype** - procesa izmantotais atmiņas tips

Komentāri:

- Parametrs `memtype` tiek atbalstīts tikai Linux, FreeBSD, Solaris[6] un AIX.
- Ja vairāki procesi izmanto koplietojamo atmiņu, procesu izmantotās atmiņas summa var radīt lielas, nereālistiskas vērtības.

Skatiet piezīmes par procesu atlasi ar parametriem `name` un `cmdline` (attiecas tikai uz Linux).

Ja šis vienums tiek izsaukts no komandrindas un satur komandrindas parametru (piemēram, izmantojot aģenta testa režīmu: `zabbix_agentd -t proc.mem[,,,apache2]`), tiks ieskaitīts viens papildu process, jo aģents ieskaitīs pats sevi.

Piemēri:

```
proc.mem[,root] #atmiņa, ko izmanto visi procesi, kas darbojas ar "root" lietotāju
proc.mem[zabbix_server,zabbix] #atmiņa, ko izmanto visi zabbix_server procesi, kas darbojas ar zabbix liet
proc.mem[,oracle,max,oracleZABBIX] #atmiņa, ko izmanto process ar lielāko atmiņas patēriņu, kas darbojas a
```

proc.num[<name>,<user>,<state>,<cmdline>,<zone>]

 Procesu skaits.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris[6], HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parametri:

- **name** — procesa nosaukums (noklusējums ir *visi procesi*);
- **user** — lietotāja vārds (noklusējums ir *visi lietotāji*);
- **state** — iespējamās vērtības:
all (noklusējums),
disk — nepārtraucams miega stāvoklis,
run — darbojas,
sleep — pārtraucams miega stāvoklis,
trace — apturēts,
zomb — zombijs;
- **cmdline** — filtrēšana pēc komandrindas (tā ir regulāra izteiksme);
- **zone** — mērķa zona: *current* (noklusējums) vai *all*. Šis parametrs tiek atbalstīts tikai Solaris.

Komentāri:

- Stāvokļa parametri *disk* un *trace* tiek atbalstīti tikai Linux, FreeBSD, OpenBSD, NetBSD.
- Windows vidē tiek atbalstīti tikai parametri `name` un `user`.
- Ja šis vienums tiek izsaukts no komandrindas un satur komandrindas parametru (piemēram, izmantojot aģents testa režīmu: `zabbix_agentd -t proc.num[,,,apache2]`), tiks uzskaitīts par vienu procesu vairāk, jo aģents uzskaitīs arī pats sevi.
- Ja parametrs `zone` ir iestatīts uz `current` (vai pēc noklusējuma) gadījumā, kad aģents ir kompilēts Solaris sistēmā bez zonu atbalsta, bet darbojas jaunākā Solaris versijā, kur zonas tiek atbalstītas, tad aģents atgriezīs NOTSUPPORTED (aģents nevar ierobežot rezultātus tikai ar pašreizējo zonu). Tomēr šajā gadījumā tiek atbalstīts *all*.
- Skatiet piezīmes par procesu atlasi ar parametriem `name` un `cmdline` (attiecas tikai uz Linux).

Piemēri:

```
proc.num[,mysql] #procesu skaits, kas darbojas mysql lietotāja vārdā
proc.num[apache2,www-data] #apache2 procesu skaits, kas darbojas www-data lietotāja vārdā
proc.num[,oracle,sleep,oracleZABBIX] #procesu skaits miega stāvoklī, kas darbojas Oracle lietotāja vārdā u
```

sensor[device,sensor,<mode>]

 Aparatūras sensora rādījums.
 Atgrieztā vērtība: *Float*.
 Atbalstītās platformas: Linux, OpenBSD.

Parametri:

- **device** - ierīces nosaukums;
- **sensor** - sensora nosaukums;
- **mode** - iespējamās vērtības: *avg*, *max* vai *min*.

Komentāri:

- Nolasa /proc/sys/dev/sensors Linux 2.4 vidē.
- Nolasa /sys/class/hwmon Linux 2.6+ vidē. Skatiet detalizētāku sensor vienuma aprakstu Linux vidē.
- Nolasa *hw.sensors* MIB OpenBSD vidē.
- Linux 2.4 vidē, ja norādāt mode, tad device un sensor tiek interpretēti kā regulārās izteiksmes.
- Linux 2.6+ un OpenBSD vidē vai, ja mode nav norādīts, device un sensor tiek interpretēti burtiski.

Piemēri:

```
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

 Sistēmas sāknēšanas laiks.
 Atgrieztā vērtība: *Integer (Unix timestamp)*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, MacOS X, OpenBSD, NetBSD.

system.cpu.discovery

 Atklāto CPU/CPU kodolu saraksts. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

system.cpu.intr

 Ierīces pārtraukumi.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.load[<cpu>,<mode>]

 CPU slodze.
 Atgrieztā vērtība: *Float*.
 Skatiet atbalstītās platformas.

Parametri:

- **cpu** — iespējamās vērtības: *all* (noklusējums) vai *percpu* (kopējā slodze dalīta ar tiešsaistē esošo CPU skaitu);
- **mode** — iespējamās vērtības: *avg1* (vienas minūtes vidējā vērtība, noklusējums), *avg5* vai *avg15*.

Piemērs:

```
system.cpu.load[,avg5]
```

system.cpu.num[<type>]

 CPU skaits.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Parametrs:

- **type** — iespējamās vērtības: *online* (noklusējums) vai *max*

Parametrs *max* tiek atbalstīts tikai Linux, FreeBSD, Solaris, MacOS X.

Piemērs:

```
system.cpu.num
```

system.cpu.switches

 Konteksta pārslēgšanu skaits.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD, NetBSD.

system.cpu.util[<cpu>,<type>,<mode>,<logical or physical>]

 CPU izmantošanas procents.
 Atgrieztā vērtība: *Float*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD, Windows.

Parametri:

- **cpu** — <*CPU number*> vai *all* (noklusējums);

- **type** — iespējamās vērtības: *user* (noklusējums), *idle*, *nice*, *system*, *iowait*, *interrupt*, *softirq*, *steal*, *spin* (OpenBSD), *guest* (Linux kodoliem 2.6.24 un jaunākiem) vai *guest_nice* (Linux kodoliem 2.6.33 un jaunākiem);
- **mode** — iespējamās vērtības: *avg1* (vienas minūtes vidējais, noklusējums), *avg5* vai *avg15*;
- **logical or physical** — iespējamās vērtības: *logical* (noklusējums) vai *physical*. Šis parametrs tiek atbalstīts tikai AIX.

Komentāri:

- Parametrs *nice* tipam tiek atbalstīts tikai Linux, FreeBSD, HP-UX, OpenBSD, NetBSD.
- Parametrs *iowait* tipam tiek atbalstīts tikai Linux 2.6 un jaunākās versijās, Solaris, AIX.
- Parametrs *interrupt* tipam tiek atbalstīts tikai Linux 2.6 un jaunākās versijās, FreeBSD, OpenBSD.
- Parametri *softirq*, *steal*, *guest*, *guest_nice* tipam tiek atbalstīti tikai Linux 2.6 un jaunākās versijās.
- Režīma parametri *avg5* un *avg15* tiek atbalstīti Linux, FreeBSD, Solaris, HP-UX, AIX, OpenBSD, NetBSD.

Piemērs:

```
system.cpu.util[0,user,avg5]
```

system.hostname[<type>,<transform>]

 Sistēmas hosta nosaukums.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Parametri:

- **type** — iespējamās vērtības: *netbios* (noklusējums operētājsistēmā Windows), *host* (noklusējums operētājsistēmā Linux), *shorthost* (atgriež hosta nosaukuma daļu pirms pirmā punkta, pilnu virkni nosaukumiem bez punktiem), *fqdn* (atgriež pilnībā kvalificētu domēna nosaukumu);
- **transform** — iespējamās vērtības: *none* (noklusējums) vai *lower* (pārveido uz mazajiem burtiem).

Vērtība tiek iegūta, izmantojot `nodename` no uname() sistēmas API izvades.

Atgriezto vērtību piemēri:

```
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>]

 Šasijas informācija.
 Atgrieztā vērtība: *String*.
 Atbalstītās platformas: Linux.

Parametrs:

- **info** — iespējamās vērtības: *full* (noklusējums), *model*, *serial*, *type* vai *vendor*

Komentāri:

- Šī vienuma atslēga ir atkarīga no SMBIOS tabulas pieejamības.
- Tiks mēģināts nolasīt DMI tabulu no sysfs; ja piekļuve sysfs neizdosies, tad tiks mēģināts nolasīt to tieši no atmiņas.
- Nepieciešamas **root tiesības**, jo vērtība tiek iegūta, nolasot datus no sysfs vai atmiņas.

Piemērs:

```
system.hw.chassis[full] → Hewlett-Packard HP Pro 3010 Small Form Factor PC CZXXXXXXXX Desktop
```

system.hw.cpu[<cpu>,<info>]

 CPU informācija.
 Atgrieztā vērtība: *String* vai *Integer*.
 Atbalstītās platformas: Linux.

Parametri:

- **cpu** — <*CPU numurs*> vai *all* (noklusējums);
- **info** — iespējamās vērtības: *full* (noklusējums), *curfreq*, *maxfreq*, *model* vai *vendor*.

Komentāri:

- Apkopo informāciju no `/proc/cpuinfo` un `/sys/devices/system/cpu/[cpunum]/cpufreq/cpuinfo_max_freq`.
- Ja ir norādīts CPU numurs un *curfreq* vai *maxfreq*, tiek atgriezta skaitliska vērtība (Hz).

Piemērs:

```
system.hw.cpu[0,vendor] → AuthenticAMD
```

system.hw.devices[<type>]

 PCI vai USB ierīču saraksts.
 Atgrieztā vērtība: *Teksts*.
 Atbalstītās platformas: Linux.

Parametrs:

- **type** - *pci* (noklusējums) vai *usb*

Atgriež vai nu lspci, vai lsusb utilītas izvadi (izpildītas bez jebkādiem parametriem).

Piemērs:

```
system.hw.devices → 00:00.0 Host bridge: Advanced Micro Devices [AMD] RS780 Host Bridge
```

system.hw.macaddr[<interface>,<format>]

 MAC adrešu saraksts.
 Atgrieztā vērtība: *String*.
 Atbalstītās platformas: Linux.

Parametri:

- **interface** — *all* (noklusējums) vai regulāra izteiksme;
- **format** — *full* (noklusējums) vai *short*

Komentāri:

- Uzskaita to saskarņu MAC adreses, kuru nosaukums atbilst norādītajai `interface` regulārajai izteiksmei (*all* uzskaita visām saskarnēm).
- Ja `format` ir norādīts kā *short*, saskarņu nosaukumi un vienādas MAC adreses netiek uzskaitītas.

Piemērs:

```
system.hw.macaddr["eth0$",full] → [eth0] 00:11:22:33:44:55
```

system.localtime[<type>]

 Sistēmas laiks.
 Atgrieztā vērtība: *Integer* - ja `type` ir *utc*; *String* - ja `type` ir *local*.
 Skatiet atbalstītās platformas.

Parametri:

- **type** - iespējamās vērtības: *utc* - (noklusējums) laiks kopš Epoch (00:00:00 UTC, 1970. gada 1. janvāris), mērīts sekundēs, vai *local* - laiks formātā 'yyyy-mm-dd,hh:mm:ss.nnn,+hh:mm'

Jāizmanto kā pasīvā pārbaude Zabbix aģentam; Zabbix aģentā 2 to var konfigurēt kā aktīvo pārbaudi.

Piemērs:

```
system.localtime[local] #izveidojiet vienumu, izmantojot šo atslēgu, un pēc tam izmantojiet to, lai parādī
```

system.run[command,<mode>]

 Izpildīt norādīto komandu hostā.
 Atgrieztā vērtība: komandas *teksta* rezultāts vai 1 — ja `mode` ir *nowait* (neatkarīgi no komandas rezultāta).
 Skatiet atbalstītās platformas.

Parametri:

- **command** — komanda izpildei;
- **mode** — iespējamās vērtības: *wait* — gaidīt izpildes beigas (noklusējums) vai *nowait* — negaidīt.

Komentāri:

- Šis vienums pēc noklusējuma ir atspējots. Uzziniet, kā to iespējot.
- Vienuma atgrieztā vērtība ir standarta izvade kopā ar standarta kļūdu izvadi, ko rada komanda. Izvades koda pārbaude netiek veikta.
- Lai komandas atgrieztā vērtība tiktu pareizi apstrādāta, tai jābūt `text` datu tipam. Ir pieļaujams arī tukšs rezultāts.
- Atgrieztās vērtības lielums ir ierobežots līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas); attiecas arī datubāzes ierobežojumi.
- Skatiet arī: Komandu izpilde.

Piemērs:

```
system.run[ls -l /] #atgriež detalizētu saknes direktorija failu sarakstu
```

system.stat[resource,<type>]

 Sistēmas statistika.
 Atgrieztā vērtība: *Integer* vai *float*.
 Atbalstītās platformas: AIX.

Parametri:

- **ent** - procesora vienību skaits, ko šai nodalījuma instancei ir tiesības saņemt (float);
- **kthr,<type>** - informācija par kodola pavedienu stāvokļiem:
r - vidējais izpildāmo kodola pavedienu skaits (float)
b - vidējais kodola pavedienu skaits, kas ievietoti Virtual Memory Manager gaidīšanas rindā (float)
- **memory,<type>** - informācija par virtuālās un reālās atmiņas lietojumu:
avm - aktīvās virtuālās lapas (integer)
fre - brīvā saraksta lielums (integer)
- **page,<type>** - informācija par lapu kļūdām un lappušošanas aktivitāti:
fi - failu lapu ielasīšana sekundē (float)
fo - failu lapu izrakstīšana sekundē (float)
pi - lapas, kas ielappušotas no lappušošanas vietas (float)
po - lapas, kas izlapušotas uz lappušošanas vietu (float)
fr - atbrīvotās lapas (lapu aizvietošana) (float)
sr - lapas, ko skenē lapu aizvietošanas algoritms (float)
- **faults,<type>** - izņēmumu un pārtraukumu biežums:
in - ierīču pārtraukumi (float)
sy - sistēmas izsaukumi (float)
cs - kodola pavedienu konteksta pārslēgšanās (float)
- **cpu,<type>** - procesora laika izmantošanas procentuālais sadalījums:
us - lietotāja laiks (float)
sy - sistēmas laiks (float)
id - dīkstāves laiks (float)
wa - dīkstāves laiks, kura laikā sistēmai bija neapkalpots(-i) diska/NFS I/O pieprasījums(-i) (float)
pc - patērēto fizisko procesoru skaits (float)
ec - patērētās piešķirtās jaudas procentuālā daļa (float)
lbusy - norāda loģiskā(-o) procesora(-u) izmantošanas procentuālo daļu, kas radās izpildes laikā lietotāja un sistēmas līmenī (float)
app - norāda pieejamos fiziskos procesorus koplietojamajā pūlā (float)
- **disk,<type>** - diska statistika:
bps - norāda uz disku pārsūtīto datu apjomu (nolasīti vai ierakstīti) baitos sekundē (integer)
tps - norāda pārsūtījumu skaitu sekundē, kas tika nosūtīti fiziskajam diskam/lentei (float)

Komentāri:

- Ņemiet vērā šādus ierobežojumus šiem vienumiem:
 `system.stat[cpu,app]` - tiek atbalstīts tikai AIX LPAR tipam "Shared";
 `system.stat[cpu,ec]` - tiek atbalstīts AIX LPAR tipiem "Shared" un "Dedicated" ("Dedicated" vienmēr atgriež 100 (procentus));
 `system.stat[cpu,lbusy]` - tiek atbalstīts tikai AIX LPAR tipam "Shared";
 `system.stat[cpu,pc]` - tiek atbalstīts AIX LPAR tipiem "Shared" un "Dedicated";
 `system.stat[ent]` - tiek atbalstīts AIX LPAR tipiem "Shared" un "Dedicated".

system.sw.arch

 Programmatūras arhitektūras informācija.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Informācija tiek iegūta no funkcijas `uname()`.

Piemērs:

```
system.sw.arch → i686
```

system.sw.os[<info>]

 Operētājsistēmas informācija.
 Atgrieztā vērtība: *String*.
 Atbalstītās platformas: Linux, Windows.

Parametrs:

- **info** - iespējamās vērtības: *full* (noklusējums), *short* vai *name*

Informācija tiek iegūta no (ņemiet vērā, ka ne visi faili un opcijas ir pieejami visās distribūcijās):

- `/proc/version` (*full*) Linux sistēmās;
- `/proc/version_signature` (*short*) Linux sistēmās;
- parametrs PRETTY_NAME no `/etc/os-release` Linux sistēmās, kas to atbalsta, vai `/etc/issue.net` (*name*);
- reģistra atslēga `HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion` Windows sistēmās.

Piemēri:

```
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

 Detalizēta informācija par operētājsistēmu (versija, tips, distribūcijas nosaukums, mazākā un lielākā versija utt.).
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, Windows.

system.sw.packages[<regexp>,<manager>,<format>]

 Instalēto pakotņu saraksts.
 Atgrieztā vērtība: *Teksts*.
 Atbalstītās platformas: Linux.

Parametri:

- **regexp** — *all* (noklusējums) vai regulāra izteiksme;
- **manager** — *all* (noklusējums) vai pakotņu pārvaldnieks;
- **format** — *full* (noklusējums) vai *short*.

Komentāri:

- Uzskaita (alfabētiskā secībā) instalētās pakotnes, kuru nosaukums atbilst norādītajai regulārajai izteiksmei (*all* uzskaita tās visas).
- Atbalstītie pakotņu pārvaldnieki (izpildītā komanda):
  - dpkg (dpkg --get-selections)
  - pkgtool (ls /var/log/packages)
  - rpm (rpm -qa)
  - pacman (pacman -Q)
portage
- Ja `format` ir norādīts kā *full*, pakotnes tiek grupētas pēc pakotņu pārvaldniekiem (katrs pārvaldnieks atsevišķā rindā, kas sākas ar tā nosaukumu kvadrātiekavās).
- Ja `format` ir norādīts kā *short*, pakotnes netiek grupētas un tiek uzskaitītas vienā rindā.

Piemērs:

```
system.sw.packages[mini,dpkg,short] → python-minimal, python2.6-minimal, ubuntu-minimal
```

system.sw.packages.get[<regexp>,<manager>]

 Detalizēts instalēto pakotņu saraksts.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux.

Parametri:

- **regexp** — *all* (noklusējums) vai regulāra izteiksme;
- **manager** — *all* (noklusējums) vai pakotņu pārvaldnieks (iespējamās vērtības: *rpm*, *dpkg*, *pkgtool*, *pacman* vai *portage*).

Komentāri:

- Atgriež neformatētu JSON ar instalētajām pakotnēm, kuru nosaukums atbilst norādītajai regulārajai izteiksmei.
- Izvade ir objektu masīvs, kur katrs objekts satur šādas atslēgas: name, manager, version, size, architecture, buildtime un installtime (skatiet sīkāku informāciju).

system.swap.in[<device>,<type>]

 Ielādes maiņatmiņā (no ierīces atmiņā) statistika.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - norādiet ierīci, kas tiek izmantota maiņatmiņai (tikai Linux), vai *all* (noklusējums);
- **type** - iespējamās vērtības: *count* (ielāžu maiņatmiņā skaits, noklusējums platformās, kas nav Linux), *sectors* (ielādētie sektori), vai *pages* (ielādētās lapas, noklusējums Linux).

Komentāri:

- Šīs informācijas avots ir:
  - `/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4)
  - `/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6)
- Ņemiet vērā, ka *pages* darbosies tikai tad, ja ierīce nav norādīta.
- Parametrs *sectors* tiek atbalstīts tikai Linux.

Piemērs:

```
system.swap.in[,pages]
```

system.swap.out[<device>,<type>]

 Swap-out (no atmiņas uz ierīci) statistika.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, OpenBSD.

Parametri:

- **device** - norādiet ierīci, kas tiek izmantota swap darbībām (tikai Linux), vai *all* (noklusējums);
- **type** - iespējamās vērtības: *count* (swap-out reižu skaits, noklusējums platformās, kas nav Linux), *sectors* (uz ierīci pārvietoto sektoru skaits) vai *pages* (uz ierīci pārvietoto lapu skaits, noklusējums Linux).

Komentāri:

- Šīs informācijas avots ir:
`/proc/swaps`, `/proc/partitions`, `/proc/stat` (Linux 2.4)
`/proc/swaps`, `/proc/diskstats`, `/proc/vmstat` (Linux 2.6)
- Ņemiet vērā, ka *pages* darbosies tikai tad, ja ierīce nav norādīta.
- Parametrs *sectors* tiek atbalstīts tikai Linux.

Piemērs:

```
system.swap.out[,pages]
```

system.swap.size[<device>,<type>]

 Mijmaiņas vietas lielums baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD, Windows.

Parametri:

- **device** - norādiet mijmaiņai izmantoto ierīci (tikai FreeBSD) vai *all* (noklusējums);
- **type** - iespējamās vērtības: *free* (brīvā mijmaiņas vieta, noklusējums), *pfree* (brīvā mijmaiņas vieta procentos), *pused* (izmantotā mijmaiņas vieta procentos), *total* (kopējā mijmaiņas vieta) vai *used* (izmantotā mijmaiņas vieta).

Komentāri:

- Ņemiet vērā, ka *pfree*, *pused* netiek atbalstīti Windows, ja mijmaiņas vietas lielums ir 0.
- Ja ierīce nav norādīta, Zabbix aģents ņems vērā tikai mijmaiņas ierīces (failus), fiziskā atmiņa tiks ignorēta. Piemēram, Solaris sistēmās komanda `swap -s` ietver daļu fiziskās atmiņas un mijmaiņas ierīces (atšķirībā no `swap -l`).

Piemērs:

```
system.swap.size[,pfree] → brīvās mijmaiņas vietas procentuālā daļa
```

system.uname

 Sistēmas identifikācija.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Komentāri:

- UNIX sistēmās šī vienuma vērtība tiek iegūta, izmantojot uname() sistēmas izsaukumu.
- Windows sistēmās vienums atgriež OS arhitektūru, savukārt UNIX sistēmās tas atgriež CPU arhitektūru.

Piemēri:

```
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

 Sistēmas darbības laiks sekundēs.
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Vienuma konfigurācijā izmantojiet **s** vai **uptime** mērvienības, lai iegūtu cilvēkam lasāmas vērtības.

system.users.num

 Pieteikušos lietotāju skaits.
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Vērtības iegūšanai aģenta pusē tiek izmantota komanda **who**.

vfs.dev.discovery

 Blokierīču un to tipa saraksts. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux.

vfs.dev.read[<device>,<type>,<mode>]

 Diska nolasīšanas statistika.
 Atgrieztā vērtība: *Integer* - ja `type` ir *sectors*, *operations*, *bytes*; *Float* - ja `type` ir *sps*, *ops*, *bps*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parametri:

- **device** - diska ierīce (noklusējuma vērtība ir *all* [3]);
- **type** - iespējamās vērtības: *sectors*, *operations*, *bytes*, *sps*, *ops* vai *bps* (*sps*, *ops*, *bps* attiecīgi nozīmē: sektori, operācijas, baiti sekundē);
- **mode** - iespējamās vērtības: *avg1* (vienas minūtes vidējais, noklusējums), *avg5* vai *avg15*. Šis parametrs tiek atbalstīts tikai tad, ja `type` ir: sps, ops, bps.

Komentāri:

- Ja tiek izmantots atjaunināšanas intervāls trīs stundas vai ilgāk[2], šis vienums vienmēr atgriezīs '0'.
- *sectors* un *sps* tipa parametri tiek atbalstīti tikai Linux.
- *ops* tipa parametrs tiek atbalstīts tikai Linux un FreeBSD.
- *bps* tipa parametrs tiek atbalstīts tikai FreeBSD.
- *bytes* tipa parametrs tiek atbalstīts tikai FreeBSD, Solaris, AIX, OpenBSD.
- `mode` parametrs tiek atbalstīts tikai Linux, FreeBSD.

- Varat izmantot relatīvos ierīču nosaukumus (piemēram, `sda`), kā arī neobligātu `/dev/` prefiksu (piemēram, `/dev/sda`).
- Tiek atbalstīti LVM loģiskie sējumi.
- Parametra 'type' noklusējuma vērtības dažādām OS:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* un *bps* atbalstītajās platformās ir ierobežoti līdz 1024 ierīcēm (1023 individuālām un viena priekš *all*).

Piemērs:

```
vfs.dev.read[,operations]
```

vfs.dev.write[<device>,<type>,<mode>]

 Diska rakstīšanas statistika.
 Atgrieztā vērtība: *Integer* - ja `type` ir *sectors*, *operations*, *bytes*; *Float* - ja `type` ir *sps*, *ops*, *bps*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, AIX, OpenBSD.

Parametri:

- **device** - diska ierīce (noklusējums ir *all* [3]);
- **type** - iespējamās vērtības: *sectors*, *operations*, *bytes*, *sps*, *ops* vai *bps* (*sps*, *ops*, *bps* attiecīgi nozīmē: sektori, operācijas, baiti sekundē);
- **mode** - iespējamās vērtības: *avg1* (vienas minūtes vidējais, noklusējums), *avg5* vai *avg15*. Šis parametrs tiek atbalstīts tikai tad, ja `type` ir: sps, ops, bps.

Komentāri:

- Ja tiek izmantots atjaunināšanas intervāls trīs stundas vai ilgāks[2], šis vienums vienmēr atgriezīs '0'.
- Parametri *sectors* un *sps* tiek atbalstīti tikai Linux.
- Parametrs *ops* tiek atbalstīts tikai Linux un FreeBSD.
- Parametrs *bps* tiek atbalstīts tikai FreeBSD.
- Parametrs *bytes* tiek atbalstīts tikai FreeBSD, Solaris, AIX, OpenBSD.
- Parametrs `mode` tiek atbalstīts tikai Linux un FreeBSD.
- Var izmantot relatīvos ierīču nosaukumus (piemēram, `sda`), kā arī neobligātu prefiksu `/dev/` (piemēram, `/dev/sda`).
- Tiek atbalstīti LVM loģiskie sējumi.
- Parametra 'type' noklusējuma vērtības dažādām OS:
  - *AIX* - operations
  - *FreeBSD* - bps
  - *Linux* - sps
  - *OpenBSD* - operations
  - *Solaris* - bytes
- *sps*, *ops* un *bps* atbalstītajās platformās ir ierobežoti līdz 1024 ierīcēm (1023 atsevišķas un viena priekš *all*).

Piemērs:

```
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>]

 Direktorija ierakstu skaits.
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **dir** - absolūtais ceļš uz direktoriju;
- **regex incl** - regulāra izteiksme, kas apraksta iekļaujamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, iekļauj visus (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, neizslēdz nevienu (noklusējuma vērtība);
- **types incl** - skaitāmo direktorija ierakstu tipi, iespējamās vērtības: *file* - parasts fails, *dir* - apakšdirektorijs, *sym* - simboliskā saite, *sock* - ligzda, *bdev* - bloka ierīce, *cdev* - rakstzīmju ierīce, *fifo* - FIFO, *dev* - sinonīms "bdev,cdev", *all* - visi tipi (noklusējums), t.i. "file,dir,sym,sock,bdev,cdev,fifo". Vairāki tipi jāatdala ar komatu un jāliek pēdiņās.
- **types excl** - direktorija ierakstu tipi (skatiet `types incl`), kurus NEskaīt. Ja kāds ieraksta tips ir norādīts gan `types incl`, gan `types excl`, šī tipa direktorija ieraksti NETIEK skaitīti.
- **max depth** - maksimālais apstaigājamo apakšdirektoriju dziļums:
-1 (noklusējums) - neierobežots,
0 - neiet apakšdirektorijos.
- **min size** - minimālais faila izmērs (baitos), lai tas tiktu skaitīts. Mazāki faili netiks skaitīti. Var izmantot atmiņas sufiksus.

- **max size** - maksimālais faila izmērs (baitos), lai tas tiktu skaitīts. Lielāki faili netiks skaitīti. Var izmantot atmiņas sufiksus.
- **min age** - minimālais direktorija ieraksta vecums (sekundēs), lai tas tiktu skaitīts. Jaunāki ieraksti netiks skaitīti. Var izmantot laika sufiksus.
- **max age** - maksimālais direktorija ieraksta vecums (sekundēs), lai tas tiktu skaitīts. Tik veci un vecāki ieraksti netiks skaitīti (modifikācijas laiks). Var izmantot laika sufiksus.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamā direktorija nosaukuma šablonu. Tiks izslēgts viss direktorija saturs (atšķirībā no regex_excl)

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Vides mainīgie, piemēram, %APP_HOME%, $HOME un %TEMP%, netiek atbalstīti.
- Pseido-direktoriji ”.” un ”..” nekad netiek skaitīti.
- Simboliskajām saitēm nekad netiek sekots, pārlūkojot direktorijus.
- Gan `regex incl`, gan `regex excl` tiek piemēroti failiem un direktorijiem, aprēķinot ierakstu skaitu, taču tie tiek ignorēti, izvēloties apakšdirektorijus, kurus apstaigāt (ja `regex incl` ir ”(?i)^.+\.zip$” un `max depth` nav iestatīts, tad tiks apstaigāti visi apakšdirektoriji, bet skaitīti tiks tikai zip tipa faili).
- Izpildes laiku ierobežo aģents konfigurācijas timeout vērtība (3 sek.). Tā kā liela direktoriju apstaigāšana var aizņemt ilgāku laiku, dati netiks atgriezti un vienums kļūs neatbalstīts. Daļējs skaits netiks atgriezts.
- Filtrējot pēc izmēra, jēgpilni izmēri ir tikai parastajiem failiem. Linux un BSD vidē arī direktorijiem ir nenulles izmēri (parasti daži KB). Ierīcēm izmērs ir nulle, piemēram, **/dev/sda1** izmērs neatspoguļo attiecīgā nodalījuma izmēru. Tāpēc, izmantojot <min_size> un <max_size>, ieteicams norādīt <types_incl> kā *”file”*, lai izvairītos no negaidītiem rezultātiem.

Piemēri:

```
vfs.dir.count[/dev] #uzrauga ierīču skaitu direktorijā /dev (Linux)
vfs.dir.count["C:\Users\ADMINI~1\AppData\Local\Temp"] #uzrauga failu skaitu pagaidu direktorijā
```

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>]

 Direktorija ierakstu saraksts.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

Parametri:

- **dir** - absolūtais ceļš uz direktoriju;
- **regex incl** - regulāra izteiksme, kas apraksta iekļaujamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, iekļaut visu (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, neizslēgt neko (noklusējuma vērtība);
- **types incl** - uzskaitāmo direktorija ierakstu tipi, iespējamās vērtības: *file* - parasts fails, *dir* - apakšdirektorijs, *sym* - simboliskā saite, *sock* - ligzda, *bdev* - bloka ierīce, *cdev* - rakstzīmju ierīce, *fifo* - FIFO, *dev* - sinonīms ”bdev,cdev”, *all* - visi tipi (noklusējums), t.i. ”file,dir,sym,sock,bdev,cdev,fifo”. Vairāki tipi jāatdala ar komatu un jāliek pēdiņās.
- **types excl** - direktorija ierakstu tipi (skatiet `types incl`), kurus NEuzskaitīt. Ja kāds ieraksta tips ir norādīts gan `types incl`, gan `types excl`, šī tipa direktorija ieraksti NETIEK uzskaitīti.
- **max depth** - maksimālais apakšdirektoriju dziļums, ko pārlūkot:
-1 (noklusējums) - neierobežots,
0 - neiet apakšdirektorijos.
- **min size** - minimālais faila izmērs (baitos), lai tas tiktu uzskaitīts. Mazāki faili netiks uzskaitīti. Var izmantot atmiņas sufiksus.
- **max size** - maksimālais faila izmērs (baitos), lai tas tiktu uzskaitīts. Lielāki faili netiks uzskaitīti. Var izmantot atmiņas sufiksus.
- **min age** - minimālais direktorija ieraksta vecums (sekundēs), lai tas tiktu uzskaitīts. Jaunāki ieraksti netiks uzskaitīti. Var izmantot laika sufiksus.
- **max age** - maksimālais direktorija ieraksta vecums (sekundēs), lai tas tiktu uzskaitīts. Tik veci un vecāki ieraksti netiks uzskaitīti (modifikācijas laiks). Var izmantot laika sufiksus.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamā direktorija nosaukuma šablonu. Tiks izslēgts viss direktorija saturs (atšķirībā no `regex excl`)

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Vides mainīgie, piemēram, %APP_HOME%, $HOME un %TEMP%, netiek atbalstīti.
- Pseido-direktoriji ”.” un ”..” nekad netiek uzskaitīti.
- Simboliskajām saitēm nekad netiek sekots, pārlūkojot direktorijus.
- Gan `regex incl`, gan `regex excl` tiek piemēroti failiem un direktorijiem, ģenerējot ierakstu sarakstu, taču tie tiek ignorēti, izvēloties apakšdirektorijus pārlūkošanai (ja `regex incl` ir ”(?i)^.+\.zip$” un `max depth` nav iestatīts, tad tiks pārlūkoti visi apakšdirektoriji, bet tiks skaitīti tikai zip tipa faili).
- Izpildes laiku ierobežo taimauta vērtība aģents konfigurācijā. Tā kā lielu direktoriju pārlūkošana var aizņemt vairāk laika, dati netiks atgriezti un vienums kļūs neatbalstīts. Daļējs saraksts netiks atgriezts.

- Filtrējot pēc izmēra, jēgpilni izmēri ir tikai parastajiem failiem. Linux un BSD vidē arī direktorijiem ir nenulles izmēri (parasti daži KB). Ierīcēm izmērs ir nulle, piemēram, **/dev/sda1** izmērs neatspoguļo attiecīgā nodalījuma izmēru. Tāpēc, izmantojot `min size` un `max size`, ieteicams norādīt `types incl` kā "*file*", lai izvairītos no negaidītiem rezultātiem.

Piemēri:

```
vfs.dir.get[/dev] #iegūst ierīču sarakstu direktorijā /dev (Linux)
vfs.dir.get["C:\Users\ADMINI~1\AppData\Local\Temp"] #iegūst failu sarakstu pagaidu direktorijā
```

vfs.dir.size[dir,<regex incl>,<regex excl>,<mode>,<max depth>,<regex excl dir>]

 Direktorija izmērs (baitos).
 Atgrieztā vērtība: *Integer*.
 Atbalstītās platformas: Linux, Windows. Vienums var darboties arī citās UNIX tipa platformās.

Parametri:

- **dir** - absolūtais ceļš uz direktoriju;
- **regex incl** - regulāra izteiksme, kas apraksta iekļaujamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, tiek iekļauti visi (noklusējuma vērtība);
- **regex excl** - regulāra izteiksme, kas apraksta izslēdzamās entītijas (faila, direktorija, simboliskās saites) nosaukuma šablonu; ja tukšs, netiek izslēgts nekas (noklusējuma vērtība);
- **mode** - iespējamās vērtības: *apparent* (noklusējums) - iegūst redzamos failu izmērus, nevis diska lietojumu (darbojas kā `du -sb dir`), *disk* - iegūst diska lietojumu (darbojas kā `du -s -B1 dir`). Atšķirībā no du komandas, vfs.dir.size vienums, aprēķinot direktorija izmēru, ņem vērā arī slēptos failus (darbojas kā `du -sb .[^.]* *` direktorijā dir).
- **max depth** - maksimālais apstaigājamo apakšdirektoriju dziļums: **-1** (noklusējums) - neierobežots, **0** - neiet apakšdirektorijos.
- **regex excl dir** - regulāra izteiksme, kas apraksta izslēdzamā direktorija nosaukuma šablonu. Tiks izslēgts viss direktorija saturs (atšķirībā no `regex excl`)

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Tiek aprēķināti tikai tie direktoriji, kuriem lietotājam *zabbix* ir vismaz lasīšanas atļauja. Direktorijiem, kuriem ir tikai lasīšanas atļauja, tiek aprēķināts tikai paša direktorija izmērs. Direktoriji ar lasīšanas un izpildes atļaujām tiek aprēķināti kopā ar saturu.
- Lielu direktoriju vai lēnu disku gadījumā šim vienumam var iestāties noildze Timeout iestatījuma dēļ aģents un serveris/starpniekserveris konfigurācijas failos. Palieliniet noildzes vērtības pēc nepieciešamības.
- Faila izmēra ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemēri:

```
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>]

 Faila kontrolsumma, kas aprēķināta, izmantojot UNIX cksum algoritmu.
 Atgrieztā vērtība: *Integer* - ja `mode` ir *crc32*, *String* - ja `mode` ir *md5*, *sha256*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **mode** - *crc32* (noklusējums), *md5* vai *sha256*.

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.cksum[/etc/passwd]
```

Atgriezto vērtību piemēri (attiecīgi crc32/md5/sha256):

```
675436101
9845acf68b73991eb7fd7ee0ded23c44
ae67546e4aac995e5c921042d0cf0f1f7147703aa42bfbfb65404b30f238f2dc
```

vfs.file.contents[file,<encoding>]

 Faila satura iegūšana[7].
 Atgrieztā vērtība: *Teksts*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **encoding** - kodu lapas identifikators.

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Atgrieztā vērtība ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek nogrieztas); attiecas arī datu bāzes ierobežojumi.
- Ja fails ir tukšs vai satur tikai LF/CR rakstzīmes, tiek atgriezta tukša virkne.
- Baitu secības atzīme (BOM) netiek iekļauta izvadē.

Piemērs:

```
vfs.file.contents[/etc/passwd]
```

vfs.file.exists[file,<types incl>,<types excl>]

 Pārbauda, vai fails eksistē.
 Atgrieztā vērtība: 0 - nav atrasts; 1 - eksistē norādītā tipa fails.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **types incl** - iekļaujamo failu tipu saraksts, iespējamās vērtības: *file* (parasts fails, noklusējums (ja types_excl nav iestatīts)), *dir* (direktorijs), *sym* (simboliskā saite), *sock* (ligzda), *bdev* (bloku ierīce), *cdev* (rakstzīmju ierīce), *fifo* (FIFO), *dev* (sinonīms "bdev,cdev"), *all* (visi minētie tipi, noklusējums, ja types_excl ir iestatīts).
- **types excl** - izslēdzamo failu tipu saraksts, iespējamās vērtības skatiet pie types_incl (pēc noklusējuma netiek izslēgts neviens tips)

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Vairāki tipi jāatdala ar komatu, un viss kopums jāieliek pēdiņās "".
- Ja viens un tas pats tips ir norādīts gan <types_incl>, gan <types_excl>, šī tipa faili tiek izslēgti.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemēri:

```
vfs.file.exists[/tmp/application.pid]
vfs.file.exists[/tmp/application.pid,"file,dir,sym"]
vfs.file.exists[/tmp/application_dir,dir]
```

vfs.file.get[file]

 Atgriež informāciju par failu.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

Parametrs:

- **file** - pilns ceļš uz failu

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Atbalstītie failu tipi UNIX līdzīgās sistēmās: parasts fails, direktorijs, simboliskā saite, ligzda, bloku ierīce, rakstzīmju ierīce, FIFO.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.get[/etc/passwd] #atgriež JSON ar informāciju par /etc/passwd failu (tips, lietotājs, atļaujas, S
```

vfs.file.md5sum[file]

 Faila MD5 kontrolsumma.
 Atgrieztā vērtība: Rakstzīmju virkne (faila MD5 jaucējkods).
 Skatiet atbalstītās platformas.

Parametrs:

- **file** — pilns ceļš uz failu

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.md5sum[/usr/local/etc/zabbix_agentd.conf]
```

Atgrieztās vērtības piemērs:

```
b5052decb577e0fffd622d6ddc017e82
```

vfs.file.owner[file,<ownertype>,<resulttype>]

 Iegūst faila īpašnieku.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **ownertype** - *user* (noklusējums) vai *group* (tikai Unix);
- **resulttype** - *name* (noklusējums) vai *id*; *id* gadījumā — atgriež uid/gid Unix sistēmās, SID Windows sistēmās.

Komentāri:

- Windows sistēmās šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemēri:

```
vfs.file.owner[/tmp/zabbix_server.log] #return the file owner of /tmp/zabbix_server.log
vfs.file.owner[/tmp/zabbix_server.log,,id] #return the file owner ID of /tmp/zabbix_server.log
```

vfs.file.permissions[file]

 Atgriež 4 ciparu virkni, kas satur oktālo skaitli ar UNIX atļaujām.
 Atgrieztā vērtība: *Virkne*.
 Atbalstītās platformas: Linux. Vienums var darboties arī citās UNIX tipa platformās.

Parametri:

- **file** - pilns ceļš uz failu

Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.permissions[/etc/passwd] #atgriež /etc/passwd atļaujas, piemēram, '0644'
```

vfs.file.regexp[file,regexp,<encoding>,<start line>,<end line>,<output>]

 Iegūt virkni failā[7].
 Atgrieztā vērtība: rinda, kas satur atbilstošo virkni, vai arī vērtība, kā norādīts neobligātajā `output` parametrā.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu;
- **encoding** - kodlapas identifikators;
- **start line** - pirmās meklējamās rindas numurs (pēc noklusējuma faila pirmā rinda);
- **end line** - pēdējās meklējamās rindas numurs (pēc noklusējuma faila pēdējā rinda);
- **output** - neobligāta izvades formatēšanas veidne. Aizvietošanas secība **\0** tiek aizstāta ar atbilstošo teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt aizvietošanas secība **\N** (kur N=1...9) tiek aizstāta ar N-to atrasto grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu).

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.
- Tiek atgriezta tikai pirmā atbilstošā rinda.
- Ja neviena rinda neatbilst izteiksmei, tiek atgriezta tukša virkne.
- Baitu secības atzīme (BOM) netiek iekļauta izvadē.
- Satura izvilkšana, izmantojot `output` parametru, notiek aģentā.

Piemēri:

```
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] → lietotāja *zabbix* ID iegūšana
```

vfs.file.regmatch[file,regexp,<encoding>,<start line>,<end line>]

 Atrod virkni failā[7].
 Atgrieztās vērtības: 0 - atbilstība nav atrasta; 1 - atrasta.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu;
- **encoding** - kodu lapas identifikators;
- **start line** - pirmās rindas numurs, no kuras meklēt (pēc noklusējuma faila pirmā rinda);
- **end line** - pēdējās rindas numurs, līdz kurai meklēt (pēc noklusējuma faila pēdējā rinda).

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.
- Baitu secības atzīme (BOM) tiek ignorēta.

Piemērs:

```
vfs.file.regmatch[/var/log/app.log,error]
```

vfs.file.size[file,<mode>]

 Faila lielums (baitos).
 Atgrieztā vērtība: *Integer*.
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **mode** - iespējamās vērtības: *bytes* (noklusējums) vai *lines* (tiek skaitītas arī tukšās rindas).

Komentāri:

- Windows vidē šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Failam jābūt lasīšanas atļaujām lietotājam *zabbix*.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.size[/var/log/syslog]
```

vfs.file.time[file,<mode>]

 Faila laika informācija.
 Atgrieztā vērtība: *Integer* (Unix laikspiedols).
 Skatiet atbalstītās platformas.

Parametri:

- **file** - pilns ceļš uz failu;
- **mode** - iespējamās vērtības:
modify (noklusējums) - pēdējais faila satura modificēšanas laiks,
access - pēdējais faila lasīšanas laiks,
change - pēdējais faila rekvizītu mainīšanas laiks

Komentāri:

- Operētājsistēmā Windows šis vienums seko UNC ceļiem, kas var radīt drošības risku.
- Faila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Piemērs:

```
vfs.file.time[/etc/passwd,modify]
```

vfs.fs.discovery

 Pievienoto failu sistēmu saraksts ar to tipu un montēšanas opcijām. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

vfs.fs.get

 Pievienoto failu sistēmu saraksts ar to tipu, pieejamo diska vietu, inode statistiku un montēšanas opcijām. Var izmantot zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON virkne*.
 Atbalstītās platformas: Linux, FreeBSD, Solaris, HP-UX, AIX, MacOS X, OpenBSD, NetBSD, Windows.

Komentāri:

- Tiek ziņots arī par failu sistēmām, kurām inode skaits ir vienāds ar nulli, kas var būt raksturīgi failu sistēmām ar dinamiskiem inode (piemēram, btrfs).
- Skatiet arī: Pievienoto failu sistēmu atklāšana.

vfs.fs.inode[fs,<mode>]

 Inodu skaits vai procentuālā daļa.
 Atgrieztā vērtība: *Integer* - skaitam; *Float* - procentuālajai daļai.
 Skatiet atbalstītās platformas.

Parametri:

- **fs** - failu sistēma;
- **mode** - iespējamās vērtības: *total* (noklusējums), *free*, *used*, *pfree* (brīvs, procentos) vai *pused* (izmantots, procentos).

Ja inodu skaits ir nulle, kas var būt failu sistēmās ar dinamiskiem inodiem (piemēram, btrfs), pfree/pused vērtības tiks ziņotas attiecīgi kā "100" un "0".

Piemērs:

```
vfs.fs.inode[/,pfree]
```

vfs.fs.size[fs,<mode>]

 Diska vieta baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.
 Skatiet atbalstītās platformas.

Parametri:

- **fs** - failu sistēma;
- **mode** - iespējamās vērtības: *total* (noklusējums), *free*, *used*, *pfree* (brīvs, procentos) vai *pused* (izmantots, procentos).

Komentāri:

- Ja failu sistēma nav piemontēta, tiek atgriezts tās lokālās failu sistēmas lielums, kurā atrodas montēšanas punkts.
- Failu sistēmas rezervētā vieta tiek ņemta vērā un netiek iekļauta, izmantojot *free* režīmu.

Piemērs:

```
vfs.fs.size[/tmp,free]
```

vm.memory.size[<mode>]

 Atmiņas lielums baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.
 Skatiet atbalstītās platformas.

Parametrs:

- **mode** - iespējamās vērtības: *total* (noklusējums), *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*, *used*, *pused* (izmantots, procentos), *available* vai *pavailable* (pieejams, procentos).

Komentāri:

- Šis vienums pieņem trīs parametru kategorijas:
  - *total* - kopējais atmiņas apjoms;
  - platformai specifiski atmiņas tipi: *active*, *anon*, *buffers*, *cached*, *exec*, *file*, *free*, *inactive*, *pinned*, *shared*, *slab*, *wired*;
  - lietotāja līmeņa novērtējumi par to, cik daudz atmiņas ir izmantots un pieejams: *used*, *pused*, *available*, *pavailable*.
- Parametrs *active* tiek atbalstīts tikai FreeBSD, HP-UX, MacOS X, OpenBSD, NetBSD.
- Parametri *anon*, *exec*, *file* tiek atbalstīti tikai NetBSD.
- Parametrs *buffers* tiek atbalstīts tikai Linux, FreeBSD, OpenBSD, NetBSD.
- Parametrs *cached* tiek atbalstīts tikai Linux, FreeBSD, AIX, OpenBSD, NetBSD.
- Parametri *inactive*, *wired* tiek atbalstīti tikai FreeBSD, MacOS X, OpenBSD, NetBSD.
- Parametrs *pinned* tiek atbalstīts tikai AIX.
- Parametrs *shared* tiek atbalstīts tikai Linux 2.4, FreeBSD, OpenBSD, NetBSD.
- Skatiet arī papildu informāciju par šo vienumu.

Piemērs:

```
vm.memory.size[pavailable]
```

web.page.get[host,<path>,<port>]

 Iegūt tīmekļa lapas saturu.
 Atgrieztā vērtība: tīmekļa lapas pirmkods kā teksts (ieskaitot galvenes).
 Skatiet atbalstītās platformas.

Parametri:

- **host** — resursdatora nosaukums vai URL (formā `scheme://host:port/path`, kur obligāts ir tikai *host*). Atļautās URL shēmas: *http*, *https*[4]. Ja shēma nav norādīta, tā tiks uzskatīta par *http*. Ja ir norādīts URL, `path` un `port` jābūt tukšiem. Lietotājvārda/paroles norādīšana, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram: `http://user:password@www.example.com`, ir iespējama tikai ar cURL atbalstu [4]. Punycode tiek atbalstīts resursdatoru nosaukumos.
- **path** — ceļš uz HTML dokumentu (noklusējums ir /);
- **port** — porta numurs (noklusējums HTTP gadījumā ir 80)

Komentāri:

- Šis vienums kļūst neatbalstīts, ja `host` norādītais resurss neeksistē vai nav pieejams.
- `host` var būt resursdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Taču IPv6 adreses gadījumā Zabbix aģents ir jākompilē ar iespējotu IPv6 atbalstu.

Piemēri:

```
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>]

 Pilnas tīmekļa lapas ielādes laiks (sekundēs).
 Atgrieztā vērtība: *Float*.
 Skatiet atbalstītās platformas.

Parametri:

- **host** - resursdatora nosaukums vai URL (kā `scheme://host:port/path`, kur obligāts ir tikai *host*). Atļautās URL shēmas: *http*, *https*[4]. Ja shēma nav norādīta, tā tiks uzskatīta par *http*. Ja ir norādīts URL, `path` un `port` jābūt tukšiem. Lietotājvārda/paroles norādīšana, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram: `http://user:password@www.example.com`, ir iespējama tikai ar cURL atbalstu [4]. Resursdatoru nosaukumos tiek atbalstīts Punycode.
- **path** - ceļš uz HTML dokumentu (noklusējums ir /);
- **port** - porta numurs (noklusējums HTTP gadījumā ir 80)

Komentāri:

- Šis vienums kļūst neatbalstīts, ja `host` norādītais resurss neeksistē vai nav pieejams.
- `host` var būt resursdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Taču IPv6 adreses gadījumā Zabbix aģents jākompilē ar iespējotu IPv6 atbalstu.

Piemēri:

```
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>]

 Atrod virkni tīmekļa lapā.
 Atgrieztā vērtība: Atrastā virkne vai, kā norādīts neobligātajā `output` parametrā.
 Skatiet atbalstītās platformas.

Parametri:

- **host** - resursdatora nosaukums vai URL (kā `scheme://host:port/path`, kur obligāts ir tikai *host*). Atļautās URL shēmas: *http*, *https*[4]. Ja shēma nav norādīta, tā tiks uzskatīta par *http*. Ja ir norādīts URL, `path` un `port` jābūt tukšiem. Lietotājvārda/paroles norādīšana, pieslēdzoties serveriem, kuriem nepieciešama autentifikācija, piemēram: `http://user:password@www.example.com`, ir iespējama tikai ar cURL atbalstu [4]. Resursdatoru nosaukumos tiek atbalstīts Punycode.
- **path** - ceļš uz HTML dokumentu (noklusējums ir /);
- **port** - porta numurs (noklusējums HTTP gadījumā ir 80)
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu;
- **length** - maksimālais atgriežamo rakstzīmju skaits;
- **output** - neobligāta izvades formatēšanas veidne. Aizvietošanas secība **\0** tiek aizstāta ar atrasto teksta daļu (no pirmās rakstzīmes, kur sākas atbilstība, līdz rakstzīmei, kur atbilstība beidzas), savukārt aizvietošanas secība **\N** (kur N=1...9) tiek aizstāta ar N-to atrasto grupu (vai tukšu virkni, ja N pārsniedz uztverto grupu skaitu).

Komentāri:

- Šis vienums kļūst par neatbalstītu, ja `host` norādītais resurss neeksistē vai nav pieejams.
- `host` var būt resursdatora nosaukums, domēna nosaukums, IPv4 vai IPv6 adrese. Taču IPv6 adreses gadījumā Zabbix aģents ir jākompilē ar iespējotu IPv6 atbalstu.
- Satura izvilkšana, izmantojot `output` parametru, notiek aģentā.

Piemēri:

```
web.page.regexp[www.example.com,index.php,80,OK,2]
web.page.regexp[https://www.example.com,,,OK,2]
```

agent.hostmetadata

 Aģenta hosta metadati.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Atgriež parametru HostMetadata vai HostMetadataItem vērtību, vai tukšu virkni, ja neviens no tiem nav definēts.

agent.hostname

 Aģenta hosta nosaukums.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Atgriež:

- Kā pasīvā pārbaude — pirmā hosta nosaukumu, kas norādīts aģenta konfigurācijas faila parametrā Hostname;
- Kā aktīvā pārbaude — pašreizējā hosta nosaukumu.

agent.ping

 Aģenta pieejamības pārbaude.
 Atgrieztā vērtība: nekas — nav pieejams; 1 — pieejams.
 Skatiet atbalstītās platformas.

Izmantojiet **nodata()** trigera funkciju, lai pārbaudītu hosta nepieejamību.

agent.variant

 Zabbix aģents variants (Zabbix aģents vai Zabbix aģents 2).
 Atgrieztā vērtība: 1 - Zabbix aģents; 2 - Zabbix aģents 2.
 Skatiet atbalstītās platformas.

aģents.version

 Zabbix aģenta versija.
 Atgrieztā vērtība: *String*.
 Skatiet atbalstītās platformas.

Atgrieztās vērtības piemērs:

```
6.0.3
```

zabbix.stats[<ip>,<port>]

 Atgriež Zabbix servera vai starpniekservera iekšējo metriku kopu. Ja ir norādīti `ip` un `port`, metrikas tiek iegūtas no attālās instances; pretējā gadījumā — no lokālās instances.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** - attālināti vaicājamo serveru/starpniekserveru IP/DNS/tīkla masku saraksts (noklusējuma vērtība ir 127.0.0.1);
- **port** - attālināti vaicājamā servera/starpniekservera ports (noklusējuma vērtība ir 10051)

Komentāri:

- Šis vienums atgriež atlasītu iekšējo metriku kopu. Sīkāku informāciju skatiet sadaļā Zabbix statistikas attālā uzraudzība.
- Ņemiet vērā, ka statistikas pieprasījums tiks pieņemts tikai no adresēm, kas mērķa instancē ir norādītas parametra 'StatsAllowedIP' serveris/starpniekserveris iestatījumā.

zabbix.stats[<ip>,<port>,queue,<from>,<to>]

 Atgriež uzraudzīto vienumu skaitu rindā, kuru apstrāde Zabbix serverī vai starpniekserverī kavējas. Ja ir norādīti `ip` un `port`, metriskie dati tiek iegūti no attālās instances; pretējā gadījumā — no lokālās instances.
 Atgrieztā vērtība: *JSON virkne*.
 Skatiet atbalstītās platformas.

Parametri:

- **ip** - attālināti vaicājamo serveru/starpniekserveru IP/DNS/tīkla masku saraksts (noklusējuma vērtība ir 127.0.0.1);
- **port** - attālināti vaicājamā servera/starpniekservera ports (noklusējuma vērtība ir 10051)
- **queue** - konstante (jāizmanto bez izmaiņām)
- **from** - kavējas vismaz par (noklusējuma vērtība ir 6 sekundes)
- **to** - kavējas ne vairāk par (noklusējuma vērtība ir bezgalība)

Ņemiet vērā, ka statistikas pieprasījums tiks pieņemts tikai no adresēm, kas mērķa instancē ir norādītas parametra 'StatsAllowedIP' serveris/starpniekserveris iestatījumā.

Piezīmes

[1] Linux specifiska piezīme. Zabbix aģentam jābūt tikai lasīšanas piekļuvei failu sistēmai */proc*. Kodola ielāpi no www.grsecurity.org ierobežo nepriviliģētu lietotāju piekļuves tiesības.

[2] `vfs.dev.read[]`, `vfs.dev.write[]`: Zabbix aģents pārtrauks "novecojušus" ierīču savienojumus, ja vienumu vērtībām nepiekļūst ilgāk par 3 stundām. Tas var notikt, ja sistēmā ir ierīces ar dinamiski mainīgiem ceļiem vai ja ierīce tiek manuāli noņemta. Ņemiet vērā arī to, ka šie vienumi, ja tiek izmantots atjaunināšanas intervāls 3 stundas vai ilgāks, vienmēr atgriezīs '0'.

[3] `vfs.dev.read[]`, `vfs.dev.write[]`: Ja pirmajam parametram tiek izmantota noklusējuma vērtība *all*, tad atslēga atgriezīs kopsavilkuma statistiku, iekļaujot visas bloku ierīces, piemēram, sda, sdb, un to nodalījumus (sda1, sda2, sdb3...), kā arī

vairākas ierīces (MD raid), kas balstītas uz šīm bloku ierīcēm/nodalījumiem, un loģiskos sējumus (LVM), kas balstīti uz šīm bloku ierīcēm/nodalījumiem. Šādos gadījumos atgrieztās vērtības būtu jāuzskata tikai par relatīvām vērtībām (dinamiskām laikā), nevis par absolūtām vērtībām.

[4] SSL (HTTPS) tiek atbalstīts tikai tad, ja aģents ir kompilēts ar cURL atbalstu. Pretējā gadījumā vienums kļūs neatbalstīts.

[5] `bytes` un `errors` vērtības netiek atbalstītas loopback saskarnēm Solaris sistēmās līdz Solaris 10 6/06 ieskaitot, jo baitu, kļūdu un noslodzes statistiku kodols neuzglabā un/vai neziņo. Tomēr, ja uzraugāt Solaris sistēmu, izmantojot net-snmp, vērtības var tikt atgrieztas, jo net-snmp satur mantotu kodu no cmu-snmp, kas datēts pat ar 1997. gadu un kas, nespējot nolasīt baitu vērtības no saskarnes statistikas, atgriež pakešu skaitītāju (kas loopback saskarnēs pastāv), reizinātu ar patvaļīgu vērtību 308. Tas balstās uz pieņēmumu, ka paketes vidējais garums ir 308 okteti, kas ir ļoti aptuvens novērtējums, jo Solaris sistēmās loopback saskarņu MTU ierobežojums ir 8892 baiti. Nevajadzētu pieņemt, ka šīs vērtības ir pareizas vai pat tuvu precīzas. Tie ir aptuveni minējumi. Zabbix aģents neveic nekādu minēšanu, taču net-snmp šiem laukiem atgriezīs vērtību.

[6] Komandrinda Solaris sistēmā, kas iegūta no /proc/pid/psinfo, ir ierobežota līdz 80 baitiem un satur komandrindu tādu, kāda tā bija procesa palaišanas brīdī.

[7] `vfs.file.contents[]`, `vfs.file.regexp[]`, `vfs.file.regmatch[]` vienumus var izmantot faila satura iegūšanai. Ja vēlaties ierobežot piekļuvi konkrētiem failiem ar sensitīvu informāciju, palaidiet Zabbix aģentu zem lietotāja, kuram nav piekļuves tiesību šo failu skatīšanai.

Lietošana ar komandrindas utilītām

Ņemiet vērā, ka, testējot vai izmantojot vienumu atslēgas ar zabbix_agentd vai zabbix_get no komandrindas, jāņem vērā arī čaulas sintakse.

Piemēram, ja noteikts atslēgas parametrs ir jāiekļauj dubultpēdiņās, jums ir skaidri jāekranē dubultpēdiņas, pretējā gadījumā čaula tās apstrādās kā speciālās rakstzīmes un tās netiks nodotas Zabbix utilītai.

Piemēri:

```
zabbix_agentd -t 'vfs.dir.count[/var/log,,,"file,dir",,0]'

zabbix_agentd -t vfs.dir.count[/var/log,,,\"file,dir\",,0]
```

Kodējuma iestatījumi

Lai nodrošinātu, ka iegūtie dati netiek bojāti, varat norādīt pareizo kodējumu pārbaudes apstrādei (piemēram, `vfs.file.contents`) parametrā `encoding`. Atbalstīto kodējumu sarakstu (kodu lapu identifikatorus) var atrast libiconv (GNU Project) dokumentācijā vai Microsoft Windows SDK dokumentācijā par "Code Page Identifiers". Ņemiet vērā, ka Microsoft dažkārt dažas kodu lapas atzīmē kā "pieejamas tikai pārvaldītām lietojumprogrammām" — tas ir, pieejamas tikai caur .NET izpildlaika vidi un nav pieejamas, izmantojot vietējās Win32 API. Zabbix aģents īsteno savu kodējuma pārveidošanas loģiku, tāpēc šīs kodu lapas Zabbix aģents atbalsta pat tad, ja vietējās Windows funkcijas tās nenodrošina.

Ja parametrā `encoding` nav norādīts kodējums, tiek piemērotas šādas noteikšanas stratēģijas:

- Ja kodējums nav norādīts (vai ir tukša virkne), tiek pieņemts, ka tas ir UTF-8, un dati tiek apstrādāti "kā ir";
- BOM analīze — attiecas uz vienumiem 'vfs.file.contents', 'vfs.file.regexp', 'vfs.file.regmatch'. Tiek mēģināts noteikt pareizo kodējumu, izmantojot baita secības atzīmi (BOM) faila sākumā. Ja BOM nav, tā vietā tiek piemērota standarta noteikšana (skatiet iepriekš).

Aģenta vienumu problēmu novēršana

Pasīvo pārbaužu gadījumā, lai novērstu situāciju, kad vienums nesaņem nekādu vērtību tāpēc, ka servera pieprasījumam uz aģentu vispirms iestājas noildze, jāņem vērā tālāk norādītais:

- Ja aģenta versija ir vecāka par servera versiju, *Timeout* vērtībai vienuma konfigurācijā (vai globālajai noildzei) var būt jābūt lielākai par `Timeout` vērtību aģenta konfigurācijas failā.
- Ja aģenta versija ir jaunāka par servera versiju, `Timeout` vērtībai servera konfigurācijas failā var būt jābūt lielākai par `Timeout` vērtību aģenta konfigurācijas failā.

1 Zabbix aģents 2

Zabbix aģents 2 atbalsta visas vienumu atslēgas, ko atbalsta Zabbix aģents Unix un Windows vidēs. Šajā lapā ir sniegta informācija par papildu vienumu atslēgām, kuras var izmantot tikai ar Zabbix aģents 2, sagrupējot tās pēc spraudņa, kuram tās pieder.

Vienumu atslēgas ir norādītas bez parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts	Spraudnis
ceph.df.details	Klastera datu lietojums un sadalījums starp pūliem.	Ceph (ielādējams)
ceph.osd.stats	Apkopotā statistika un statistika par katru OSD.	
ceph.osd.discovery	Atklāto OSD saraksts.	
ceph.osd.dump	OSD lietojuma sliekšņi un statusi.	
ceph.ping	Pārbauda, vai var izveidot savienojumu ar Ceph.	
ceph.pool.discovery	Atklāto pūlu saraksts.	
ceph.status	Kopējais klastera statuss.	
docker.container_info	Zema līmeņa informācija par konteineru.	Docker
docker.container_stats	Konteinera resursu lietojuma statistika.	
docker.containers	Atgriež konteineru sarakstu.	
docker.containers.discovery	Atgriež konteineru sarakstu. Tiek izmantots zema līmeņa atklāšanai.	
docker.data.usage	Informācija par pašreizējo datu lietojumu.	
docker.images	Atgriež attēlu sarakstu.	
docker.images.discovery	Atgriež attēlu sarakstu. Tiek izmantots zema līmeņa atklāšanai.	
docker.info	Sistēmas informācija.	
docker.ping	Pārbauda, vai Docker dēmons darbojas.	
ember.get	Atgriež pieprasītās ierīces rezultātu.	Ember+ (ielādējams)
memcached.ping	Pārbauda, vai savienojums ir aktīvs.	Memcached
memcached.stats	Iegūst komandas STATS izvadi.	
mongodb.cfg.discovery	Atgriež atklāto konfigurācijas serveru sarakstu.	MongoDB (ielādējams)
mongodb.collection.stats	Atgriež dažādu krātuves statistiku norādītajai kolekcijai.	
mongodb.collections.discovery	Atgriež atklāto kolekciju sarakstu.	
mongodb.collections.usage	Atgriež kolekciju lietojuma statistiku.	
mongodb.connpool.stats	Atgriež informāciju par atvērtajiem izejošajiem savienojumiem no pašreizējās datubāzes instances uz citiem šardētā klastera vai repliku kopas dalībniekiem.	
mongodb.db.stats	Atgriež statistiku, kas atspoguļo noteiktas datubāzes sistēmas stāvokli.	
mongodb.db.discovery	Atgriež atklāto datubāzu sarakstu.	
mongodb.jumbo_chunks.count	Atgriež jumbo gabalu skaitu.	
mongodb.oplog.stats	Atgriež repliku kopas statusu, izmantojot no oplog iegūtos datus.	
mongodb.ping	Pārbauda, vai savienojums ir aktīvs.	
mongodb.rs.config	Atgriež pašreizējo repliku kopas konfigurāciju.	
mongodb.rs.status	Atgriež repliku kopas statusu no tā dalībnieka skatpunkta, kurā metode tiek izpildīta.	
mongodb.server.status	Atgriež datubāzes stāvokli.	
mongodb.sh.discovery	Atgriež klasterī esošo atklāto šardu sarakstu.	
mongodb.version	Atgriež datubāzes servera versiju.	
mqtt.get	Abonē norādītā brokera konkrētu tēmu vai tēmas (ar aizstājējzīmēm) un gaida publikācijas.	MQTT
mssql.availability.group.get	Atgriež pieejamības grupas.	MSSQL (ielādējams)
mssql.custom.query	Atgriež pielāgota vaicājuma rezultātu.	
mssql.db.get	Atgriež visas pieejamās MSSQL datubāzes.	
mssql.job.status.get	Atgriež uzdevumu statusu.	
mssql.last.backup.get	Atgriež pēdējās dublējumkopijas laiku visām datubāzēm.	
mssql.local.db.get	Atgriež datubāzes, kas piedalās Always On pieejamības grupā un replikā (primārajā vai sekundārajā) un atrodas serverī, ar kuru tika izveidots savienojums.	
mssql.mirroring.get	Atgriež spoguļošanas informāciju.	
mssql.nonlocal.db.get	Atgriež datubāzes, kas piedalās Always On pieejamības grupā un replikā (primārajā vai sekundārajā), kas atrodas citos serveros (datubāze nav lokāla SQL Server instancei, ar kuru tika izveidots savienojums).	
mssql.perfcounter.get	Atgriež veiktspējas skaitītājus.	
mssql.ping	Pārbauda, vai savienojums ir aktīvs.	
mssql.quorum.get	Atgriež kvoruma informāciju.	
mssql.quorum.member.get	Atgriež kvoruma dalībniekus.	
mssql.replica.get	Atgriež replikas.	
mssql.version	Atgriež MSSQL versiju.	
mysql.custom.query	Atgriež pielāgota vaicājuma rezultātu.	MySQL
mysql.db.discovery	Atgriež MySQL datubāzu sarakstu.	

Vienuma atslēga	Apraksts	Spraudnis
mysql.db.size	Datubāzes lielums baitos.	
mysql.get_status_variables	Globālo statusa mainīgo vērtības.	
mysql.ping	Pārbauda, vai savienojums ir aktīvs.	
mysql.replication.discovery	Atgriež MySQL replikāciju sarakstu.	
mysql.replication.get_slave_status	Replikācijas statuss.	
mysql.version	MySQL versija.	
net.dns.get	Izpilda DNS vaicājumu un atgriež detalizētu DNS ierakstu informāciju.	Network
nvml.device.count	GPU ierīču skaits.	NVIDIA GPU (ielādējams)
nvml.device.decoder.utilization	GPU ierīces dekodera noslodze.	
nvml.device.ecc.mode	GPU ierīces ECC režīms.	
nvml.device.encoder.stats.get	GPU ierīces kodētāja statistika.	
nvml.device.encoder.utilization	GPU ierīces kodētāja noslodze.	
nvml.device.energy.consumption	GPU ierīces enerģijas patēriņš.	
nvml.device.errors.memory	GPU ierīces ECC atmiņas kļūdu informācija.	
nvml.device.errors.register	GPU ierīces ECC reģistra kļūdu informācija.	
nvml.device.fan.speed.avg	GPU ierīces ventilatora ātruma vidējā vērtība.	
nvml.device.get	Atklātie GPU ar UUID un nosaukumu.	
nvml.device.graphics.frequency	GPU ierīces grafiskā pulksteņa frekvence.	
nvml.device.memory.bar1.get	GPU ierīces BAR1 atmiņas informācija.	
nvml.device.memory.fb.get	GPU ierīces kadru bufera atmiņas informācija.	
nvml.device.memory.frequency	GPU ierīces atmiņas pulksteņa frekvence.	
nvml.device.pci.utilization	GPU ierīces PCI noslodze.	
nvml.device.performance.state	GPU ierīces veiktspējas stāvoklis.	
nvml.device.power.limit	GPU ierīces jaudas limits.	
nvml.device.power.usage	GPU ierīces jaudas lietojums.	
nvml.device.serial	GPU ierīces sērijas numurs.	
nvml.device.sm.frequency	GPU ierīces straumēšanas daudzprocesoru pulksteņa frekvence.	
nvml.device.temperature	GPU ierīces temperatūra.	
nvml.device.utilization	GPU ierīces noslodzes statistika.	
nvml.device.video.frequency	GPU ierīces video pulksteņa frekvence.	
nvml.system.driver.version	NVIDIA draivera versija.	
nvml.version	NVML bibliotēkas versija.	
oracle.diskgroups.stats	Atgriež Automatic Storage Management (ASM) disku grupu statistiku.	Oracle
oracle.diskgroups.discovery	Atgriež ASM disku grupu sarakstu.	
oracle.archive.info	Arhīva žurnālu statistika.	
oracle.cdb.info	Container Databases (CDB) informācija.	
oracle.custom.query	Pielāgota vaicājuma rezultāts.	
oracle.datafiles.stats	Atgriež datu failu statistiku.	
oracle.db.discovery	Atgriež datubāzu sarakstu.	
oracle.fra.stats	Atgriež Fast Recovery Area (FRA) statistiku.	
oracle.instance.info	Instances statistika.	
oracle.pdb.info	Pluggable Databases (PDB) informācija.	
oracle.pdb.discovery	Atgriež PDB sarakstu.	
oracle.pga.stats	Atgriež Program Global Area (PGA) statistiku.	
oracle.ping	Pārbauda, vai var izveidot savienojumu ar Oracle.	
oracle.proc.stats	Atgriež procesu statistiku.	
oracle.redolog.info	Žurnālfaila informācija no vadības faila.	
oracle.sga.stats	Atgriež System Global Area (SGA) statistiku.	
oracle.sessions.stats	Atgriež sesiju statistiku.	
oracle.sys.metrics	Atgriež sistēmas metriku vērtību kopu.	
oracle.sys.params	Atgriež sistēmas parametru vērtību kopu.	
oracle.ts.stats	Atgriež tabultelpu statistiku.	
oracle.ts.discovery	Atgriež tabultelpu sarakstu.	
oracle.user.info	Atgriež Oracle lietotāja informāciju.	
oracle.version	Atgriež datubāzes servera versiju.	
pgsql.autovacuum.count	Autovacuum darbinieku skaits.	PostgreSQL (ielādējams)
pgsql.archive	Informācija par arhivētajiem failiem.	
pgsql.bgwriter	Kopējais kontrolpunktu skaits datubāzes klasterī, sadalīts pēc kontrolpunkta tipa.	

Vienuma atslēga	Apraksts	Spraudnis
pgsql.cache.hit	PostgreSQL bufera kešatmiņas trāpījumu rādītājs.	
pgsql.connections	Atgriež savienojumus pēc tipa.	
pgsql.custom.query	Atgriež pielāgota vaicājuma rezultātu.	
pgsql.db.age	Datubāzes vecākā FrozenXID vecums.	
pgsql.db.bloating_tables	Uzpūsto tabulu skaits katrā datubāzē.	
pgsql.db.discovery	PostgreSQL datubāzu saraksts.	
pgsql.db.size	Datubāzes lielums baitos.	
pgsql.dbstat	Apkopo statistiku par katru datubāzi.	
pgsql.dbstat.sum	Apkopotie dati par visām klastera datubāzēm.	
pgsql.locks	Informācija par piešķirtajām bloķēšanām katrā datubāzē.	
pgsql.oldest.xid	Vecākā XID vecums.	
pgsql.ping	Pārbauda, vai savienojums ir aktīvs.	
pgsql.queries	Vaicājumu metrikas pēc izpildes laika.	
pgsql.replication.count	Gaidstāves serveru skaits.	
pgsql.replication.process	Flush lag, write lag un replay lag katram sūtītāja procesam.	
pgsql.replication.process.discovery	Replikācijas procesa nosaukumu atklāšana.	
pgsql.replication.recovery_role	Atkopšanas statuss.	
pgsql.replication.status	Replikācijas statuss.	
pgsql.replication_lag.b	Replikācijas aizture baitos.	
pgsql.replication_lag.sec	Replikācijas aizture sekundēs.	
pgsql.uptime	PostgreSQL darbības laiks milisekundēs.	
pgsql.version	Atgriež PostgreSQL versiju.	
pgsql.wal.stat	WAL statistika.	
redis.config	Iegūst Redis instances konfigurācijas parametrus, kas atbilst šablonam.	Redis
redis.info	Iegūst komandas INFO izvadi.	
redis.ping	Pārbauda, vai savienojums ir aktīvs.	
redis.slowlog.count	Lēnā žurnāla ierakstu skaits kopš Redis palaišanas.	
smart.attribute.discovery	Atgriež S.M.A.R.T. ierīču atribūtu sarakstu.	S.M.A.R.T.
smart.disk.discovery	Atgriež S.M.A.R.T. ierīču sarakstu.	
smart.disk.get	Atgriež visas pieejamās S.M.A.R.T. ierīču īpašības.	
systemd.unit.get	Atgriež visas systemd vienības īpašības.	Systemd
systemd.unit.info	Systemd vienības informācija.	
systemd.unit.discovery	Systemd vienību saraksts un to detalizētā informācija.	
web.certificate.get	Validē sertifikātus un atgriež sertifikātu detalizētu informāciju.	Web certificates

Vienuma atslēgas detaļas

Parametri bez leņķiekavām ir obligāti. Parametri, kas atzīmēti ar leņķiekavām **< >**, ir neobligāti.

ceph.df.details[connString,<user>,<apikey>]

 Klastera datu lietojums un sadalījums starp pūliem.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

ceph.osd.stats[connString,<user>,<apikey>]

 Apkopotā un katra OSD statistika.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

ceph.osd.discovery[connString,<user>,<apikey>]

 Atklāto OSD saraksts. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

ceph.osd.dump[connString,<user>,<apikey>]

 OSD lietojuma sliekšņi un statusi.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

ceph.ping[connString,<user>,<apikey>]

 Pārbauda, vai var izveidot savienojumu ar Ceph.
 Atgrieztā vērtība: *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas); *1* - savienojums ir veiksmīgs.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojuma izveidei ar Ceph.

ceph.pool.discovery[connString,<user>,<apikey>]

 Atklāto pūlu saraksts. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

ceph.status[connString,<user>,<apikey>]

 Kopējā klastera statuss.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, apikey** - lietotājvārds un API atslēga savienojumam ar Ceph.

docker.container_info[<ID>,<info>]

 Zema līmeņa informācija par konteineru.
 Atgrieztā vērtība: ContainerInspect API izsaukuma izvade, serializēta kā JSON.

Parametri:

- **ID** - konteinera ID vai nosaukums;
- **info** - atgrieztās informācijas apjoms. Atbalstītās vērtības: *short* (noklusējums) vai *full*.

Agent 2 lietotājs (’zabbix’) ir jāpievieno ’docker’ grupai, lai nodrošinātu pietiekamas privilēģijas. Pretējā gadījumā pārbaude neizdosies.

docker.container_stats[<ID>]

 Konteinera resursu izmantošanas statistika.
 Atgrieztā vērtība: ContainerStats API izsaukuma izvade un CPU izmantošanas procentuālā vērtība, serializēta kā JSON.

Parametrs:

- **ID** - konteinera ID vai nosaukums.

Agent 2 lietotājs (’zabbix’) ir jāpievieno ’docker’ grupai, lai nodrošinātu pietiekamas privilēģijas. Pretējā gadījumā pārbaude neizdosies.

docker.containers

 Konteineru saraksts.
 Atgrieztā vērtība: ContainerList API izsaukuma izvade, serializēta kā JSON.

Agent 2 lietotājs (’zabbix’) ir jāpievieno ’docker’ grupai, lai nodrošinātu pietiekamas privilēģijas. Pretējā gadījumā pārbaude neizdosies.

docker.containers.discovery[<options>]

 Atgriež konteineru sarakstu. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametrs:

- **options** - norāda, vai jāatklāj visi konteineri vai tikai darbojošies konteineri. Atbalstītās vērtības: *true* - atgriezt visus konteinerus; *false* - atgriezt tikai darbojošos konteinerus (noklusējums).

Lai nodrošinātu pietiekamas privilēģijas, Agent 2 lietotājs (’zabbix’) jāpievieno ’docker’ grupai. Pretējā gadījumā pārbaude neizdosies.

docker.data.usage

 Informācija par pašreizējo datu lietojumu.
 Atgrieztā vērtība: SystemDataUsage API izsaukuma izvade, serializēta kā JSON.

Lai nodrošinātu pietiekamas privilēģijas, Agent 2 lietotājs (’zabbix’) jāpievieno ’docker’ grupai. Pretējā gadījumā pārbaude neizdosies.

docker.images

 Atgriež attēlu sarakstu.
 Atgrieztā vērtība: ImageList API izsaukuma izvade, serializēta kā JSON.

Lai nodrošinātu pietiekamas privilēģijas, Agent 2 lietotājs (’zabbix’) jāpievieno ’docker’ grupai. Pretējā gadījumā pārbaude neizdosies.

docker.images.discovery

 Atgriež attēlu sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Lietotājs Agent 2 (’zabbix’) ir jāpievieno ’docker’ grupai, lai nodrošinātu pietiekamas privilēģijas. Pretējā gadījumā pārbaude neizdosies.

docker.info

 Sistēmas informācija.
 Atgrieztā vērtība: SystemInfo API izsaukuma izvade, serializēta kā JSON.

Lai nodrošinātu pietiekamas privilēģijas, Agent 2 lietotājs (’zabbix’) jāpievieno grupai ’docker’. Pretējā gadījumā pārbaude neizdosies.

docker.ping

 Pārbauda, vai Docker dēmons darbojas vai nē.
 Atgrieztā vērtība: *1* — savienojums darbojas; *0* — savienojums ir pārtraukts.

Aģenta 2 lietotājs (’zabbix’) ir jāpievieno ’docker’ grupai, lai nodrošinātu pietiekamas privilēģijas. Pretējā gadījumā pārbaude neizdosies.

ember.get[<uri>,<path>]

 Atgriež pieprasītās ierīces rezultātu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - Ember+ ierīces URI. Noklusējums: 127.0.0.1:9998;
- **path** - OID ceļš uz ierīci. Pēc noklusējuma tukšs, atgriež saknes kolekcijas datus.

memcached.ping[connString,<user>,<password>]

 Pārbauda, vai savienojums ir aktīvs vai ne.
 Atgrieztā vērtība: *1* — savienojums ir aktīvs; *0* — savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **connString** — URI vai sesijas nosaukums;
- **user, password** — Memcached pieteikšanās akreditācijas dati.

memcached.stats[connString,<user>,<password>,<type>]

 Iegūst komandas STATS izvadi.
 Atgrieztā vērtība: *JSON* — izvade tiek serializēta kā JSON.

Parametri:

- **connString** — URI vai sesijas nosaukums;
- **user, password** — Memcached pieteikšanās akreditācijas dati;
- **type** — atgriežamās statistikas tips: *items*, *sizes*, *slabs* vai *settings* (pēc noklusējuma tukšs, atgriež vispārīgo statistiku).

mongodb.collection.stats[connString,<user>,<password>,<database>,collection]

 Atgriež dažādu krātuves statistiku norādītajai kolekcijai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati;
- **database** - datubāzes nosaukums (noklusējums: admin);

- **collection** - kolekcijas nosaukums.

mongodb.cfg.discovery[connString,<user>,<password>]

 Atgriež atrasto konfigurācijas serveru sarakstu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.collections.discovery[connString,<user>,<password>]

 Atgriež atrasto kolekciju sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** — URI vai sesijas nosaukums;
- **user, password** — MongoDB pieteikšanās akreditācijas dati.

mongodb.collections.usage[connString,<user>,<password>]

 Atgriež kolekciju lietojuma statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.connpool.stats[connString,<user>,<password>]

 Atgriež informāciju par atvērtajiem izejošajiem savienojumiem no pašreizējās datubāzes instances uz citiem šardētā klastera vai repliku kopas dalībniekiem.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.db.stats[connString,<user>,<password>,<database>]

 Atgriež statistiku, kas atspoguļo dotās datubāzes sistēmas stāvokli.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati;
- **database** - datubāzes nosaukums (noklusējums: admin).

mongodb.db.discovery[connString,<user>,<password>]

 Atgriež atrasto datubāzu sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.jumbo_chunks.count[connString,<user>,<password>]

 Atgriež jumbo gabalu skaitu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.oplog.stats[connString,<user>,<password>]

 Atgriež replicēšanas kopas statusu, izmantojot no oplog nolasītos datus.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.ping[connString,<user>,<password>]

 Pārbauda, vai savienojums ir aktīvs vai nē.
 Atgrieztā vērtība: *1* - savienojums ir aktīvs; *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.rs.config[connString,<user>,<password>]

 Atgriež pašreizējo repliku kopas konfigurāciju.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.rs.status[connString,<user>,<password>]

 Atgriež repliku kopas statusu no tā dalībnieka skatpunkta, kurā metode tiek izpildīta.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.server.status[connString,<user>,<password>]

 Atgriež datubāzes stāvokli.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.sh.discovery[connString,<user>,<password>]

 Atgriež klasterī esošo atklāto šardu sarakstu.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mongodb.version[connString,<user>,<password>]

 Atgriež datubāzes servera versiju.
 Atgrieztā vērtība: *String*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MongoDB pieteikšanās akreditācijas dati.

mqtt.get[<broker url>,topic,<user>,<password>]

 Abonē norādītā brokera konkrētu tēmu vai tēmas (ar aizstājējzīmēm) un gaida publikācijas.
 Atgrieztā vērtība: atkarīga no tēmas satura. Ja tiek izmantotas aizstājējzīmes, atgriež tēmas saturu JSON formātā.

Parametri:

- **broker url** - MQTT brokera URL formātā `protocol://host:port` bez vaicājuma parametriem (atbalstītie protokoli: `tcp`, `ssl`, `ws`). Ja vērtība nav norādīta, aģents izmantos `tcp://localhost:1883`. Ja protokols vai ports nav norādīts, tiks izmantots noklusējuma protokols (`tcp`) vai ports (1883);
- **topic** - MQTT tēma (obligāta). Tiek atbalstītas aizstājējzīmes (+,#);
- **user, password** - autentifikācijas akreditācijas dati (ja nepieciešams).

Komentāri:

- vienums jākonfigurē kā aktīvā pārbaude ('Zabbix aģents (aktīvs)' vienuma tips);
- TLS šifrēšanas sertifikātus var izmantot, saglabājot tos noklusējuma atrašanās vietā (piemēram, Ubuntu sistēmā direktorijā `/etc/ssl/certs/`). TLS izmantošanai lietojiet `tls://` shēmu.

mssql.availability.group.get[URI,<user>,<password>]

 Atgriež pieejamības grupas.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **URI** - MSSQL servera URI (vienīgā atbalstītā shēma ir `sqlserver://`). Iegultie akreditācijas dati tiks ignorēti. Ir iespējams norādīt instances nosaukumu kā daļu no URI, piemēram: `sqlserver://localhost/InstanceName` (bez porta). Ja ir norādīts ports, instances nosaukums tiek ignorēts;

- **user, password** - lietotājvārds, parole nosūtīšanai uz aizsargātu MSSQL serveri.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.custom.query[URI,<user>,<password>,queryName,<args...>]

 Atgriež pielāgota vaicājuma rezultātu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **URI** - MSSQL servera URI (vienīgā atbalstītā shēma ir `sqlserver://`). Iegultie akreditācijas dati tiks ignorēti. Ir iespējams norādīt instances nosaukumu kā daļu no URI, piemēram: `sqlserver://localhost/InstanceName` (bez porta). Ja ir norādīts ports, instances nosaukums tiek ignorēts;
- **user, password** - lietotājvārds, parole nosūtīšanai uz aizsargātu MSSQL serveri;
- **queryName** - pielāgota vaicājuma nosaukums, kas konfigurēts `Plugins.MSSQL.CustomQueriesDir`, bez .sql paplašinājuma;
- **args** - viens vai vairāki ar komatiem atdalīti argumenti, kas jānodod vaicājumam.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.db.get

 Atgriež visas pieejamās MSSQL datubāzes.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.job.status.get

 Atgriež darbu statusu.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.last.backup.get

 Atgriež pēdējās dublējumkopijas laiku visām datubāzēm.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.local.db.get

 Atgriež datubāzes, kas piedalās Always On pieejamības grupā un replikā (primārajā vai sekundārajā) un atrodas serverī, ar kuru tika izveidots savienojums.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.mirroring.get

 Atgriež spoguļošanas informāciju.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.nonlocal.db.get

 Atgriež datubāzes, kas piedalās Always On pieejamības grupā un replikā (primārajā vai sekundārajā), kas atrodas citos serveros (datubāze nav lokāla SQL Server instancei, ar kuru tika izveidots savienojums).
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.perfcounter.get

 Atgriež veiktspējas skaitītājus.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.ping

 Pārbauda datubāzes sasniedzamību. Pārbauda, vai savienojums ir pareizi konfigurēts.
 Atgrieztā vērtība: *1* — darbojas, *0* — nedarbojas.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.quorum.get

 Atgriež quorum informāciju.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL spraudņa readme.

mssql.quorum.member.get

 Atgriež kvoruma dalībniekus.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL sprauduņa readme.

mssql.replica.get

 Atgriež replikas.
 Atgrieztā vērtība: *JSON objekts*.

Papildinformāciju skatiet MSSQL sprauduņa readme.

mssql.version

 Atgriež MSSQL versiju.
 Atgrieztā vērtība: *String*.

Papildinformāciju skatiet MSSQL sprauduņa readme.

mysql.custom.query[connString,<user>,<password>,queryName,<args...>]

 Atgriež pielāgota vaicājuma rezultātu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati;
- **queryName** - pielāgota vaicājuma nosaukums; tam jāsakrīt ar SQL faila nosaukumu bez paplašinājuma;
- **args** - viens vai vairāki ar komatiem atdalīti argumenti, kas jānodod vaicājumam.

Papildinformāciju skatiet MySQL sprauduņa readme faila sadaļā *Pielāgotie vaicājumi*.

mysql.db.discovery[connString,<user>,<password>]

 Atgriež MySQL datubāzu sarakstu. Tiek izmantots zema līmeņa atklāšanai.
 Atgrieztā vērtība: SQL vaicājuma ”show databases” rezultāts LLD JSON formātā.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati.

mysql.db.size[connString,<user>,<password>,<database name>]

 Datubāzes izmērs baitos.
 Atgrieztā vērtība: SQL vaicājuma ”select coalesce(sum(data_length + index_length),0) as size from information_schema.tables where table_schema=?” rezultāts konkrētajai datubāzei baitos.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati;
- **database name** - datubāzes nosaukums.

mysql.get_status_variables[connString,<user>,<password>]

 Globālo statusa mainīgo vērtības.
 Atgrieztā vērtība: SQL vaicājuma ”show global status” rezultāts JSON formātā.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati.

mysql.ping[connString,<user>,<password>]

 Pārbauda, vai savienojums ir aktīvs vai nē.
 Atgrieztā vērtība: *1* - savienojums ir aktīvs; *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati.

mysql.replication.discovery[connString,<user>,<password>]

 Atgriež MySQL replikāciju sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: `SHOW SLAVE STATUS` vai `SHOW REPLICA STATUS` SQL vaicājuma rezultāts LLD JSON formātā.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati.

mysql.replication.get_slave_status[connString,<user>,<password>,<master host>]

 Replikācijas statuss.
 Atgrieztā vērtība: `SHOW SLAVE STATUS` vai `SHOW REPLICA STATUS` SQL vaicājuma rezultāts JSON formātā.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati;
- **master host** - replikācijas galvenā hosta nosaukums. Ja nekas netiek atrasts, tiek atgriezta kļūda. Ja šis parametrs nav norādīts, tiek atgriezti visi hosti.

mysql.version[connString,<user>,<password>]

 MySQL versija.
 Atgrieztā vērtība: *String* (ar MySQL instances versiju).

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **user, password** - MySQL pieteikšanās akreditācijas dati.

net.dns.get[<ip>,name,<type>,<timeout>,<count>,<protocol>,"<flags>"]

Veic DNS vaicājumu un atgriež detalizētu DNS ierakstu informāciju.
 Šis vienums ir paplašināta `net.dns.record` Zabbix aģents vienuma versija ar atbalstu lielākam ierakstu tipu skaitam un pielāgojamiem karodziņiem.
 Atgrieztās vērtības: *JSON objekts*

Parametri:

- **ip** — DNS servera IP adrese (atstājiet tukšu, lai izmantotu noklusējuma DNS serveri);
- **name** — DNS nosaukums, kuram veikt vaicājumu;
- **type** — vaicājamā ieraksta tips (noklusējums ir *SOA*);
- **timeout** — pieprasījuma noildze sekundēs (noklusējums ir 1 sekunde);
- **count** — pieprasījuma mēģinājumu skaits (noklusējums ir 2);
- **protocol** — protokols, ko izmanto DNS vaicājumu veikšanai: *udp* (noklusējums) vai *tcp*;
- **flags** — viens vai vairāki ar komatiem atdalīti argumenti, kas jānodod vaicājumam.

Komentāri:

- Iespējamās `type` vērtības ir: *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*. Ņemiet vērā, ka vērtībām jābūt tikai ar lielajiem burtiem; vērtības ar mazajiem burtiem vai jauktā reģistrā netiek atbalstītas.
- Reversajiem DNS uzmeklējumiem (ja `type` ir iestatīts uz *PTR*) DNS nosaukumu var norādīt gan apgrieztā, gan neapgrieztā formātā (skatiet piemērus zemāk). Ņemiet vērā, ka, pieprasot PTR ierakstu, DNS nosaukums faktiski ir IP adrese.
- Iespējamās `flags` vērtības ir: *cdflag* vai *nocdflag* (noklusējums), *rdflag* (noklusējums) vai *nordflag*, *dnssec* vai *nodnssec* (noklusējums), *nsid* vai *nonsid* (noklusējums), *edns0* (noklusējums) vai *noedns0*, *aaflag* vai *noaaflag* (noklusējums), *adflag* vai *noadflag* (noklusējums). `flags` vērtības *dnssec* un *nsid* nevar izmantot kopā ar *noedns0*, jo abām ir nepieciešams *edns0*. Ņemiet vērā, ka vērtībām jābūt tikai ar mazajiem burtiem; vērtības ar lielajiem burtiem vai jauktā reģistrā netiek atbalstītas.
- Internacionalizēti domēnu nosaukumi netiek atbalstīti; tā vietā izmantojiet IDNA kodētus nosaukumus.
- Izvade ir objekts, kas satur DNS ierakstu informāciju atbilstoši norādītajiem parametriem (skatiet sīkāku informāciju).

Piemēri:

```
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

 GPU ierīču skaits.
 Atgrieztā vērtība: *vesels skaitlis*.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.decoder.utilization[<deviceUUID>]

 GPU ierīces dekodētāja noslodze procentos.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.ecc.mode[<deviceUUID>]

 GPU ierīces ECC režīma informācija (pašreizējais, gaidošais).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.encoder.stats.get[<deviceUUID>]

 GPU ierīces kodētāja statistika.
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** — GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.encoder.utilization[<deviceUUID>]

 GPU ierīces kodētāja izmantojums procentos.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.energy.consumption[<deviceUUID>]

 GPU ierīces kopējais enerģijas patēriņš milidžoulos (mJ) kopš pēdējās draivera pārlādes.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.errors.memory[<deviceUUID>]

 GPU ierīces ECC atmiņas kļūdu informācija (izlabotas, neizlabotas).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.errors.register[<deviceUUID>]

 GPU ierīces ECC reģistra kļūdu informācija (izlabotas, neizlabotas).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.fan.speed.avg[<deviceUUID>]

 GPU ierīces vidējais ventilatora ātrums procentos no maksimālā ātruma.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.get

 Atklātie GPU ar UUID un nosaukumu.
 Atgrieztā vērtība: *JSON*.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.graphics.frequency[<deviceUUID>]

 GPU ierīces grafiskā pulksteņa frekvence MHz.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.memory.fb.get[<deviceUUID>]

 GPU ierīces kadru bufera atmiņas statistika (kopā, rezervēta, brīva, izmantota).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.memory.bar1.get[<deviceUUID>]

 GPU ierīces BAR1 atmiņas statistika (kopā, brīva, izmantota).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.memory.frequency[<deviceUUID>]

 GPU ierīces atmiņas takts frekvence MHz.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.pci.utilization[<deviceUUID>]

 GPU ierīces PCI izmantojums (nosūtīšanas/saņemšanas caurlaidspēja KBps).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.performance.state[<deviceUUID>]

 GPU ierīces veiktspējas stāvoklis.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.power.limit[<deviceUUID>]

 GPU ierīces jaudas ierobežojums milivatos.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.power.usage[<deviceUUID>]

 GPU ierīces pašreizējais enerģijas patēriņš milivatos.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.serial[<deviceUUID>]

 GPU ierīces sērijas numurs.
 Atgrieztā vērtība: *String*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.sm.frequency[<deviceUUID>]

 GPU ierīces straumēšanas daudzprocesoru takts frekvence MHz.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.temperature[<deviceUUID>]

 GPU ierīces temperatūra Celsija grādos.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.utilization[<deviceUUID>]

 GPU ierīces izmantošanas statistika (GPU/atmiņas izmantošana procentos).
 Atgrieztā vērtība: *JSON*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.device.video.frequency[<deviceUUID>]

 GPU ierīces video pulksteņa frekvence MHz.
 Atgrieztā vērtība: *Integer*.

Parametrs:

- **deviceUUID** - GPU ierīces UUID.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.system.driver.version

 NVIDIA draivera versija.
 Atgrieztā vērtība: *String*.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

nvml.version

 NVML bibliotēkas versija.
 Atgrieztā vērtība: *String*.

Papildinformāciju skatiet NVIDIA GPU sprauduņa readme.

oracle.diskgroups.stats[connString,<user>,<password>,<service>,<diskgroup>]

 Atgriež Automatic Storage Management (ASM) disku grupu statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt beigu atstarpi).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **diskgroup** - vaicājamās ASM disku grupas nosaukums.

oracle.diskgroups.discovery[connString,<user>,<password>,<service>]

 Atgriež ASM disku grupu sarakstu. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un jebkurā gadījumā tas tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.archive.info[connString,<user>,<password>,<service>,<destination>]

 Arhīva žurnālu statistika.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **destination** - vaicājamā mērķa nosaukums.

oracle.cdb.info[connString,<user>,<password>,<service>,<database>]

 Konteinerdatubāzu (CDB) informācija.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** — tas var būt:
  - URI — ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums — ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga — ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
  - TNS vērtība — ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** — Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt beigu atstarpi).
- **password** — Oracle parole;
- **service** — Oracle servisa nosaukums;
- **database** — vaicājamās datubāzes nosaukums.

oracle.custom.query[connString,<user>,<password>,<service>,queryName,<args...>]

 Pielāgota vaicājuma rezultāts.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:

    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
    - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **queryName** - pielāgota vaicājuma nosaukums; tam jāsakrīt ar SQL faila nosaukumu bez paplašinājuma;
- **args** - viens vai vairāki ar komatiem atdalīti argumenti, kas jānodod vaicājumam.

Komentāri: - Atgrieztie dati tiek automātiski pārveidoti par JSON. - Izvairieties no JSON tiešas atgriešanas no vaicājumiem, jo tas tiks bojāts, kad spraudnis mēģinās to vēlreiz pārveidot par JSON.

oracle.datafiles.stats[connString,<user>,<password>,<service>]

 Atgriež datņu failu statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.db.discovery[connString,<user>,<password>,<service>]

 Atgriež datubāzu sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.fra.stats[connString,<user>,<password>,<service>]

 Atgriež Fast Recovery Area (FRA) statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;

    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.instance.info[connString,<user>,<password>,<service>]

 Instances statistika.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.pdb.info[connString,<user>,<password>,<service>,<database>]

 Informācija par Pluggable Databases (PDB).
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** — tas var būt:
    - URI — ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums — ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga — ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
    - TNS vērtība — ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** — Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** — Oracle parole;
- **service** — Oracle servisa nosaukums;
- **database** — vaicājamās datubāzes nosaukums.

oracle.pdb.discovery[connString,<user>,<password>,<service>]

 Atgriež PDB sarakstu. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** — tas var būt:
    - URI — ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums — ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
    - TNS atslēga — ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
    - TNS vērtība — ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** — Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** — Oracle parole;
- **service** — Oracle servisa nosaukums.

oracle.pga.stats[connString,<user>,<password>,<service>]

 Atgriež Program Global Area (PGA) statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un jebkurā gadījumā tas tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.ping[connString,<user>,<password>,<service>]

 Pārbauda, vai var izveidot savienojumu ar Oracle.
 Atgrieztā vērtība: *1* - savienojums ir veiksmīgs; *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
  - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

Piemēri:

```
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>]

 Atgriež procesu statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, ”tcp://”), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts sprauduņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu “(“ (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.redolog.info[connString,<user>,<password>,<service>]

 Žurnālfaila informācija no vadības faila.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.sga.stats[connString,<user>,<password>,<service>]

 Atgriež System Global Area (SGA) statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.sessions.stats[connString,<user>,<password>,<service>,<lockMaxTime>]

 Atgriež sesiju statistiku.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga, un tās beigās nedrīkst būt atstarpe).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **lockMaxTime** - maksimālais sesijas bloķēšanas ilgums sekundēs, lai sesiju uzskatītu par ilgstoši bloķētu. Noklusējums: 600 sekundes.

oracle.sys.metrics[connString,<user>,<password>,<service>,<duration>]

 Atgriež sistēmas metriku vērtību kopu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;

    - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **duration** - sistēmas metriku vērtību iegūšanas intervāls (sekundēs). Iespējamās vērtības: *60* — garš ilgums (noklusējums), *15* — īss ilgums.

oracle.sys.params[connString,<user>,<password>,<service>]

 Atgriež sistēmas parametru vērtību kopu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
    - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.ts.stats[connString,<user>,<password>,<service>,<tablespace>,<type>,<conname>]

 Atgriež tabulvietu statistiku.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **connString** - tas var būt:
    - URI - ja netiek atrasta sesija ar norādīto nosaukumu un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
    - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
    - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neattiecas neviens no iepriekš minētajiem nosacījumiem;
    - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **tablespace** - vaicājamās tabulvietas nosaukums. Noklusējums (ja atstāts tukšs un `type` ir iestatīts):
    - "TEMP" (ja `type` ir iestatīts uz "TEMPORARY");
    - "USERS" (ja `type` ir iestatīts uz "PERMANENT").
- **type** - vaicājamās tabulvietas tips. Noklusējums (ja `tablespace` ir iestatīts): "PERMANENT".
- **conname** - konteinera nosaukums, kuram ir nepieciešama informācija.

Ja `tablespace`, `type` vai `conname` nav norādīts, vienums atgriezīs tabulvietu statistiku visiem atbilstošajiem konteineriem (ieskaitot PDB un CDB).

oracle.ts.discovery[connString,<user>,<password>,<service>]

 Atgriež tabulvietu sarakstu. Tiek izmantots zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
    - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas

jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

oracle.user.info[connString,<user>,<password>,<service>,<username>]

 Atgriež Oracle lietotāja informāciju.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; tiek atbalstīta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošana formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums;
- **username** - lietotājvārds, kuram ir nepieciešama informācija. Lietotājvārdi ar mazajiem burtiem netiek atbalstīti. Noklusējums: pašreizējais lietotājs.

oracle.version[connString,<user>,<password>,<service>]

 Atgriež datubāzes servera versiju.
 Atgrieztā vērtība: *String*.

Parametri:

- **connString** - tas var būt:
  - URI - ja sesija ar norādīto nosaukumu netiek atrasta un `Plugins.Oracle.ResolveTNS` ir iestatīts uz false. Ja tas satur shēmu (piemēram, "tcp://"), portu (piemēram, 1521) vai abus, opcija ResolveTNS netiks ņemta vērā, un tas jebkurā gadījumā tiks apstrādāts kā URI;
  - sesijas nosaukums - ja šāds nosaukums ir atrasts spraudņa konfigurācijas failā;
  - TNS atslēga - ja opcija ResolveTNS ir iestatīta uz true un neviens no iepriekš minētajiem nosacījumiem nav spēkā;
  - TNS vērtība - ja tā sākas ar atverošo iekavu "(" (sākuma atstarpes tiek ignorētas). Šajā gadījumā opcija ResolveTNS netiek ņemta vērā.
- **user** - Oracle lietotājvārds; atbalsta vienas no pieteikšanās opcijām `as sysdba`, `as sysoper`, `as sysasm`, `as sysbackup`, `as sysdg`, `as syskm` vai `as sysrac` pievienošanu formātā `user as sysdba` (pieteikšanās opcija nav reģistrjutīga un nedrīkst saturēt atstarpi beigās).
- **password** - Oracle parole;
- **service** - Oracle servisa nosaukums.

pgsql.autovacuum.count[uri,<username>,<password>,<database name>]

 Autovacuum darbinieku skaits.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.archive[uri,<username>,<password>,<database name>]

 Informācija par arhivētajiem failiem.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;

- **database name** — datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.bgwriter[uri,<username>,<password>,<database name>]

 Kopējais kontrolpunktu skaits datubāzu klasterim, sadalīts pēc kontrolpunktu tipa.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;
- **database name** — datubāzes nosaukums (ja tas ir izlaists, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.cache.hit[uri,<username>,<password>,<database name>]

 PostgreSQL bufera kešatmiņas trāpījumu koeficients.
 Atgrieztā vērtība: *Float*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;
- **database name** — datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.connections[uri,<username>,<password>,<database name>]

 Atgriež savienojumus pēc tipa.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.custom.query[uri,<username>,<password>,queryName,<args...>]

 Atgriež pielāgota vaicājuma rezultātu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **queryName** - pielāgota vaicājuma nosaukums; tam jāsakrīt ar SQL faila nosaukumu bez paplašinājuma;
- **args** - viens vai vairāki ar komatiem atdalīti argumenti, ko nodot vaicājumam.

pgsql.db.age[uri,<username>,<password>,<database name>]

 Datubāzes vecākā FrozenXID vecums.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.db.bloating_tables[uri,<username>,<password>,<database name>]

 Uzpūsto tabulu skaits katrā datubāzē.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;
- **database name** — datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.db.discovery[uri,<username>,<password>,<database name>]

 PostgreSQL datubāzu saraksts. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;

- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.db.size[uri,<username>,<password>,<database name>]

 Datubāzes izmērs baitos.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;
- **database name** — datubāzes nosaukums (ja nav norādīts, tiek izmantota Plugins.PostgreSQL.Default.Database vērtība no postgresql.conf).

pgsql.dbstat[uri,<username>,<password>,<database name>]

 Apkopo statistiku par katru datubāzi. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.dbstat.sum[uri,<username>,<password>,<database name>]

 Apkopotie dati par visām datubāzēm klasterī.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja tas nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.locks[uri,<username>,<password>,<database name>]

 Informācija par piešķirtajām bloķēšanām katrā datubāzē. Izmanto zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.oldest.xid[uri,<username>,<password>,<database name>]

 Vecākā XID vecums.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.ping[uri,<username>,<password>,<database name>]

 Pārbauda, vai savienojums ir aktīvs vai nē.
 Atgrieztā vērtība: *1* - savienojums ir aktīvs; *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.queries[uri,<username>,<password>,<database name>,time period]

 Vaicājumu metrikas pēc izpildes laika.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** — URI vai sesijas nosaukums;
- **username, password** — PostgreSQL akreditācijas dati;
- **database name** — datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf);
- **time period** — izpildes laika ierobežojums lēno vaicājumu skaitam (jābūt pozitīvam veselam skaitlim).

pgsql.replication.count[uri,<username>,<password>]

 Gaidstāves serveru skaits.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication.process[uri,<username>,<password>]

 Flush aizture, rakstīšanas aizture un atskaņošanas aizture katram sūtītāja procesam.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication.process.discovery[uri,<username>,<password>]

 Replikācijas procesa nosaukuma atklāšana.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication.recovery_role[uri,<username>,<password>]

 Atkopšanas statuss.
 Atgrieztā vērtība: *0* - galvenā režīms; *1* - atkopšana joprojām turpinās (gaidstāves režīms).

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication.status[uri,<username>,<password>]

 Replikācijas statuss.
 Atgrieztā vērtība: *0* - straumēšana nedarbojas; *1* - straumēšana darbojas; *2* - galvenā mezgla režīms.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication_lag.b[uri,<username>,<password>]

 Replikācijas aizture baitos.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.replication_lag.sec[uri,<username>,<password>]

 Replikācijas aizture sekundēs.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati.

pgsql.uptime[uri,<username>,<password>,<database name>]

 PostgreSQL darbības laiks milisekundēs.
 Atgrieztā vērtība: *Float*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;

- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.version[uri,<username>,<password>,<database name>]

 Atgriež PostgreSQL versiju.
 Atgrieztā vērtība: *String*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

pgsql.wal.stat[uri,<username>,<password>,<database name>]

 WAL statistika.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **uri** - URI vai sesijas nosaukums;
- **username, password** - PostgreSQL akreditācijas dati;
- **database name** - datubāzes nosaukums (ja nav norādīts, tiek izmantota vērtība Plugins.PostgreSQL.Default.Database no postgresql.conf).

redis.config[connString,<password>,<pattern>]

 Iegūst Redis instances konfigurācijas parametrus, kas atbilst šablonam.
 Atgrieztā vērtība: *JSON* - ja tika izmantots glob stila šablons; viena vērtība - ja šablons nesaturēja nevienu aizstājējzīmi.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **password** - Redis parole;
- **pattern** - glob stila šablons (* pēc noklusējuma).

Komentārs:

- Drošības brīdinājums: `redis.config` izpilda Redis komandu `CONFIG GET` un var atgriezt sensitīvus konfigurācijas parametrus (piemēram, `requirepass`, `masterauth`, ar TLS saistītus iestatījumus un citus noslēpumus). Izmantojot noklusējuma šablonu (`*`) vai plašus šablonus, šie noslēpumi var tikt atklāti ikvienam, kam ir atļauja skatīt vienumu vērtības Zabbix.

redis.info[connString,<password>,<section>]

 Iegūst komandas INFO izvadi.
 Atgrieztā vērtība: *JSON* — izvade tiek serializēta kā JSON.

Parametri:

- **connString** — URI vai sesijas nosaukums;
- **password** — Redis parole;
- **section** — informācijas sadaļa (*default* pēc noklusējuma).

redis.ping[connString,<password>]

 Pārbauda, vai savienojums ir aktīvs vai nē.
 Atgrieztā vērtība: *1* - savienojums ir aktīvs; *0* - savienojums ir pārtraukts (ja ir radusies jebkāda kļūda, tostarp AUTH un konfigurācijas problēmas).

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **password** - Redis parole.

redis.slowlog.count[connString,<password>]

 Lēnā žurnāla ierakstu skaits kopš Redis palaišanas.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **connString** - URI vai sesijas nosaukums;
- **password** - Redis parole.

smart.attribute.discovery

 Atgriež S.M.A.R.T. ierīces atribūtu sarakstu.
 Atgrieztā vērtība: *JSON object*.

Komentāri:

- Tiek atgriezti šādi makrosi un to vērtības: {#NAME}, {#DISKTYPE}, {#ID}, {#ATTRNAME}, {#THRESH};
- Tiek atbalstīti HDD, SSD un NVME disku tipi. Diski var būt atsevišķi vai apvienoti RAID masīvā. {#NAME} būs ar papildinājumu RAID gadījumā, piemēram: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.discovery[<type>]

 Atgriež S.M.A.R.T. ierīču sarakstu.
 Atgrieztā vērtība: *JSON objekts*.

Parametrs:

- **type** — norāda vērtību, pēc kuras skenēt diskus. Iespējamās vērtības: *id*, *name* (noklusējums). *id* netiek atbalstīts Windows vidē, un, ja tas ir atlasīts, tiks atgriezta kļūda

Komentāri:

- Tiek atgrieztas šādas makrokomandas un to vērtības: {#NAME}, {#DISKTYPE}, {#MODEL}, {#SN}, {#PATH}, {#ATTRIBUTES}, {#RAIDTYPE};
- Tiek atbalstīti HDD, SSD un NVME disku tipi. Ja disks nepieder RAID, {#RAIDTYPE} būs tukšs. RAID gadījumā {#NAME} būs ar papildinājumu, piemēram: {"{#NAME}": "/dev/sda cciss,2"}.

smart.disk.get[<path>,<raid type>]

 Atgriež visas pieejamās S.M.A.R.T. ierīču īpašības.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **path** - diska ceļš; kā vērtību var izmantot makrosu {#PATH};
- **raid_type** - RAID tips; kā vērtību var izmantot makrosu {#RAID}

Komentāri:

- Tiek atbalstīti HDD, SSD un NVME disku tipi. Diski var būt atsevišķi vai apvienoti RAID masīvā;
- Dati ietver smartctl versiju un izsaukuma argumentus, kā arī papildu laukus:
disk_name - satur nosaukumu ar RAID atklāšanai nepieciešamajiem papildinājumiem, piemēram: {"disk_name": "/dev/sda cciss,2"}
disk_type - satur diska tipu HDD, SSD vai NVME, piemēram: {"disk_type": "ssd"};
- Ja nav norādīts neviens parametrs, vienums atgriezīs informāciju par visiem diskiem.

systemd.unit.get[vienuma nosaukums,<saskarne>]

 Atgriež visas systemd vienuma īpašības.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **vienuma nosaukums** - vienuma nosaukums (iespējams, vēlēsieties izmantot {#UNIT.NAME} makrosu vienuma prototipā, lai atklātu nosaukumu);
- **saskarne** - vienuma saskarnes tips, iespējamās vērtības: *Unit* (noklusējums), *Service*, *Socket*, *Device*, *Mount*, *Automount*, *Swap*, *Target*, *Path*.

Komentāri:

- Šis vienums tiek atbalstīts tikai Linux platformā;
- LoadState, ActiveState un UnitFileState saskarnei Unit tiek atgriezti kā teksts un vesels skaitlis: `"ActiveState":{"state":1,"text"`

systemd.unit.info[unit name,<property>,<interface>]

 Systemd vienuma informācija.
 Atgrieztā vērtība: *String*.

Parametri:

- **unit name** - vienuma nosaukums (iespējams, vēlēsieties izmantot makrosu {#UNIT.NAME} vienuma prototipā, lai atklātu nosaukumu);
- **property** - vienuma īpašība (piemēram, ActiveState (noklusējums), LoadState, Description);
- **interface** - vienuma saskarnes tips (piemēram, Unit (noklusējums), Socket, Service).

Komentāri:

- Šis vienums tiek atbalstīts tikai Linux platformā;
- Šis vienums ļauj iegūt noteiktu īpašību no noteikta saskarnes tipa, kā aprakstīts dbus API.

Piemēri:

```
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>]

 systemd vienību un to detalizētās informācijas saraksts. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON object*.

Parametrs:

- **type** — iespējamās vērtības: *all*, *automount*, *device*, *mount*, *path*, *service* (noklusējums), *socket*, *swap*, *target*.

Šis vienums tiek atbalstīts tikai Linux platformā.

web.certificate.get[hostname,<port>,<address>]

 Validē sertifikātus un atgriež sertifikāta detalizētu informāciju.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **hostname** — var būt gan IP, gan DNS.
Var ietvert URL shēmu (tikai *https*), ceļu (tas tiks ignorēts) un portu.
Ja ports ir norādīts gan pirmajā, gan otrajā parametrā, to vērtībām jāsakrīt.
Ja ir norādīts address (3. parametrs), hostname tiek izmantots tikai SNI un hosta nosaukuma pārbaudei;
- **port** — porta numurs (HTTPS noklusējuma vērtība ir 443);
- **address** — var būt gan IP, gan DNS. Ja tas ir norādīts, tas tiks izmantots savienojumam, bet hostname (1. parametrs) tiks izmantots SNI un hosta pārbaudei. Ja 1. parametrs ir IP un 3. parametrs ir DNS, 1. parametrs tiks izmantots savienojumam, bet 3. parametrs tiks izmantots SNI un hosta pārbaudei.

Komentāri:

- Šis vienums kļūst neatbalstīts, ja hosta konfigurācijā norādītais galamērķis neeksistē, nav pieejams vai ja TLS rokasspiediens neizdodas ar jebkādu kļūdu, izņemot nederīgu sertifikātu;
- Pašlaik netiek atbalstīts AIA (Authority Information Access) X.509 paplašinājums, CRL un OCSP (ieskaitot OCSP stapling), kā arī Certificate Transparency;
- JSON atbildes lauki:
  - *x509*: satur X.509 sertifikāta detalizētu informāciju.
    - *version*: X.509 versija (piemēram, 3).
    - *serial_number*: sertifikāta sērijas numurs.
    - *signature_algorithm*: algoritms, kas izmantots sertifikāta parakstīšanai (piemēram, `SHA256-RSA`).
    - *issuer*: sertifikāta izdevējs.
    - *not_before*: sertifikāta derīguma sākuma datums.
    - *not_after*: sertifikāta derīguma beigu datums.
    - *subject*: sertifikāta subjekts.
    - *public_key_algorithm*: algoritms, kas izmantots publiskajai atslēgai (piemēram, `RSA`).
    - *alternative_names*: subjekta alternatīvie nosaukumi (ja ir), pretējā gadījumā `null`.
  - result: satur validācijas rezultātu.
    - *value*: validācijas statuss (skatiet iespējamās vērtības zemāk).
    - *message*: detalizēts validācijas ziņojums (piemēram, `"certificate verified successfully"`).
  - *sha1_fingerprint*: sertifikāta SHA-1 nospiedums.
  - *sha256_fingerprint*: sertifikāta SHA-256 nospiedums.
- Lauks `$.result.value` norāda sertifikāta validācijas rezultātu. Iespējamās vērtības:
  - *valid* – sertifikāts ir derīgs un uzticams.
  - *valid-but-self-signed* – sertifikāts ir derīgs, bet pašparakstīts, tas nozīmē, ka tā subjekts sakrīt ar tā izdevēju.
  - *invalid* – sertifikāts nav derīgs tādas problēmas dēļ kā derīguma termiņa beigas, nepareizs hosta nosaukums vai nezināma parakstīšanas iestāde.

Piemērs:

```
web.certificate.get[example.com,443]
```

JSON atbilde:

```
{
  "x509": {
    "version": 3,
    "serial_number": "0ad893bafa68b0b7fb7a404f06ecaf9a",
    "signature_algorithm": "ECDSA-SHA384",
    "issuer": "CN=DigiCert Global G3 TLS ECC SHA384 2020 CA1,O=DigiCert Inc,C=US",
    "not_before": {
```

```
      "value": "Jan 15 00:00:00 2025 GMT",
      "timestamp": 1736899200
    },
    "not_after": {
      "value": "Jan 15 23:59:59 2026 GMT",
      "timestamp": 1768521599
    },
    "subject": "CN=*.example.com,O=Internet Corporation for Assigned Names and Numbers,L=Los Angeles,ST=Ca
    "public_key_algorithm": "ECDSA",
    "alternative_names": [
      "*.example.com",
      "example.com"
    ]
  },
  "result": {
    "value": "valid",
    "message": "certificate verified successfully"
  },
  "sha1_fingerprint": "310db7af4b2bc9040c8344701aca08d0c69381e3",
  "sha256_fingerprint": "455943cf819425761d1f950263ebf54755d8d684c25535943976f488bc79d23b"
}
```


2 Windows Zabbix aģents

Pārskats

Windows Zabbix aģenta vienumi ir parādīti divos sarakstos:

- Koplietotie vienumi - vienumu atslēgas, kas ir koplietotas ar UNIX Zabbix aģentu;
- Windows specifiskie vienumi - vienumu atslēgas, kas tiek atbalstītas **tikai** Windows vidē.

Ņemiet vērā, ka visas vienumu atslēgas, ko Windows vidē atbalsta Zabbix aģents, atbalsta arī jaunās paaudzes Zabbix aģents 2. Skatiet papildu vienumu atslēgas, kuras varat izmantot tikai ar aģentu 2.

Skatiet arī: Minimālās atļaujas Windows vienumiem

Koplietojamie vienumi

Tālāk esošajā tabulā ir uzskaitīti Zabbix aģenta vienumi, kas tiek atbalstīti Windows vidē un ir koplietojami ar UNIX Zabbix aģentu:

- Vienuma atslēga ir saite uz pilnu informāciju par UNIX Zabbix aģenta vienumu
- Ir iekļauti ar Windows saistītie vienumu komentāri

Item key	Description	Item group
log	Žurnālfaila uzraudzība. Šis vienums netiek atbalstīts Windows Event Log. Parametrs `persistent_dir` netiek atbalstīts Windows vidē.	Žurnālu uzraudzība
log.count	Atbilstošo rindu skaits uzraudzītā žurnālfailā. Šis vienums netiek atbalstīts Windows Event Log. Parametrs `persistent_dir` netiek atbalstīts Windows vidē.	
logrt	Rotēta žurnālfaila uzraudzība. Šis vienums netiek atbalstīts Windows Event Log. Parametrs `persistent_dir` netiek atbalstīts Windows vidē.	
logrt.count	Atbilstošo rindu skaits uzraudzītā rotētā žurnālfailā. Šis vienums netiek atbalstīts Windows Event Log. Parametrs `persistent_dir` netiek atbalstīts Windows vidē.	
modbus.get	Nolasa Modbus datus.	Modbus
net.dns	Pārbauda, vai DNS pakalpojums darbojas. Parametri `ip`, `timeout` un `count` Windows vidē tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	Tīkls
net.dns.perf	Pārbauda DNS pakalpojuma veiktspēju. Parametri `ip`, `timeout` un `count` Windows vidē tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	
net.dns.record	Izpilda DNS vaicājumu. Parametri `ip`, `timeout` un `count` Windows vidē tiek ignorēti, ja vien netiek izmantots Zabbix aģents 2.	

Item key	Description	Item group
net.if.discovery	Tīkla saskarņu saraksts. Dažām Windows versijām (piemēram, Server 2008) var būt nepieciešami jaunākie instalētie atjauninājumi, lai atbalstītu ne-ASCII rakstzīmes saskarņu nosaukumos.	
net.if.in	Ienākošās datplūsmas statistika tīkla saskarnē. Windows vidē vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu saskarņu statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Windows vidē tiek atbalstīti vairāku baitu saskarņu nosaukumi. Tīkla saskarņu aprakstus Windows vidē var iegūt ar net.if.discovery vai net.if.list vienumiem.	
net.if.out	Izejošās datplūsmas statistika tīkla saskarnē. Windows vidē vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu saskarņu statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Windows vidē tiek atbalstīti vairāku baitu saskarņu nosaukumi. Tīkla saskarņu aprakstus Windows vidē var iegūt ar net.if.discovery vai net.if.list vienumiem.	
net.if.total	Ienākošās un izejošās datplūsmas statistikas summa tīkla saskarnē. Windows vidē vienums iegūst vērtības no 64 bitu skaitītājiem, ja tie ir pieejami. 64 bitu saskarņu statistikas skaitītāji tika ieviesti Windows Vista un Windows Server 2008. Ja 64 bitu skaitītāji nav pieejami, aģents izmanto 32 bitu skaitītājus. Tīkla saskarņu aprakstus Windows vidē var iegūt ar net.if.discovery vai net.if.list vienumiem.	
net.tcp.listen	Pārbauda, vai šis TCP ports ir LISTEN stāvoklī.	
net.tcp.port	Pārbauda, vai ir iespējams izveidot TCP savienojumu ar norādīto portu.	
net.tcp.service	Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus. LDAP un HTTPS pārbaude Windows vidē tiek atbalstīta tikai ar Zabbix aģentu 2.	
net.tcp.service.perf	Pārbauda TCP pakalpojuma veiktspēju. LDAP un HTTPS pārbaude Windows vidē tiek atbalstīta tikai ar Zabbix aģentu 2.	
net.tcp.socket.count	Atgriež TCP ligzdu skaitu, kas atbilst parametriem. Šis vienums Linux vidē tiek atbalstīts ar Zabbix aģentu, bet Windows vidē tas tiek atbalstīts tikai ar Zabbix aģentu 2 64 bitu Windows sistēmās.	
net.udp.service	Pārbauda, vai pakalpojums darbojas un atbild uz UDP pieprasījumiem.	
net.udp.service.perf	Pārbauda UDP pakalpojuma veiktspēju.	
net.udp.socket.count	Atgriež UDP ligzdu skaitu, kas atbilst parametriem. Šis vienums Linux vidē tiek atbalstīts ar Zabbix aģentu, bet Windows vidē tas tiek atbalstīts tikai ar Zabbix aģentu 2 64 bitu Windows sistēmās.	
proc.get	OS procesu un to parametru saraksts. Parametrs `cmdline` netiek atbalstīts Windows vidē.	Procesi
proc.num	Procesu skaits. Windows vidē tiek atbalstīti tikai parametri `name` un `user`.	
system.cpu.discovery	Atklāto CPU/CPU kodolu saraksts.	Sistēma
system.cpu.load	CPU slodze. Kad Zabbix aģentā tiek palaists kolektora process, tiek inicializēti un vēlāk šim vienumam izmantoti šādi veiktspējas skaitītāji: `\System\Processor Queue Length`	
system.cpu.num	CPU skaits.	
system.cpu.util	CPU izmantojuma procentuālā vērtība. Vērtība tiek iegūta, izmantojot *Processor Time* veiktspējas skaitītāju. Ņemiet vērā, ka начиная ar Windows 8, Task Manager rāda CPU izmantojumu, pamatojoties uz *Processor Utility* veiktspējas skaitītāju, savukārt iepriekšējās versijās tas bija *Processor Time* skaitītājs (skatiet sīkāku informāciju). *system* ir vienīgais `type` parametrs, kas tiek atbalstīts Windows vidē.	
system.hostname	Sistēmas hosta nosaukums. Windows vidē šī vienuma vērtība tiek iegūta, izmantojot funkcijas GetComputerName() (priekš **netbios**), GetComputerNameExA() (priekš **fqdn**) vai gethostname() (priekš **host**). Skatiet arī detalizētāku aprakstu.	
system.localtime	Sistēmas laiks.	
system.run	Izpilda norādīto komandu hostā.	
system.sw.arch	Programmatūras arhitektūras informācija.	

Item key	Description	Item group
system.sw.os	Operētājsistēmas informācija.	
system.sw.os.get	Detalizēta informācija par operētājsistēmu (versija, tips, distribūcijas nosaukums, mazā un lielā versija utt.).	
system.swap.size	Maiņvietas lielums baitos vai procentos no kopējā apjoma. pused tipa parametrs Linux vidē tiek atbalstīts ar Zabbix aģentu, bet Windows vidē tas tiek atbalstīts tikai ar Zabbix aģentu 2. Ņemiet vērā, ka šī atslēga var ziņot nepareizu maiņvietas lielumu/procentuālo vērtību virtualizētās (VMware ESXi, VirtualBox) Windows platformās. Šādā gadījumā varat izmantot atslēgu `perf_counter[\700(_Total)\702]`, lai iegūtu pareizu maiņvietas procentuālo vērtību.	
system.uname	Sistēmas identifikācija. Windows vidē šī vienuma vērtība tiek iegūta no Win32_OperatingSystem un Win32_Processor WMI klasēm. OS nosaukums (ieskaitot izlaidumu) var būt tulkots lietotāja attēlošanas valodā. Dažās Windows versijās tas satur preču zīmju simbolus un papildu atstarpes.	
system.uptime	Sistēmas darbības laiks sekundēs.	
vfs.dir.count	Direktorija ierakstu skaits. Windows vidē direktoriju simboliskās saites tiek izlaistas, un cietās saites tiek skaitītas tikai vienu reizi.	Virtuālās failu sistēmas
vfs.dir.get	Direktorija ierakstu saraksts. Windows vidē direktoriju simboliskās saites tiek izlaistas, un cietās saites tiek skaitītas tikai vienu reizi.	
vfs.dir.size	Direktorija lielums. Windows vidē jebkura simboliskā saite tiek izlaista, un cietās saites tiek ņemtas vērā tikai vienu reizi.	
vfs.file.cksum	Faila kontrolsumma, kas aprēķināta pēc UNIX cksum algoritma.	
vfs.file.contents	Faila satura iegūšana.	
vfs.file.exists	Pārbauda, vai fails eksistē. Windows vidē dubultās pēdiņas ir jāekranē ar atpakaļslīpsvītru '\', un visa vienuma atslēga jāiekļauj dubultajās pēdiņās, izmantojot komandrindas utilītu zabbix_get.exe vai agent2 izsaukšanai. Ņemiet vērā, ka vienums Windows vidē var kļūt neatbalstīts, ja direktorijs tiek meklēts neeksistējošā direktorijā, piemēram, `vfs.file.exists[C:\no\dir,dir]` (kur 'no' neeksistē).	
vfs.file.get	Atgriež informāciju par failu. Atbalstītie failu tipi Windows vidē: parasts fails, direktorijs, simboliskā saite	
vfs.file.md5sum	Faila MD5 kontrolsumma.	
vfs.file.owner	Iegūst faila īpašnieku.	
vfs.file.regexp	Iegūst virkni failā.	
vfs.file.regmatch	Atrod virkni failā.	
vfs.file.size	Faila lielums.	
vfs.file.time	Faila laika informācija. Windows XP vidē `vfs.file.time[file,change]` var būt vienāds ar `vfs.file.time[file,access]`.	
vfs.fs.discovery	Pievienoto failu sistēmu saraksts ar to tipu un montēšanas opcijām. Makross {#FSLABEL} tiek atbalstīts Windows vidē.	
vfs.fs.get	Pievienoto failu sistēmu saraksts ar to tipu, pieejamo diska vietu, inode statistiku un montēšanas opcijām. Makross {#FSLABEL} tiek atbalstīts Windows vidē.	
vfs.fs.size	Diska vieta baitos vai procentos no kopējā apjoma.	
vm.memory.size	Atmiņas lielums baitos vai procentos no kopējā apjoma.	Virtuālā atmiņa
web.page.get	Iegūst tīmekļa lapas saturu.	Tīmekļa uzraudzība
web.page.perf	Pilnas tīmekļa lapas ielādes laiks.	
web.page.regexp	Atrod virkni tīmekļa lapā.	
agent.hostmetadata	Aģenta hosta metadati.	Zabbix
agent.hostname	Aģenta hosta nosaukums.	
agent.ping	Aģenta pieejamības pārbaude.	
agent.variant	Zabbix aģenta variants (Zabbix aģents vai Zabbix aģents 2).	
agent.version	Zabbix aģenta versija.	

Item key	Description	Item group
zabbix.stats	Attālināti atgriež Zabbix servera vai starpniekservera iekšējo metriku kopu.	
zabbix.stats	Attālināti atgriež uzraudzīto vienumu skaitu rindā, kuru apstrāde Zabbix serverī vai starpniekserverī kavējas.	

Tikai Windows paredzētie vienumi

Tabulā ir sniegta informācija par vienumu atslēgām, kuras atbalsta **tikai** Windows Zabbix aģents.

Tikai Windows paredzētie vienumi dažkārt ir aptuvens līdzīga aģenta vienuma ekvivalents; piemēram, `proc_info`, ko atbalsta Windows, aptuveni atbilst `proc.mem` vienumam, kas netiek atbalstīts Windows vidē.

Vienuma atslēga ir saite uz pilnu informāciju par vienuma atslēgu.

Item key	Description	Item group
eventlog	Windows notikumu žurnāla uzraudzība.	Žurnālu uzraudzība
eventlog.count	Rindu skaits Windows notikumu žurnālā.	
net.if.list	Tīkla saskarņu saraksts (ietver saskarnes tipu, statusu, IPv4 adresi, aprakstu).	Tīkls
perf_counter	Jebkura Windows veiktspējas skaitītāja vērtība.	Veiktspējas skaitītāji
perf_counter_en	Jebkura Windows veiktspējas skaitītāja vērtība angļu valodā.	
perf_instance.discovery	Windows veiktspējas skaitītāju objektu instanču saraksts.	
perf_instance_en.discovery	Windows veiktspējas skaitītāju objektu instanču saraksts, kas noteikts, izmantojot objektu nosaukumus angļu valodā.	
proc_info	Dažāda informācija par konkrētu procesu vai procesiem.	Procesi
registry.data	Atgriež datus norādītajam vērtības nosaukumam Windows reģistra atslēgā.	Reģistrs
registry.get	Windows reģistra vērtību vai atslēgu saraksts, kas atrodas norādītajā atslēgā.	
service.discovery	Windows pakalpojumu saraksts.	Pakalpojumi
service.info	Informācija par pakalpojumu.	
services	Pakalpojumu uzskaitījums.	
vm.vmemory.size	Virtuālās atmiņas lielums baitos vai procentos no kopējā apjoma.	Virtuālā atmiņa
wmi.get	Izpilda WMI vaicājumu un atgriež pirmo atlasīto objektu.	WMI
wmi.getall	Izpilda WMI vaicājumu un atgriež visu atbildi.	

Vienuma atslēgas detaļas

Parametri bez leņķiekavām ir obligāti. Parametri, kas apzīmēti ar leņķiekavām **< >**, ir neobligāti.

eventlog[name,<regexp>,<severity>,<source>,<eventid>,<maxlines>,<mode>]

 Notikumu žurnāla uzraudzība.
 Atgrieztā vērtība: *Log*.

Parametri:

- **name** - notikumu žurnāla kanāla nosaukums (*Log Name* notikumu skatītāja GUI);
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu (reģistrjutīga);
- **severity** - regulāra izteiksme, kas apraksta svarīguma līmeni (nereģistrjutīga). Šis parametrs pieņem regulāru izteiksmi, kas balstīta uz šādām vērtībām: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (darbojas uz Windows Vista vai jaunākas versijas).
- **source** - regulāra izteiksme, kas apraksta avota identifikatoru (nereģistrjutīga);
- **eventid** - regulāra izteiksme, kas apraksta notikuma identifikatoru(-us) (reģistrjutīga);
- **maxlines** - maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveris vai starpniekserveris. Šis parametrs pārraksta ’MaxLinesPerSecond’ vērtību failā zabbix_agentd.conf.
- **mode** - iespējamās vērtības: *all* (noklusējums) vai *skip* - izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem).

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvā pārbaude;
- Aģents nevar nosūtīt notikumus no ”Forwarded events” žurnāla;
- Tiek atbalstīts Windows Eventing 6.0;
- Izvēloties šim vienumam informācijas tipu, kas nav Log, tiks zaudēts lokālais laikspiedols, kā arī žurnāla svarīguma līmeņa un avota informācija;

- Skatiet arī papildu informāciju par žurnālu uzraudzību.

Piemēri:

```
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] #šeit ir norādīta pielāgota regulāra izteiksme ar nosaukumu `TWOSHORT` (defi
```

eventlog.count[name,<regexp>,<severity>,<source>,<eventid>,<maxproclines>,<mode>]

 Rindu skaits Windows notikumu žurnālā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **name** - notikumu žurnāla kanāla nosaukums (*Log Name* notikumu skatītāja GUI);
- **regexp** - regulāra izteiksme, kas apraksta nepieciešamo šablonu (reģistrjutīga);
- **severity** - regulāra izteiksme, kas apraksta nopietnības līmeni (nereģistrjutīga). Šis parametrs pieņem regulāru izteiksmi, kas balstīta uz šādām vērtībām: ”Information”, ”Warning”, ”Error”, ”Critical”, ”Verbose” (darbojoties uz Windows Vista vai jaunākas versijas).
- **source** - regulāra izteiksme, kas apraksta avota identifikatoru (nereģistrjutīga);
- **eventid** - regulāra izteiksme, kas apraksta notikuma identifikatoru(-us) (reģistrjutīga);
- **maxproclines** - maksimālais jauno rindu skaits sekundē, ko aģents analizēs (nedrīkst pārsniegt 10000). Noklusējuma vērtība ir 10*’MaxLinesPerSecond’ failā zabbix_agentd.conf.
- **mode** - iespējamās vērtības: *all* (noklusējums) vai *skip* - izlaist vecāku datu apstrādi (attiecas tikai uz jaunizveidotiem vienumiem).

Komentāri:

- Vienumam jābūt konfigurētam kā aktīvā pārbaude;
- aģents nevar nosūtīt notikumus no ”Forwarded events” žurnāla;
- Tiek atbalstīts Windows Eventing 6.0;
- Izvēloties šim vienumam informācijas tipu, kas nav Log type of information, tiks zaudēts lokālais laikspiedols, kā arī žurnāla nopietnības līmeņa un avota informācija;
- Skatiet arī papildu informāciju par žurnālu uzraudzību.

Piemēri:

```
eventlog.count[System,,"Warning|Error"]
eventlog.count[Windows PowerShell,,,,,,skip]
```

net.if.list

 Tīkla saskarņu saraksts (ietver saskarnes tipu, statusu, IPv4 adresi, aprakstu).
 Atgrieztā vērtība: *Teksts*.

Komentāri:

- Tiek atbalstīti vairāku baitu saskarņu nosaukumi;
- Atspējotās saskarnes netiek uzskaitītas;
- Dažu komponentu iespējošana/atspējošana var mainīt to secību Windows saskarnes nosaukumā;
- Dažām Windows versijām (piemēram, Server 2008) var būt nepieciešami instalēti jaunākie atjauninājumi, lai atbalstītu ne-ASCII rakstzīmes saskarņu nosaukumos.

perf_counter[counter,<interval>]

 Jebkura Windows veiktspējas skaitītāja vērtība.
 Atgrieztā vērtība: *Integer*, *float*, *string* vai *text* (atkarībā no pieprasījuma).

Parametri:

- **counter** - ceļš uz skaitītāju;
- **interval** - pēdējās N sekundes vidējās vērtības glabāšanai. `interval` jābūt no 1 līdz 900 sekundēm (ieskaitot), un noklusējuma vērtība ir 1.

Komentāri:

- Šis vienums seko UNC ceļiem, kas var radīt drošības risku;
- `interval` tiek izmantots skaitītājiem, kuriem nepieciešams vairāk nekā viens paraugs (piemēram, CPU noslodze), tāpēc pārbaude katru reizi atgriež vidējo vērtību par pēdējām ”interval” sekundēm;

- Veiktspējas pārraugu var izmantot, lai iegūtu pieejamo skaitītāju sarakstu.
- Skatiet arī: Windows veiktspējas skaitītāji.

perf_counter_en[counter,<interval>]

 Jebkura Windows veiktspējas skaitītāja vērtība angļu valodā.
 Atgrieztā vērtība: *Integer*, *float*, *string* vai *text* (atkarībā no pieprasījuma).

Parametri:

- **counter** — ceļš uz skaitītāju angļu valodā;
- **interval** — pēdējās N sekundes vidējās vērtības glabāšanai. `interval` jābūt no 1 līdz 900 sekundēm (ieskaitot), un noklusējuma vērtība ir 1.

Komentāri:

- Šis vienums izmanto UNC ceļus, kas var radīt drošības risku;
- `interval` tiek izmantots skaitītājiem, kuriem nepieciešams vairāk nekā viens paraugs (piemēram, CPU noslodze), tāpēc pārbaude katru reizi atgriež vidējo vērtību par pēdējām ”interval” sekundēm;
- Šis vienums tiek atbalstīts tikai **Windows Server 2008/Vista** un jaunākās versijās;
- Angļu virkņu sarakstu var atrast, apskatot šādu reģistra atslēgu: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

perf_instance.discovery[object]

 Windows veiktspējas skaitītāju objektu instanču saraksts. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametrs:

- **object** - objekta nosaukums (lokalizēts).

perf_instance_en.discovery[object]

 Windows veiktspējas skaitītāju objektu instanču saraksts, kas atklāts, izmantojot objektu nosaukumus angļu valodā. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

Parametrs:

- **object** - objekta nosaukums (angļu valodā).

proc_info[process,<attribute>,<type>]

 Dažāda informācija par konkrētu procesu(-iem).
 Atgrieztā vērtība: *Float*.

Parametri:

- **process** - procesa nosaukums;
- **attribute** - pieprasītais procesa atribūts;
- **type** - attēlojuma tips (nozīmīgs, ja pastāv vairāk nekā viens process ar vienādu nosaukumu)

Komentāri:

- Tiek atbalstīti šādi `attributes`:
vmsize (noklusējums) - procesa virtuālās atmiņas izmērs Kbaitos
wkset - procesa darba kopas izmērs (procesa izmantotās fiziskās atmiņas apjoms) Kbaitos
pf - lappušu kļūdu skaits
ktime - procesa kodola laiks milisekundēs
utime - procesa lietotāja laiks milisekundēs
io_read_b - procesa I/O operāciju laikā nolasīto baitu skaits
io_read_op - procesa veikto lasīšanas operāciju skaits
io_write_b - procesa I/O operāciju laikā ierakstīto baitu skaits
io_write_op - procesa veikto rakstīšanas operāciju skaits
io_other_b - procesa pārsūtīto baitu skaits operāciju laikā, kas nav lasīšanas un rakstīšanas operācijas
io_other_op - procesa veikto I/O operāciju skaits, izņemot lasīšanas un rakstīšanas operācijas
gdiobj - procesa izmantoto GDI objektu skaits
userobj - procesa izmantoto USER objektu skaits;
- Derīgie `types` ir:
avg (noklusējums) - vidējā vērtība visiem procesiem ar nosaukumu <process>
min - minimālā vērtība starp visiem procesiem ar nosaukumu <process>
max - maksimālā vērtība starp visiem procesiem ar nosaukumu <process>
sum - vērtību summa visiem procesiem ar nosaukumu <process>;
- 64 bitu sistēmā šī vienuma pareizai darbībai ir nepieciešams 64 bitu Zabbix aģents.

Piemēri:

```
proc_info[iexplore.exe,wkset,sum] #iegūt visu Internet Explorer procesu izmantotās fiziskās atmiņas apjomu
proc_info[iexplore.exe,pf,avg] #iegūt Internet Explorer procesu vidējo lappušu kļūdu skaitu
```

registry.data[key,<value name>]

 Atgriež datus norādītajam vērtības nosaukumam Windows reģistra atslēgā.
 Atgrieztā vērtība: *Integer*, *string* vai *text* (atkarībā no vērtības tipa)

Parametri:

- **key** - reģistra atslēga, ieskaitot saknes atslēgu; ir atļauti saknes saīsinājumi (piemēram, HKLM);
- **value name** - reģistra vērtības nosaukums atslēgā (pēc noklusējuma tukša virkne ""). Ja vērtības nosaukums nav norādīts, tiek atgriezta noklusējuma vērtība.

Komentāri:

- Atbalstītie saknes saīsinājumi:
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
- Atslēgas ar atstarpēm jāliek dubultpēdiņās.

Piemēri:

```
registry.data["HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting"] #atgriež šīs atslēg
registry.data["HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting","EnableZip"] #atgriež šīs atslēgas
```

registry.get[key,<mode>,<name regexp>]

 Windows reģistra vērtību vai atslēgu saraksts, kas atrodas norādītajā atslēgā.
 Atgrieztā vērtība: *JSON object*.

Parametri:

- **key** — reģistra atslēga, ieskaitot saknes atslēgu; ir atļauti saknes saīsinājumi (piemēram, HKLM) (skatiet komentārus pie registry.data[], lai redzētu pilnu saīsinājumu sarakstu);
- **mode** — iespējamās vērtības:
values (noklusējums) vai *keys*;
- **name regexp** — atklāt tikai tās vērtības, kuru nosaukumi atbilst regulārajai izteiksmei (noklusējums — atklāt visas vērtības). Atļauts tikai ar *values* kā `mode`.

Atslēgām ar atstarpēm jābūt ietvertām dubultpēdiņās.

Piemēri:

```
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values,"^DisplayName|DisplayVersion$
JSON iekļaus informāciju par atslēgu, pēdējo apakšatslēgu, vērtības nosaukumu, vērtības tipu un vērtības d
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,values] #atgriež visu šajā atslēgā e
JSON iekļaus informāciju par atslēgu, pēdējo apakšatslēgu, vērtības nosaukumu, vērtības tipu un vērtības d
registry.get[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall,keys] #atgriež visas šīs atslēgas ap
JSON iekļaus informāciju par atslēgu un pēdējo apakšatslēgu.
```

service.discovery

 Windows servisu saraksts. Izmanto zemā līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*.

service.info[service,<param>]

 Informācija par servisu.
 Atgrieztā vērtība: *Integer* - ja `param` ir *state*, *startup*; *String* - ja `param` ir *displayname*, *path*, *user*; *Text* - ja `param` ir *description*
Konkrēti attiecībā uz *state*: 0 - darbojas, 1 - pauzēts, 2 - gaida palaišanu, 3 - gaida pauzēšanu, 4 - gaida turpināšanu, 5 - gaida apturēšanu, 6 - apturēts, 7 - nezināms, 255 - šāds serviss neeksistē
Konkrēti attiecībā uz *startup*: 0 - automātisks, 1 - automātisks ar aizkavi, 2 - manuāls, 3 - atspējots, 4 - nezināms, 5 - automātiska palaišana pēc trigera, 6 - automātiska palaišana ar aizkavi pēc trigera, 7 - manuāla palaišana pēc trigera

Parametri:

- **service** - reāls servisa nosaukums vai tā parādāmais nosaukums, kā redzams MMC Services papildmodulī;
- **param** - *state* (noklusējums), *displayname*, *path*, *user*, *startup* vai *description*.

Komentāri:

- Tādi vienumi kā `service.info[service,state]` un `service.info[service]` atgriezīs vienu un to pašu informāciju;
- Tikai tad, ja `param` ir *state*, šis vienums atgriež vērtību neeksistējošiem servisiem (255).

Piemēri:

```
service.info[SNMPTRAP] - SNMPTRAP servisa stāvoklis;
service.info[SNMP Trap] - tā paša servisa stāvoklis, bet norādot parādāmo nosaukumu;
service.info[EventLog,startup] - EventLog servisa palaišanas tips
```

services[<type>,<state>,<exclude>]

 Pakalpojumu saraksts.
 Atgrieztā vērtība: *0* - ja tukšs; *Text* - pakalpojumu saraksts, atdalīts ar jaunu rindu.

Parametri:

- **type** - *all* (noklusējums), *automatic*, *manual* vai *disabled*;
- **state** - *all* (noklusējums), *stopped*, *started*, *start_pending*, *stop_pending*, *running*, *continue_pending*, *pause_pending* vai *paused*;
- **exclude** - pakalpojumi, kas jāizslēdz no rezultāta. Izslēdzamie pakalpojumi jānorāda dubultpēdiņās, atdalot ar komatu, bez atstarpēm.

Piemēri:

```
services[,started] #atgriež palaisto pakalpojumu sarakstu;
services[automatic, stopped] #atgriež apturēto pakalpojumu sarakstu, kuriem vajadzētu darboties;
services[automatic, stopped, "service1,service2,service3"] #atgriež apturēto pakalpojumu sarakstu, kuriem
```

vm.vmemory.size[<type>]

 Virtuālās atmiņas lielums baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *float* - procentiem.

Parametrs:

- **type** - iespējamās vērtības: *available* (pieejamā virtuālā atmiņa), *pavailable* (pieejamā virtuālā atmiņa procentos), *pused* (izmantotā virtuālā atmiņa procentos), *total* (kopējā virtuālā atmiņa, noklusējums) vai *used* (izmantotā virtuālā atmiņa)

Komentāri:

- Virtuālās atmiņas statistikas uzraudzība ir balstīta uz:
  - Kopējo virtuālo atmiņu Windows sistēmā (kopējā fiziskā atmiņa + lappušošanas faila lielums);
  - Maksimālo atmiņas apjomu, ko Zabbix aģents var rezervēt;
  - Pašreizējo rezervētās atmiņas ierobežojumu sistēmai vai Zabbix aģentam atkarībā no tā, kurš ir mazāks.

Piemērs:

```
vm.vmemory.size[pavailable] #atgriež pieejamo virtuālo atmiņu procentos
```

wmi.get[<namespace>,<query>]

 Izpildīt WMI vaicājumu un atgriezt pirmo atlasīto objektu.
 Atgrieztā vērtība: *Integer*, *float*, *string* vai *text* (atkarībā no pieprasījuma).

Parametri:

- **namespace** - WMI nosaukumvieta;
- **query** - WMI vaicājums, kas atgriež vienu objektu.

WMI vaicājumi tiek izpildīti ar WQL.

Piemērs:

```
wmi.get[root\cimv2,select status from Win32_DiskDrive where Name like '%PHYSICALDRIVE0%'] #atgriež pirmā f
```

wmi.getall[<namespace>,<query>]

 Izpildīt WMI vaicājumu un atgriezt visu atbildi. Var izmantot zema līmeņa atklāšanai.
 Atgrieztā vērtība: *JSON objekts*

Parametri:

- **namespace** - WMI nosaukumtelpa;
- **query** - WMI vaicājums.

Komentāri:

- WMI vaicājumi tiek izpildīti ar WQL.
- JSONPath priekšapstrādi var izmantot, lai norādītu uz konkrētākām vērtībām atgrieztajā JSON.

Piemērs:

```
wmi.getall[root\cimv2,select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'] #atgriež fizisko di
```

Windows pakalpojumu uzraudzība

Šī pamācība sniedz soli pa solim norādījumus Windows pakalpojumu uzraudzības iestatīšanai. Tiek pieņemts, ka Zabbix serveris un aģents ir konfigurēti un darbojas.

1. solis

Iegūstiet servisa nosaukumu.

Jūs varat iegūt servisa nosaukumu, atverot MMC Services papildprogrammu un atverot servisa rekvizītus. Cilnē *General* jums vajadzētu redzēt lauku ar nosaukumu "Service name". Tālāk norādītā vērtība ir nosaukums, ko izmantosiet, iestatot vienumu uzraudzībai. Piemēram, ja vēlaties uzraudzīt "workstation" servisu, tad jūsu serviss varētu būt: **lanmanworkstation**.

2. darbība

Konfigurējiet vienumu pakalpojuma uzraudzībai.

Vienums `service.info[service,<param>]` izgūst informāciju par konkrētu pakalpojumu. Atkarībā no nepieciešamās informācijas norādiet opciju `param`, kas pieņem šādas vērtības: *displayname*, *state*, *path*, *user*, *startup* vai *description*. Noklusējuma vērtība ir *state*, ja `param` nav norādīts (`service.info[service]`).

Atgrieztās vērtības tips ir atkarīgs no izvēlētā `param`: vesels skaitlis priekš *state* un *startup*; rakstzīmju virkne priekš *displayname*, *path* un *user*; teksts priekš *description*.

Piemērs:

- *Atslēga:* `service.info[lanmanworkstation]`
- *Informācijas tips:* Numeric (unsigned)

Vienums `service.info[lanmanworkstation]` izgūs informāciju par pakalpojuma stāvokli kā skaitlisku vērtību. Lai lietotāja saskarnē skaitlisku vērtību attēlotu kā tekstuālu reprezentāciju ("0" kā "Running", "1" kā "Paused" utt.), varat konfigurēt vērtību kartēšanu hostā, kurā vienums ir konfigurēts. Lai to izdarītu, vai nu piesaistiet veidni *Windows services by Zabbix agent* vai *Windows services by Zabbix agent active* hostam, vai arī konfigurējiet hostā jaunu vērtību karti, kas balstīta uz minētajās veidnēs konfigurēto vērtību karti *Windows service state*.

Ņemiet vērā, ka abās minētajās veidnēs ir konfigurēts atklāšanas noteikums, kas automātiski atklās pakalpojumus. Ja to nevēlaties, varat atspējot atklāšanas noteikumu hosta līmenī pēc tam, kad veidne ir piesaistīta hostam.

Windows pakalpojumu atklāšana

Zema līmeņa atklāšana nodrošina veidu, kā automātiski izveidot vienumus, trigerus un grafikus dažādām datora entītijām. Zabbix var automātiski sākt uzraudzīt Windows pakalpojumus jūsu datorā, bez nepieciešamības zināt precīzu pakalpojuma nosaukumu vai manuāli izveidot vienumus katram pakalpojumam. Filtru var izmantot, lai ģenerētu reālus vienumus, trigerus un grafikus tikai tiem pakalpojumiem, kas jūs interesē.

2 SNMP aģents

Pārskats

Iespējams, vēlēsieties izmantot SNMP uzraudzību tādās ierīcēs kā printeri, tīkla komutatori, maršrutētāji vai UPS, kurām parasti ir iespējots SNMP un kurās būtu nepraktiski mēģināt uzstādīt pilnvērtīgas operētājsistēmas un Zabbix aģentus.

Lai varētu iegūt datus, ko šajās ierīcēs nodrošina SNMP aģenti, Zabbix serverim sākotnēji jābūt konfigurētam ar SNMP atbalstu, norādot karodziņu `--with-net-snmp`. Ieteicams arī instalēt MIB failus, lai nodrošinātu, ka vienumu vērtības tiek attēlotas pareizajā formātā. Bez MIB failiem var rasties formatēšanas problēmas, piemēram, vērtību attēlošana HEX formātā UTF-8 vietā vai otrādi.

SNMP pārbaudes tiek veiktas tikai, izmantojot UDP protokolu.

Zabbix servera un starpniekservera dēmoni reģistrē šādām līdzīgas žurnāla rindas, ja tie saņem nepareizu SNMP atbildi:

```
SNMP response from host "gateway" does not contain all of the requested variable bindings
```

Lai gan tās neaptver visus problemātiskos gadījumus, tās ir noderīgas, lai identificētu atsevišķas SNMP ierīces, kurām būtu jāatspējo kombinētie pieprasījumi.

Zabbix serveris/starpniekserveris atkārtos SNMP `walk` un `get` vienumu pārbaudi līdz 5 reizēm. Atkārtošanas mehānisms neattiecas uz DNS atrisināšanas kļūmēm.

Mantotajām SNMP pārbaudēm (viens OID numurs vai virkne) Zabbix serveris/starpniekserveris pēc neveiksmīga vaicājuma mēģinājuma atkārtos pārbaudi vismaz vienu reizi: vai nu izmantojot SNMP bibliotēkas atkārtošanas mehānismu, vai arī izmantojot iekšējo kombinētās apstrādes mehānismu.

Warning:
Ja uzraugāt SNMPv3 ierīces, pārliecinieties, ka msgAuthoritativeEngineID (zināms arī kā snmpEngineID vai "Engine ID") nekad netiek koplietots starp divām ierīcēm. Saskaņā ar RFC 2571 (3.1.1.1. sadaļa) tam jābūt unikālam katrai ierīcei.

Warning:
RFC3414 pieprasa, lai SNMPv3 ierīces saglabātu savu engineBoots vērtību. Dažas ierīces to nedara, kā rezultātā to SNMP ziņojumi pēc restartēšanas tiek noraidīti kā novecojuši. Šādā situācijā SNMP kešatmiņa serverī/starpniekserverī ir manuāli jānotīra (izmantojot -R snmp_cache_reload) vai arī serveris/starpniekserveris ir jāpārstartē.

Note:
Zabbix kešo SNMPv3 EngineID → IP atbilstības un atkārtoti izmanto kešotos EngineID turpmākajām pārbaudēm, nevis katru reizi nosūta zondēšanas pieprasījumu, tādējādi samazinot tīkla datplūsmu. Ja EngineID nevar izmantot atkārtoti, tiks veikts atkārtots mēģinājums ar zondēšanu, lai noteiktu jauno EngineID.

SNMP uzraudzības konfigurēšana

Lai sāktu ierīces uzraudzību, izmantojot SNMP, ir jāveic šādas darbības:

1. darbība

Noskaidrojiet tā vienuma SNMP virkni (vai OID), kuru vēlaties uzraudzīt.

Lai iegūtu SNMP virkņu sarakstu, izmantojiet komandu **snmpwalk** (daļu no net-snmp programmatūras, kurai jābūt instalētai kā daļai no Zabbix instalācijas) vai līdzvērtīgu rīku:

```
snmpwalk -v 2c -c public <host IP> .
```

Tā kā šeit '2c' apzīmē SNMP versiju, varat to aizstāt arī ar '1', lai norādītu, ka ierīcē tiek izmantota SNMP 1. versija.

Tam vajadzētu atgriezt SNMP virkņu sarakstu un to pēdējo vērtību. Ja tā nenotiek, iespējams, ka SNMP 'community' atšķiras no standarta 'public', un tādā gadījumā jums būs jānoskaidro, kāda tā ir.

Pēc tam varat pārskatīt sarakstu, līdz atrodat virkni, kuru vēlaties uzraudzīt, piemēram, ja vēlaties uzraudzīt baitus, kas ienāk jūsu komutatorā 3. portā, jums jāizmanto virkne `IF-MIB::ifHCInOctets.3` no šīs rindas:

```
IF-MIB::ifHCInOctets.3 = Counter64: 3409739121
```

Tagad varat izmantot komandu **snmpget**, lai noskaidrotu 'IF-MIB::ifHCInOctets.3' skaitlisko OID:

```
snmpget -v 2c -c public -On <host IP> IF-MIB::ifHCInOctets.3
```

Ņemiet vērā, ka pēdējais skaitlis virknē ir porta numurs, kuru vēlaties uzraudzīt. Skatiet arī: Dinamiskie indeksi.

Tam vajadzētu atgriezt kaut ko līdzīgu šim:

```
.1.3.6.1.2.1.31.1.1.1.6.3 = Counter64: 3472126941
```

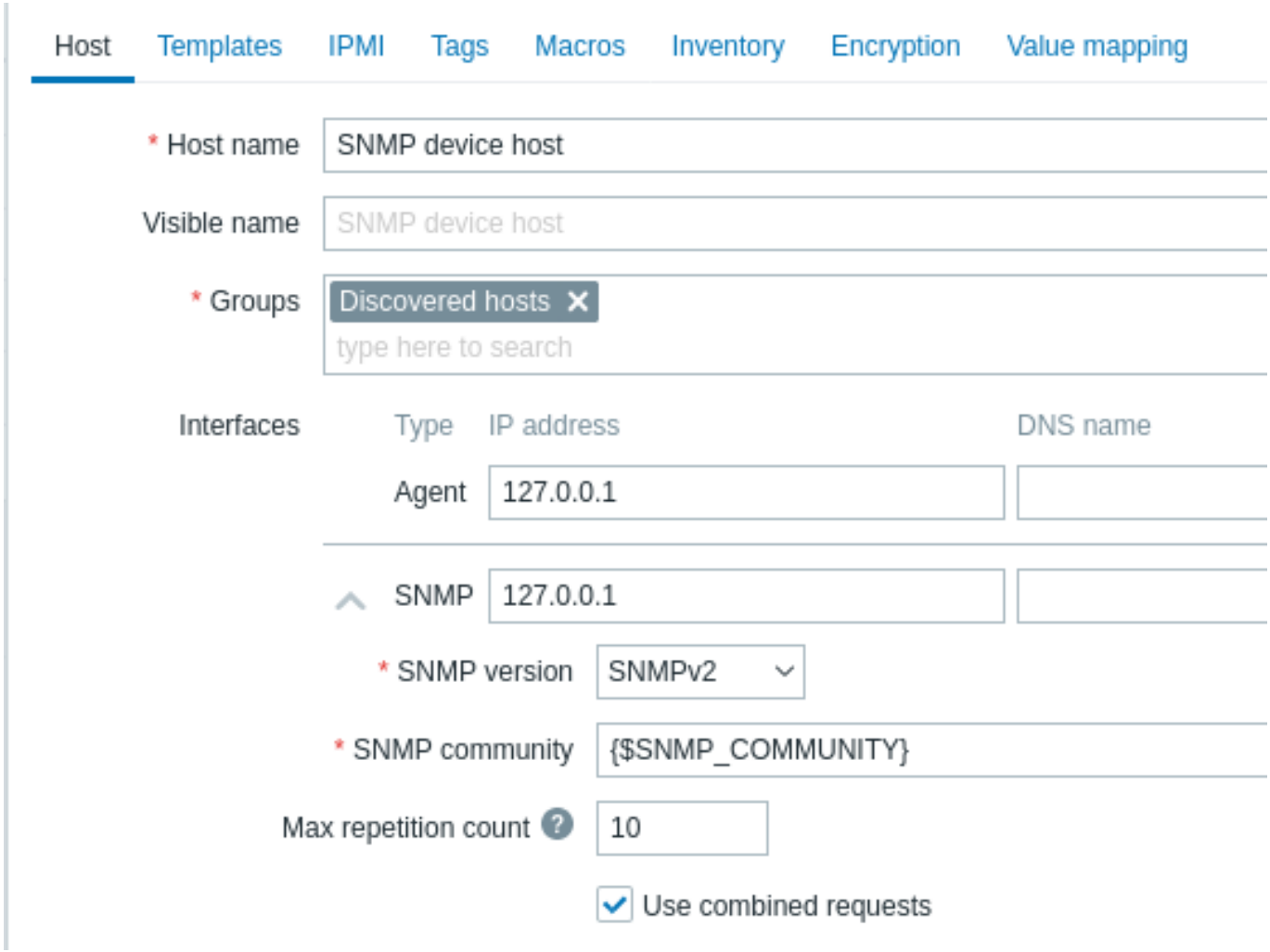
Atkal, pēdējais skaitlis OID ir porta numurs.

Note:
Daži no visbiežāk izmantotajiem SNMP OID tiek automātiski pārveidoti skaitliskā attēlojumā Zabbix sistēmā.

Iepriekšējā pēdējā piemērā vērtības tips ir "Counter64", kas iekšēji atbilst tipam ASN_COUNTER64. Pilns atbalstīto tipu saraksts ir 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 un ASN_OBJECT_ID. Šie tipi aptuveni atbilst "Counter32", "Counter64", "UInteger32", "INTEGER", "Float", "Double", "Timeticks", "Gauge32", "IpAddress", "OCTET STRING", "OBJECT IDENTIFIER" **snmpget** izvadē, taču atkarībā no attēlošanas norādes tie var tikt parādīti arī kā "STRING", "Hex-STRING", "OID" un citi.

2. solis

Izveidojiet hostu, kas atbilst ierīcei.



Pievienojiet hostam SNMP saskarni:

- Ievadiet IP adresi/DNS nosaukumu un porta numuru.
- Nolaižamajā sarakstā atlasiet *SNMP version*.
- Pievienojiet saskarnes akreditācijas datus atkarībā no atlasītās SNMP versijas:
  - SNMPv1, v2 nepieciešama tikai kopiena (parasti 'public').
  - SNMPv3 nepieciešamas specifiskākas opcijas (skatiet tālāk).
- Norādiet maksimālo atkārtojumu vērtību (noklusējums: 10) vietējiem SNMP lielapjoma pieprasījumiem (GetBulkRequest-PDUs); tikai `discovery[]` un `walk[]` vienumiem SNMPv2 un v3. Ņemiet vērā, ka pārāk augstas šīs vērtības iestatīšana var izraisīt SNMP aģents pārbaudes noildzi.
- Atzīmējiet izvēles rūtiņu *Use combined requests*, lai atļautu SNMP pieprasījumu kombinēto apstrādi (nav saistīta ar vietējiem SNMP lielapjoma pieprasījumiem "walk" un "get").

SNMPv3 parametrs	Apraksts
Context name	Ievadiet konteksta nosaukumu, lai identificētu vienumu SNMP apakštīklā. Lietotāja makrosi tiek atrisināti šajā laukā.
Security name	Ievadiet drošības nosaukumu. Lietotāja makrosi tiek atrisināti šajā laukā.
Security level	Atlasiet drošības līmeni: **noAuthNoPriv** - netiek izmantoti ne autentifikācijas, ne privātuma protokoli **AuthNoPriv** - tiek izmantots autentifikācijas protokols, privātuma protokols netiek izmantots **AuthPriv** - tiek izmantoti gan autentifikācijas, gan privātuma protokoli
Authentication protocol	Atlasiet autentifikācijas protokolu - *MD5*, *SHA1*; ar net-snmp 5.8 un jaunāku arī *SHA224*, *SHA256*, *SHA384* vai *SHA512*.
Authentication passphrase	Ievadiet autentifikācijas paroli. Lietotāja makrosi tiek atrisināti šajā laukā.
Privacy protocol	Atlasiet privātuma protokolu - *DES*, *AES128*, *AES192*, *AES256*, *AES192C* (Cisco) vai *AES256C* (Cisco). Skatiet piezīmes par privātuma protokola atbalstu

SNMPv3 parametrs	Apraksts
Privacy passphrase	Ievadiet privātuma paroli. Lietotāja makrosi tiek atrisināti šajā laukā.

Nepareizu SNMPv3 akreditācijas datu gadījumā (drošības nosaukums, autentifikācijas protokols/parole, privātuma protokols):

- Zabbix saņem ERROR no net-snmp, izņemot nepareizu *Privacy passphrase* gadījumu, kad Zabbix saņem TIMEOUT kļūdu no net-snmp.
- SNMP saskarnes pieejamība pārslēgsies uz sarkanu (nav pieejama).

Note:
Izmaiņas *Authentication protocol*, *Authentication passphrase*, *Privacy protocol* vai *Privacy passphrase*, kas veiktas, nemainot *Security name*, parasti tiek piemērotas automātiski, kad atbilstošā SNMPv3 saskarne tiek atjaunināta Zabbix. Gadījumos, kad tiek mainīts arī *Security name*, visi parametri tiks atjaunināti nekavējoties.

Varat izmantot kādu no pieejamajām SNMP veidnēm, kas automātiski pievienos vienumu kopu. Pirms veidnes izmantošanas pārliecinieties, ka tā ir saderīga ar hostu.

Noklikšķiniet uz *Add*, lai saglabātu hostu.

Privātuma protokolu atbalsts

Atkarībā no jūsu operētājsistēmas un net-snmp konfigurācijas daži privātuma protokoli var nebūt pieejami:

- Dažās jaunākās operētājsistēmās (piemēram, RHEL9) DES atbalsts net-snmp pakotnei ir noņemts.
- Šifrēšanas protokoli AES192 un spēcīgāki netiek atbalstīti pēc noklusējuma operētājsistēmās, kas ir vecākas par RHEL 8, CentOS 8, Oracle Linux 8, Debian 12, Ubuntu LTS 22.04, openSUSE Leap 15.5.

Lai pārbaudītu, vai net-snmp bibliotēka atbalsta AES192+, izmantojiet vienu no šīm iespējām:

1. `net-snmp-config`:

```
net-snmp-config --configure-options
```

Ja izvade satur `--enable-blumenthal-aes`, AES192+ tiek atbalstīts.

Ņemiet vērā, ka net-snmp-config ir daļa no SNMP izstrādes pakotnes (libsnmp-dev Debian/Ubuntu, net-snmp-devel CentOS/RHEL/OL/SUSE) un pēc noklusējuma var nebūt instalēta.

2. `snmpget`:

```
snmpget -v 3 -x AES-256
```

Ja izvade satur `Invalid privacy protocol specified after -3x flag: AES-256`, AES192+ netiek atbalstīts. Ja izvade satur `No hostname specified.`, AES192+ netiek atbalstīts.

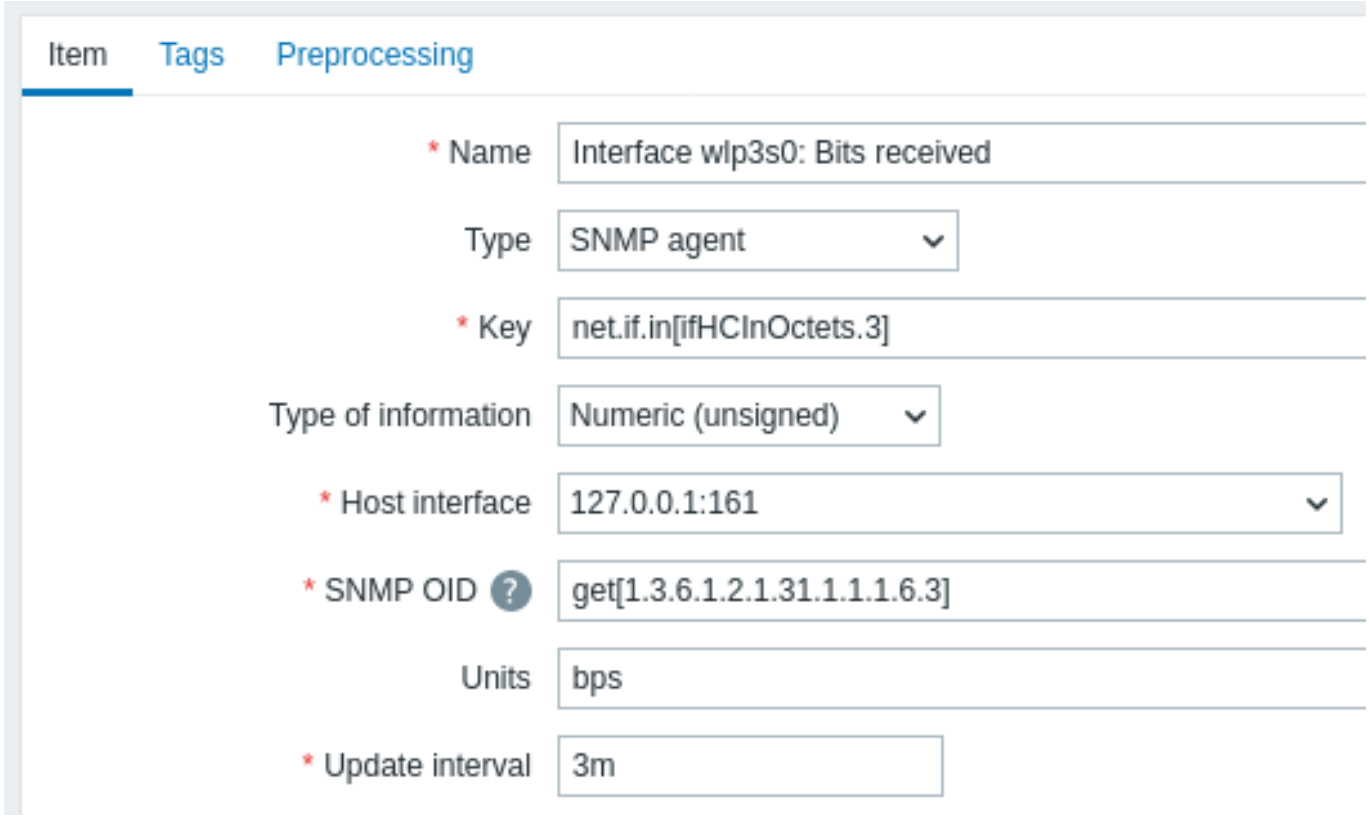
Ja jūsu net-snmp bibliotēka neatbalsta AES192 un augstākus protokolus, pārkompilējiet net-snmp ar opciju `--enable-blumenthal-aes`, pēc tam pārkompilējiet Zabbix serveris, norādot opciju `--with-net-snmp=/home/user/yourcustomnetsnmp/bin/net-snmp-config`.

3. solis

Izveidojiet vienumu uzraudzībai.

Tagad atgriezieties Zabbix un noklikšķiniet uz *Items* SNMP hostam, ko izveidojāt iepriekš. Atkarībā no tā, vai hosta izveides laikā izmantojāt veidni vai ne, jums būs vai nu ar jūsu hostu saistītu SNMP vienumu saraksts, vai vienkārši tukšs saraksts. Mēs pieņemsim, ka jūs izveidosiet vienumu pats, izmantojot informāciju, ko tikko ieguvāt ar snmpwalk un snmpget, tāpēc noklikšķiniet uz *Create item*.

Aizpildiet obligātos parametrus jaunā vienuma formā:



Parametrs	Apraksts
Name	Ievadiet vienuma nosaukumu.
Type	Šeit atlasiet **SNMP aģents**.
Key	Ievadiet atslēgu ar jēgpilnu nosaukumu.
Host interface	Pārliecinieties, ka ir atlasīta SNMP saskarne, piemēram, jūsu komutatora/maršrutētāja saskarne.

Parametrs	Apraksts
SNMP OID	Izmantojiet vienu no atbalstītajiem formātiem, lai ievadītu OID vērtību(-as): **walk[OID1,OID2,...]** — iegūt vērtību apakškoku. Piemēram: `walk[1.3.6.1.2.1.2.2.1.2,1.3.6.1.2.1.2.2.1.3]`. Šī opcija asinhroni izmanto vietējos SNMP bulk pieprasījumus (GetBulkRequest-PDUs). Šī vienuma noildzes iestatījumus var norādīt vienuma konfigurācijas formā. Apsveriet iespēju iestatīt zemu noildzes vērtību, lai izvairītos no ilgām aizturēm, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtotiem mēģinājumiem, ja iepriekšējie beidzas ar noildi vai neizdodas (piemēram, 3 sekunžu noilde var radīt 15 sekunžu gaidīšanas laiku). Varat to izmantot kā galveno vienumu ar atkarīgajiem vienumiem, kas iegūst datus no galvenā vienuma, izmantojot priekšapstrādi. Vienā snmp walk ir iespējams norādīt vairākus OID, piemēram, `walk[OID1,OID2,...]`, lai asinhroni apstrādātu pa vienam OID vienlaikus. Ja bulk pieprasījums neatgriež rezultātus, tiek mēģināts iegūt vienu ierakstu bez bulk pieprasījuma. Kā parametri tiek atbalstīti MIB nosaukumi; tādējādi `walk[1.3.6.1.2.1.2.2.1.2]` un `walk[ifDescr]` atgriezīs vienādu izvadi. Ja ir norādīti vairāki OID/MIB, piemēram, `walk[ifDescr,ifType,ifPhysAddress]`, tad izvade būs apvienots saraksts. GetBulk pieprasījumi tiek izmantoti ar SNMPv2 un v3 saskarnēm, bet GetNext — ar SNMPv1 saskarnēm; maksimālais atkārtojumu skaits bulk pieprasījumiem tiek konfigurēts saskarnes līmenī. Maksimālo atkārtojumu parametrs ietekmē bulk pieprasījumus, nosakot maksimālo OID skaitu, kas tiek atgriezts vienā bulk atbildē. Lielāka vērtība rada lielākas bulk atbildes, samazinot nepieciešamo pārraižu skaitu. Tomēr ne visas ierīces var atbalstīt ļoti lielas vērtības, un tas var radīt problēmas. Šis vienums atgriež snmpwalk utilītas izvadi ar parametriem -Oe -Ot -On. Varat izmantot šo vienumu kā galveno vienumu SNMP atklāšanā. **get[OID]** — asinhroni iegūt *vienu* vērtību. Piemēram: `get[1.3.6.1.2.1.31.1.1.1.6.3]` Šī vienuma noildzes iestatījumus var norādīt vienuma konfigurācijas formā. Apsveriet iespēju iestatīt zemu noildzes vērtību, lai izvairītos no ilgām aizturēm, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtotiem mēģinājumiem, ja iepriekšējie beidzas ar noildi vai neizdodas (piemēram, 3 sekunžu noilde var radīt 15 sekunžu gaidīšanas laiku). **OID** — (mantotais variants) ievadiet vienu teksta vai skaitlisku OID, lai sinhroni iegūtu vienu vērtību, pēc izvēles kombinējot to ar citām vērtībām. Piemēram: `1.3.6.1.2.1.31.1.1.1.6.3`. Šai opcijai vienuma pārbaudes noilde būs vienāda ar vērtību, kas iestatīta servera konfigurācijas failā. Labākai veiktspējai **ieteicams** izmantot `walk[OID]` un `get[OID]` vienumus. Visi `walk[OID]` un `get[OID]` vienumi tiek izpildīti asinhroni — nav nepieciešams saņemt atbildi uz vienu pieprasījumu, pirms tiek sāktas citas pārbaudes. Arī DNS atrisināšana notiek asinhroni. Asinhrono pārbaužu maksimālā vienlaicība ir 1000 (noteikta ar MaxConcurrentChecksPerPoller). Asinhrono SNMP aptaujātāju skaitu nosaka parametrs StartSNMPPollers. Ņemiet vērā, ka tīkla datplūsmas statistikai, kas tiek atgriezta ar jebkuru no metodēm, cilnē *Preprocessing* ir jāpievieno solis *Change per second*; pretējā gadījumā jūs saņemsiet kumulatīvo vērtību no SNMP ierīces, nevis jaunākās izmaiņas.

Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Tagad saglabājiet vienumu un dodieties uz *Monitoring* > *Latest data*, lai skatītu savus SNMP datus.

Piemērs 1

Vispārīgs piemērs:

Parametrs	Apraksts
OID	1.2.3.45.6.7.8.0 (vai .1.2.3.45.6.7.8.0)
Atslēga	<Unikāla virkne, kas tiks izmantota kā atsauce uz trigeriem> Piemēram, ”my_param”.

Ņemiet vērā, ka OID var norādīt gan skaitliskā, gan virknes formā. Tomēr dažos gadījumos virknes OID ir jāpārveido skaitliskā attēlojumā. Šim nolūkam var izmantot utilītu snmpget:

```
snmpget -On localhost public enterprises.ucdavis.memory.memTotalSwap.0
```

Piemērs 2

Darbspējas laika uzraudzība:

Parametrs	Apraksts
OID	MIB::sysUpTime.0
Atslēga	router.uptime
Vērtības tips	Peldošā komata skaitlis
Mērvienības	darbspējas laiks
Pirmsapstrādes solis: Pielāgots reizinātājs	0.01

Vietējie SNMP lielapjoma pieprasījumi

walk[OID1,OID2,...] vienums ļauj izmantot SNMP vietējo lielapjoma pieprasījumu funkcionalitāti (GetBulkRequest-PDUs), kas ir pieejama SNMP 2./3. versijā.

GetBulk pieprasījums SNMP vidē izpilda vairākus GetNext pieprasījumus un atgriež rezultātu vienā atbildē. To var izmantot gan parastajiem SNMP vienumiem, gan SNMP atklāšanai, lai samazinātu tīkla apmaiņas reižu skaitu.

SNMP **walk[OID1,OID2,...]** vienumu var izmantot kā galveno vienumu, kas vienā pieprasījumā savāc datus, kopā ar atkarīgajiem vienumiem, kuri pēc vajadzības parsē atbildi, izmantojot priekšapstrādi.

Ņemiet vērā, ka SNMP vietējo lielapjoma pieprasījumu izmantošana nav saistīta ar SNMP pieprasījumu apvienošanas opciju, kas ir Zabbix paša veids, kā apvienot vairākus SNMP pieprasījumus (skatiet nākamo sadaļu).

SNMP lielapjoma vienumiem tiks veikti līdz pieciem atkārtotiem mēģinājumiem, lai izvairītos no kļūmes, ja kāda no paketēm tiek pazaudēta. SNMP vienumu ar `get` un `walk` noildze (iestatīta vienuma konfigurācijas formā) tiek noteikta visai sesijai. Noildze tiek piemērota neatkarīgi no tā, vai dati ir iegūti pilnībā; ja dati tiek saņemti tikai daļēji (piemēram, dati tiek veiksmīgi savākti tikai vienam no vairākiem OID), tad vienums kļūst neatbalstīts ar ziņojumu ”Only partial data received”. Ja tiek sasniegta noildze, notiks atkārtots mēģinājums, noildze tiks atiestatīta, un pēdējais pieprasījums tiks nosūtīts vēlreiz, ļaujot turpināt sesiju no pēdējā pieprasījuma, ja viena pakete ir pazaudēta vai pienākusi pārāk vēlu. Apsveriet iespēju iestatīt zemu noildzes vērtību, lai izvairītos no ilgām aizturēm, ja ierīce nav sasniedzama, jo tiks veikti līdz 5 atkārtoti mēģinājumi, ja iepriekšējie beigsies ar noildi vai kļūmi (piemēram, 3 sekunžu noildze var radīt 15 sekunžu gaidīšanas laiku).

Kombinētās apstrādes iekšējā darbība

Zabbix serveris un starpniekserveris var vaicāt SNMP ierīcēm vairākas vērtības vienā pieprasījumā. Tas ietekmē vairākus SNMP vienumu tipus:

- parastos SNMP vienumus
- SNMP vienumus ar dinamiskiem indeksiem
- SNMP zemā līmeņa atklāšanas noteikumus

Visi SNMP vienumi vienā saskarnē ar identiskiem parametriem tiek ieplānoti vaicāšanai vienlaikus. Pirmos divus vienumu tipus aptaujātāji apstrādā paketēs, kurās ir ne vairāk kā 128 vienumi, savukārt zemā līmeņa atklāšanas noteikumi, tāpat kā iepriekš, tiek apstrādāti atsevišķi.

Zemākā līmenī vērtību vaicāšanai tiek veiktas divu veidu darbības: vairāku norādītu objektu iegūšana un OID koka pārlūkošana.

”Iegūšanai” tiek izmantots GetRequest-PDU ar ne vairāk kā 128 mainīgo piesaistēm. ”Pārlūkošanai” SNMPv1 gadījumā tiek izmantots GetNextRequest-PDU, bet SNMPv2 un SNMPv3 gadījumā tiek izmantots GetBulkRequest ar lauku ”max-repetitions”, kura vērtība nepārsniedz 128.

Tādējādi kombinētās apstrādes priekšrocības katram SNMP vienuma tipam ir šādas:

- parastie SNMP vienumi gūst labumu no ”iegūšanas” uzlabojumiem;
- SNMP vienumi ar dinamiskiem indeksiem gūst labumu gan no ”iegūšanas”, gan ”pārlūkošanas” uzlabojumiem: ”iegūšana” tiek izmantota indeksu pārbaudei, bet ”pārlūkošana” — kešatmiņas veidošanai;

- SNMP zemā līmeņa atklāšanas noteikumi gūst labumu no ”pārlūkošanas” uzlabojumiem.

Tomēr pastāv tehniska problēma — ne visas ierīces spēj atgriezt 128 vērtības vienā pieprasījumā. Dažas vienmēr atgriež korektu atbildi, bet citas vai nu atbild ar kļūdu ”tooBig(1)”, vai arī neatbild vispār, tiklīdz iespējamā atbilde pārsniedz noteiktu robežu.

Lai noteiktu optimālo vaicājamo objektu skaitu konkrētai ierīcei, Zabbix izmanto šādu stratēģiju. Tas piesardzīgi sāk ar 1 vērtības vaicāšanu vienā pieprasījumā. Ja tas izdodas, tas vaicā 2 vērtības vienā pieprasījumā. Ja arī tas izdodas, tas vaicā 3 vērtības vienā pieprasījumā un turpina līdzīgi, reizinot vaicājamo objektu skaitu ar 1.5, iegūstot šādu pieprasījumu izmēru secību: 1, 2, 3, 4, 6, 9, 13, 19, 28, 42, 63, 94, 128.

Tomēr, tiklīdz ierīce atsakās sniegt korektu atbildi (piemēram, 42 mainīgajiem), Zabbix veic divas darbības.

Pirmkārt, pašreizējai vienumu paketei tas uz pusi samazina objektu skaitu vienā pieprasījumā un vaicā 21 mainīgo. Ja ierīce ir pieejama, tad vaicājumam vajadzētu darboties lielākajā daļā gadījumu, jo ir zināms, ka 28 mainīgie darbojās, bet 21 ir ievērojami mazāk. Tomēr, ja arī tas neizdodas, Zabbix pāriet uz vērtību vaicāšanu pa vienai. Ja arī šajā brīdī tas neizdodas, tad ierīce noteikti neatbild un pieprasījuma izmērs nav problēma.

Otrkārt, nākamajām vienumu paketēm Zabbix sāk ar pēdējo veiksmīgo mainīgo skaitu (mūsu piemērā 28) un turpina palielināt pieprasījuma izmēru par 1, līdz tiek sasniegta robeža. Piemēram, pieņemot, ka lielākais atbildes izmērs ir 32 mainīgie, nākamie pieprasījumi būs ar izmēriem 29, 30, 31, 32 un 33. Pēdējais pieprasījums neizdosies, un Zabbix vairs nekad neizsniegs pieprasījumu ar izmēru 33. No šī brīža Zabbix šai ierīcei vaicās ne vairāk kā 32 mainīgos.

Ja lieli vaicājumi neizdodas ar šādu mainīgo skaitu, tas var nozīmēt vienu no divām lietām. Precīzus kritērijus, ko ierīce izmanto atbildes izmēra ierobežošanai, nav iespējams zināt, taču mēs cenšamies to aptuveni noteikt, izmantojot mainīgo skaitu. Tātad pirmā iespēja ir tāda, ka šis mainīgo skaits vispārīgā gadījumā ir tuvu ierīces faktiskajai atbildes izmēra robežai: dažreiz atbilde ir mazāka par robežu, dažreiz tā ir lielāka. Otrā iespēja ir tāda, ka UDP pakete vienā no virzieniem vienkārši ir pazudusi. Šo iemeslu dēļ, ja Zabbix saņem neveiksmīgu vaicājumu, tas samazina maksimālo mainīgo skaitu, lai mēģinātu nonākt dziļāk ierīces drošajā diapazonā, bet ne vairāk kā divas reizes.

Iepriekš minētajā piemērā, ja vaicājums ar 32 mainīgajiem nejauši neizdodas, Zabbix samazinās skaitu līdz 31. Ja arī tas nejauši neizdodas, Zabbix samazinās skaitu līdz 30. Tomēr Zabbix nesamazinās skaitu zem 30, jo pieņems, ka turpmākās kļūmes izraisa UDP pakešu zudums, nevis ierīces ierobežojums.

Ja tomēr ierīce citu iemeslu dēļ nespēj korekti apstrādāt kombinētos pieprasījumus un iepriekš aprakstītā heiristika nedarbojas, katrai saskarnei ir iestatījums ”Use combined requests”, kas ļauj šai ierīcei atspējot kombinētos pieprasījumus.

Ja kombinētie pieprasījumi izraisa daļējas vai kļūdainas atbildes, kuru rezultātā rodas nepareizi aprēķini sekundē (delta) (piemēram, šķietami lēcieni saskarnes skaitītājos), atspējojiet *Use combined requests* ietekmētajai saskarnei, lai piespiestu atsevišķus katra vienuma vaicājumus; tas bieži novērš nepatiesus lēcienus. Alternatīvi apsveriet asinhronu `get[]` vai `walk[]` vienumu izmantošanu, kas tiek izpildīti asinhroni un uz kuriem neattiecas katras saskarnes *Use combined requests* paketēšana — tos var izmantot mantoto sinhrono OID pārbaužu vietā, lai izvairītos no problēmām, kas saistītas ar kombinētajiem pieprasījumiem. Meklējiet servera/starpniekservera žurnāla ierakstus, kas ir līdzīgi Pārskata sadaļā parādītajam, lai palīdzētu noteikt ietekmētās ierīces.

Papildus tam, ja saskarne bieži kļūst nepieejama, var būt nepieciešams palielināt `UnavailableDelay` parametru Zabbix serveris vai Zabbix starpniekserveris konfigurācijas failos, lai samazinātu pieprasījumu biežumu. Vienumi var kļūt neatbalstīti, ja atklāšanas vai OID pārlūkošanas laikā tiek saņemti daļēji dati.

1 Dinamiskie indeksi

Pārskats

Lai gan nepieciešamo indeksa numuru (piemēram, tīkla saskarnes) var atrast starp SNMP OID, dažkārt nevar pilnībā paļauties uz to, ka indeksa numurs vienmēr paliks nemainīgs.

Indeksa numuri var būt dinamiski — tie laika gaitā var mainīties, un tādēļ jūsu vienums var pārstāt darboties.

Lai izvairītos no šādas situācijas, ir iespējams definēt OID, kas ņem vērā iespēju, ka indeksa numurs var mainīties.

Piemēram, ja jums ir jāiegūst indeksa vērtība, ko pievienot **ifInOctets**, kas atbilst **GigabitEthernet0/1** saskarnei Cisco ierīcē, izmantojiet šādu OID:

```
ifInOctets["index","ifDescr","GigabitEthernet0/1"]
```

Sintakse

OID tiek izmantota īpaša sintakse:

<datu OID>[”index”,”<indeksa bāzes OID>”,”<meklējamā virkne>”]

Parametrs	Apraksts
Datu OID	Galvenais OID, ko izmantot datu iegūšanai vienumā.
index	Apstrādes metode. Pašlaik tiek atbalstīta viena metode: **index** – meklēt indeksu un pievienot to datu OID
Indeksa bāzes OID	Šis OID tiks uzmeklēts, lai iegūtu virknei atbilstošo indeksa vērtību.
Meklējamā virkne	Virkne, ko izmantot precīzai atbilstībai ar vērtību uzmeklēšanas laikā. Reģistrjutīga.

Piemērs

apache procesa atmiņas lietojuma iegūšana.

Ja tiek izmantota šī OID sintakse:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem["index","HOST-RESOURCES-MIB::hrSWRunPath", "/usr/sbin/apache2"]
```

tad indeksa numurs tiks uzmeklēts šeit:

```
...
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"
...
```

Tagad mums ir indekss — 5388. Indekss tiks pievienots datu OID, lai saņemtu mūs interesējošo vērtību:

```
HOST-RESOURCES-MIB::hrSWRunPerfMem.5388 = INTEGER: 31468 KBytes
```

Indeksu uzmeklēšanas kešošana

Kad tiek pieprasīts dinamiska indeksa vienums, Zabbix izgūst un kešo visu SNMP tabulu zem indeksa bāzes OID, pat ja atbilstība tiktu atrasta ātrāk. Tas tiek darīts gadījumam, ja cits vienums vēlāk atsauksies uz to pašu bāzes OID — Zabbix meklēs indeksu kešatmiņā, nevis atkārtoti vaicās uzraudzītajam hostam. Ņemiet vērā, ka katrs aptaujātāja process izmanto atsevišķu kešatmiņu.

Visās turpmākajās vērtību izgūšanas darbībās tiek pārbaudīts tikai atrastais indekss. Ja tas nav mainījies, tiek pieprasīta vērtība. Ja tas ir mainījies, kešatmiņa tiek pārbūvēta — katrs aptaujātājs, kas konstatē mainītu indeksu, vēlreiz iziet cauri indeksa SNMP tabulai.

2 Īpašie OID

Daži no visbiežāk izmantotajiem SNMP OID tiek automātiski pārveidoti uz skaitlisku attēlojumu Zabbix sistēmā. Piemēram, **ifIndex** tiek pārveidots uz **1.3.6.1.2.1.2.2.1.1**, bet **ifIndex.0** tiek pārveidots uz **1.3.6.1.2.1.2.2.1.1.0**.

Tabulā ir ietverts īpašo OID saraksts.

Īpašais OID	Identifikators	Apraksts
ifIndex	1.3.6.1.2.1.2.2.1.1	Unikāla vērtība katrai saskarnei.
ifDescr	1.3.6.1.2.1.2.2.1.2	Teksta virkne, kas satur informāciju par saskarni. Šajā virknē jāiekļauj ražotāja nosaukums, produkta nosaukums un aparatūras saskarnes versija.
ifType	1.3.6.1.2.1.2.2.1.3	Saskarnes tips, kas noteikts atbilstoši fiziskajam/saites protokolam(-iem), kas protokolu stekā atrodas tieši ”zem” tīkla slāņa.
ifMtu	1.3.6.1.2.1.2.2.1.4	Lielākās datagrammas izmērs, ko var nosūtīt/saņemt caur saskarni, norādīts oktetos.
ifSpeed	1.3.6.1.2.1.2.2.1.5	Saskarnes pašreizējā joslas platuma novērtējums bitos sekundē.
ifPhysAddress	1.3.6.1.2.1.2.2.1.6	Saskarnes adrese protokolu slānī, kas protokolu stekā atrodas tieši ‘zem’ tīkla slāņa.
ifAdminStatus	1.3.6.1.2.1.2.2.1.7	Saskarnes pašreizējais administratīvais stāvoklis.
ifOperStatus	1.3.6.1.2.1.2.2.1.8	Saskarnes pašreizējais darbības stāvoklis.
ifInOctets	1.3.6.1.2.1.2.2.1.10	Kopējais saskarnē saņemto oktetu skaits, ieskaitot kadrēšanas rakstzīmes.
ifInUcastPkts	1.3.6.1.2.1.2.2.1.11	Apakštīkla unicast pakešu skaits, kas piegādātas augstāka līmeņa protokolam.
ifInNUcastPkts	1.3.6.1.2.1.2.2.1.12	Ne-unicast (t.i., apakštīkla apraides vai apakštīkla multiraides) pakešu skaits, kas piegādātas augstāka līmeņa protokolam.

Īpašais OID	Identifikators	Apraksts
ifInDiscards	1.3.6.1.2.1.2.2.1.13	Ienākošo pakešu skaits, kuras tika izvēlētas atmešanai, lai gan kļūdas netika konstatētas, tādējādi nepieļaujot to piegādi augstāka līmeņa protokolam. Viens no iespējamiem šādas paketes atmešanas iemesliem varētu būt bufera vietas atbrīvošana.
ifInErrors	1.3.6.1.2.1.2.2.1.14	Ienākošo pakešu skaits, kurās bija kļūdas, kas neļāva tās piegādāt augstāka līmeņa protokolam.
ifInUnknownProtos	1.3.6.1.2.1.2.2.1.15	Caur saskarni saņemto pakešu skaits, kas tika atmestas nezināma vai neatbalstīta protokola dēļ.
ifOutOctets	1.3.6.1.2.1.2.2.1.16	Kopējais no saskarnes nosūtīto oktetu skaits, ieskaitot kadrēšanas rakstzīmes.
ifOutUcastPkts	1.3.6.1.2.1.2.2.1.17	Kopējais pakešu skaits, kuru nosūtīšanu pieprasīja augstāka līmeņa protokoli un kuras šajā apakšslānī nebija adresētas multiraides vai apraides adresei, ieskaitot tās, kas tika atmestas vai netika nosūtītas.
ifOutNUcastPkts	1.3.6.1.2.1.2.2.1.18	Kopējais pakešu skaits, kuru nosūtīšanu pieprasīja augstāka līmeņa protokoli un kuras šajā apakšslānī bija adresētas multiraides vai apraides adresei, ieskaitot tās, kas tika atmestas vai netika nosūtītas.
ifOutDiscards	1.3.6.1.2.1.2.2.1.19	Izejošo pakešu skaits, kuras tika izvēlētas atmešanai, lai gan kļūdas netika konstatētas, tādējādi nepieļaujot to nosūtīšanu. Viens no iespējamiem šādas paketes atmešanas iemesliem varētu būt bufera vietas atbrīvošana.
ifOutErrors	1.3.6.1.2.1.2.2.1.20	Izejošo pakešu skaits, kuras nevarēja nosūtīt kļūdu dēļ.
ifOutQLen	1.3.6.1.2.1.2.2.1.21	Izvades pakešu rindas garums (paketēs).

3 MIB faili

Ievads

MIB nozīmē Management Information Base. MIB faili ļauj izmantot OID (Object Identifier) teksta attēlojumu. Uzraugot SNMP ierīces ar Zabbix, ir iespējams izmantot neapstrādātus OID, taču, ja jums ērtāk ir izmantot teksta attēlojumu, jums ir jāinstalē MIB faili.

Piemēram,

`ifHCOutOctets`

ir OID teksta attēlojums

`1.3.6.1.2.1.31.1.1.1.10`

MIB failu instalēšana

Debian bāzētās sistēmās:

```
apt install snmp-mibs-downloader
download-mibs
```

RedHat bāzētās sistēmās:

```
dnf install net-snmp-libs
```

MIB failu iespējošana

Sistēmās, kuru pamatā ir RedHat, MIB failiem pēc noklusējuma jābūt iespējotiem. Sistēmās, kuru pamatā ir Debian, jums ir jārediģē fails `/etc/snmp/snmp.conf` un jāizkomentē rinda `mibs :`

```
# Tā kā snmp pakotnes tiek piegādātas bez MIB failiem licencēšanas iemeslu dēļ, ielāde
# MIB failiem pēc noklusējuma ir atspējota. Ja esat pievienojis MIB failus, varat atkārtoti iespējot
# to ielādi, izkomentējot šādu rindu.
mibs :
```

MIB failu testēšana

SNMP MIB testēšanu var veikt, izmantojot `snmpwalk` utilītu. Ja tā nav instalēta, izmantojiet tālāk norādītās instrukcijas.

Sistēmās, kuru pamatā ir Debian:

```
apt install snmp
```

Sistēmās, kuru pamatā ir RedHat:

```
dnf install net-snmp-utils
```

Pēc tam, vaicājot tīkla ierīci, šādai komandai nevajadzētu atgriezt kļūdu:

```
$ 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
[...]
```

MIB izmantošana Zabbix

Vissvarīgākais, kas jāņem vērā, ir tas, ka Zabbix procesi nesaņem informāciju par MIB failos veiktajām izmaiņām. Tāpēc pēc katrām izmaiņām jums ir jāpārstartē Zabbix serveris vai starpniekserveris, piemēram:

```
systemctl restart zabbix-server
```

Pēc tam MIB failos veiktās izmaiņas stājas spēkā.

Pielāgotu MIB failu izmantošana

Katrā GNU/Linux distribūcijā ir iekļauti standarta MIB faili. Taču daži ierīču ražotāji nodrošina arī savus.

Pieņemsim, ka vēlaties izmantot CISCO-SMI MIB failu. Tālāk norādītās instrukcijas to lejupielādēs un instalēs:

```
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
```

Tagad jums vajadzētu būt iespējai to izmantot. Mēģiniet pārtulkot objekta *ciscoProducts* nosaukumu no MIB faila uz OID:

```
snmptranslate -IR -On CISCO-SMI::ciscoProducts
.1.3.6.1.4.1.9.1
```

Ja OID vietā saņemat kļūdas, pārliecinieties, ka visas iepriekšējās komandas tika izpildītas bez kļūdām.

Objekta nosaukuma tulkošana izdevās, un jūs esat gatavs izmantot pielāgotu MIB failu. Ņemiet vērā vaicājumā izmantoto MIB nosaukuma prefiksu (*CISCO-SMI::*). Tas būs nepieciešams, izmantojot komandrindas rīkus, kā arī Zabbix.

Neaizmirstiet pārstartēt Zabbix serveri/starpniekserveri, pirms izmantot šo MIB failu Zabbix vidē.

> **Attention:**
> Ņemiet vērā, ka MIB failiem var būt atkarības. Tas nozīmē, ka vienam MIB var būt nepieciešams cits. Lai izpildītu šīs atkarības, jums jāinstalē visi attiecīgie MIB faili.

3 SNMP slazdi

Pārskats

SNMP trigeru saņemšana ir pretēja SNMP iespējotu ierīču vaicāšanai.

Šajā gadījumā informācija tiek nosūtīta no SNMP iespējotas ierīces uz snmptrapd, un to no faila savāc jeb "pārtver" Zabbix serveris vai Zabbix starpniekserveris.

Parasti trigeri tiek nosūtīti, mainoties kādam nosacījumam, un aģents izveido savienojumu ar serveri, izmantojot 162. portu (pretstatā 161. portam aģenta pusē, kas tiek izmantots vaicājumiem). Trigeru izmantošana var palīdzēt noteikt īslaicīgas problēmas, kas rodas starp vaicājumu intervāliem un kuras vaicājumu dati var nepamanīt.

SNMP trigeru saņemšana Zabbix ir paredzēta darbam ar **snmptrapd** un vienu no mehānismiem trigeru nodošanai Zabbix — vai nu Bash vai Perl skriptu, vai SNMPTT.

> **Note:**
> Vienkāršākais veids, kā iestatīt trigeru uzraudzību pēc Zabbix konfigurēšanas, ir izmantot Bash skripta risinājumu, jo Perl un SNMPTT mūsdienu distribūcijās bieži nav pieejami un prasa sarežģītāku konfigurēšanu. Tomēr šis risinājums izmanto skriptu, kas konfigurēts kā `traphandle`. Lai nodrošinātu labāku veiktspēju produkcijas sistēmās, izmantojiet iegulto Perl risinājumu (vai nu skriptu ar do `perl` opciju, vai SNMPTT).

Trigera saņemšanas darbplūsma:

1. `snmptrapd` saņem trigeri
2. `snmptrapd` nodod trigeri saņēmēja skriptam (Bash, Perl) vai SNMPTT
3. Saņēmējs parsē, formatē un ieraksta trigeri failā
4. Zabbix SNMP trapper nolasa un parsē trigeru failu
5. Katram trigerim Zabbix atrod visus ”SNMP trapper” vienumus ar hostu saskarnēm, kas atbilst saņemtā trigera adresei. Ņemiet vērā, ka salīdzināšanas laikā tiek izmantots tikai hosta saskarnē atlasītais ”IP” vai ”DNS”.
6. Katram atrastajam vienumam trigeris tiek salīdzināts ar regulāro izteiksmi `snmptrap[regexp]`. Trigeris tiek iestatīts kā vērtība **visiem** atbilstošajiem vienumiem. Ja netiek atrasts neviens atbilstošs vienums un ir `snmptrap.fallback` vienums, trigeris tiek iestatīts kā tā vērtība.
7. Ja trigeris netika iestatīts kā neviena vienuma vērtība, Zabbix pēc noklusējuma reģistrē neatbilstošo trigeri žurnālā. (To konfigurē ar ”Log unmatched SNMP traps” sadaļā Administration > General > Other.)

Piezīmes par HA pārslēgšanos kļūmes gadījumā

Augstas pieejamības (HA) mezgla pārslēgšanās laikā Zabbix turpinās apstrādi pēc pēdējā ieraksta pēdējā ISO 8601 laikspiedola ietvaros; ja tas pats ieraksts netiks atrasts, tad pēdējās pozīcijas noteikšanai tiks izmantots tikai laikspiedols.

SNMP slazdu konfigurēšana

Šim vienuma tipam ir nepieciešama šāda lietotāja saskarnes konfigurācija.

1. Izveidojiet SNMP saskarni savam hostam

- Sadaļā *Data collection* > *Hosts* izveidojiet/rediģējiet hostu un laukā *Interfaces* pievienojiet saskarnes tipu ”SNMP”, norādot IP vai DNS adresi.

Adrese no katra saņemtā slazda tiks salīdzināta ar visu SNMP saskarņu IP un DNS adresēm, lai atrastu atbilstošos hostus.

2. Konfigurējiet vienumu

- Sadaļā *Data collection* > *Hosts* izveidojiet/rediģējiet nepieciešamo vienumu.
- Laukā *Key* izmantojiet vienu no SNMP slazdu atslēgām:

Key		
Description	Return value	Comments
snmptrap[regexp] Uztver visus SNMP slazdus, kas atbilst regulārajai izteiksmei, kas norādīta **regexp**. Ja regexp nav norādīts, tiek uztverts jebkurš slazds.	SNMP slazds	Šo vienumu var iestatīt tikai SNMP saskarnēm. Šī vienuma atslēgas parametrā tiek atbalstīti lietotāja makrosi un globālās regulārās izteiksmes.
snmptrap.fallback Uztver visus SNMP slazdus, kurus šai saskarnei neuztvēra neviens no snmptrap[] vienumiem.	SNMP slazds	Šo vienumu var iestatīt tikai SNMP saskarnēm.

Note:
Vairākrindu regulāro izteiksmju atbilstības pārbaude pašlaik netiek atbalstīta.

- Lai laikspiedoli tiktu parsēti, iestatiet *Type of information* uz ”Log”. Ir pieļaujami arī citi formāti, piemēram, ”Numeric”, taču tiem var būt nepieciešams pielāgots slazdu apstrādātājs.

SNMP trap monitoringa iestatīšana

Zabbix servera/starpniekservera konfigurēšana

Lai nolasītu slazdus, Zabbix serverim vai starpniekserverim jābūt konfigurētam tā, lai palaistu SNMP trapper procesu un norādītu uz slazdu failu, kurā raksta SNMPTT vai Bash/Perl slazdu uztvērējs. Lai to izdarītu, rediģējiet konfigurācijas failu (zabbix_server.conf vai zabbix_proxy.conf):

```
StartSNMPTrapper=1
SNMPTrapperFile=[TRAP FILE]
```

Warning:
Ja tiek izmantots systemd parametrs **PrivateTmp** , maz ticams, ka šis fails darbosies mapē */tmp*.

Bash slazdu uztvērēja konfigurēšana

Prasības: tikai snmptrapd.

Bash slazdu uztvērēja skriptu var izmantot, lai pārsūtītu slazdus uz Zabbix serveri no snmptrapd, izmantojot trapper failu. Lai to konfigurētu, pievienojiet opciju `traphandle` snmptrapd konfigurācijas failam (`snmptrapd.conf`), skatiet piemēru.

Note:
Iespējams, būs jāpārstartē snmptrapd, lai tas ielādētu konfigurācijas izmaiņas.

Perl trap uztvērēja konfigurēšana

Prasības: Perl, Net-SNMP kompilēts ar --enable-embedded-perl (tas tiek darīts pēc noklusējuma kopš Net-SNMP 5.4)

Perl trap uztvērēju (skatiet misc/snmptrap/zabbix_trap_receiver.pl) var izmantot, lai pārsūtītu trap uz Zabbix serveri tieši no snmptrapd. Lai to konfigurētu:

- pievienojiet Perl skriptu snmptrapd konfigurācijas failam (snmptrapd.conf), piemēram:

```
perl do "[FULL PATH TO PERL RECEIVER SCRIPT]";
```

- konfigurējiet uztvērēju, piemēram:

```
$SNMPTrapperFile = '[TRAP FILE]';
$DateTimeFormat = '[DATE TIME FORMAT]';
```

Note:
Lai piemērotu izmaiņas konfigurācijā, var būt nepieciešams restartēt snmptrapd.

Note:
Ja skripta nosaukums nav ievietots pēdiņās, snmptrapd atteiksies startēties, parādot ziņojumus, kas ir līdzīgi šiem:


```
Regexp modifiers "/l" and "/a" are mutually exclusive at (eval 2) line 1, at end of line
Regexp modifier "/l" may not appear twice at (eval 2) line 1, at end of line
```

SNMPTT konfigurēšana

Vispirms snmptrapd jākonfigurē tā, lai tas izmantotu SNMPTT.

Note:
Lai nodrošinātu vislabāko veiktspēju, SNMPTT jākonfigurē kā dēmons, izmantojot **snmptthandler-embedded**, lai nodotu tam trapi. Skatiet norādījumus par SNMPTT konfigurēšanu.

Kad SNMPTT ir konfigurēts trapu saņemšanai, konfigurējiet `snmptt.ini`:

1. iespējojiet Perl moduļa izmantošanu no NET-SNMP pakotnes:

```
net_snmp_perl_enable = 1
```

2. reģistrējiet trapus trapu failā, ko nolasīs Zabbix:

```
log_enable = 1
log_file = [TRAP FILE]
```

3. iestatiet datuma un laika formātu:

```
date_time_format = %Y-%m-%dT%H:%M:%S%z
```

Warning:
Pakotne "net-snmp-perl" tika noņemta RHEL 8.0-8.2 versijās; atkārtoti pievienota RHEL 8.3. Plašāku informāciju skatiet zināmajās problēmās.

Tagad formatējiet trapus, lai Zabbix tos varētu atpazīt (rediģējiet snmptt.conf):

1. Katram FORMAT priekšrakstam jāsākas ar "ZBXTRAP [address]", kur [address] tiks salīdzināta ar Zabbix SNMP saskarņu IP un DNS adresēm. Piemēram:

```
EVENT coldStart .1.3.6.1.6.3.1.1.5.1 "Status Events" Normal
FORMAT ZBXTRAP $aA Device reinitialized (coldStart)
```

2. Plašāku informāciju par SNMP trapu formātu skatiet zemāk.

Attention:
Neizmantojiet nezināmus trapus — Zabbix tos nevarēs atpazīt. Nezināmus trapus var apstrādāt, definējot vispārīgu notikumu failā snmptt.conf:


```
EVENT general .* "General event" Normal
```

SNMP trap formāts

Visiem pielāgotajiem Perl trap uztvērējiem un SNMPTT trap konfigurācijai trap ir jāformatē šādi:

```
[timestamp] [trap, 1. daļa] ZBXTRAP [address] [trap, 2. daļa]
```

kur

- [timestamp] - laika zīmogs formātā "%Y-%m-%dT%H:%M:%S%z"
- ZBXTRAP - galvene, kas norāda, ka šajā rindā sākas jauns trap
- [address] - IP adrese, kas tiek izmantota, lai atrastu šim trap atbilstošo hostu

Ņemiet vērā, ka apstrādes laikā "ZBXTRAP" un "[address]" tiks izgriezti no ziņojuma. Ja trap ir formatēts citādi, Zabbix var neparedzēti parsēt trap.

Trap piemērs:

```
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
```

Tā rezultātā SNMP saskarnei ar IP=192.168.1.1 tiks iegūts šāds trap:

```
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
```

Sistēmas prasības

Note:
Ieteicams instalēt MIB failus, lai nodrošinātu, ka vienumu vērtības tiek attēlotas pareizajā formātā. Bez MIB failiem var rasties formatēšanas problēmas, piemēram, vērtību attēlošana HEX formātā UTF-8 vietā vai otrādi.

Lielu failu atbalsts

Zabbix atbalsta lielus failus SNMP trapper failiem. Maksimālais faila izmērs, ko Zabbix var nolasīt, ir 2^63 (8 EiB). Ņemiet vērā, ka failu sistēma var noteikt zemāku faila izmēra ierobežojumu.

Žurnālfailu rotācija

Zabbix nenodrošina nekādu žurnālfailu rotācijas sistēmu — tas jānodrošina lietotājam. Žurnālfailu rotācijai vispirms jāpārdēvē vecais fails un tikai pēc tam tas jādzēš, lai netiktu zaudēti nekādi slazdi:

1. Zabbix atver slazdu failu pēdējā zināmajā atrašanās vietā un pāriet uz 3. soli
2. Zabbix pārbauda, vai pašlaik atvērtais fails ir rotēts, salīdzinot inode numuru ar norādītā slazdu faila inode numuru. Ja nav atvērta faila, Zabbix atiestata pēdējo atrašanās vietu un pāriet uz 1. soli.
3. Zabbix nolasa datus no pašlaik atvērtā faila un iestata jauno atrašanās vietu.
4. Jaunie dati tiek parsēti. Ja tas bija rotētais fails, fails tiek aizvērts un process atgriežas pie 2. soļa.
5. Ja nav jaunu datu, Zabbix gaida 1 sekundi un atgriežas pie 2. soļa.

Failu sistēma

Trap faila ieviešanas dēļ Zabbix nepieciešama failu sistēma, kas atbalsta inode, lai atšķirtu failus (informācija tiek iegūta ar stat() izsaukumu).

Iestatīšanas piemēri, izmantojot dažādas SNMP protokola versijas

Šajā piemērā tiek izmantots snmptrapd un Bash uztvērēja skripts, lai pārsūtītu trapus uz Zabbix serveri.

Iestatīšana:

1. Konfigurējiet Zabbix, lai palaistu SNMP trapper un iestatītu trapu failu. Pievienojiet `zabbix_server.conf`:

```
StartSNMPTrapper=1
SNMPTrapperFile=/var/lib/zabbix/snmptraps/snmptraps.log
```

2. Lejupielādējiet Bash skriptu uz `/usr/sbin/zabbix_trap_handler.sh`:

```
curl -o /usr/sbin/zabbix_trap_handler.sh https://raw.githubusercontent.com/zabbix/zabbix-docker/trunk/Dock
```

Ja nepieciešams, pielāgojiet skriptā mainīgo ZABBIX_TRAPS_FILE. Lai izmantotu noklusējuma vērtību, vispirms izveidojiet vecākdirektoriju:

```
mkdir -p /var/lib/zabbix/snmptraps
```

3. Pievienojiet tālāk norādīto `snmtrapd.conf` (skatiet darbojošos piemēru)

```
traphandle default /bin/bash /usr/sbin/zabbix_trap_handler.sh
```

> **Note:**
> Lai piemērotu konfigurācijas izmaiņas, snmptrapd, iespējams, būs jāpārstartē.

4. Izveidojiet SNMP vienumu TEST (ņemiet vērā sākotnējās konfigurācijas prasības):

   Tips: SNMP trap
 Informācijas tips: Log
 Hosta saskarne: SNMP 127.0.0.1
 Atslēga: `snmptrap["linkUp"]`
 Žurnāla laika formāts: yyyyMMdd.hhmmss

Ņemiet vērā, ka tiek izmantots ISO 8601 datuma un laika formāts.

5. Tālāk konfigurēsim `snmptrapd` izvēlētajai SNMP protokola versijai un nosūtīsim testa trapus, izmantojot utilītu `snmptrap`.

SNMPv1, SNMPv2

SNMPv1 un SNMPv2 protokoli izmanto autentifikāciju ar "community string". Tālāk redzamajā piemērā izmantosim "secret" kā community string. Tai jābūt iestatītai uz to pašu vērtību arī SNMP trapu sūtītājos.

Lūdzu, ņemiet vērā, ka, lai gan SNMPv2 joprojām tiek plaši izmantots produkcijas vidēs, tas nepiedāvā nekādu šifrēšanu un reālu sūtītāja autentifikāciju. Dati tiek sūtīti atklātā tekstā, tādēļ šīs protokola versijas drīkst izmantot tikai drošās vidēs, piemēram, privātā tīklā, un tās nekad nevajadzētu izmantot publiskā vai trešo pušu tīklā.

SNMP 1. versija mūsdienās faktiski vairs netiek izmantota, jo tā neatbalsta 64 bitu skaitītājus un tiek uzskatīta par mantotu protokolu.

Lai iespējotu SNMPv1 vai SNMPv2 trapu pieņemšanu, `snmptrapd.conf` jāpievieno šāda rinda. Aizstājiet "secret" ar SNMP community string, kas konfigurēta SNMP trapu sūtītājos:

```
authCommunity log,execute,net secret
```

Pēc tam varam nosūtīt testa trapu, izmantojot `snmptrap`. Šajā piemērā izmantosim bieži sastopamo "link up" OID:

```
snmptrap -v 2c -c secret localhost '' linkUp.0
```

SNMPv3

SNMPv3 novērš SNMPv1/v2 drošības problēmas un nodrošina autentifikāciju un šifrēšanu. Varat izmantot MD5 vai vairākas SHA autentifikācijas metodes un DES/vairākus AES kā šifru.

Lai iespējotu SNMPv3 pieņemšanu, pievienojiet `snmptrapd.conf` šādas rindas:

```
createUser -e 0x8000000001020304 traptest SHA mypassword AES
authuser log,execute traptest
```

> **Attention:**
> Lūdzu, ņemiet vērā atslēgvārdu "execute", kas ļauj izpildīt skriptus šim lietotāja drošības modelim.

```
snmptrap -v 3 -n "" -a SHA -A mypassword -x AES -X mypassword -l authPriv -u traptest -e 0x800000000102030
```

> **Warning:**
> Ja vēlaties izmantot spēcīgas šifrēšanas metodes, piemēram, AES192 vai AES256, lūdzu, izmantojiet net-snmp, sākot ar versiju 5.8. Iespējams, tas būs jāpārkompilē ar `configure` opciju: `--enable-blumenthal-aes`. Vecākas net-snmp versijas neatbalsta AES192/AES256. Skatiet arī: Strong Authentication or Encryption.

Verifikācija

Abos piemēros failā `/var/lib/zabbix/snmptraps/snmptraps.log` redzēsiet līdzīgas rindas:

```
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
```

Vienuma vērtība Zabbix būs:

```
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
```

Piemērs ar 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
```

Skatiet arī

- Zabbix emuāra raksts par SNMP slazdiem
- snmptrapd konfigurēšana (oficiālā net-snmp dokumentācija)
- snmptrapd konfigurēšana SNMPv3 paziņojumu saņemšanai (oficiālā net-snmp dokumentācija)

4 IPMI pārbaudes

Pārskats

Jūs varat uzraudzīt Intelligent Platform Management Interface (IPMI) ierīču veselības stāvokli un pieejamību Zabbix. Lai veiktu IPMI pārbaudes, Zabbix serverim sākotnēji jābūt konfigurētam ar IPMI atbalstu.

IPMI ir standartizēta saskarne datoru sistēmu attālinātai "lights-out" vai "out-of-band" pārvaldībai. Tā ļauj uzraudzīt aparatūras statusu tieši no tā sauktajām "out-of-band" pārvaldības kartēm neatkarīgi no operētājsistēmas vai no tā, vai iekārta vispār ir ieslēgta.

Zabbix IPMI uzraudzība darbojas tikai ierīcēm ar IPMI atbalstu (HP iLO, DELL DRAC, IBM RSA, Sun SSP u.c.).

IPMI pārvaldnieka process plāno IPMI pārbaudes, ko veic IPMI aptaujātāji. Vienu hostu vienlaikus vienmēr aptaujā tikai viens IPMI aptaujātājs, samazinot atvērto savienojumu skaitu ar BMC kontrolieriem. Tādējādi ir droši palielināt IPMI aptaujātāju skaitu, neuztraucoties par BMC kontrolieru pārslodzi. IPMI pārvaldnieka process tiek automātiski palaists, kad tiek palaists vismaz viens IPMI aptaujātājs.

Skatiet arī zināmās problēmas attiecībā uz IPMI pārbaudēm.

Konfigurācija

Hosta konfigurācija

Lai apstrādātu IPMI pārbaudes, hosts ir jākonfigurē. Jāpievieno IPMI saskarne, norādot attiecīgo IP adresi un porta numuru, kā arī jādefinē IPMI autentifikācijas parametri.

Plašāku informāciju skatiet hostu konfigurācijā.

Servera konfigurācija

Pēc noklusējuma Zabbix serveris nav konfigurēts palaist nevienu IPMI aptaujātāju, tādēļ visi pievienotie IPMI vienumi nedarbosies. Lai to mainītu, atveriet Zabbix servera konfigurācijas failu (zabbix_server.conf) ar root tiesībām un atrodiet šādu rindu:

```
# StartIPMIPollers=0
```

Noņemiet komentāra zīmi un iestatiet aptaujātāju skaitu, piemēram, uz 3, lai rinda būtu šāda:

```
StartIPMIPollers=3
```

Saglabājiet failu un pēc tam restartējiet zabbix_server.

Vienuma konfigurācija

Kad konfigurējat vienumu hosta līmenī:

- Atlasiet 'IPMI aģents' kā *Type*
- Ievadiet atslēgu, kas ir unikāla hosta ietvaros (piemēram, ipmi.fan.rpm)
- Laukā *Host interface* atlasiet atbilstošo IPMI saskarni (IP adrese un ports). Ņemiet vērā, ka hostā ir jābūt izveidotai IPMI saskarnei.
- Norādiet *IPMI sensor* (piemēram, 'FAN MOD 1A RPM' Dell Poweredge serverī), no kura iegūt metriku. Pēc noklusējuma jānorāda sensora ID. Ir iespējams arī izmantot prefiksus pirms vērtības:
  - `id:` - lai norādītu sensora ID;
  - `name:` - lai norādītu sensora pilno nosaukumu. Tas var būt noderīgi situācijās, kad sensorus var atšķirt, tikai norādot pilno nosaukumu.
- Atlasiet atbilstošo informācijas tipu ('Numeric (float)' šajā gadījumā; diskrētiem sensoriem — 'Numeric (unsigned)'), mērvienības (visticamāk, 'rpm') un jebkurus citus nepieciešamos vienuma atribūtus

Atbalstītās pārbaudes

IPMI aģents atbalsta iebūvēto vienumu **ipmi.get**, kas atgriež ar IPMI sensoriem saistītu informāciju un ko var izmantot IPMI sensoru atklāšanai.
 Atgrieztā vērtība: *JSON objekts*

Noildze un sesijas pārtraukšana

IPMI ziņojumu noildzes un atkārtoto mēģinājumu skaits ir definēts OpenIPMI bibliotēkā. Pašreizējā OpenIPMI dizaina dēļ nav iespējams padarīt šīs vērtības konfigurējamas Zabbix, ne saskarnes, ne vienuma līmenī.

IPMI sesijas neaktivitātes noildze LAN savienojumam ir 60 +/-3 sekundes. Pašlaik nav iespējams ieviest periodisku komandas Activate Session nosūtīšanu, izmantojot OpenIPMI. Ja no Zabbix uz konkrētu BMC ilgāk par BMC konfigurēto sesijas noildzi netiek veikta neviena IPMI vienuma pārbaude, tad nākamā IPMI pārbaude pēc noildzes beigām beigsies ar noildzi atsevišķu ziņojumu noildžu, atkārtotu mēģinājumu vai saņemšanas kļūdas dēļ. Pēc tam tiek atvērta jauna sesija un tiek uzsākta pilna BMC atkārtota skenēšana. Jaunās sesijas pārvaldībai var tikt atvērts jauns UDP ports. Neaktivitāte tiek definēta kā gan izejošo pieprasījumu, gan ienākošo atbilžu neesamība. Ja vēlaties izvairīties no nevajadzīgas BMC atkārtotas skenēšanas, ieteicams iestatīt IPMI vienuma aptaujas intervālu zem BMC konfigurētās IPMI sesijas neaktivitātes noildzes.

Piezīmes par IPMI diskrētajiem sensoriem

Lai atrastu sensorus hostā, palaidiet Zabbix serveri ar iespējotu **DebugLevel=4**. Pagaidiet dažas minūtes un atrodiet sensoru atklāšanas ierakstus Zabbix servera žurnālfailā:

```
$ 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
```

Lai atšifrētu IPMI sensoru tipus un stāvokļus, ir pieejama IPMI 2.0 specifikāciju kopija (lūdzu, ņemiet vērā, ka turpmāki atjauninājumi IPMI specifikācijai nav plānoti).

Pirmais parametrs, ar kuru sākt, ir ”reading_type”. Izmantojiet specifikācijas ”Table 42-1, Event/Reading Type Code Ranges”, lai atšifrētu ”reading_type” kodu. Lielākajai daļai sensoru mūsu piemērā ir ”reading_type:0x1”, kas nozīmē ”threshold” sensoru. ”Table 42-3, Sensor Type Codes” parāda, ka ”type:0x1” nozīmē temperatūras sensoru, ”type:0x2” — sprieguma sensoru, ”type:0x4” — ventilatoru utt. Threshold sensorus dažkārt sauc par ”analog” sensoriem, jo tie mēra nepārtrauktus parametrus, piemēram, temperatūru, spriegumu, apgriezienus minūtē.

Vēl viens piemērs — sensors ar ”reading_type:0x3”. ”Table 42-1, Event/Reading Type Code Ranges” norāda, ka nolasīšanas tipa kodi 02h-0Ch nozīmē ”Generic Discrete” sensoru. Diskrētajiem sensoriem ir līdz 15 iespējamiem stāvokļiem (citiem vārdiem — līdz 15 nozīmīgiem bitiem). Piemēram, sensoram ’CATERR’ ar ”type:0x7” ”Table 42-3, Sensor Type Codes” parāda, ka šis tips nozīmē ”Processor”, un atsevišķo bitu nozīme ir: 00h (vismazāk nozīmīgais bits) — IERR, 01h — Thermal Trip utt.

Mūsu piemērā ir daži sensori ar ”reading_type:0x6f”. Šiem sensoriem ”Table 42-1, Event/Reading Type Code Ranges” iesaka izmantot ”Table 42-3, Sensor Type Codes”, lai atšifrētu bitu nozīmi. Piemēram, sensoram ’Power Unit Stat’ ir tips ”type:0x9”, kas nozīmē ”Power Unit”. Nobīde 00h nozīmē ”PowerOff/Power Down”. Citiem vārdiem, ja vismazāk nozīmīgais bits ir 1, tad serveris ir izslēgts. Lai pārbaudītu šo bitu, var izmantot funkciju `bitand` ar masku ’1’. Trigera izteiksme varētu būt šāda:

```
bitand(last(/www.example.com/Power Unit Stat,#1),1)=1
```

lai brīdinātu par servera izslēgšanu.

Piezīmes par diskrēto sensoru nosaukumiem OpenIPMI-2.0.16, 2.0.17, 2.0.18 un 2.0.19

Diskrēto sensoru nosaukumiem OpenIPMI-2.0.16, 2.0.17 un 2.0.18 versijās bieži beigās ir pievienots papildu ”0” (vai kāds cits cipars vai burts). Piemēram, kamēr `ipmitool` un OpenIPMI-2.0.19 attēlo sensoru nosaukumus kā ”`PhysicalSecurity`” vai ”`CATERR`”, OpenIPMI-2.0.16, 2.0.17 un 2.0.18 versijās nosaukumi attiecīgi ir ”`PhysicalSecurity0`” vai ”`CATERR0`”.

Konfigurējot IPMI vienums ar Zabbix serveris, kas izmanto OpenIPMI-2.0.16, 2.0.17 un 2.0.18, izmantojiet šos nosaukumus, kas beidzas ar ”0”, IPMI aģents vienumu *IPMI sensor* laukā. Kad jūsu Zabbix serveris tiks jauninātsuz jaunu Linux distributīvu, kas izmanto OpenIPMI-2.0.19 (vai jaunāku versiju), vienumi ar šiem IPMI diskrētajiem sensoriem kļūs par ”NOT SUPPORTED”. Jums būs jāmaina to *IPMI sensor* nosaukumi (jānoņem beigās esošais ’0’) un kādu laiku jāpagaida, līdz tie atkal kļūs ”Enabled”.

Piezīmes par sliekšņa un diskrētā sensora vienlaicīgu pieejamību

Daži IPMI aģenti nodrošina gan sliekšņa sensoru, gan diskrēto sensoru ar vienu un to pašu nosaukumu. Priekšroka vienmēr tiek dota sliekšņa sensoram.

Piezīmes par savienojuma pārtraukšanu

Ja IPMI pārbaudes netiek veiktas (jebkāda iemesla dēļ: visi hosta IPMI vienumi ir atspējoti/nav atbalstīti, hosts ir atspējots/dzēsts, hosts ir apkopes režīmā utt.), IPMI savienojums tiks pārtraukts no Zabbix servera vai starpniekservera pēc 3 līdz 4 stundām atkarībā no laika, kad tika palaists Zabbix serveris/starpniekserveris.

5 Vienkāršās pārbaudes

Pārskats

Vienkāršās pārbaudes parasti tiek izmantotas attālinātām pakalpojumu pārbaudēm bez aģenta.

Ņemiet vērā, ka vienkāršajām pārbaudēm Zabbix aģents nav nepieciešams. Par vienkāršo pārbaužu apstrādi (ārējo savienojumu izveidi u.c.) ir atbildīgs Zabbix serveris/starpniekserveris.

Vienkāršo pārbaužu izmantošanas piemēri:

```
net.tcp.service[ftp,,155]
net.tcp.service[http]
net.tcp.service.perf[http,,8080]
net.udp.service.perf[ntp]
```

Note:
Lauki *Lietotājvārds* un *Parole* (ierobežoti līdz 255 rakstzīmēm) vienkāršās pārbaudes vienums konfigurācijā tiek izmantoti VMware uzraudzības vienumiem; citos gadījumos tie tiek ignorēti.

Atbalstītās pārbaudes

Vienumu atslēgas ir uzskaitītas bez neobligātajiem parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Skatiet arī VMware uzraudzības vienumu atslēgas.

Vienuma atslēga	Apraksts
icmpping	Hosta pieejamība, izmantojot ICMP ping.
icmppingloss	Zaudēto pakešu procentuālā daļa.
icmppingretry	Hosta pieejamība, izmantojot ICMP ping ar atkārtotiem mēģinājumiem.
icmppingsec	ICMP ping atbildes laiks.
net.tcp.service	Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus.
net.tcp.service.perf	Pārbauda TCP pakalpojuma veiktspēju.
net.udp.service	Pārbauda, vai pakalpojums darbojas un atbild uz UDP pieprasījumiem.
net.udp.service.perf	Pārbauda UDP pakalpojuma veiktspēju.

Vienuma atslēgas detaļas

Parametri bez leņķiekavām ir obligāti. Parametri, kas atzīmēti ar leņķiekavām **< >**, ir neobligāti.

icmpping[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Hosta pieejamība ar ICMP ping palīdzību.
 Atgrieztā vērtība: *0* - ICMP ping neizdevās; *1* - ICMP ping veiksmīgs.

Parametri:

- **target** - hosta IP adrese vai DNS nosaukums;
- **packets** - pakešu skaits;
- **interval** - laiks starp secīgām paketēm milisekundēs;
- **size** - paketes lielums baitos;
- **timeout** - noildze milisekundēs;
- **options** - izmanto, lai atļautu pāradresāciju: ja tukšs (noklusējuma vērtība), pāradresētās atbildes tiek uzskatītas par mērķa hosta nepieejamību; ja iestatīts uz *allow_redirect*, pāradresētās atbildes tiek uzskatītas par mērķa hosta pieejamību.

Skatiet arī noklusējuma vērtību tabulu.

Piemērs:

```
icmpping[,4] #Ja tiek saņemta vismaz viena no četrām paketēm, vienums atgriezīs 1.
```

icmppingloss[<target>,<packets>,<interval>,<size>,<timeout>,<options>]

 Zaudēto pakešu procentuālā daļa.
 Atgrieztā vērtība: *Float*.

Parametri:

- **target** - hosta IP vai DNS nosaukums;
- **packets** - pakešu skaits;
- **interval** - laiks starp secīgām paketēm milisekundēs;
- **size** - paketes izmērs baitos;
- **timeout** - noildze milisekundēs;
- **options** - tiek izmantots, lai atļautu pāradresāciju: ja tukšs (noklusējuma vērtība), pāradresētās atbildes tiek uzskatītas par mērķa hosta nepieejamību; ja iestatīts uz *allow_redirect*, pāradresētās atbildes tiek uzskatītas par mērķa hosta pieejamību.

Skatiet arī noklusējuma vērtību tabulu.

icmppingretry[<target>,<retries>,<backoff>,<size>,<timeout>,<options>]

 Hosta pieejamība ar ICMP ping ar atkārtotiem mēģinājumiem. Ja pirmā pakete ir veiksmīga, apstāties; ja pakete neizdodas, mēģināt atkārtoti, līdz tiek sasniegts parametrā `retries` definētais maksimālais mēģinājumu skaits. Šis vienums ir noderīgs, lai samazinātu tīklā nosūtīto pakešu skaitu.
 Atgrieztā vērtība: *0* - ICMP ping neizdodas; *1* - ICMP ping veiksmīgs.

Parametri:

- **target** - hosta IP adrese vai DNS nosaukums;
- **retries** - reižu skaits, cik tiks veikts mēģinājums pingot mērķi, neieskaitot pirmo mēģinājumu (0 vai vairāk; noklusējums 1);
- **backoff** - skaitlis, ar kuru gaidīšanas laiks tiek reizināts katrā nākamajā pieprasījumā (diapazons 1.0-5.0; noklusējums 1.0);
- **size** - paketes lielums baitos;
- **timeout** - noildze milisekundēs;
- **options** - izmanto, lai atļautu pāradresāciju: ja tukšs (noklusētā vērtība), pāradresētās atbildes tiek uzskatītas par mērķa hosta nepieejamību; ja iestatīts uz *allow_redirect*, pāradresētās atbildes tiek uzskatītas par mērķa hosta pieejamību.

Skatiet arī noklusēto vērtību tabulu.

icmppingsec[<target>,<packets>,<interval>,<size>,<timeout>,<mode>,<options>]

 ICMP ping atbildes laiks (sekundēs).
 Atgrieztā vērtība: *Float*.

Parametri:

- **target** - hosta IP adrese vai DNS nosaukums;
- **packets** - pakešu skaits;
- **interval** - laiks starp secīgām paketēm milisekundēs;
- **size** - paketes izmērs baitos;
- **timeout** - noildze milisekundēs;
- **mode** - iespējamās vērtības: *min*, *max* vai *avg* (noklusējums);
- **options** - izmanto, lai atļautu pāradresāciju: ja tukšs (noklusējuma vērtība), pāradresētās atbildes tiek uzskatītas par mērķa hosta nepieejamību; ja iestatīts uz *allow_redirect*, pāradresētās atbildes tiek uzskatītas par mērķa hosta pieejamību.

Komentāri:

- Aprēķinā netiek izmantotas paketes, kas ir zudušas vai kurām iestājusies noildze;
- Ja hosts nav pieejams (sasniegta noildze), vienums atgriezīs 0;
- Ja atgrieztā vērtība ir mazāka par 0.0001 sekundēm, vērtība tiks iestatīta uz 0.0001 sekundēm;
- Skatiet arī noklusējuma vērtību tabulu.

net.tcp.service[service,<ip>,<port>]

 Pārbauda, vai pakalpojums darbojas un pieņem TCP savienojumus.
 Atgrieztā vērtība: *0* - pakalpojums nav pieejams; *1* - pakalpojums darbojas.

Parametri:

- **service** - iespējamās vērtības: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (pēc noklusējuma tiek izmantota hosta IP/DNS adrese);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Komentāri:

- Ņemiet vērā, ka pakalpojumam *tcp* porta norādīšana ir obligāta;
- Šo pārbaužu rezultātā sistēmas dēmona žurnālfailos var parādīties papildu ziņojumi (parasti tiek reģistrētas SMTP un SSH sesijas);
- Šifrētu protokolu pārbaude (piemēram, IMAP portā 993 vai POP portā 995) pašlaik netiek atbalstīta. Kā risinājumu, lūdzu, izmantojiet `net.tcp.service[tcp,<ip>,port]` šādām pārbaudēm.

Piemērs:

```
net.tcp.service[ftp,,45] #Šo vienumu var izmantot, lai pārbaudītu FTP servera pieejamību TCP portā 45.
```

Attention:
Ja SELinux darbojas piespiedu režīmā, pielāgotās TCP/UDP vienkāršās pārbaudes var tikt bloķētas ar politiku. Lai pārbaudītu un atļautu jauno izejošo savienojumu, pārskatiet auditēšanas atteikumus: `ar grep denied /var/log/audit/audit.log`

net.tcp.service.perf[service,<ip>,<port>]

 Pārbauda TCP pakalpojuma veiktspēju.
 Atgrieztā vērtība: *Float*: *0.000000* - pakalpojums nav pieejams; *sekundes* - sekunžu skaits, kas pavadīts, veidojot savienojumu ar pakalpojumu.

Parametri:

- **service** - iespējamās vērtības: *ssh*, *ldap*, *smtp*, *ftp*, *http*, *pop*, *nntp*, *imap*, *tcp*, *https*, *telnet* (skatiet details);
- **ip** - IP adrese vai DNS nosaukums (pēc noklusējuma tiek izmantota hosta IP/DNS vērtība);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Komentāri:

- Ņemiet vērā, ka pakalpojumam *tcp* porta norādīšana ir obligāta;
- Šifrētu protokolu pārbaude (piemēram, IMAP uz porta 993 vai POP uz porta 995) pašlaik netiek atbalstīta. Kā apiešanas risinājumu, lūdzu, izmantojiet `net.tcp.service[tcp,<ip>,port]` šādām pārbaudēm.

Piemērs:

```
net.tcp.service.perf[ssh] #Šo vienumu var izmantot, lai pārbaudītu sākotnējās atbildes ātrumu no SSH serve
```

net.udp.service[service,<ip>,<port>]

 Pārbauda, vai pakalpojums darbojas un atbild uz UDP pieprasījumiem.
 Atgrieztā vērtība: *0* - pakalpojums nav pieejams; *1* - pakalpojums darbojas.

Parametri:

- **service** - iespējamās vērtības: *ntp* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (pēc noklusējuma tiek izmantota hosta IP/DNS vērtība);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta pakalpojuma porta numurs).

Piemērs:

```
net.udp.service[ntp,,45] #Šo vienumu var izmantot, lai pārbaudītu NTP pakalpojuma pieejamību UDP 45. portā
```

net.udp.service.perf[service,<ip>,<port>]

 Pārbauda UDP servisa veiktspēju.
 Atgrieztā vērtība: *Float*: *0.000000* - serviss nav pieejams; *seconds* - sekunžu skaits, kas pavadīts, gaidot atbildi no servisa.

Parametri:

- **service** - iespējamās vērtības: *ntp* (skatiet detalizētu informāciju);
- **ip** - IP adrese vai DNS nosaukums (pēc noklusējuma tiek izmantota hosta IP/DNS);
- **port** - porta numurs (pēc noklusējuma tiek izmantots standarta servisa porta numurs).

Piemērs:

```
net.udp.service.perf[ntp] #Šo vienumu var izmantot, lai pārbaudītu atbildes laiku no NTP servisa.
```

Attention:
Lai LDAP vienkāršajās pārbaudēs atbalstītu SourceIP (piemēram, `net.tcp.service[ldap]`), ir nepieciešama OpenLDAP versija 2.6.1 vai jaunāka.

Noildzes apstrāde

Elastīgas vienumu noildzes, lai gan tās tiek atbalstītas vienkāršajām pārbaudēm, neattiecas uz `icmpping*` un VMware vienumiem. Skatiet Elastīgu noildžu atbalsts.

ICMP pingi

Zabbix izmanto ārēju utilītu **fping**, lai apstrādātu ICMP pingus (**icmpping**, **icmppingloss**, **icmppingretry**, **icmppingsec**).

Instalēšana

fping nav iekļauts Zabbix komplektācijā, un tas ir jāinstalē atsevišķi:

- Dažādās Unix bāzētās platformās fping pakotne ir pieejama noklusējuma repozitorijos, taču tā nav iepriekš instalēta. Šādā gadījumā fping instalēšanai varat izmantot pakotņu pārvaldnieku.
- Zabbix nodrošina fping pakotnes priekš RHEL un tā atvasinājumiem. Lūdzu, ņemiet vērā, ka šīs pakotnes tiek nodrošinātas bez oficiāla atbalsta.
- fping var arī kompilēt no pirmkoda.

Konfigurācija

Norādiet fping atrašanās vietu Zabbix servera/starpniekservera konfigurācijas faila parametrā *FpingLocation* (vai parametrā *Fping6Location*, ja tiek izmantotas IPv6 adreses).

fping jābūt izpildāmam lietotājam, kura vārdā darbojas Zabbix serveris/starpniekserveris, un šim lietotājam jābūt pietiekamām tiesībām.

Skatiet arī: Zināmās problēmas par vienkāršo pārbaužu apstrādi ar fping versijām, kas ir zemākas par 3.10.

Noklusējuma vērtības

Noklusējuma vērtības, ierobežojumi un vērtību apraksts ICMP pārbaudes parametriem:

Parametrs	Mērvienība	Apraksts	Fping karo-dziņš	Noklusējuma vērtības iestata		Zabbix atļautie ierobežojumi	
				fping	**Zabbix**	**min**	**max**
packets	skaits	Uz mērķi nosūtīto pieprasījuma pakešu skaits	-C		3	1	10000
interval	milisekundes	Laiks, ko gaidīt starp secīgām paketēm uz atsevišķu mērķi	-p	1000		20	neierobežots
size	baiti	Paketes izmērs baitos 56 baiti uz x86, 68 baiti uz x86_64	-b	56 vai 68		24	65507

Parametrs	Mērvienība	Apraksts	Fping karo-dziņš	Noklusējuma vērtības iestata		Zabbix atļautie ierobežojumi	
timeout	milisekundes	**fping v3.x** - taimauts gaidīšanai pēc pēdējās nosūtītās paketes, ko ietekmē *-C* karodziņš **fping v4.x** - individuāls taimauts katrai paketei	-t	**fping v3.x** - 500 **fping v4.x** un jaunākās versijās - mantots no *-p* karodz-iņa, bet ne vairāk par 2000		50	neierobežots
retries	skaits	Reižu skaits, cik tiks mēģināts pingot mērķi, neskaitot pirmo mēģinājumu	-r	3	1	0	neierobežots
backoff factor	skaits	Skaitlis, ar kuru gaidīšanas laiks tiek reizināts katrā nākamajā pieprasījumā	-B	1.5	1.0	1.0	5.0

Noklusējuma vērtības var nedaudz atšķirties atkarībā no platformas un versijas.

Papildus tam Zabbix izmanto fping opcijas *-i interval ms* (nejaukt ar vienuma parametru *interval*, kas minēts tabulā iepriekš, kurš atbilst fping opcijai *-p*) un *-S source IP address* (vai *-I* vecākās fping versijās). Šīs opcijas tiek automātiski noteiktas, palaižot pārbaudes ar dažādām opciju kombinācijām. Zabbix mēģina noteikt minimālo vērtību milisekundēs, ko fping atļauj izmantot ar *-i*, pārbaudot 3 vērtības: 0, 1 un 10. Vērtība, kas pirmā izdodas, pēc tam tiek izmantota turpmākajām ICMP pārbaudēm. Šo procesu katrs ICMP pinger process veic atsevišķi.

Automātiski noteiktās fping opcijas tiek anulētas katru stundu un noteiktas no jauna nākamajā mēģinājumā veikt ICMP pārbaudi. Iestatiet DebugLevel>=4, lai skatītu šī procesa detaļas servera vai starpniekservera žurnālfailā.

Zabbix ieraksta IP adreses, kas jāpārbauda ar jebkuru no *icmpping** atslēgām, pagaidu failā, kas pēc tam tiek nodots fping. Ja vienumiem ir atšķirīgi atslēgas parametri, vienā failā tiek ierakstīti tikai tie, kuriem ir identiski atslēgas parametri. Visas IP adreses, kas ierakstītas vienā failā, fping pārbaudīs paralēli, tāpēc Zabbix ICMP pinger process patērēs fiksētu laiku neatkarīgi no IP adrešu skaita failā.

1 VMware uzraudzības vienumu atslēgas

VMware uzraudzības vienumu atslēgu saraksts ir pārvietots uz sadaļu VMware uzraudzība.

6 Žurnālfailu uzraudzība

Pārskats

Zabbix var izmantot centralizētai žurnālfailu uzraudzībai un analīzei ar/bez žurnālfailu rotācijas atbalsta.

Paziņojumus var izmantot, lai brīdinātu lietotājus, kad žurnālfails satur noteiktas virknes vai virkņu šablonus.

Lai uzraudzītu žurnālfailu, jums ir jābūt:

- Zabbix aģentam, kas darbojas hostā
- konfigurētam žurnālfaila uzraudzības vienumam

Attention:
Uzraudzītā žurnālfaila lieluma ierobežojums ir atkarīgs no lielu failu atbalsta.

Konfigurācija

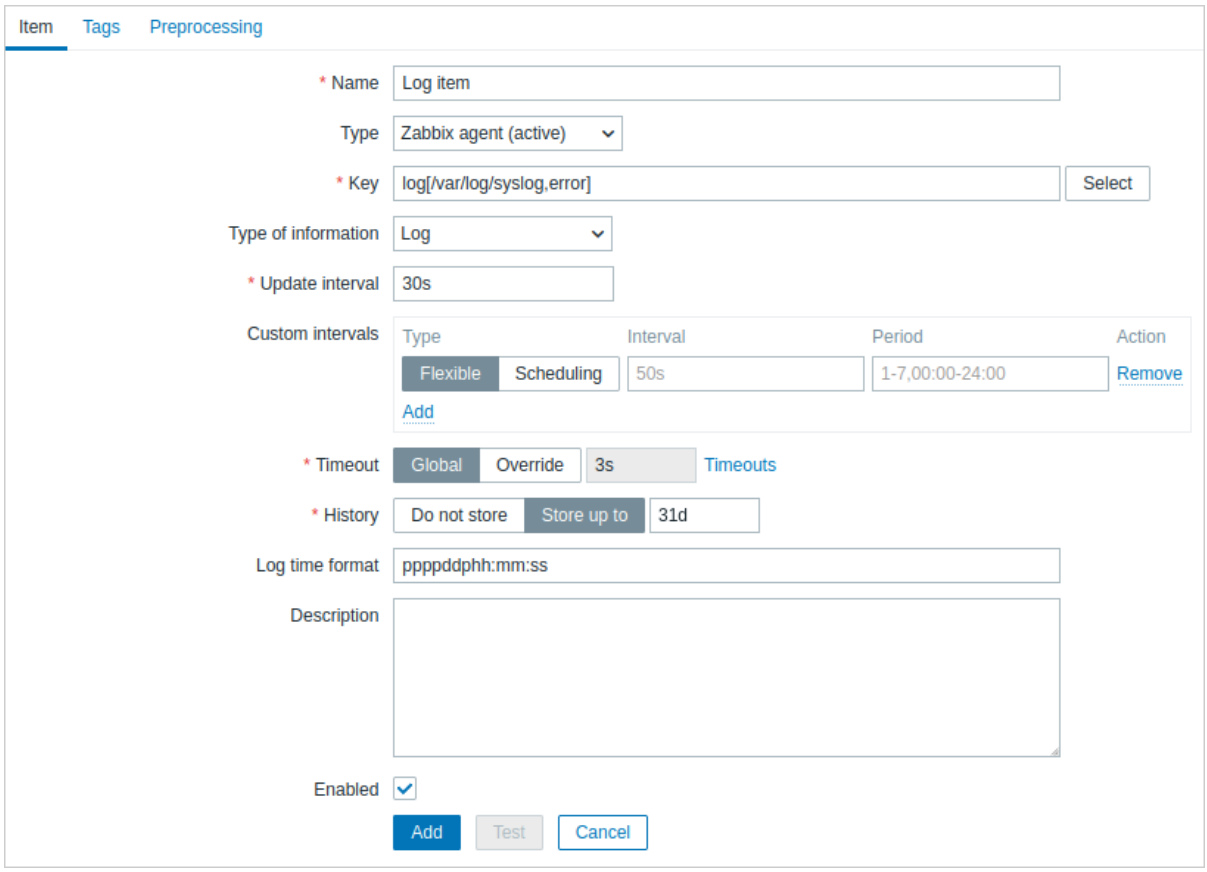
Pārbaudiet aģenta parametrus

Pārliecinieties, ka aģenta konfigurācijas failā:

- `Hostname` parametrs atbilst hosta nosaukumam lietotāja saskarnē.
- `ServerActive` parametrā ir norādīti serveri aktīvo pārbaužu apstrādei.

Vienuma konfigurācija

Konfigurējiet žurnālu uzraudzības vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Tieši žurnālu uzraudzības vienumiem ievadiet:

Type	Šeit atlasiet **Zabbix aģents (active)**.
Key	Izmantojiet vienu no šādām vienuma atslēgām: **log[]** vai **logrt[]**: Šīs divas vienuma atslēgas ļauj uzraudzīt žurnālus un filtrēt žurnāla ierakstus pēc satura regulārās izteiksmes, ja tā ir norādīta. Piemēram: `log[/var/log/syslog,error]`. Pārliecinieties, ka failam ir lasīšanas atļaujas lietotājam 'zabbix', pretējā gadījumā vienuma statuss tiks iestatīts kā 'unsupported'. **log.count[]** vai **logrt.count[]**: Šīs divas vienuma atslēgas ļauj atgriezt tikai atbilstošo rindu skaitu. Detalizētu informāciju par šo vienuma atslēgu un to parametru izmantošanu skatiet atbalstīto Zabbix aģents vienumu atslēgu sadaļā.
Type of information	Aizpildās automātiski: log[] vai logrt[] vienumiem - Log; log.count[] vai logrt.count[] vienumiem - `Numeric (unsigned)`. Ja pēc izvēles izmantojat parametru `output`, varat manuāli atlasīt atbilstošu informācijas tipu, kas nav Log. Ņemiet vērā, ka, izvēloties informācijas tipu, kas nav Log, tiks zaudēts lokālais laikspiedols.
Update interval (in sec)	Šis parametrs nosaka, cik bieži Zabbix aģents pārbaudīs, vai žurnāla failā ir kādas izmaiņas. Iestatot to uz 1 sekundi, tiks nodrošināts, ka jaunus ierakstus saņemat pēc iespējas ātrāk.

Log time format	Šajā laukā pēc izvēles varat norādīt šablonu žurnāla rindas laikspiedola parsēšanai. Atbalstītie vietturi: * **y**: *Gads (1970-2038)* * **M**: *Mēnesis (01-12)* * **d**: *Diena (01-31)* * **h**: *Stunda (00-23)* * **m**: *Minūte (00-59)* * **s**: *Sekunde (00-59)* Ja lauks tiek atstāts tukšs, laikspiedols Unix laikā tiks iestatīts uz 0, kas atbilst 1970. gada 1. janvārim. Piemēram, aplūkojiet šādu rindu no Zabbix aģenta žurnāla faila: " 23480:20100328:154718.045 Zabbix agent started. Zabbix 1.8.2 (revision 11211)." Tā sākas ar sešām rakstzīmju pozīcijām PID, kam seko datums, laiks un pārējais ziņojums. Šīs rindas žurnāla laika formāts būtu "pppppp:yyyyMMdd:hhmmss". Ņemiet vērā, ka rakstzīmes "p" un ":" ir vietturi un var būt jebkuras rakstzīmes, izņemot "yMdhms".

Svarīgas piezīmes

- Serveris un aģents uztur divus skaitītājus, kuros tiek saglabāts uzraudzītā žurnālfaila izmērs un pēdējās modificēšanas laiks (attiecībā uz `logrt`). Papildus:
  - Aģents iekšēji izmanto arī inode numurus (uz UNIX/GNU/Linux), failu indeksus (uz Microsoft Windows) un pirmo 512 žurnālfaila baitu MD5 kontrolsummas, lai uzlabotu lēmumu pieņemšanu gadījumos, kad žurnālfaili tiek saīsināti un rotēti.
  - UNIX/GNU/Linux sistēmās tiek pieņemts, ka failu sistēmas, kurās tiek glabāti žurnālfaili, ziņo inode numurus, kurus var izmantot failu izsekošanai.
  - Uz Microsoft Windows Zabbix aģents nosaka failu sistēmas tipu, kurā atrodas žurnālfaili, un izmanto:
    * NTFS failu sistēmās 64 bitu failu indeksus.
    * ReFS failu sistēmās (tikai начиная ar Microsoft Windows Server 2012) 128 bitu failu ID.
    * Failu sistēmās, kur failu indeksi mainās (piemēram, FAT32, exFAT), tiek izmantots rezerves algoritms, lai nenoteiktos apstākļos piemērotu saprātīgu pieeju, kad žurnālfaila rotācijas rezultātā rodas vairāki žurnālfaili ar vienādu pēdējās modificēšanas laiku.
  - Inode numurus, failu indeksus un MD5 kontrolsummas Zabbix aģents apkopo iekšēji. Tie netiek nosūtīti uz Zabbix serveris un tiek zaudēti, kad Zabbix aģents tiek apturēts.
  - Nemainiet žurnālfaila pēdējās modificēšanas laiku (piemēram, ar `touch`) un neaizstājiet uzraudzītu žurnālfailu, kopējot failu atpakaļ ar tā sākotnējo nosaukumu (tas izveido jaunu inode). Abos gadījumos Zabbix var uzskatīt failu par citu failu un atkārtoti nolasīt to no sākuma, kas var radīt dublētus brīdinājumus.
  - Ja `logrt[]` vienums atbilst vairākiem žurnālfailiem un Zabbix aģents seko jaunākajam no tiem, un šis jaunākais žurnālfails tiek dzēsts, tiek reģistrēts brīdinājuma ziņojums `"there are no files matching "<regexp mask>" in "<directory>"`. Zabbix aģents ignorē žurnālfailus, kuru modificēšanas laiks ir mazāks par jaunāko modificēšanas laiku, ko aģents ir redzējis pārbaudāmajam `logrt[]` vienums.
- Aģents sāk lasīt žurnālfailu no vietas, kur tas iepriekšējā reizē apstājās.
- Jau analizēto baitu skaits (izmēra skaitītājs) un pēdējās modificēšanas laiks (laika skaitītājs) tiek glabāti Zabbix datubāzē un nosūtīti aģentam, lai nodrošinātu, ka aģents sāk lasīt žurnālfailu no šī punkta gadījumos, kad aģents ir tikko palaists vai ir saņēmis vienumus, kas iepriekš bija atspējoti vai neatbalstīti. Tomēr, ja aģents no serveris ir saņēmis izmēra skaitītāju, kas nav nulle, bet `logrt[]` vai `logrt.count[]` vienums nespēj atrast atbilstošus failus, izmēra skaitītājs tiek atiestatīts uz 0, lai analizēšanu sāktu no sākuma, ja faili parādās vēlāk.
- Ikreiz, kad žurnālfails kļūst mazāks par aģentam zināmo žurnāla izmēra skaitītāju, skaitītājs tiek atiestatīts uz nulli un aģents sāk lasīt žurnālfailu no sākuma, ņemot vērā laika skaitītāju.
- Ja direktorijā ir vairāki atbilstoši faili ar vienādu pēdējās modificēšanas laiku, aģents cenšas korekti analizēt visus žurnālfailus ar vienādu modificēšanas laiku un izvairīties no datu izlaišanas vai vienu un to pašu datu analizēšanas divreiz, lai gan to nevar garantēt visās situācijās. Aģents nepieņem nekādu konkrētu žurnālfailu rotācijas shēmu un arī to nenosaka. Ja tiek uzrādīti vairāki žurnālfaili ar vienādu pēdējās modificēšanas laiku, aģents tos apstrādās leksikogrāfiski dilstošā secībā. Tādējādi dažām rotācijas shēmām žurnālfaili tiks analizēti un ziņoti to sākotnējā secībā. Citām rotācijas shēmām sākotnējā žurnālfailu secība netiks ievērota, kas var novest pie tā, ka atbilstošie žurnālfaila ieraksti tiek ziņoti mainītā secībā (šī problēma nerodas, ja žurnālfailiem ir atšķirīgi pēdējās modificēšanas laiki).
- Zabbix aģents apstrādā jaunus žurnālfaila ierakstus reizi *Update interval* sekundēs.
- Zabbix aģents nenosūta vairāk par **maxlines** žurnālfaila rindām sekundē. Šis ierobežojums novērš tīkla un CPU resursu pārslodzi un ignorē noklusējuma vērtību, ko nodrošina parametrs **MaxLinesPerSecond** aģenta konfigurācijas failā.
- Lai atrastu nepieciešamo virkni, Zabbix apstrādās 10 reizes vairāk jaunu rindu, nekā iestatīts MaxLinesPerSecond. Tādējādi, piemēram, ja `log[]` vai `logrt[]` vienums ir ar *Update interval* 1 sekunde, pēc noklusējuma aģents analizēs ne vairāk kā 200 žurnālfaila ierakstus un vienā pārbaudē nosūtīs uz Zabbix serveris ne vairāk kā 20 atbilstošus ierakstus. Palielinot **Max-**

LinesPerSecond aģenta konfigurācijas failā vai iestatot **maxlines** parametru vienuma atslēgā, ierobežojumu var palielināt līdz 10000 analizētiem žurnālfaila ierakstiem un 1000 atbilstošiem ierakstiem, kas vienā pārbaudē tiek nosūtīti uz Zabbix serveris. Ja *Update interval* ir iestatīts uz 2 sekundēm, vienas pārbaudes ierobežojumi būs 2 reizes lielāki nekā ar *Update interval* 1 sekunde.

- Papildus tam `log` un `log.count` vērtības vienmēr ir ierobežotas līdz 50% no aģenta sūtīšanas bufera izmēra, pat ja tajā nav nevienas ne-žurnāla vērtības. Tāpēc, lai **maxlines** vērtības tiktu nosūtītas vienā savienojumā (nevis vairākos savienojumos), aģenta parametrs BufferSize ir jābūt vismaz maxlines x 2. Zabbix aģents var augšupielādēt datus žurnālu apkopošanas laikā un tādējādi atbrīvot buferi, savukārt Zabbix aģents 2 pārtrauks žurnālu apkopošanu, līdz dati tiks augšupielādēti un buferis atbrīvots; tas tiek veikts asinhroni.
- Ja nav žurnāla vienumu, viss aģenta bufera izmērs tiek izmantots ne-žurnāla vērtībām. Kad parādās žurnāla vērtības, tās pēc vajadzības aizstāj vecākās ne-žurnāla vērtības līdz noteiktajiem 50%.
- Žurnālfaila ierakstiem, kas ir garāki par 256kB, tikai pirmie 256kB tiek salīdzināti ar regulāro izteiksmi, bet pārējā ieraksta daļa tiek ignorēta. Tomēr, ja Zabbix aģents tiek apturēts, kamēr tas apstrādā garu ierakstu, aģenta iekšējais stāvoklis tiek zaudēts un garais ieraksts pēc aģenta atkārtotas palaišanas var tikt analizēts vēlreiz un citādi.
- Īpaša piezīme par ceļa atdalītājiem "\": ja file_format ir "file\.log", tad nedrīkst būt direktorija "file", jo nav iespējams vienozīmīgi noteikt, vai "." ir atsoļots vai arī tas ir faila nosaukuma pirmais simbols.
- Regulārās izteiksmes `logrt` gadījumā tiek atbalstītas tikai faila nosaukumā; direktorija regulārās izteiksmes atbilstība netiek atbalstīta.
- UNIX platformās `logrt[]` vienums kļūst par NOTSUPPORTED, ja direktorija, kurā paredzēts atrast žurnālfailus, neeksistē.
- Uz Microsoft Windows, ja direktorija neeksistē, vienums nekļūs par NOTSUPPORTED (piemēram, ja direktorijas nosaukums vienuma atslēgā ir uzrakstīts kļūdaini).
- Žurnālfailu neesamība `logrt[]` vienumam nepadara to par NOTSUPPORTED. `logrt[]` vienuma žurnālfailu lasīšanas kļūdas tiek reģistrētas kā brīdinājumi Zabbix aģenta žurnālfailā, bet nepadara vienumu par NOTSUPPORTED.
- Zabbix aģenta žurnālfails var būt noderīgs, lai noskaidrotu, kāpēc `log[]` vai `logrt[]` vienums kļuva par NOTSUPPORTED. Zabbix var uzraudzīt savu aģenta žurnālfailu, izņemot gadījumus, kad DebugLevel=4 vai DebugLevel=5.
- Meklējot jautājuma zīmi ar regulāro izteiksmi, piemēram, \?, var rasties kļūdaini pozitīvi rezultāti, ja teksta fails satur NUL simbolus, jo Zabbix tos aizstāj ar "?", lai turpinātu rindas apstrādi līdz jaunās rindas rakstzīmei.

Atbilstošās regulārās izteiksmes daļas izgūšana

Dažreiz mēs varam vēlēties izgūt tikai interesējošo vērtību no mērķa faila, nevis atgriezt visu rindu, kad tiek atrasta atbilstība regulārajai izteiksmei.

Žurnāla vienumiem ir iespēja izgūt vajadzīgās vērtības no atbilstošajām rindām. Tas tiek panākts ar papildu parametru **output** `log` un `logrt` vienumos.

Izmantojot parametru "output", ir iespējams norādīt atbilstības "tveršanas grupu", kas mūs var interesēt.

Tātad, piemēram,

```
log[/path/to/the/file,"large result buffer allocation.*Entries: ([0-9]+)",,,,\1]
```

vajadzētu ļaut atgriezt ierakstu skaitu, kāds ir atrodams šādā saturā:

```
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
```

Tiks atgriezts tikai skaitlis, jo **\1** attiecas uz pirmo un vienīgo tveršanas grupu: **([0-9]+)**.

Turklāt, pateicoties iespējai izgūt un atgriezt skaitli, šo vērtību var izmantot, lai definētu trigerus.

Parametra maxdelay izmantošana

Parametrs `maxdelay` žurnālu vienumos ļauj ignorēt dažas vecākas rindas no žurnālfailiem, lai nodrošinātu, ka jaunākās rindas tiek analizētas `maxdelay` sekunžu laikā.

Warning:
Norādot `maxdelay` > 0, var tikt **ignorēti svarīgi žurnālfaila ieraksti un palaisti garām brīdinājumi**. Izmantojiet to piesardzīgi un uz savu risku tikai tad, kad tas ir nepieciešams.

Pēc noklusējuma vienumi žurnālu uzraudzībai seko visām jaunajām rindām, kas parādās žurnālfailos. Tomēr ir lietojumprogrammas, kas noteiktās situācijās sāk rakstīt savos žurnālfailos milzīgu skaitu ziņojumu. Piemēram, ja datubāze vai DNS serveris nav pieejams, šādas lietojumprogrammas pārpludina žurnālfailus ar tūkstošiem gandrīz identisku kļūdu ziņojumu, līdz tiek atjaunota normāla darbība. Pēc noklusējuma visi šie ziņojumi tiks rūpīgi analizēti, un atbilstošās rindas tiks nosūtītas uz serveri, kā konfigurēts `log` un `logrt` vienumos.

Iebūvētā aizsardzība pret pārslodzi sastāv no konfigurējama parametra `maxlines` (aizsargā serveri no pārāk liela ienākošo atbilstošo žurnāla rindu skaita) un ierobežojuma 10*`maxlines` (aizsargā hosta CPU un I/O no pārslodzes, ko vienas pārbaudes laikā

rada aģents). Tomēr iebūvētajai aizsardzībai ir 2 problēmas. Pirmkārt, uz serveri tiek ziņots liels skaits potenciāli ne pārāk informatīvu ziņojumu, un tie aizņem vietu datubāzē. Otrkārt, ierobežotā sekundē analizējamo rindu skaita dēļ aģents var atpalikt no jaunākajiem žurnāla ierakstiem vairākas stundas. Ļoti iespējams, ka jūs vēlētos ātrāk saņemt informāciju par pašreizējo situāciju žurnālfailos, nevis stundām ilgi pārmeklēt vecus ierakstus.

Abu problēmu risinājums ir parametra `maxdelay` izmantošana. Ja ir norādīts `maxdelay` > 0, katras pārbaudes laikā tiek mērīts apstrādāto baitu skaits, atlikušo baitu skaits un apstrādes laiks. Pamatojoties uz šiem skaitļiem, aģents aprēķina aptuveno aizkavi — cik sekundes būtu nepieciešamas, lai analizētu visus atlikušos ierakstus žurnālfailā.

Ja aizkave nepārsniedz `maxdelay`, tad aģents turpina analizēt žurnālfailu kā parasti.

Ja aizkave ir lielāka par `maxdelay`, tad aģents **ignorē žurnālfaila daļu, to "pārlecot"** uz jaunu aptuvenu pozīciju, lai atlikušās rindas varētu tikt analizētas `maxdelay` sekunžu laikā.

Ņemiet vērā, ka aģents ignorētās rindas pat nenolasa buferī, bet aprēķina aptuvenu pozīciju, uz kuru pārlēkt failā.

Fakts, ka žurnālfaila rindas tiek izlaistas, tiek reģistrēts aģenta žurnālfailā šādi:

```
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
```

Skaitlis "to byte" ir aptuvens, jo pēc "pārlēciena" aģents pielāgo pozīciju failā uz žurnāla rindas sākumu, kas failā var atrasties tālāk vai agrāk.

Atkarībā no tā, kā žurnālfaila pieauguma ātrums salīdzinās ar tā analizēšanas ātrumu, jūs varat neredzēt nevienu "pārlēcienu", redzēt retus vai biežus "pārlēcienus", lielus vai mazus "pārlēcienus", vai pat nelielu "pārlēcienu" katrā pārbaudē. Sistēmas slodzes svārstības un tīkla latentums arī ietekmē aizkaves aprēķinu un līdz ar to "pārlēkšanu" uz priekšu, lai ievērotu parametru `maxdelay`.

Nav ieteicams iestatīt `maxdelay` < `update interval` (tas var izraisīt biežus nelielus "pārlēcienus").

Piezīmes par `copytruncate` žurnālfaila rotācijas apstrādi

`logrt` ar opciju `copytruncate` pieņem, ka dažādiem žurnālfailiem ir atšķirīgi ieraksti (vismaz to laikspiedoli atšķiras), tāpēc sākotnējo bloku MD5 summas (līdz pirmajiem 512 baitiem) būs atšķirīgas. Divi faili ar vienādām sākotnējo bloku MD5 summām nozīmē, ka viens no tiem ir oriģināls, bet otrs — kopija.

`logrt` ar opciju `copytruncate` cenšas korekti apstrādāt žurnālfailu kopijas, neziņojot par dublikātiem. Tomēr nav ieteicamas tādas situācijas kā vairāku žurnālfaila kopiju izveide ar vienādu laikspiedolu, žurnālfaila rotācija biežāk nekā logrt[] vienuma atjaunināšanas intervāls, kā arī bieža aģenta restartēšana. Aģents cenšas visas šīs situācijas apstrādāt iespējami korekti, taču labus rezultātus nevar garantēt visos apstākļos.

Piezīmes par pastāvīgajiem failiem log*[] vienumiem

Pastāvīgo failu mērķis

Kad tiek palaists Zabbix aģents, tas saņem aktīvo pārbaužu sarakstu no Zabbix servera vai starpniekservera. Attiecībā uz log*[] metrikām tas saņem apstrādātā žurnāla izmēru un modificēšanas laiku, lai noteiktu, no kurienes sākt žurnālfaila uzraudzību. Atkarībā no faktiskā žurnālfaila izmēra un modificēšanas laika, par ko ziņo failu sistēma, aģents izlemj vai nu turpināt žurnālfaila uzraudzību no apstrādātā žurnāla izmēra, vai atkārtoti analizēt žurnālfailu no sākuma.

Darbojošs aģents uztur plašāku atribūtu kopu, lai starp pārbaudēm izsekotu visus uzraudzītos žurnālfailus. Šis atmiņā glabātais stāvoklis tiek zaudēts, kad aģents tiek apturēts.

Jaunais neobligātais parametrs **persistent_dir** norāda direktoriju, kurā failā glabāt šī log[], log.count[], logrt[] vai logrt.count[] vienuma stāvokli. Žurnāla vienuma stāvoklis tiek atjaunots no pastāvīgā faila pēc Zabbix aģenta restartēšanas.

Galvenais lietošanas gadījums ir žurnālfaila uzraudzība, kas atrodas spoguļotā failu sistēmā. Līdz noteiktam brīdim žurnālfailā tiek rakstīts abos spoguļos. Pēc tam spoguļi tiek atdalīti. Aktīvajā kopijā žurnālfails turpina augt, tajā tiek pievienoti jauni ieraksti. Zabbix aģents to analizē un nosūta serverim apstrādāto žurnālu izmēru un modificēšanas laiku. Pasīvajā kopijā žurnālfails paliek nemainīgs un ievērojami atpaliek no aktīvās kopijas. Vēlāk operētājsistēma un Zabbix aģents tiek pārstartēti no pasīvās kopijas. Apstrādātā žurnāla izmērs un modificēšanas laiks, ko Zabbix aģents saņem no servera, var nebūt derīgi situācijai pasīvajā kopijā. Lai turpinātu žurnālfaila uzraudzību no vietas, kur aģents to pārtrauca failu sistēmas spoguļa atdalīšanas brīdī, aģents atjauno savu stāvokli no pastāvīgā faila.

Aģenta darbība ar pastāvīgo failu

Palaišanas laikā Zabbix aģents neko nezina par pastāvīgajiem failiem. Tikai pēc aktīvo pārbaužu saraksta saņemšanas no Zabbix servera (starpniekservera) aģents redz, ka daži žurnāla vienumi ir jānodrošina ar pastāvīgajiem failiem norādītajos direktorijos.

Aģenta darbības laikā pastāvīgie faili tiek atvērti rakstīšanai (ar fopen(filename, "w")) un pārrakstīti ar jaunākajiem datiem. Iespēja zaudēt pastāvīgā faila datus, ja pārrakstīšana un failu sistēmas spoguļa sadalīšanās notiek vienlaikus, ir ļoti maza, tādēļ tam nav

paredzēta īpaša apstrāde. Rakstīšanai pastāvīgajā failā NETIEK veikta piespiedu sinhronizācija ar datu nesēju (fsync() netiek izsaukts).

Pārrakstīšana ar jaunākajiem datiem tiek veikta pēc tam, kad atbilstošais žurnālfaila ieraksts vai metadati (apstrādātā žurnāla izmērs un modifikācijas laiks) ir veiksmīgi nosūtīti uz Zabbix serveri. Tas var notikt pat tik bieži kā katrā vienuma pārbaudē, ja žurnālfails turpina mainīties.

Aģenta apturēšanas laikā netiek veiktas nekādas īpašas darbības.

Pēc aktīvo pārbaužu saraksta saņemšanas aģents atzīmē novecojušos pastāvīgos failus noņemšanai. Pastāvīgais fails kļūst novecojis, ja:

1. Atbilstošais žurnāla vienums vairs netiek uzraudzīts.
2. Žurnāla vienums tiek pārkonfigurēts ar citu **persistent_dir** atrašanās vietu nekā iepriekš.

Noņemšana tiek veikta ar 24 stundu aizkavi, jo žurnālfaili stāvoklī NOTSUPPORTED netiek iekļauti aktīvo pārbaužu sarakstā, taču vēlāk tie var kļūt par SUPPORTED, un to pastāvīgie faili būs noderīgi.

Ja aģents tiek apturēts, pirms ir pagājušas 24 stundas, novecojušie faili netiks dzēsti, jo Zabbix aģents vairs nesaņem informāciju par to atrašanās vietu no Zabbix serveris.

Warning:
Ja žurnāla vienuma **persistent_dir** tiek pārkonfigurēts atpakaļ uz iepriekšējo **persistent_dir** atrašanās vietu laikā, kad aģents ir apturēts, un lietotājs nav izdzēsis veco pastāvīgo failu, aģenta stāvoklis tiks atjaunots no vecā pastāvīgā faila, kā rezultātā var tikt izlaisti ziņojumi vai radīti kļūdaini brīdinājumi.

Pastāvīgo failu nosaukšana un atrašanās vieta

Zabbix aģents atšķir aktīvās pārbaudes pēc to atslēgām. Piemēram, logrt[/home/zabbix/test.log] un logrt[/home/zabbix/test.log,] ir dažādi vienumi. Mainot vienumu logrt[/home/zabbix/test.log„,10] lietotāja saskarnē uz logrt[/home/zabbix/test.log„,20], rezultātā vienums logrt[/home/zabbix/test.log„,10] tiks dzēsts no aģenta aktīvo pārbaužu saraksta un tiks izveidots vienums logrt[/home/zabbix/test.log„,20] (daži atribūti tiek pārnesti, veicot izmaiņas lietotāja saskarnē/serverī, bet ne aģentā).

Faila nosaukums sastāv no vienuma atslēgas MD5 summas ar pievienotu vienuma atslēgas garumu, lai samazinātu kolīziju iespējamību. Piemēram, vienuma logrt[/home/zabbix50/test.log„„„„„/home/zabbix50/agent_private] stāvoklis tiks saglabāts pastāvīgajā failā c963ade4008054813bbc0a650bb8e09266.

Vairāki žurnāla vienumi var izmantot vienu un to pašu **persistent_dir** vērtību.

persistent_dir tiek norādīts, ņemot vērā konkrētus failu sistēmas izkārtojumus, montēšanas punktus un montēšanas opcijas, kā arī krātuves spoguļošanas konfigurāciju — pastāvīgajam failam jāatrodas tajā pašā spoguļotajā failu sistēmā, kur uzraudzītais žurnāla fails.

Ja direktoriju **persistent_dir** nevar izveidot vai tas neeksistē, vai arī Zabbix aģenta piekļuves tiesības neļauj izveidot/rakstīt/lasīt/dzēst failus, žurnāla vienums kļūst par NOTSUPPORTED.

Ja piekļuves tiesības uz pastāvīgās glabātuves failiem tiek noņemtas aģenta darbības laikā vai rodas citas kļūdas (piemēram, disks ir pilns), tad kļūdas tiek reģistrētas aģenta žurnālfailā, bet žurnāla vienums nekļūst par NOTSUPPORTED.

I/O slodze

Vienuma pastāvīgais fails tiek atjaunināts pēc katras datu paketes (kas satur vienuma datus) veiksmīgas nosūtīšanas uz serveri. Piemēram, noklusējuma `BufferSize` ir 100. Ja žurnāla vienums ir atradis 70 atbilstošus ierakstus, tad pirmie 50 ieraksti tiks nosūtīti vienā paketē, pastāvīgais fails tiks atjaunināts, pēc tam atlikušie 20 ieraksti tiks nosūtīti (iespējams, ar zināmu aizkavi, kamēr tiek uzkrāti papildu dati) otrajā paketē, un pastāvīgais fails tiks atjaunināts vēlreiz.

Darbības, ja saziņa starp aģentu un serveri neizdodas

Katrai atbilstošajai rindai no `log[]` un `logrt[]` vienuma, kā arī katram `log.count[]` un `logrt.count[]` vienuma pārbaudes rezultātam ir nepieciešama brīva vieta aģenta sūtīšanas bufera tam paredzētajā 50% apgabalā. Bufera elementi tiek regulāri nosūtīti uz serveri (vai starpniekserveri), un bufera vietas atkal kļūst brīvas.

Kamēr aģenta sūtīšanas bufera paredzētajā žurnālu apgabalā ir brīvas vietas un saziņa starp aģentu un serveri (vai starpniekserveri) neizdodas, žurnālu uzraudzības rezultāti tiek uzkrāti sūtīšanas buferī. Tas palīdz mazināt īslaicīgu saziņas traucējumu ietekmi.

Ilgāku saziņas traucējumu laikā visas žurnālu vietas tiek aizņemtas, un tiek veiktas šādas darbības:

- `log[]` un `logrt[]` vienumu pārbaudes tiek apturētas. Kad saziņa tiek atjaunota un buferī ir pieejamas brīvas vietas, pārbaudes tiek atsāktas no iepriekšējās pozīcijas. Neviena atbilstošā rinda netiek zaudēta, tās vienkārši tiek ziņotas vēlāk.

- `log.count[]` un `logrt.count[]` pārbaudes tiek apturētas, ja `maxdelay = 0` (noklusējums). Darbība ir līdzīga `log[]` un `logrt[]` vienumiem, kā aprakstīts iepriekš. Ņemiet vērā, ka tas var ietekmēt `log.count[]` un `logrt.count[]` rezultātus: piemēram, viena pārbaude saskaita 100 atbilstošas rindas žurnāla failā, bet, tā kā buferī nav brīvu vietu, pārbaude tiek apturēta. Kad saziņa tiek atjaunota, aģents saskaita tās pašas 100 atbilstošās rindas un vēl 70 jaunas atbilstošās rindas. Tagad aģents nosūta skaitu = 170 tā, it kā tās būtu atrastas vienā pārbaudē.
- `log.count[]` un `logrt.count[]` pārbaudes ar `maxdelay > 0`: ja pārbaudes laikā nenotika "lēciens", tad darbība ir līdzīga iepriekš aprakstītajai. Ja notika "lēciens" pāri žurnāla faila rindām, tad tiek saglabāta pozīcija pēc "lēciena", bet saskaitītais rezultāts tiek atmests. Tādējādi aģents cenšas sekot līdzi augošam žurnāla failam pat saziņas traucējumu gadījumā.

Regulāro izteiksmju kompilēšanas un izpildlaika kļūdu apstrāde

Ja regulāro izteiksmi, kas izmantota `log[]`, `logrt[]`, `log.count[]` vai `logrt.count[]` vienumā, PCRE vai PCRE2 bibliotēka nevar kompilēt, tad vienums pāriet NOTSUPPORTED stāvoklī ar kļūdas ziņojumu. Lai turpinātu žurnāla vienuma uzraudzību, regulārā izteiksme ir jāizlabo.

Ja regulārā izteiksme tiek veiksmīgi kompilēta, bet neizdodas izpildlaikā (dažiem vai visiem žurnāla ierakstiem), tad žurnāla vienums paliek atbalstīts un uzraudzība turpinās. Izpildlaika kļūda tiek reģistrēta Zabbix aģenta žurnālfailā (bez žurnālfaila ieraksta).

Reģistrēšanas biežums ir ierobežots līdz vienai izpildlaika kļūdai vienā pārbaudē, lai Zabbix aģents varētu uzraudzīt pats savu žurnālfailu. Piemēram, ja tiek analizēti 10 ieraksti un 3 ierakstiem rodas regexp izpildlaika kļūda, aģenta žurnālā tiek izveidots viens ieraksts.

Izņēmums: ja `MaxLinesPerSecond=1` un atjaunināšanas intervāls=1 (vienā pārbaudē ir atļauts analizēt tikai 1 ierakstu), tad regexp izpildlaika kļūdas netiek reģistrētas.

zabbix_agentd izpildlaika kļūdas gadījumā reģistrē vienuma atslēgu, savukārt zabbix_agent2 reģistrē vienuma ID, lai palīdzētu noteikt, kuram žurnāla vienumam rodas izpildlaika kļūdas. Izpildlaika kļūdu gadījumā ir ieteicams pārveidot regulāro izteiksmi.

7 Aprēķinātie vienumi

Pārskats

Aprēķināms vienums ļauj izveidot aprēķinu, kas balstīts uz dažu esošu vienumu vērtībām. Piemēram, jūs varat vēlēties aprēķināt kāda vienuma vērtības stundas vidējo vai aprēķināt kopējo vērtību vienumu grupai. Tieši tam ir paredzēts aprēķināms vienums.

Aprēķinos var izmantot gan:

- atsevišķas individuālu vienumu vērtības
- sarežģītu filtru, lai atlasītu vairākus vienumus agregēšanai (skatiet agregētos aprēķinus sīkākai informācijai)

Aprēķināmi vienumi ir veids, kā izveidot virtuālus datu avotus. Visi aprēķini tiek veikti tikai Zabbix serverī. Vērtības tiek periodiski aprēķinātas, pamatojoties uz izmantoto aritmētisko izteiksmi.

Iegūtie dati tiek glabāti Zabbix datubāzē tāpat kā jebkuram citam vienumam; tiek glabātas gan vēstures, gan tendenču vērtības, un var tikt ģenerēti grafiki.

> **Note:**
> Ja aprēķina rezultāts ir peldošā komata vērtība, tā tiks apgriezta līdz veselam skaitlim, ja aprēķināmā vienuma informācijas tips ir *Numeric (unsigned)*.
> Tāpat, ja kešatmiņā nav nesenu datu un funkcijā nav definēts vaicājuma periods, Zabbix pēc noklusējuma meklēs datubāzē vēsturiskās vērtības līdz pat vienai nedēļai pagātnē.

Aprēķināmiem vienumiem ir tāda pati sintakse kā trigeru izteiksmēm. Ja esat pazīstams ar trigeru izteiksmēm, varat uztvert aprēķināmus vienumus kā funkcijas, kas analizē noteiktas vērtības, bet bez salīdzināšanas ar slieksni.

Aprēķināmos vienumos ir atļauta salīdzināšana ar virkņu vērtībām. Uz aprēķināmiem vienumiem var atsaukties makrosi vai citas entītijas tāpat kā uz jebkuru citu vienuma tipu.

Lai izmantotu aprēķināmus vienumus, izvēlieties vienuma tipu **Calculated**.

Konfigurējamie lauki

Atslēga ir unikāls vienuma identifikators (katram hostam). Jūs varat izveidot jebkādu atslēgas nosaukumu, izmantojot atbalstītos simbolus.

Aprēķina definīcija jāievada laukā **Formula**. Starp formulu un atslēgu nav saistības. Atslēgas parametri formulā nekādā veidā netiek izmantoti.

Vienkāršas formulas sintakse ir šāda:

`function(/host/key,<parameter1>,<parameter2>,...)`

kur:

Elements	Apraksts
`function`	Viena no atbalstītajām funkcijām: last, min, max, avg, count u.c.
`host`	Vienuma hosts, kas tiek izmantots aprēķinam. Pašreizējo hostu var izlaist (t.i., kā `function(//key,parameter,...)`).
`key`	Vienuma atslēga, kas tiek izmantota aprēķinam. Vienumi, kas atgriež vērtības ar bināro vai JSON datu tipu, netiek atbalstīti.
`parameter(s)`	Funkcijas parametri, ja nepieciešams. Tiek atbalstīti laika sufiksi un atmiņas lieluma sufiksi.

Attention:
lietotāja makrosi formulā tiks izvērsti, ja tie tiks izmantoti, lai norādītu uz funkcijas parametru, vienuma filtra parametru vai konstanti. Lietotāja makrosi NETIKS izvērsti, ja tie norāda uz funkciju, hosta nosaukumu, vienuma atslēgu, vienuma atslēgas parametru vai operatoru.

Sarežģītākā formulā var izmantot funkciju, operatoru un iekavu kombināciju. Jūs varat izmantot visas funkcijas un operatorus, kas tiek atbalstīti trigeru izteiksmēs. Loģika un operatoru prioritāte ir tieši tāda pati.

Atšķirībā no trigeru izteiksmēm, Zabbix apstrādā aprēķinātos vienumus atbilstoši vienuma atjaunināšanas intervālam, nevis saņemot jaunu vērtību.

Visiem vienumiem, uz kuriem atsaucas vēstures funkcijas aprēķinātā vienuma formulā, ir jāeksistē un jāapkopo dati. Turklāt, ja mainât atsauces vienuma atslēgu, jums manuāli jāatjaunina visas formulas, kurās šī atslēga tiek izmantota.

Aprēķinātais vienums var kļūt neatbalstīts vairākos gadījumos:

- atsauces vienums(-i)
  - nav atrasts
  - ir atspējots
  - pieder atspējotam hostam
  - nav atbalstīts (izņemot ar nodata() funkciju un operatoriem ar nezināmām vērtībām)
- nav datu funkcijas aprēķināšanai
- dalīšana ar nulli
- izmantota nepareiza sintakse

Lietošanas piemēri

Piemērs 1

Brīvās diska vietas procentuālās daļas aprēķināšana uz '/'.

Funkcijas **last** izmantošana:

```
100*last(//vfs.fs.size[/,free])/last(//vfs.fs.size[/,total])
```

Zabbix ņems jaunākās brīvās un kopējās diska vietas vērtības un aprēķinās procentuālo daļu saskaņā ar doto formulu.

Piemērs 2

Zabbix apstrādāto vērtību skaita 10 minūšu vidējās vērtības aprēķināšana.

Funkcijas **avg** izmantošana:

```
avg(/Zabbix Server/zabbix[wcache,values],10m)
```

Ņemiet vērā, ka plaša aprēķināto vienumu izmantošana ar gariem laika periodiem var ietekmēt Zabbix servera veiktspēju.

Piemērs 3

Kopējā joslas platuma aprēķināšana uz eth0.

Divu funkciju summa:

```
last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes])
```

Piemērs 4

Ienākošās datplūsmas procentuālās daļas aprēķināšana.

Sarežģītāka izteiksme:

```
100*last(//net.if.in[eth0,bytes])/(last(//net.if.in[eth0,bytes])+last(//net.if.out[eth0,bytes]))
```

Skatiet arī: Apkopoto aprēķinu piemēri

1 Agregētie aprēķini

Pārskats

Apkopotie aprēķini ir aprēķināma vienuma tips, kas ļauj Zabbix serverim apkopot informāciju no vairākiem vienumiem un pēc tam aprēķināt apkopoto vērtību atkarībā no izmantotās apkopošanas funkcijas.

Apkopotajiem aprēķiniem nav nepieciešams, lai uzraugāmajā hostā darbotos kāds aģents.

Sintakse

Lai iegūtu agregātus, izmantojiet vienu no atbalstītajām agregācijas funkcijām: `avg`, `max`, `min`, `sum` utt. Pēc tam pievienojiet funkciju **foreach** kā vienīgo parametru un tās vienumu filtru, lai atlasītu nepieciešamos vienumus:

```
aggregate_function(function_foreach(/host/key?[group="host group"],timeperiod))
```

Funkcija **foreach** (piemēram, *avg_foreach*, *count_foreach* utt.) atgriež vienu agregētu vērtību katram atlasītajam vienumam. Vienumi tiek atlasīti, izmantojot vienumu filtru (`/host/key?[group="host group"]`) no vienumu vēstures. Plašāku informāciju skatiet sadaļā foreach functions.

Ja dažiem vienumiem pieprasītajā periodā nav datu, tie aprēķinā netiek ņemti vērā. Ja datu nav nevienam vienumam, funkcija atgriezīs kļūdu.

Alternatīvi varat uzskaitīt vairākus vienumus kā agregācijas parametrus:

```
aggregate_function(function(/host/key,parameter),function(/host2/key2,parameter),...)
```

Ņemiet vērā, ka `function` šeit jābūt vēstures/tendenču funkcijai.

Note:
Ja agregācijas rezultāts ir peldošā komata vērtība, tā tiks noapaļota līdz veselam skaitlim, ja agregētā vienuma informācijas tips ir *Numeric (unsigned)*.

Lietotāja makrosi un zema līmeņa atklāšanas makrosi tiek atbalstīti šeit:

- vienuma atslēgas parametros
- funkciju parametros
- filtra nosacījumos (hostu grupas nosaukums un birkas nosaukums)
- izteiksmes konstantēs

Agregācijas aprēķins var kļūt neatbalstīts, ja:

- nav atrasts neviens no norādītajiem vienumiem (tas var notikt, ja vienuma atslēga ir nepareiza, neviens no vienumiem nepastāv vai visas iekļautās grupas ir nepareizas)
- funkcijas aprēķinam nav datu

Lietošanas piemēri

Atslēgu piemēri apkopojošiem aprēķiniem.

Piemērs 1

Kopējā diska vieta hostu grupai 'MySQL Servers'.

```
sum(last_foreach(/*/vfs.fs.size[/,total]?[group="MySQL Servers"]))
```

Piemērs 2

Visu hosta vienumu, kas atbilst net.if.in[*], jaunāko vērtību summa.

```
sum(last_foreach(/host/net.if.in[*]))
```

Piemērs 3

Vidējā procesora slodze hostu grupai 'MySQL Servers'.

```
avg(last_foreach(/*/system.cpu.load[,avg1]?[group="MySQL Servers"]))
```

Piemērs 4

5 minūšu vidējais vaicājumu skaits sekundē hostu grupai 'MySQL Servers'.

```
avg(avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m))
```

Piemērs 5

Vidējā CPU slodze visos hostos vairākās hostu grupās, kurām ir norādītās birkas.

```
avg(last_foreach(/*/system.cpu.load?[(group="Servers A" or group="Servers B" or group="Servers C") and (ta
```

Piemērs 6

Aprēķins, kas izmantots visa hostu grupas jaunāko vienumu vērtību summām.

```
sum(last_foreach(/*/net.if.out[eth0,bytes]?[group="video"])) / sum(last_foreach(/*/nginx_stat.sh[active]?[
```

Piemērs 7

Kopējais neatbalstīto vienumu skaits hostu grupā 'Zabbix servers'.

```
sum(last_foreach(/*/zabbix[host,,items_unsupported]?[group="Zabbix servers"]))
```

Piemērs 8

Jaunāko DNS pārbaužu skaitlisko rezultātu summa visos hostos. Parādītā vienuma forma ir `net.dns[192.0.2.0,example.com,A]` kā viena iespējamā atslēgas piemērs.

```
sum(last_foreach(/*/net.dns[*,*,*]))
```

Ņemiet vērā, ka aizstājējzīmēm jāatbilst parametru skaitam atslēgā (šeit `net.dns` ir trīs parametri: ip, name, type).

Pareizas/nepareizas sintakses piemēri

Izteiksmes (ieskaitot funkciju izsaukumus) nevar izmantot kā vēstures, tendenču vai foreach funkciju parametrus. Tomēr pašas šīs funkcijas var izmantot citu (ne-vēsturisko) funkciju parametros.

Izteiksme	Piemērs
Derīgs	`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)))`

Nederīgs	`sum(/host/key,10+2)`
	`sum(/host/key, avg(10,2))`
	`sum(/host/key,last(/host/key2))`

Ņemiet vērā, ka šādā izteiksmē:

```
sum(sum_foreach(//resptime[*],5m))/sum(count_foreach(//resptime[*],5m))
```

nevar garantēt, ka abām vienādojuma daļām vienmēr būs viena un tā pati vērtību kopa. Kamēr tiek novērtēta viena izteiksmes daļa, pieprasītajam periodam var pienākt jauna vērtība, un tad otrai izteiksmes daļai būs cita vērtību kopa.

8 Iekšējās pārbaudes

Pārskats

Iekšējās pārbaudes ļauj uzraudzīt Zabbix iekšējos procesus. Citiem vārdiem sakot, varat uzraudzīt, kas notiek ar Zabbix serveri vai Zabbix starpniekserveri.

Iekšējās pārbaudes tiek aprēķinātas:

- Zabbix serverī — ja hostu uzrauga serveris
- Zabbix starpniekserverī — ja hostu uzrauga starpniekserveris

Iekšējās pārbaudes apstrādā serveris vai starpniekserveris neatkarīgi no hosta uzturēšanas statusa.

Lai izmantotu šo vienumu, izvēlieties vienuma tipu **Zabbix internal**.

Note:
Iekšējās pārbaudes apstrādā Zabbix aptaujātāji.

Veiktspēja

Dažu iekšējo vienumu izmantošana var negatīvi ietekmēt veiktspēju. Šie vienumi ir:

- `zabbix[host,,items]`
- `zabbix[host,,items_unsupported]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[queue,,]`
- `zabbix[requiredperformance]`
- `zabbix[stats,,,queue,,]`
- `zabbix[triggers]`

Tiek ietekmētas arī lietotāja saskarnes sadaļas Sistēmas informācija un Rinda.

Atbalstītās pārbaudes

Vienumu atslēgas ir norādītas bez pielāgojamiem parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts
zabbix[boottime]	Zabbix servera vai Zabbix starpniekservera procesa palaišanas laiks sekundēs.
zabbix[cluster,discovery,nodes]	Atklāj augstas pieejamības klastera mezglus.
zabbix[connector_queue]	Savienotāju rindā ievietoto vērtību skaits.
zabbix[discovery_queue]	Tīkla pārbaužu skaits, kas ievietotas atklāšanas rindā.
zabbix[host,,items]	Iespējoto vienumu skaits hostā (atbalstītie un neatbalstītie).
zabbix[host,,items_unsupported]	Iespējoto neatbalstīto vienumu skaits hostā.
zabbix[host,maintenance]	Pašreizējais hosta uzturēšanas statuss.
zabbix[host,active_agent,available]	Aktīvā aģenta pārbaužu pieejamība hostā.
zabbix[host,discovery,interfaces]	Informācija par visām hostam konfigurētajām saskarnēm Zabbix lietotāja saskarnē.
zabbix[host,available]	Noteikta tipa pārbaužu galvenās saskarnes pieejamība hostā.
zabbix[hosts]	Uzraudzīto hostu skaits.
zabbix[items]	Iespējoto vienumu skaits (atbalstītie un neatbalstītie).
zabbix[items_unsupported]	Neatbalstīto vienumu skaits.
zabbix[java,]	Informācija par Zabbix Java gateway.
zabbix[lld_queue]	Zema līmeņa atklāšanas apstrādes rindā ievietoto vērtību skaits.
zabbix[preprocessing]	Pirmsapstrādes pārvaldnieka saņemto vērtību statistika.
zabbix[preprocessing_queue]	Pirmsapstrādes rindā ievietoto vērtību skaits.
zabbix[process,,]	Laika procentuālā daļa, ko konkrēts Zabbix process vai procesu grupa (identificēta pēc <type> un <mode>) pavadīja stāvoklī <state>.
zabbix[proxy,]	Informācija par Zabbix starpniekserveri.
zabbix[proxy,discovery]	Zabbix starpniekserveru saraksts.
zabbix[proxy group,available]	Tiešsaistē esošo starpniekserveru skaits starpniekserveru grupā.
zabbix[proxy group,pavailable]	Tiešsaistē esošo starpniekserveru procentuālā daļa starpniekserveru grupā.
zabbix[proxy group,proxies]	Zabbix starpniekserveru saraksts starpniekserveru grupā.
zabbix[proxy group,state]	Starpniekserveru grupas stāvoklis.
zabbix[proxy group,discovery]	Atgriež starpniekserveru grupu sarakstu ar konfigurācijas datiem un reāllaika datiem.
zabbix[proxy_buffer,buffer,]	Atgriež starpniekservera atmiņas bufera lietojuma statistiku.
zabbix[proxy_buffer,state,changes]	Atgriež stāvokļa maiņu skaitu starp diska/atmiņas bufera režīmiem kopš palaišanas.
zabbix[proxy_buffer,state,current]	Atgriež pašreizējo darba stāvokli, kurā tiek glabāti jaunie dati.
zabbix[proxy_history]	Vērtību skaits starpniekservera vēstures tabulā, kas gaida nosūtīšanu uz serveri.
zabbix[queue,]	Uzraudzīto vienumu skaits rindā, kuru aizkave ir vismaz <from> sekundes, bet mazāka par <to> sekundēm.
zabbix[rcache,]	Zabbix konfigurācijas kešatmiņas pieejamības statistika.

Vienuma atslēga	Apraksts
zabbix[requiredperformance]	Nepieciešamā Zabbix servera vai Zabbix starpniekservera veiktspēja, sagaidāmo jauno vērtību skaits sekundē.
zabbix[stats„]	Atgriež Zabbix servera vai starpniekservera iekšējos metriskos datus. Ja ir norādīti <ip> un <port>, metriskie dati tiek iegūti no attālās instances; pretējā gadījumā — no lokālās instances.
zabbix[stats„,queue„]	Atgriež Zabbix servera vai starpniekservera iekšējos rindas metriskos datus. Ja ir norādīti <ip> un <port>, metriskie dati tiek iegūti no attālās instances; pretējā gadījumā — no lokālās instances.
zabbix[tcache„]	Zabbix tendenču funkciju kešatmiņas efektivitātes statistika.
zabbix[triggers]	Iespējoto trigeru skaits Zabbix datubāzē, kur visiem vienumiem iespējotajos hostos ir iespējots statuss.
zabbix[uptime]	Zabbix servera vai starpniekservera procesa darbības laiks sekundēs.
zabbix[vcache,buffer,]	Zabbix vērtību kešatmiņas pieejamības statistika.
zabbix[vcache,cache,]	Zabbix vērtību kešatmiņas efektivitātes statistika.
zabbix[version]	Zabbix servera vai starpniekservera versija.
zabbix[vmware,buffer,]	Zabbix vmware kešatmiņas pieejamības statistika.
zabbix[vps,written]	Kopējais datubāzē ierakstīto vēstures vērtību skaits.
zabbix[wcache„]	Zabbix rakstīšanas kešatmiņas statistika un pieejamība.

Vienuma atslēgas informācija

- Parametri bez leņķiekavām ir obligāti un jāizmanto *tieši tādi, kā ir* (piemēram, ”host” un ”available” iekš `zabbix[host,<type>,availa`
- Parametri ar leņķiekavām **< >** jāaizstāj ar derīgu vērtību. Ja parametram ir noklusējuma vērtība, to var izlaist.
- Vērtības vienumiem un vienumu parametriem, kas apzīmēti kā ”not supported on proxy”, var iegūt tikai tad, ja hostu uzrauga serveris. Savukārt vērtības ”not supported on server” var iegūt tikai tad, ja hostu uzrauga starpniekserveris.

zabbix[boottime]

 Zabbix serveris vai Zabbix starpniekserveris procesa palaišanas laiks sekundēs.
 Atgrieztā vērtība: *Integer*.

zabbix[cluster,discovery,nodes]

 Atklāj augstas pieejamības klastera mezglus.
 Atgrieztā vērtība: *JSON objekts*.

Komentāri:

- Šo vienumu var izmantot zema līmeņa atklāšanā.
- Atgrieztie lauki lielākoties atbilst augstas pieejamības mezgla objekta īpašībām.
- Papildu lauki:
  - `db_timestamp` — pašreizējais servera laiks (Unix laikspiedols), kas iekļauts atbildē
  - `lastaccess_age` — sekundes kopš pēdējā mezgla sirdspuksta (db_timestamp - lastaccess)
  - `status` — mezgla stāvoklis:
    - 0 — gaidstāve
    - 1 — apturēts manuāli
    - 2 — nav pieejams
    - 3 — aktīvs

Atgriezto vērtību piemērs:


```
[
  {
    "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]

 Savienotāja rindā ievietoto vērtību skaits.
 Atgrieztā vērtība: *Integer*.

zabbix[discovery_queue]

 Tīkla pārbaužu skaits, kas ievietotas atklāšanas rindā.
 Atgrieztā vērtība: *Integer*.

zabbix[host,„items]

 Iespējoto vienumu skaits (atbalstīto un neatbalstīto) hostā.
 Atgrieztā vērtība: *Integer*.

zabbix[host,„items_unsupported]

 Iespējoto neatbalstīto vienumu skaits hostā.
 Atgrieztā vērtība: *Integer*.

zabbix[host,„maintenance]

 Pašreizējais hosta uzturēšanas statuss.
 Atgrieztās vērtības: *0* - normāls stāvoklis; *1* - uzturēšana ar datu vākšanu; *2* - uzturēšana bez datu vākšanas.

Komentāri:

- Šo vienumu vienmēr apstrādā Zabbix serveris neatkarīgi no hosta atrašanās vietas (uz servera vai starpniekservera). Starpniekserveris nesaņems šo vienumu kopā ar konfigurācijas datiem.
- Otrajam parametram jābūt tukšam, un tas ir rezervēts izmantošanai nākotnē.

zabbix[host,active_agent,available]

 Aktīvo aģenta pārbaužu pieejamība hostā.
 Atgrieztās vērtības: *0* - nezināms; *1* - pieejams; *2* - nav pieejams.

zabbix[host,discovery,interfaces]

 Visu Zabbix lietotāja saskarnē konfigurēto hosta saskarņu detalizēta informācija.
 Atgrieztā vērtība: *JSON objekts*.

Komentāri:

- Šo vienumu var izmantot zemā līmeņa atklāšanā.
- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[host,<type>,available]

 Noteikta pārbaužu tipa galvenās saskarnes pieejamība hostā.
 Atgrieztās vērtības: *0* - nezināma; *1* - pieejama; *2* - nav pieejama.

Parametri:

- **type** - *aģents*, *snmp*, *ipmi* vai *jmx*.

Komentāri:

- Vienuma vērtība tiek aprēķināta saskaņā ar konfigurācijas parametriem attiecībā uz hosta nesasniedzamību/nepieejamību.

zabbixhosts

 Uzraudzīto hostu skaits.
 Atgrieztā vērtība: *Integer*.

zabbix[items]

 Iespējoto vienumu skaits (atbalstīto un neatbalstīto).
 Atgrieztā vērtība: *Integer*.

zabbix[items_unsupported]

 Neatbalstīto vienumu skaits.
 Atgrieztā vērtība: *Integer*.

zabbix[java,,<param>]

 Informācija par Zabbix Java gateway.
 Atgrieztās vērtības: *1* — ja <param> ir *ping*; *Java gateway version* — ja <param> ir *version* (piemēram: "8.0.0").

Parametri:

- **param** — *ping* vai *version*.

Komentāri:

- Šo vienumu var izmantot, lai pārbaudītu Java gateway pieejamību, izmantojot `nodata()` trigera funkciju.
- Otrajam parametram jābūt tukšam; tas ir rezervēts izmantošanai nākotnē.

zabbix[lld_queue]

 Zema līmeņa atklāšanas apstrādes rindā ievietoto vērtību skaits.
 Atgrieztā vērtība: *Integer*.

Komentāri:

- Šo vienumu var izmantot, lai uzraudzītu zema līmeņa atklāšanas apstrādes rindas garumu.

zabbix[preprocessing]

 Statistika par vērtībām, ko saņēmis priekšapstrādes pārvaldnieks:

- *queued* - rindā esošo vērtību skaits un apjoms, kurām nepieciešama priekšapstrāde (skaitītājs)
- *direct* - rindā esošo vērtību skaits un apjoms, kurām priekšapstrāde nav nepieciešama (skaitītājs)
- *queue* - priekšapstrādes rindā ievietoto vērtību skaits (tas pats, kas `zabbix[preprocessing_queue]`)

Atgrieztā vērtība: *JSON*.

Atgriezto vērtību piemērs:

```
{"data":
    {
    "queued": {
        "count": 106,
        "size": 58620
    },
    "direct": {
        "count": 395,
        "size": 33843
    },
    "queue": 0
    }
}
```

zabbix[preprocessing_queue]

 Priekšapstrādes rindā ievietoto vērtību skaits.
 Atgrieztā vērtība: *Integer*.

Komentāri:

- Šo vienumu var izmantot, lai uzraudzītu priekšapstrādes rindas garumu.

zabbix[process,<type>,<mode>,<state>]

 Laika procentuālā daļa, ko konkrēts Zabbix process vai procesu grupa (identificēta pēc <type> un <mode>) pavadīja stāvoklī <state>. Tā tiek aprēķināta tikai par pēdējo minūti.
 Atgrieztā vērtība: *Float*.

Parametri:

- **type** — servera procesiem: *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*;
starpniekservera procesiem: *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* — vidējā vērtība visiem dotā tipa procesiem (noklusējums);
count — atgriež atzarojumu skaitu dotajam procesa tipam, <state> nav jānorāda;
max — maksimālā vērtība;
min — minimālā vērtība;
<*process number*> — procesa numurs (no 1 līdz iepriekš atzaroto instanču skaitam; piemēram, ja darbojas 4 trapper procesi, vērtība ir no 1 līdz 4);
- **state** — *busy* — process ir aizņemtā stāvoklī, piemēram, apstrādā pieprasījumu (noklusējums);
idle — process ir dīkstāves stāvoklī un neko nedara.

Komentāri:

- Ja <mode> ir Zabbix procesa numurs, kas nedarbojas (piemēram, ja darbojas 5 poller procesi, bet <mode> ir norādīts kā 6), šāds vienums kļūs neatbalstīts.
- Minimālā un maksimālā vērtība attiecas uz viena procesa izmantošanas procentuālo daļu. Tātad, ja 3 poller procesu grupā izmantošanas procenti katram procesam bija 2, 18 un 66, *min* atgriezīs 2, bet *max* atgriezīs 66.
- Procesi koplietojamajā atmiņā ziņo, ko tie dara, un self-monitoring process apkopo šos datus katru sekundi. Stāvokļa izmaiņas (busy/idle) tiek reģistrētas izmaiņu brīdī — tātad process, kas kļūst busy, tiek reģistrēts kā tāds un nemaina vai neatjaunina stāvokli, līdz tas kļūst idle. Tas nodrošina, ka pat pilnībā uzkārušies procesi tiks korekti reģistrēti kā 100% busy.
- Pašlaik "busy" nozīmē "nav miega režīmā", taču nākotnē var tikt ieviesti papildu stāvokļi — gaidīšana uz bloķēšanām, datubāzes vaicājumu izpilde utt. Ņemiet vērā, ka asinhronie poller procesi tiek uzskatīti par busy, ja tie ir sasnieguši limitu, kas iestatīts MaxConcurrentChecksPerPoller serveris/starpniekserveris konfigurācijas parametrā.
- Linux un lielākajā daļā citu sistēmu izšķirtspēja ir 1/100 sekundes.

Piemēri:

```
zabbix[process,poller,avg,busy] #poller procesu vidējais laiks, kas pēdējās minūtes laikā pavadīts, kaut k
zabbix[process,"icmp pinger",max,busy] #maksimālais laiks, ko jebkurš ICMP pinger process pēdējās minūtes
zabbix[process,"history syncer",2,busy] #laiks, ko history syncer process ar numuru 2 pēdējās minūtes laik
zabbix[process,trapper,count] #pašlaik darbojošos trapper procesu skaits
```

zabbix[proxy,<name>,<param>]

 Informācija par Zabbix starpniekserveri.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **name** - starpniekservera nosaukums;
- **param** - *lastaccess* - pēdējā no starpniekservera saņemtā sirdspukstu ziņojuma laikspiedols;
delay - cik ilgi savāktās vērtības nav nosūtītas; aprēķināts kā "starpniekservera aizkave" + ("pašreizējais servera laiks" - "starpniekservera lastaccess"), kur "starpniekservera aizkave" ir starpība starp pašreizējo starpniekservera laiku un vecākās nenosūtītās vērtības laikspiedolu starpniekserverī.

Komentāri:

- Šo vienumu vienmēr apstrādā Zabbix serveris neatkarīgi no hosta atrašanās vietas (uz servera vai starpniekservera).
- Funkciju `fuzzytime()` var izmantot, lai pārbaudītu starpniekservera pieejamību.

Piemērs:

```
zabbix[proxy,"Germany",lastaccess] #pēdējā no "Germany" starpniekservera saņemtā sirdspukstu ziņojuma laik
```

zabbix[proxy,discovery]

 Zabbix starpniekserveru saraksts ar nosaukumu, režīmu, šifrēšanu, saspiešanu, versiju, pēdējo redzēšanas reizi, hostu skaitu, vienumu skaitu, nepieciešamajām vērtībām sekundē (vps), versijas statusu (current/outdated/unsupported), noildzēm pēc vienuma tipa, starpniekserveru grupas nosaukumu (ja starpniekserveris pieder grupai), stāvokli (unknown/offline/online).
 Atgrieztā vērtība: *JSON objekts*.

zabbix[proxy group,<name>,available]

 Tiešsaistē esošo starpniekserveru skaits starpniekserveru grupā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

zabbix[proxy group,<name>,pavailable]

 Tiešsaistē esošo starpniekserveru procentuālā daļa starpniekserveru grupā.
 Atgrieztā vērtība: *Float*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

zabbix[proxy group,<name>,proxies]

 Zabbix starpniekserveru saraksts starpniekserveru grupā ar nosaukumu, režīmu, šifrēšanu, saspiešanu, versiju, pēdējo redzēšanas reizi, hostu skaitu, vienumu skaitu, nepieciešamajām vērtībām sekundē (vps), versijas statusu (current/outdated/unsupported), taimautiem, starpniekserveru grupas nosaukumu, stāvokli (unknown/offline/online).
 Atgrieztā vērtība: *JSON*.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

zabbix[proxy group,<name>,state]

 Starpniekserveru grupas stāvoklis.
 Atgrieztā vērtība: *0* - nezināms; *1* - bezsaistē; *2* - atkopjas; *3* - tiešsaistē; *4* - pasliktinās.

Parametri:

- **name** - starpniekserveru grupas nosaukums.

zabbix[proxy group,discovery]

 Atgriež starpniekserveru grupu sarakstu ar konfigurācijas datiem un reāllaika datiem. Konfigurācijas dati ietver starpniekserveru grupas nosaukumu, pārslēgšanās aizkavi un minimālo nepieciešamo tiešsaistē esošo starpniekserveru skaitu. Reāllaika dati ietver starpniekserveru grupas stāvokli (sīkāku informāciju skatiet komentāros), tiešsaistē esošo starpniekserveru skaitu un tiešsaistē esošo starpniekserveru procentuālo daļu.
 Atgrieztā vērtība: *JSON*.

Komentāri:

- Šis vienums neatgriež starpniekserverus, kas nepieder nevienai grupai.
- Ja vērtībai "failover_delay" vai "min_online" ir nederīga vērtība, tad tiek ziņota īpaša vērtība *-1*, lai to norādītu. Nederīgas vērtības var rasties, ja konfigurācijā tiek izmantoti makrosi un tos nevar izvērst par derīgu vērtību.
- Starpniekserveru grupas stāvoklis tiek ziņots kā vesels skaitlis: *0* - nezināms; *1* - bezsaistē; *2* - atkopjas; *3* - tiešsaistē; *4* - degradējas.

Atgriezto vērtību piemērs:

```
{
    "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>]

 Starpniekservera atmiņas bufera izmantošanas statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **mode**: *total* - kopējais bufera izmērs (var izmantot, lai pārbaudītu, vai atmiņas buferis ir iespējots);
free - brīvā bufera izmērs;
pfree - brīvā bufera procentuālā daļa;
used - izmantotā bufera izmērs;
pused - izmantotā bufera procentuālā daļa.

Komentāri:

- Atgriež kļūdu 'Proxy memory buffer is disabled', ja atmiņas buferis ir atspējots;
- Šis vienums netiek atbalstīts Zabbix serverī.

zabbix[proxy_buffer,state,changes]

 Atgriež stāvokļa izmaiņu skaitu starp diska/atmiņas bufera režīmiem kopš palaišanas.
 Atgrieztās vērtības: *Integer*; *0* - atmiņas buferis ir atspējots.

Komentāri:

- Biežas stāvokļa izmaiņas norāda, ka ir jāpalielina vai nu atmiņas bufera izmērs, vai vecums.

- Ja atmiņas bufera stāvoklis tiek uzraudzīts reti (piemēram, reizi minūtē), buferis var mainīt savu stāvokli, to nereģistrējot.

zabbix[proxy_buffer,state,current]

 Atgriež pašreizējo darba stāvokli, kurā tiek glabāti jaunie dati.
 Atgrieztās vērtības: *0* - disks; *1* - atmiņa.

Komentāri:

- ”0” tiek atgriezts arī tad, ja atmiņas buferis ir atspējots.

zabbix[proxy_history]

 Vērtību skaits starpniekservera vēstures tabulā, kas gaida nosūtīšanu uz serveri.
 Atgrieztās vērtības: *Integer*.

Komentāri:

- Šis vienums netiek atbalstīts Zabbix serverī.

zabbix[queue,<from>,<to>]

 Uzraudzīto vienumu skaits rindā, kuru aizkave ir vismaz <from> sekundes, bet mazāka par <to> sekundēm.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **from** - aizkave ir vismaz tik liela (noklusējums ir 6 sekundes);
- **to** - aizkave ir ne vairāk kā tik liela (noklusējums ir bezgalība).

Komentāri:

- Parametros tiek atbalstīti laika sufiksi (s,m,h,d,w).

Attention:
Zabbix starpniekserveris neņem vērā uzturēšanas periodus; sīkāku informāciju skatiet sadaļā Rindu aprēķināšana uzturēšanas laikā.

zabbix[rcache,<cache>,<mode>]

 Zabbix konfigurācijas kešatmiņas pieejamības statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **cache** - *buffer*;
- **mode** - *total* - kopējais bufera izmērs;
free - brīvā bufera izmērs;
pfree - brīvā bufera procentuālā daļa;
used - izmantotā bufera izmērs;
pused - izmantotā bufera procentuālā daļa.

zabbix[requiredperformance]

 Nepieciešamā Zabbix serveris vai Zabbix starpniekserveris veiktspēja, sagaidāmo jauno vērtību sekundē izteiksmē.
 Atgrieztā vērtība: *Float*.

Komentāri:

- Aptuveni atbilst ”Nepieciešamā servera veiktspēja, jaunas vērtības sekundē” sadaļā *Pārskati* > *Sistēmas informācija*.

zabbix[stats,<ip>,<port>]

 Atgriež Zabbix servera vai starpniekservera iekšējos metriku datus. Ja ir norādīti <ip> un <port>, metrikas tiek iegūtas no attālās instances; pretējā gadījumā — no lokālās instances.
 Atgrieztās vērtības: *JSON objekts*.

Parametri:

- **ip** - attāli vaicājamo serveru/starpniekserveru IP/DNS/tīkla masku saraksts (noklusējuma vērtība ir 127.0.0.1);
- **port** - attāli vaicājamā servera/starpniekservera ports (noklusējuma vērtība ir 10051).

Komentāri:

- Statistikas pieprasījums tiks pieņemts tikai no adresēm, kas mērķa instancē ir norādītas parametrā ’StatsAllowedIP’ serveris/starpniekserveris.
- Šis vienums atgriež atlasītu iekšējo metriku kopu. Papildinformāciju skatiet sadaļā Zabbix statistikas attālā uzraudzība.

zabbix[stats,<ip>,<port>,queue,<from>,<to>]

 Atgriež Zabbix servera vai starpniekservera iekšējās rindas metriku vērtības (skatiet `zabbix[queue,<from>,<to>]`). Ja ir norādīti <ip> un <port>, metriku vērtības tiek iegūtas no attālās instances; pretējā gadījumā — no lokālās instances.
 Atgrieztās vērtības: *JSON objekts*.

Parametri:

- **ip** - attāli vaicājamo serveru/starpniekserveru IP/DNS/tīkla masku saraksts (noklusējums ir 127.0.0.1);
- **port** - attāli vaicājamā servera/starpniekservera ports (noklusējums ir 10051);
- **from** - aizkavēts vismaz par (noklusējums ir 6 sekundes);
- **to** - aizkavēts ne vairāk kā par (noklusējums ir bezgalība).

Komentāri:

- Statistikas pieprasījums tiks pieņemts tikai no adresēm, kas mērķa instancē ir norādītas parametrā 'StatsAllowedIP' serveris/starpniekserveris.
- Šis vienums atgriež atlasītu iekšējo metriku kopu. Sīkāku informāciju skatiet Zabbix statistikas attālā uzraudzība.

Attention:
Zabbix starpniekserveris neņem vērā uzturēšanas periodus; sīkāku informāciju skatiet Rindu aprēķināšana uzturēšanas laikā.

zabbix[tcache,cache,<parameter>]

 Zabbix tendenču funkcijas kešatmiņas efektivitātes statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **parameter** - *all* - kopējais kešatmiņas pieprasījumu skaits (noklusējums);
hits - trāpījumu skaits kešatmiņā;
phits - trāpījumu procentuālā daļa kešatmiņā;
misses - netrāpījumu skaits kešatmiņā;
pmisses - netrāpījumu procentuālā daļa kešatmiņā;
items - kešatmiņā saglabāto vienumu skaits;
requests - kešatmiņā saglabāto pieprasījumu skaits;
pitems - kešatmiņā saglabāto vienumu procentuālā daļa no kešatmiņā saglabātajiem vienumiem + pieprasījumiem. Zems procentuālais rādītājs, visticamāk, nozīmē, ka kešatmiņas izmēru var samazināt.

Komentāri:

- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[triggers]

 Iespējoto trigeru skaits Zabbix datubāzē, kuriem visi vienumi ir iespējoti iespējotos hostos.
 Atgrieztā vērtība: *Integer*.

Komentāri:

- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[uptime]

 Zabbix servera vai starpniekservera procesa darbības laiks sekundēs.
 Atgrieztā vērtība: *Integer*.

zabbix[vcache,buffer,<mode>]

 Zabbix vērtību kešatmiņas pieejamības statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **mode** - *total* - kopējais bufera izmērs;
free - brīvā bufera izmērs;
pfree - brīvā bufera procentuālā daļa;
used - izmantotā bufera izmērs;
pused - izmantotā bufera procentuālā daļa.

Komentāri:

- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[vcache,cache,<parameter>]

 Zabbix vērtību kešatmiņas efektivitātes statistika.
 Atgrieztā vērtība: *Integer*. Ja <parameter> ir iestatīts uz *mode*, atgrieztā vērtība ir: *0* - normālais režīms; *1* - zemas atmiņas režīms.

Parametri:

- **parameter** - *requests* - kopējais pieprasījumu skaits;
hits - kešatmiņas trāpījumu skaits (vēstures vērtības, kas ņemtas no kešatmiņas);
misses - kešatmiņas netrāpījumu skaits (vēstures vērtības, kas ņemtas no datubāzes);
mode - vērtību kešatmiņas darbības režīms.

Komentāri:

- Kad zemas atmiņas režīms ir ieslēgts, vērtību kešatmiņa paliks šajā stāvoklī 24 stundas, pat ja problēma, kas izraisīja šo režīmu, tiek atrisināta agrāk.
- Jūs varat izmantot šo atslēgu ar *Change per second* priekšapstrādes soli, lai iegūtu vērtību skaita sekundē statistiku.
- Šis vienums netiek atbalstīts Zabbix starpniekserverī.

zabbix[version]

 Zabbix serveris vai starpniekserveris versija.
 Atgrieztā vērtība: *String*. Piemēram: `8.0.0`.

zabbix[vmware,buffer,<mode>]

 Zabbix vmware kešatmiņas pieejamības statistika.
 Atgrieztās vērtības: *Integer* (izmēram); *Float* (procentuālajai daļai).

Parametri:

- **mode** - *total* - kopējais bufera izmērs;
free - brīvā bufera izmērs;
pfree - brīvā bufera procentuālā daļa;
used - izmantotā bufera izmērs;
pused - izmantotā bufera procentuālā daļa.

zabbix[vps,written]

 Kopējais datubāzē ierakstīto vēstures vērtību skaits.
 Atgrieztā vērtība: *Integer*.

zabbix[wcache,<cache>,<mode>]

 Zabbix rakstīšanas kešatmiņas statistika un pieejamība.
 Atgrieztās vērtības: *Integer* (skaitam/izmēram); *Float* (procentuālajai vērtībai).

Parametri:

- **cache** - *values*, *history*, *index* vai *trend*;
- **mode** - (ar *values*) *all* (noklusējums) - kopējais Zabbix servera/starpniekservera apstrādāto vērtību skaits, izņemot neatbalstītos vienumus (skaitītājs);
float - apstrādāto peldošā komata vērtību skaits (skaitītājs);
uint - apstrādāto neparakstīto veselo skaitļu vērtību skaits (skaitītājs);
str - apstrādāto rakstzīmju/virkņu vērtību skaits (skaitītājs);
log - apstrādāto žurnāla vērtību skaits (skaitītājs);
text - apstrādāto teksta vērtību skaits (skaitītājs);
bin - apstrādāto bināro vērtību skaits (skaitītājs);
json - apstrādāto JSON vērtību skaits (skaitītājs);
not supported - reižu skaits, kad vienuma apstrādes rezultātā vienums kļuva neatbalstīts vai saglabāja šo stāvokli (skaitītājs);
(ar *history*, *index*, *trend* kešatmiņu) *pfree* (noklusējums) - brīvā bufera procentuālā daļa;
total - kopējais bufera izmērs;
free - brīvā bufera izmērs;
used - izmantotā bufera izmērs;
pused - izmantotā bufera procentuālā daļa.

Komentāri:

- Parametrs `trend` netiek atbalstīts Zabbix starpniekserverī.
- Vēstures kešatmiņa tiek izmantota vienumu vērtību glabāšanai. Zems skaits norāda uz veiktspējas problēmām datubāzes pusē.
- Vēstures indeksa kešatmiņa tiek izmantota vēstures kešatmiņā saglabāto vērtību indeksēšanai.
- Pēc tam, kad vēstures kešatmiņa ir aizpildīta un pēc tam iztīrīta, vēstures indeksa kešatmiņā joprojām paliks daļa datu. Šāda darbība ir paredzēta un palīdz sistēmai darboties efektīvāk, izvairoties no papildu apstrādes, kas nepieciešama, lai pastāvīgi mainītu atmiņas izmēru.
- Tendenču kešatmiņa glabā pašreizējās stundas apkopotos datus visiem vienumiem, kas saņem datus.
- Lai iegūtu vērtību sekundē statistiku, varat izmantot atslēgu zabbix[wcache,values] kopā ar *Change per second* priekšapstrādes soli.

9 SSH pārbaudes

Pārskats

SSH pārbaudes tiek veiktas kā uzraudzība bez aģenta. SSH pārbaudēm Zabbix aģents nav nepieciešams.

Lai veiktu SSH pārbaudes, Zabbix serverim sākotnēji jābūt konfigurētam ar SSH2 atbalstu (libssh vai libssh2). Skatiet arī: Prasības.

> **Attention:**
> Sākot ar RHEL 8, tiek atbalstīts tikai libssh. Citām distribūcijām libssh ir ieteicams libssh2 vietā.

Konfigurācija

Autentifikācija ar ieejas frāzi

SSH pārbaudes nodrošina divas autentifikācijas metodes — lietotājvārda/paroles pāri un autentifikāciju, kas balstīta uz atslēgas failu.

Ja neplānojat izmantot atslēgas, papildu konfigurācija nav nepieciešama, izņemot libssh vai libssh2 sasaisti ar Zabbix, ja veidojat no pirmkoda.

Autentifikācija ar atslēgas failu

Lai SSH vienumiem izmantotu autentifikāciju, kas balstīta uz atslēgām, ir nepieciešamas noteiktas izmaiņas servera konfigurācijā.

Atveriet Zabbix servera konfigurācijas failu (*zabbix_server.conf*) kā `root` un atrodiet šādu rindu:

```
##### SSHKeyLocation=
```

Noņemiet komentāru un norādiet pilnu ceļu uz mapi, kurā atradīsies publiskās un privātās atslēgas:

```
SSHKeyLocation=/home/zabbix/.ssh
```

Saglabājiet failu un pēc tam pārstartējiet Zabbix serveri.

Ceļš */home/zabbix* šeit ir *zabbix* lietotāja konta mājas direktorijs, bet *.ssh* ir direktorijs, kurā pēc noklusējuma publiskās un privātās atslēgas tiks ģenerētas ar komandu ssh-keygen mājas direktorijā.

Parasti Zabbix servera instalācijas pakotnes no dažādām OS distribūcijām izveido *zabbix* lietotāja kontu ar mājas direktoriju citā vietā, piemēram, */var/lib/zabbix* (kā sistēmas kontiem).

Pirms atslēgu ģenerēšanas varat pārvietot mājas direktoriju uz */home/zabbix*, lai tas atbilstu iepriekš minētajam Zabbix servera konfigurācijas parametram `SSHKeyLocation`.

Note:
Tālāk norādītās darbības var izlaist, ja *zabbix* konts ir pievienots manuāli saskaņā ar instalēšanas sadaļu. Šādā gadījumā *zabbix* konta mājas direktorijs, visticamāk, jau ir */home/zabbix*.

Lai mainītu *zabbix* lietotāja konta mājas direktoriju, ir jāaptur visi darbojošies procesi, kas to izmanto:

```
systemctl stop zabbix-agent
systemctl stop zabbix-server
```

Lai mainītu mājas direktorija atrašanās vietu, mēģinot to arī pārvietot (ja tas eksistē), jāizpilda šāda komanda:

```
usermod -m -d /home/zabbix zabbix
```

Iespējams arī, ka mājas direktorijs vecajā vietā nav eksistējis, tāpēc tas jāizveido jaunajā vietā. Drošs veids, kā to izdarīt, ir:

```
test -d /home/zabbix || mkdir /home/zabbix
```

Lai pārliecinātos, ka viss ir droši, var izpildīt papildu komandas, lai iestatītu mājas direktorija atļaujas:

```
chown zabbix:zabbix /home/zabbix
chmod 700 /home/zabbix
```

Iepriekš apturētos procesus tagad var atkal palaist:

```
systemctl start zabbix-agent
systemctl start zabbix-server
```

Tagad publiskās un privātās atslēgas var ģenerēt, izmantojot šādas komandas (labākai lasāmībai komandu uzvednes ir komentētas):

```
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):
<Atstājiet tukšu>
##### Enter same passphrase again:
<Atstājiet tukšu>
##### 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:
Publiskās un privātās atslēgas (*id_rsa.pub* un *id_rsa*) pēc noklusējuma ir ģenerētas direktorijā */home/zabbix/.ssh*, kas atbilst Zabbix servera konfigurācijas parametram `SSHKeyLocation`.

Attention:
Atslēgu tipi, kas nav "rsa", var tikt atbalstīti ar ssh-keygen rīku un SSH serveriem, taču tie var nebūt atbalstīti libssh2, ko izmanto Zabbix.

Shell konfigurācijas forma

Šī darbība jāveic tikai vienu reizi katram hostam, kas tiks uzraudzīts ar SSH pārbaudēm.

Izmantojot tālāk norādītās komandas, **publiskās** atslēgas failu var instalēt attālajā hostā *10.10.10.10*, lai SSH pārbaudes varētu veikt, izmantojot *root* kontu (labākai lasāmībai komandu uzvednes ir komentētas):

```
sudo -u zabbix ssh-copy-id root@10.10.10.10
##### Hosta '10.10.10.10 (10.10.10.10)' autentiskumu nevar noteikt.
##### RSA atslēgas pirkstu nospiedums ir 38:ba:f2:a4:b5:d9:8f:52:00:09:f7:1f:75:cc:0b:46.
##### Vai tiešām vēlaties turpināt savienošanos (yes/no)?
yes
##### Brīdinājums: '10.10.10.10' (RSA) ir neatgriezeniski pievienots zināmo hostu sarakstam.
##### root@10.10.10.10 parole:
<Ievadiet root paroli>
##### Tagad mēģiniet pieteikties datorā ar "ssh 'root@10.10.10.10'",
##### un pārbaudiet, vai tika pievienota tikai tā(s) atslēga(s), kuru(s) vēlējāties.
```

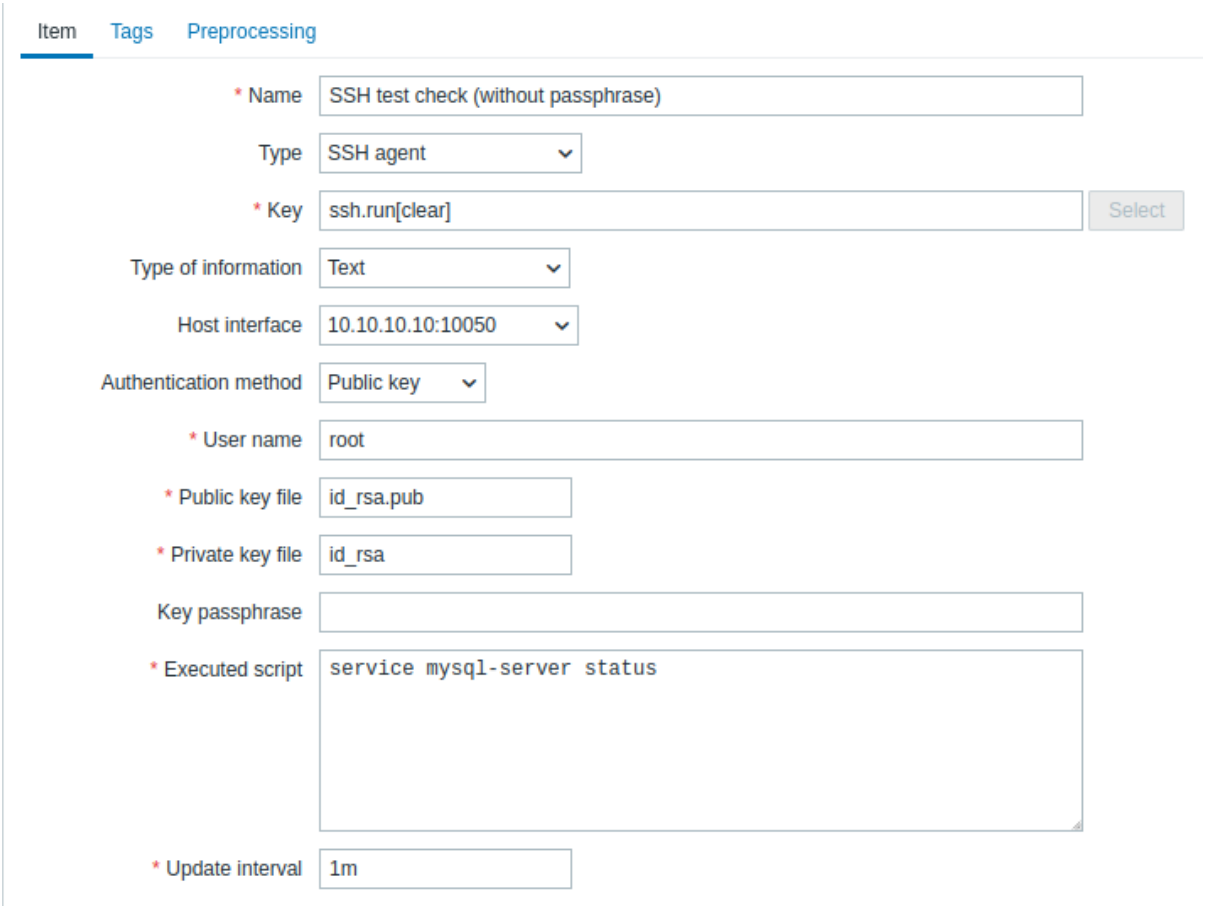
Tagad ir iespējams pārbaudīt SSH pieteikšanos, izmantojot noklusējuma privāto atslēgu (*/home/zabbix/.ssh/id_rsa*) *zabbix* lietotāja kontam:

```
sudo -u zabbix ssh root@10.10.10.10
```

Ja pieteikšanās ir veiksmīga, tad konfigurācijas daļa shell vidē ir pabeigta un attālo SSH sesiju var aizvērt.

Vienuma konfigurācija

Faktiskā(-ās) izpildāmā(-ās) komanda(-as) jānorāda vienuma konfigurācijas laukā *Executed script*. Vairākas komandas var izpildīt vienu pēc otras, ievietojot katru jaunā rindā. Šajā gadījumā arī atgrieztās vērtības tiks formatētas vairākās rindās.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem SSH vienumiem nepieciešama specifiska informācija, ir šādi:

Parameter	Description	Comments
Type	Šeit atlasiet **SSH aģents**.	

Parameter	Description	Comments

Key	Unikāla (katram hostam) vienuma atslēga formātā **ssh.run[unikāls īss apraksts,<ip>,<port>,<encoding>,<ssh options>,<subsystem>]**	**unikāls īss apraksts** ir obligāts, un tam jābūt unikālam katram SSH vienumam katrā hostā.
		
		Noklusējuma ports ir 22, nevis ports, kas norādīts saskarnē, kurai šis vienums ir piesaistīts.
		
		ssh options ļauj nodot papildu SSH opcijas formātā *key1=value1;key2=value2,value3*. Vairākas vērtības vienai atslēgai var nodot, atdalot tās ar komatu (šajā gadījumā parametrs ir jāievieto pēdiņās); vairākas opciju atslēgas var nodot, atdalot tās ar semikolu.
		
		Tiek atbalstītas šādas opciju atslēgas: `KexAlgorithms`, `HostkeyAlgorithms`, `Ciphers`, `MACs`, `PubkeyAcceptedKeyTypes`. Opciju atslēgu un vērtību atbalsts ir atkarīgs no SSH bibliotēkas (piemēram, `PubkeyAcceptedKeyTypes` tiek atbalstīts tikai ar libssh); ja opcija netiek atbalstīta, tiks atgriezta kļūda, un vienums kļūs neatbalstīts.
		
		Ņemiet vērā, ka zīme ”+” šifrēšanas iestatījumu pievienošanai un ”!” konkrētu šifrēšanas iestatījumu atspējošanai (kā GnuTLS un OpenSSL) netiek atbalstītas.
		
		Piemēri:
		=>
		`ssh.run[KexAlgorithms,127.0.0.1,,,Ciphers=aes128`
		=>
		`ssh.run[KexAlgorithms,,,,"KexAlgorithms=diffie-h`
		=>
		`ssh.run[PubkeyAcceptedKeyTypes,127.0.0.1,,,Pubke`
		
		subsystem ļauj nodot SSH apakšsistēmu, ierobežojot SSH savienojumu līdz konkrētām darbībām, ko atļauj apakšsistēma (piemēram, failu pārsūtīšana, izmantojot SFTP, vai tīkla ierīču pārvaldība, izmantojot NETCONF). Ņemiet vērā, ka apakšsistēmas izmantošanai var būt nepieciešama arī specifiska skripta sintakse parametrā *Executed script*.
		
		Piemēri:
		=>
		`ssh.run[SFTPBackup,192.0.2.18,,,,sftp]`
		=>
		`ssh.run[Cisco1234,192.0.2.18,,,,netconf]`

Authentication method	Viena no vērtībām: ”Password” vai ”Public key”.	
User name	Lietotājvārds (līdz 255 rakstzīmēm) autentifikācijai attālajā hostā. Obligāts.	
Public key file	Publiskās atslēgas faila nosaukums, ja *Authentication method* ir ”Public key”. Obligāts.	Piemērs: *id_rsa.pub* — noklusējuma publiskās atslēgas faila nosaukums, ko ģenerē komanda ssh-keygen.
Private key file	Privātās atslēgas faila nosaukums, ja *Authentication method* ir ”Public key”. Obligāts.	Piemērs: *id_rsa* — noklusējuma privātās atslēgas faila nosaukums.
Password or *Key passphrase*	Parole (līdz 255 rakstzīmēm) autentifikācijai vai Paroles frāze, **ja** tā tika izmantota privātajai atslēgai.	Atstājiet lauku *Key passphrase* tukšu, ja paroles frāze netika izmantota. Skatiet arī zināmās problēmas saistībā ar paroles frāzes izmantošanu.

Parameter	Description	Comments

Executed script	Izpildāmā(-ās) čaulas komanda(-as), izmantojot SSH attālo sesiju.	Izpildītās(-o) čaulas komandas(-u) atgrieztā vērtība ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas); attiecas arī datu bāzes ierobežojumi.
		
		Ņemiet vērā, ka libssh2 bibliotēka var saīsināt izpildāmos skriptus līdz ~32kB.
		
		Piemēri:
		`date +%s`
		`systemctl status mysql-server`
		`ps auxww | grep httpd | wc -l`
		
		Piemērs (NETCONF apakšsistēmai):
		`<rpc>`
		`<get-software-information/>`
		`</rpc>`
		`]]>]]>`
		`<rpc>`
		`<close-session/>`
		`</rpc>`
		`]]>]]>`

10 Telnet pārbaudes

Pārskats

Telnet pārbaudes tiek veiktas kā uzraudzība bez aģenta. Zabbix aģents nav nepieciešams Telnet pārbaudēm.

Konfigurējamie lauki

Faktiskā(-ās) izpildāmā(-ās) komanda(-as) jānorāda vienuma konfigurācijas laukā **Executed script**.
 Vairākas komandas var izpildīt vienu pēc otras, ievietojot katru jaunā rindā. Šajā gadījumā arī atgrieztā vērtība tiks formatēta kā vairākrindu.

Atbalstītās rakstzīmes, ar kurām var beigties čaulas uzvedne:

- $
- #
- >
- %

Note:
Telnet uzvednes rinda, kas beidzas ar vienu no šīm rakstzīmēm, tiks noņemta no atgrieztās vērtības, bet tikai pirmajai komandai komandu sarakstā, t.i., tikai telnet sesijas sākumā.

Atslēga	Apraksts
telnet.run[<unikāls īss aprakssts>,<ip>,<ports>,<kodējums>]	Izpildīt komandu attālinātā ierīcē, izmantojot telnet savienojumu

Attention:
Ja telnet pārbaude atgriež vērtību ar rakstzīmēm, kas nav ASCII, un kodējumā, kas nav UTF8, tad atslēgas parametrs *<encoding>* ir jānorāda pareizi. Plašāku informāciju skatiet lapā atgriezto vērtību kodējums.

11 Ārējās pārbaudes

Pārskats

Ārējā pārbaude ir pārbaude, ko izpilda Zabbix serveris, palaižot čaulas skriptu vai bināro failu. Tomēr, ja hosti tiek uzraudzīti ar Zabbix starpniekserveri, ārējās pārbaudes izpilda starpniekserveris.

Ārējām pārbaudēm nav nepieciešams, lai uzraugāmajā hostā darbotos kāds aģents.

Vienuma atslēgas sintakse ir šāda:

```
script[<parameter1>,<parameter2>,...]
```

Kur:

ARGUMENTS	DEFINĪCIJA
script	Čaulas skripta vai binārā faila nosaukums.
parameter(s)	Neobligāti komandrindas parametri.

Ja nevēlaties skriptam nodot nekādus parametrus, varat izmantot:

```
script[] or
script
```

Zabbix serveris vai starpniekserveris meklēs ārējiem skriptiem norādīto direktoriju un izpildīs komandu (skatiet `ExternalScripts` parametru Zabbix serveris/starpniekserveris konfigurācijas failā). Komanda tiks izpildīta ar to pašu lietotāju, ar kuru darbojas Zabbix serveris/starpniekserveris, tāpēc visas piekļuves atļaujas vai vides mainīgie, ja nepieciešams, jāapstrādā ietverošajā skriptā. Arī komandas atļaujām jāļauj šim lietotājam to izpildīt. Izpildei ir pieejamas tikai komandas norādītajā direktorijā.

Warning:
Nelietojiet ārējās pārbaudes pārmērīgi, jo katram skriptam Zabbix serverim/starpniekserverim ir jāpalaiž atsevišķs fork process, un daudzu skriptu izpilde var būtiski samazināt Zabbix veiktspēju.

Lietošanas piemērs

Izpildot skriptu **check_oracle.sh** ar pirmo parametru '-h'. Otrais parametrs tiks aizstāts ar IP adresi vai DNS nosaukumu atkarībā no izvēles hosta īpašībās.

```
check_oracle.sh["-h","{HOST.CONN}"]
```

Pieņemot, ka hosts ir konfigurēts izmantot IP adresi, Zabbix serveris/starpniekserveris izpildīs:

```
check_oracle.sh '-h' '192.168.1.4'
```

Ārējās pārbaudes rezultāts

Ārējās pārbaudes atgrieztā vērtība ir standarta izvade kopā ar standarta kļūdu, ko ģenerē pārbaude.

Attention:
Vienums, kas atgriež tekstu (rakstzīmju, žurnāla vai teksta tipa informāciju), nekļūs neatbalstīts standarta kļūdas izvades gadījumā.

Atgrieztās vērtības lielums ir ierobežots līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas); ir spēkā arī datu bāzes ierobežojumi.

Ja pieprasītais skripts netiek atrasts vai Zabbix serverim/starpniekserverim nav atļauju to izpildīt, vienums kļūs neatbalstīts un tiks parādīts atbilstošs kļūdas ziņojums.

Noildzes gadījumā vienums kļūs neatbalstīts, tiks parādīts atbilstošs kļūdas ziņojums, un skriptam izveidotais process tiks pārtraukts.

12 Trapper vienumi

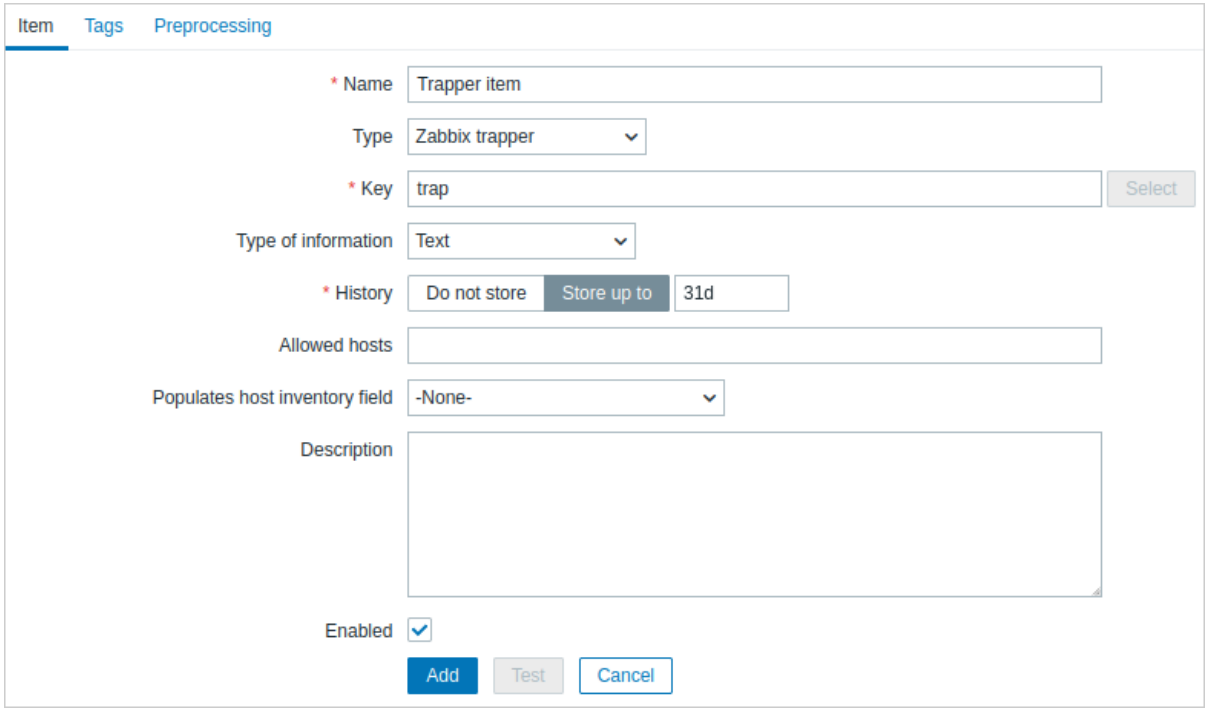
Pārskats

Trapper vienumi pieņem ienākošos datus, nevis tos pieprasa. Tas ir noderīgi jebkuriem datiem, kurus vēlaties nosūtīt uz Zabbix.

Konfigurācija

Lai konfigurētu trapper vienumu:

1. Dodieties uz *Datu ievākšana → Hosti*.
2. Noklikšķiniet uz *Vienumi* attiecīgā hosta rindā.
3. Noklikšķiniet uz *Izveidot vienumu*.
4. Ievadiet vienuma parametrus konfigurācijas formā.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem trapper vienumiem nepieciešama specifiska informācija, ir:

Tips	Atlasiet ”Zabbix trapper”.
Atslēga	Ievadiet atslēgu, kas tiks izmantota, lai atpazītu vienumu, nosūtot datus uz Zabbix serveri.
Informācijas tips	Atlasiet informācijas tipu, kas atbildīs nosūtāmo datu formātam.
Atļautie hosti	Ar komatiem atdalīts IP adrešu saraksts (pēc izvēles CIDR pierakstā) vai DNS nosaukumi. Ja norādīts, ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad ’127.0.0.1’, ’::127.0.0.1’, ’::ffff:127.0.0.1’ tiek uzskatītas par vienādām, un ’::/0’ atļaus jebkuru IPv4 vai IPv6 adresi. ’0.0.0.0/0’ var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka ”IPv4-compatible IPv6 addresses” (prefikss 0000::/96) tiek atbalstītas, taču ir novecojušas saskaņā ar RFC4291. Piemērs: 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} Atstarpes, lietotāja makrosi un hosta makrosi {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN} tiek atbalstīti.

Note:
Pirms vērtību nosūtīšanas, iespējams, būs jāuzgaida līdz 60 sekundēm pēc vienuma saglabāšanas, līdz Zabbix serveris uztvers izmaiņas no konfigurācijas kešatmiņas atjauninājuma.

Datu nosūtīšana

Datu nosūtīšana uz Zabbix serveri vai starpniekserveri ir iespējama, izmantojot utilītu Zabbix sender vai Zabbix sender protokolu.
Datu nosūtīšana uz Zabbix serveri ir iespējama arī, izmantojot API metodi `history.push`.

Zabbix sender

Lai nosūtītu datus uz Zabbix serveris vai starpniekserveris, izmantojot Zabbix sender utilītu, varat izpildīt šādu komandu, lai nosūtītu ”test value”:

```
zabbix_sender -z <server IP address> -p 10051 -s "New host" -k trap -o "test value"
```

Lai nosūtītu ”test value”, tiek izmantotas šādas komandas opcijas:

- -z lai norādītu Zabbix serveris IP adresi

- `-p` lai norādītu Zabbix serveris porta numuru (pēc noklusējuma 10051)
- `-s` lai norādītu hosts (pārliecinieties, ka izmantojat tehnisko, nevis redzamo hosta nosaukumu)
- `-k` lai norādītu trigeris vienums atslēgu, kas konfigurēta trapper vienumā
- `-o` lai norādītu nosūtāmo vērtību

Attention:
Zabbix trapper process neizvērš makrosus, kas izmantoti vienuma atslēgā, lai pārbaudītu atbilstošās vienuma atslēgas esamību mērķa hostam.

Plašāku informāciju par saziņu starp Zabbix sender un Zabbix serveris vai starpniekserveris skatiet *Zabbix sender protocol*.

history.push

Lai nosūtītu datus uz Zabbix serveris, izmantojot `history.push` API metodi, varat veikt šādu HTTP POST pieprasījumu, kas satur dažas testa vērtības:

```
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
```

Ja pieprasījums ir pareizs, API atgrieztā atbilde varētu izskatīties šādi:

```
{
    "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
}
```

Kļūdas atbildes datos norāda, ka datu nosūtīšana konkrētiem vienumiem nav izturējusi Zabbix serveris validāciju. Tas var notikt šādu iemeslu dēļ:

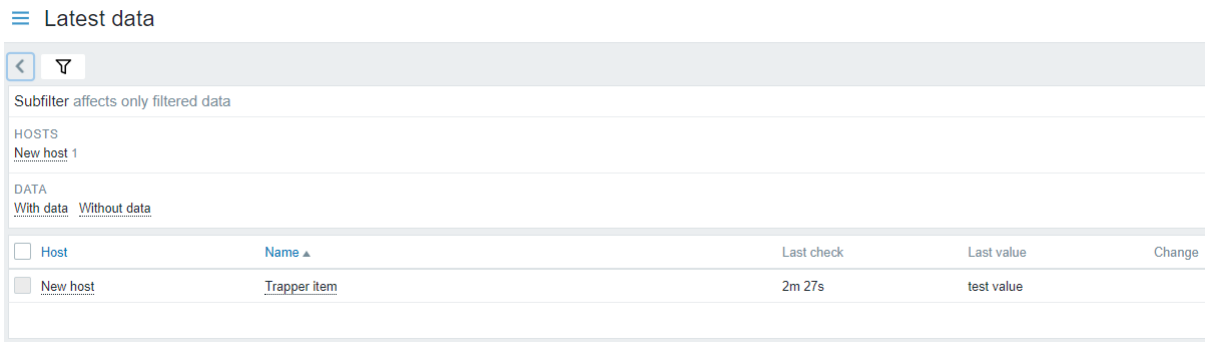
- lietotājam, kurš nosūta datus, nav *lasīšanas* atļaujas uz vienuma hostu;
- hosts ir atspējots vai atrodas uzturēšanas režīmā bez datu vākšanas;
- vienums neeksistē vai vēl nav iekļauts servera konfigurācijas kešatmiņā;
- vienums ir atspējots vai tā tips nav Zabbix trapper vai HTTP aģents (ar iespējotu trapping);
- lietotāja IP vai DNS nav iestatīts vienuma *Allowed hosts* sarakstā;
- citam vienumam ir vērtība ar dublējošu laikspiedolu nanosekunžu līmenī.

Kļūdu neesamība norāda, ka nosūtītās vērtības ir pieņemtas apstrādei, kas ietver priekšapstrādi (ja tāda ir), trigeris apstrādi un saglabāšanu datubāzē. Ņemiet vērā, ka arī pieņemtās vērtības apstrāde var neizdoties (piemēram, priekšapstrādes laikā), kā rezultātā vērtība tiek atmesta.

Plašāku informāciju par darbu ar Zabbix API skatiet sadaļā API.

Datu attēlošana

Kad dati ir nosūtīti, varat doties uz *Monitoring* → *Jaunākie dati*, lai skatītu rezultātu:



Note:
Ja tiek nosūtīta viena skaitliska vērtība, datu grafikā tiks parādīta horizontāla līnija pa kreisi un pa labi no vērtības laika punkta.

13 JMX uzraudzība

Pārskats

JMX uzraudzību var izmantot Java lietojumprogrammas JMX skaitītāju uzraudzībai.

JMX uzraudzībai ir vietējs atbalsts Zabbix sistēmā Zabbix dēmona veidā ar nosaukumu "Zabbix Java gateway".

Lai iegūtu konkrēta JMX skaitītāja vērtību hostā, Zabbix serveris vaicā Zabbix **Java gateway**, kas savukārt izmanto JMX pārvaldības API, lai attālināti vaicātu interesējošo lietojumprogrammu.

Plašāku informāciju un iestatīšanas norādes skatiet sadaļā Zabbix Java gateway.

Warning:
Saziņu starp Java gateway un uzraugāmo JMX lietojumprogrammu nedrīkst bloķēt ugunsmūris.

Attālās JMX uzraudzības iespējošana Java lietojumprogrammai

Java lietojumprogrammai nav nepieciešama papildu programmatūras instalēšana, taču tā ir jāpalaiž ar tālāk norādītajām komandrindas opcijām, lai nodrošinātu attālās JMX uzraudzības atbalstu.

Kā minimālā prasība, ja vēlaties vienkārši sākt ar vienkāršas Java lietojumprogrammas uzraudzību lokālajā hostā bez ieviestas drošības, palaidiet to ar šādām opcijām:

```
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
```

Tas liek Java klausīties ienākošos JMX savienojumus 12345. portā tikai no lokālā hosta un norāda, ka nav nepieciešama autentifikācija vai SSL.

Ja vēlaties atļaut savienojumus citā saskarnē, iestatiet -Djava.rmi.server.hostname parametru uz šīs saskarnes IP adresi.

Ja vēlaties ievērot stingrākas drošības prasības, jums ir pieejamas daudzas citas Java opcijas. Piemēram, nākamais piemērs palaiž lietojumprogrammu ar daudzpusīgāku opciju kopu un atver to plašākam tīklam, ne tikai lokālajam hostam.

```
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
```

Lielāko daļu (ja ne visus) no šiem iestatījumiem var norādīt `$JRE/lib/management/management.properties` (vai vietā, kur šis fails atrodas jūsu sistēmā).

Ņemiet vērā, ka, ja vēlaties izmantot SSL, jums ir jāmaina startup.sh skripts, pievienojot `-Djavax.net.ssl.*` opcijas Java gateway, lai tas zinātu, kur atrast atslēgu un uzticamības krātuves.

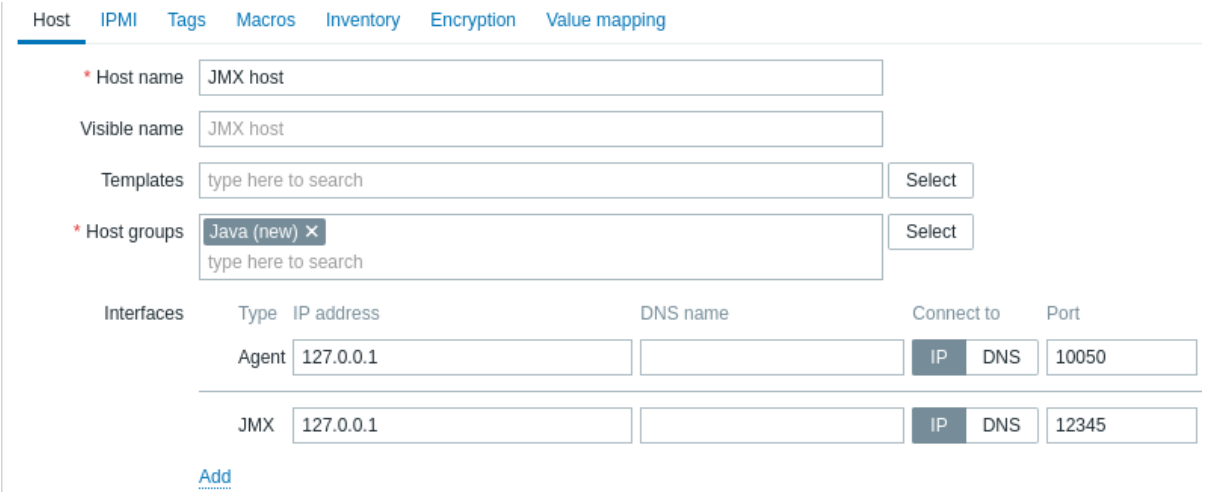
Detalizētu aprakstu skatiet Monitoring and Management Using JMX.

JMX saskarņu un vienumu konfigurēšana Zabbix lietotāja saskarnē

Kad Java gateway darbojas, serveris zina, kur to atrast, un Java lietojumprogramma ir palaista ar attālinātās JMX uzraudzības atbalstu, ir laiks konfigurēt saskarnes un vienumus Zabbix lietotāja saskarnē.

JMX saskarnes konfigurēšana

Jūs sākat, izveidojot JMX tipa saskarni interesējošajā hostā.

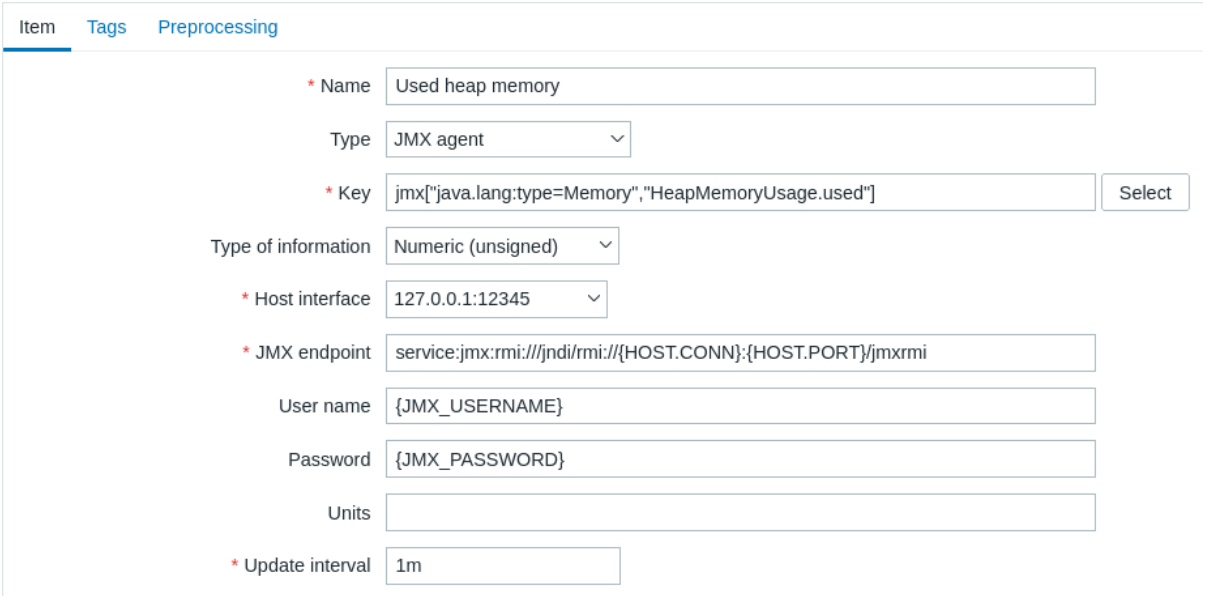


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

JMX aģenta vienuma pievienošana

Katram JMX skaitītājam, kas jūs interesē, pievienojiet **JMX aģenta** vienumu, kas piesaistīts šai saskarnei.

Atslēga zemāk redzamajā ekrānuzņēmumā ir `jmx["java.lang:type=Memory","HeapMemoryUsage.used"]`.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem JMX vienumiem nepieciešama specifiska informācija, ir šādi:

Type	Šeit iestatiet **JMX aģents**.
Key	`jmx[]` vienuma atslēga satur trīs parametrus: **objekta nosaukums** — MBean objekta nosaukums **atribūta nosaukums** — MBean atribūta nosaukums ar neobligātiem salikto datu lauku nosaukumiem, kas atdalīti ar punktiem **unikāls īss apraksts** — unikāls apraksts, kas ļauj hostā izmantot vairākus JMX vienumus ar vienādu objekta nosaukumu un atribūta nosaukumu (neobligāti) Sīkāku informāciju par JMX vienumu atslēgām skatiet zemāk. Jūs varat atklāt MBean un MBean atribūtus, izmantojot `jmx.discovery[]` zema līmeņa atklāšanas vienumu.
JMX endpoint	Varat norādīt pielāgotu JMX galapunktu. Pārliecinieties, ka JMX galapunkta savienojuma parametri atbilst JMX saskarnei. To var panākt, izmantojot {HOST.*} makrosus, kā tas darīts noklusējuma JMX galapunktā. Tiek atbalstīti {HOST.*} makrosi un lietotāja makrosi.
User name	Norādiet lietotājvārdu (līdz 255 rakstzīmēm), ja jūsu Java lietojumprogrammai ir konfigurēta autentifikācija. Tiek atbalstīti lietotāja makrosi.
Password	Norādiet paroli (līdz 255 rakstzīmēm), ja jūsu Java lietojumprogrammai ir konfigurēta autentifikācija. Tiek atbalstīti lietotāja makrosi.

Ja vēlaties uzraudzīt Būla skaitītāju, kura vērtība ir vai nu ”true”, vai ”false”, tad norādiet informācijas tipu kā ”Numeric (unsigned)” un cilnē Preprocessing atlasiet priekšapstrādes soli ”Boolean to decimal”. Serveris saglabās Būla vērtības attiecīgi kā 1 vai 0.

JMX vienumu atslēgas sīkāk

Vienkārši atribūti

MBean objekta nosaukums ir tikai virkne, ko definējat savā Java lietotnē. Savukārt atribūta nosaukums var būt sarežģītāks. Ja atribūts atgriež primitīvu datu tipu (veselu skaitli, virkni utt.), nav par ko uztraukties — atslēga izskatīsies šādi:

`jmx[com.example:Type=Hello,weight]`

Šajā piemērā objekta nosaukums ir ”com.example:Type=Hello”, atribūta nosaukums ir ”weight”, un atgrieztās vērtības tipam, visticamāk, jābūt ”Numeric (float)”.

Atribūti, kas atgriež saliktus datus

Tas kļūst sarežģītāk, ja jūsu atribūts atgriež saliktus datus. Piemēram: jūsu atribūta nosaukums ir ”apple”, un tas atgriež jaucējtabulu, kas attēlo tā parametrus, piemēram, ”weight”, ”color” utt. Jūsu atslēga var izskatīties šādi:

`jmx[com.example:Type=Hello,apple.weight]`

Šādi atribūta nosaukums un jaucējtabulas atslēga tiek atdalīti, izmantojot punkta simbolu. Tādā pašā veidā, ja atribūts atgriež ligzdotus saliktus datus, daļas tiek atdalītas ar punktu:

`jmx[com.example:Type=Hello,fruits.apple.weight]`

Atribūti, kas atgriež tabulārus datus

Tabulāru datu atribūti sastāv no viena vai vairākiem saliktiem atribūtiem. Ja šāds atribūts ir norādīts atribūta nosaukuma parametrā, tad šī vienuma vērtība atgriezīs pilnu atribūta struktūru JSON formātā. Atsevišķu elementu vērtības tabulāro datu atribūtā var iegūt, izmantojot priekšapstrādi.

Tabulāru datu atribūta piemērs:

`jmx[com.example:type=Hello,foodinfo]`

Vienuma vērtība:

```
[
  {
    "a": "apple",
    "b": "banana",
    "c": "cherry"
  },
  {
    "a": "potato",
    "b": "lettuce",
```

```
    "c": "onion"
  }
]
```

Problēma ar punktiem

Līdz šim viss ir labi. Bet ko darīt, ja atribūta nosaukums vai jaucējtabulas atslēga satur punkta simbolu? Šeit ir piemērs:

```
jmx[com.example:Type=Hello,all.fruits.apple.weight]
```

Tā ir problēma. Kā pateikt Zabbix, ka atribūta nosaukums ir "all.fruits", nevis tikai "all"? Kā atšķirt punktu, kas ir daļa no nosaukuma, no punkta, kas atdala atribūta nosaukumu un jaucējtabulas atslēgas?

Tas ir iespējams — viss, kas jums jādara, ir jāekranē punkti, kas ir daļa no nosaukuma, ar atpakaļslīpsvītru:

```
jmx[com.example:Type=Hello,all\.fruits.apple.weight]
```

Tādā pašā veidā, ja jūsu jaucējtabulas atslēga satur punktu, to ir jāekranē:

```
jmx[com.example:Type=Hello,all\.fruits.apple.total\.weight]
```

Citas problēmas

Atpakaļslīpsvītras rakstzīme atribūta nosaukumā ir jāekranē:

```
jmx[com.example:type=Hello,c:\\documents]
```

Lai apstrādātu jebkādas citas speciālās rakstzīmes JMX vienuma atslēgā, lūdzu, skatiet vienuma atslēgas formāta sadaļu.

Patiesībā tas ir viss. Veiksmīgu JMX uzraudzību!

Neprimitīvie datu tipi

Ir iespējams strādāt ar pielāgotiem MBeans, kas atgriež neprimitīvos datu tipus, kuri pārraksta **toString()** metodi.

Pielāgota galapunkta izmantošana ar JBoss EAP 6.4

Pielāgoti galapunkti ļauj strādāt ar dažādiem transporta protokoliem, izņemot noklusējuma RMI.

Lai ilustrētu šo iespēju, mēģināsim kā piemēru konfigurēt JBoss EAP 6.4 uzraudzību. Vispirms pieņemsim dažus nosacījumus:

- Jums jau ir instalēts Zabbix Java gateway. Ja nē, tad to varat izdarīt saskaņā ar dokumentāciju.
- Zabbix serveris un Java gateway ir instalēti ar prefiksu `/usr/local/`
- JBoss jau ir instalēts mapē `/opt/jboss-eap-6.4/` un darbojas standalone režīmā
- Pieņemsim, ka visi šie komponenti darbojas uz viena un tā paša hosta
- Ugunsmūris un SELinux ir atspējoti (vai attiecīgi konfigurēti)

Veiksim dažus vienkāršus iestatījumus failā zabbix_server.conf:

```
JavaGateway=127.0.0.1
StartJavaPollers=5
```

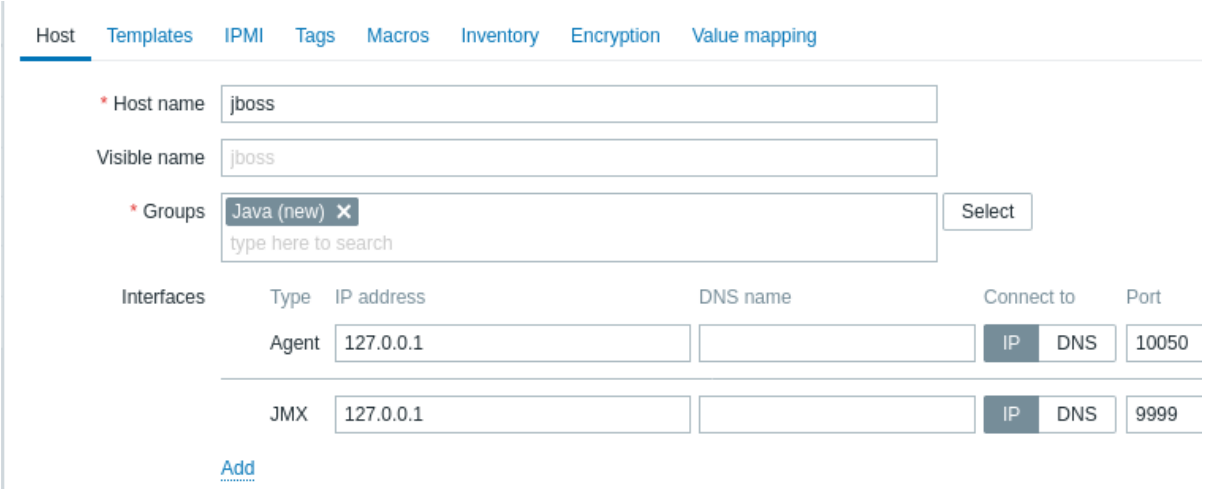
Un konfigurācijas failā `zabbix_java/settings.sh` (vai `zabbix_java_gateway.conf`):

```
START_POLLERS=5
```

Pārbaudiet, vai JBoss klausās savā standarta pārvaldības portā:

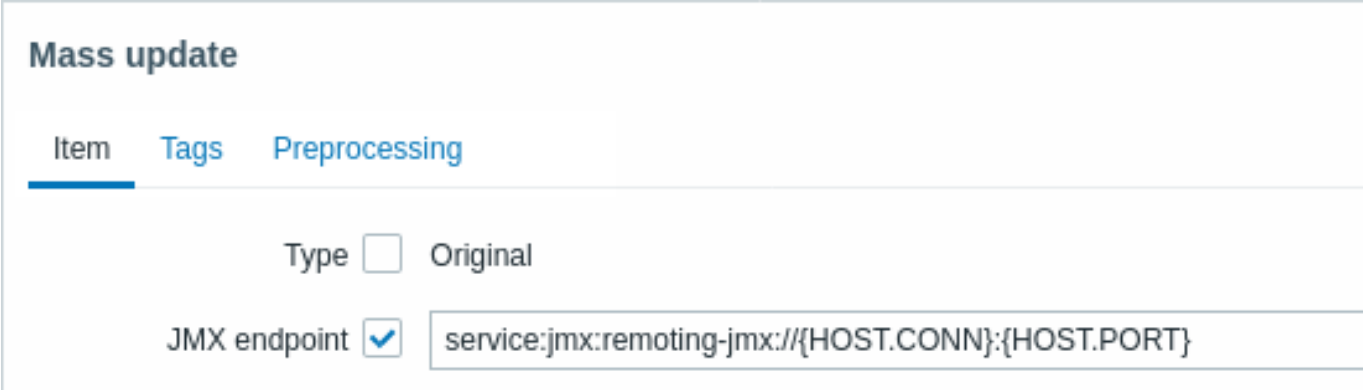
```
$ netstat -natp | grep 9999
tcp        0      0 127.0.0.1:9999          0.0.0.0:*               LISTEN      10148/java
```

Tagad izveidosim hostu ar JMX saskarni 127.0.0.1:9999 Zabbix.



Tā kā zinām, ka šī JBoss versija izmanto JBoss Remoting protokolu RMI vietā, mēs varam attiecīgi masveidā atjaunināt JMX galapunkta parametru vienumiem mūsu JMX veidnē:

```
service:jmx:remoting-jmx://{HOST.CONN}:{HOST.PORT}
```



Atjaunināsim konfigurācijas kešatmiņu:

```
/usr/local/sbin/zabbix_server -R config_cache_reload
```

Ņemiet vērā, ka sākumā var rasties kļūda.

```
3. mc [root@centos7-dev]:/home/vagrant/zabbix-3.2.6/src/zabbix_java (ssh)
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:12.644 [pool-1-thread-1] WARN  com.zabbix.gateway.SocketProcessor - error processing request
com.zabbix.gateway.ZabbixException: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at com.zabbix.gateway.JMXItemChecker.getValues(JMXItemChecker.java:97) ~[zabbix-java-gateway-3.4.2.jar:na]
        at com.zabbix.gateway.SocketProcessor.run(SocketProcessor.java:63) ~[zabbix-java-gateway-3.4.2.jar:na]
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149) [na:1.8.0_144]
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624) [na:1.8.0_144]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_144]
Caused by: java.net.MalformedURLException: Unsupported protocol: remoting-jmx
        at javax.management.remote.JMXConnectorFactory.newJMXConnector(JMXConnectorFactory.java:359) ~[na:1.8.0_144]
        at javax.management.remote.JMXConnectorFactory.connect(JMXConnectorFactory.java:269) ~[na:1.8.0_144]
        at com.zabbix.gateway.ZabbixJMXConnectorFactory$1.run(ZabbixJMXConnectorFactory.java:76) ~[zabbix-java-gatewa
-3.4.2.jar:na]
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511) ~[na:1.8.0_144]
        at java.util.concurrent.FutureTask.run(FutureTask.java:266) ~[na:1.8.0_144]
        ... 3 common frames omitted
2017-11-07 13:52:14.889 [Thread-0] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885)
as stopped
2017-11-07 13:52:26.167 [main] INFO  com.zabbix.gateway.JavaGateway - Zabbix Java Gateway 3.4.2 (revision 72885) has
tarted
```

"Unsupported protocol: remoting-jmx" nozīmē, ka Java gateway nezina, kā strādāt ar norādīto protokolu. To var novērst, izveidojot failu `~/needed_modules.txt` ar šādu saturu:

```
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
```

un pēc tam izpildot komandu:

```
for i in $(cat ~/needed_modules.txt); do find /opt/jboss-eap-6.4 -iname "${i}*.jar" -exec cp '{}' /usr/loc
```

Tādējādi Java gateway būs visi nepieciešamie moduļi darbam ar jmx-remoting. Atliek tikai restartēt Java gateway, nedaudz pagaidīt un, ja visu izdarījāt pareizi, redzēt, ka JMX uzraudzības dati sāk ienākt Zabbix (skatiet arī: Jaunākie dati).

14 ODBC uzraudzība

Pārskats

ODBC uzraudzība atbilst vienuma tipam **Datubāzes monitors** Zabbix lietotāja saskarnē.

ODBC ir C programmēšanas valodas starpslāņa API piekļuvei datubāzu pārvaldības sistēmām (DBMS). ODBC koncepciju izstrādāja Microsoft, un vēlāk tā tika pārnesta uz citām platformām.

Zabbix var vaicāt jebkuru datubāzi, ko atbalsta ODBC. Lai to izdarītu, Zabbix neveido tiešu savienojumu ar datubāzēm, bet izmanto ODBC saskarni un ODBC konfigurētos draiverus. Tas ļauj efektīvāk uzraudzīt dažādas datubāzes dažādiem mērķiem (piemēram, pārbaudīt konkrētas datubāzes rindas, lietojuma statistiku u.c.).

Zabbix atbalsta unixODBC, kas ir viena no visbiežāk izmantotajām atvērtā pirmkoda ODBC API realizācijām.

> **Attention:**
> Skatiet arī: zināmās problēmas par ODBC pārbaudēm.

unixODBC instalēšana

Ieteicamais unixODBC instalēšanas veids ir izmantot Linux operētājsistēmas noklusējuma pakotņu repozitorijus. Populārākajās Linux distribūcijās unixODBC pēc noklusējuma ir iekļauts pakotņu repozitorijā. Ja pakotnes nav pieejamas, pirmkoda failus var iegūt unixODBC mājaslapā: http://www.unixodbc.org/download.html.

Lai instalētu unixODBC, izmantojiet izvēlētās sistēmas pakotņu pārvaldnieku:

```
##### Ubuntu/Debian sistēmām:
apt install unixodbc unixodbc-dev

##### RedHat/Fedora bāzētām sistēmām:
dnf install unixODBC unixODBC-devel

##### SUSE bāzētām sistēmām:
zypper in unixODBC-devel
```

> **Attention:**
> `unixodbc-dev` vai `unixODBC-devel` pakotne ir nepieciešama, lai kompilētu Zabbix ar unixODBC atbalstu. Lai iespējotu ODBC atbalstu, Zabbix jākompilē ar šādu konfigurācijas opciju:

> `--with-unixodbc[=ARG] # Use ODBC driver against unixODBC package.`

unixODBC draiveru instalēšana

unixODBC datubāzes draiveris ir jāinstalē datubāzei, kas tiks uzraudzīta. Atbalstīto datubāzu un draiveru sarakstu skatiet unixODBC mājaslapā: http://www.unixodbc.org/drivers.html.

> **Note:**
> Dažās Linux distribūcijās datubāzu draiveri ir iekļauti pakotņu repozitorijos.

MySQL

Lai instalētu MySQL unixODBC datubāzes draiveri, izmantojiet jūsu izvēlētās sistēmas pakotņu pārvaldnieku:

```
##### Ubuntu/Debian sistēmām:
apt install odbc-mariadb

##### RedHat/Fedora bāzētām sistēmām:
dnf install mariadb-connector-odbc

##### SUSE bāzētām sistēmām:
zypper install mariadb-connector-odbc
```

Lai instalētu datubāzes draiveri bez pakotņu pārvaldnieka, lūdzu, skatiet MySQL dokumentāciju par `mysql-connector-odbc` vai MariaDB dokumentāciju par `mariadb-connector-odbc`.

PostgreSQL

Lai instalētu PostgreSQL unixODBC datubāzes draiveri, izmantojiet jūsu izvēlētās sistēmas pakotņu pārvaldnieku:

```
##### Ubuntu/Debian sistēmām:
apt install odbc-postgresql

##### RedHat/Fedora bāzētām sistēmām:
dnf install postgresql-odbc

##### SUSE bāzētām sistēmām:
zypper install psqlODBC
```

Lai instalētu datubāzes draiveri bez pakotņu pārvaldnieka, lūdzu, skatiet PostgreSQL dokumentāciju.

Oracle

Lai instalētu unixODBC datubāzes draiveri, lūdzu, skatiet Oracle dokumentāciju.

MSSQL

Lai instalētu MSSQL unixODBC datubāzes draiveri, izmantojiet jūsu izvēlētās sistēmas pakotņu pārvaldnieku:

```
##### Ubuntu/Debian sistēmām:
apt install tdsodbc

##### RedHat/Fedora bāzētām sistēmām (EPEL pakotnes: https://docs.fedoraproject.org/en-US/epel/):
dnf install epel-release
dnf install freetds

##### SUSE bāzētām sistēmām:
zypper install libtdsodbc0
```

Lai instalētu datubāzes draiveri bez pakotņu pārvaldnieka, lūdzu, skatiet FreeTDS user guide.

unixODBC konfigurēšana

Lai konfigurētu unixODBC, jums jārediģē faili `odbcinst.ini` un `odbc.ini`. Šo failu atrašanās vietu varat pārbaudīt, izpildot šādu komandu:

```
odbcinst -j
```

Komandas rezultātā jābūt informācijai, kas ir līdzīga šādai:

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

odbcinst.ini

Fails `odbcinst.ini` uzskaita instalētos ODBC datubāzu draiverus. Ja `odbcinst.ini` trūkst, tas ir jāizveido manuāli.

```
[TEST_MYSQL]
Description=ODBC for MySQL
Driver=/usr/lib/libmyodbc5.so
FileUsage=1
```

Parameter	Description
TEST_MYSQL	Datubāzes draivera nosaukums.
Description	Datubāzes draivera apraksts.
Driver	Datubāzes draivera bibliotēkas atrašanās vieta.
FileUsage	Nosaka, vai datubāzes draiveris atbalsta savienošanos ar datubāzes serveri bez atbalsta piekļuvei lokālajiem failiem (0); atbalsta datu nolasīšanu no failiem (1); atbalsta datu rakstīšanu failos (2).
Threading	Pavedienu serializācijas līmenis. Atbalstīts PostgreSQL. Kopš 1.6, ja draiveru pārvaldnieks ir būvēts ar pavedienu atbalstu, varat pievienot vēl vienu draivera ierakstu.

odbc.ini

Fails `odbc.ini` tiek izmantots, lai konfigurētu datu avotus. Ņemiet vērā, ka atbalstīto parametru saraksts ir atkarīgs no datubāzes draivera (piemēram, Oracle datubāzēs Server vietā var tikt izmantots ServerName u.c.).

```
[TEST_MYSQL]
Description=MySQL Test Database
Driver=mysql
Server=127.0.0.1
User=root
Password=
Port=3306
Socket=
Database=zabbix
```

Parametrs	Apraksts
TEST_MYSQL	Datu avota nosaukums (DSN).
Description	Datu avota apraksts.
Driver	Datubāzes draivera nosaukums (kā norādīts `odbcinst.ini`).
Server	Datubāzes servera IP/DNS.
User	Datubāzes lietotājs savienojumam.
Password	Datubāzes lietotāja parole.
Port	Datubāzes savienojuma ports.
Socket	Datubāzes savienojuma ligzda.
Database	Datubāzes nosaukums.

Par citām iespējamām konfigurācijas parametru opcijām skatiet MySQL dokumentāciju.

PostgreSQL `odbc.ini` fails var saturēt papildu parametrus:

```
[TEST_PSQL]
Description=PostgreSQL Test Database
Driver=postgresql
Username=zbx_test
Password=zabbix
Servername=127.0.0.1
Database=zabbix
Port=5432
ReadOnly=No
Protocol=8.0+
ShowOidColumn=No
FakeOidIndex=No
RowVersioning=No
ShowSystemTables=No
Fetch=Yes
BoolsAsChar=Yes
```

```
SSLmode=Require
ConnSettings=
```

Parametrs	Apraksts
ReadOnly	Norāda, vai datubāzes savienojums atļauj tikai lasīšanas darbības (SELECT vaicājumus) un ierobežo izmaiņu veikšanu (INSERT, UPDATE un DELETE priekšrakstus); noderīgi scenārijos, kuros datiem jāpaliek nemainītiem.
Protocol	PostgreSQL aizmugursistēmas protokola versija (tiek ignorēta, izmantojot SSL savienojumus).
ShowOidColumn	Norāda, vai iekļaut Object ID (OID) SQLColumns rezultātos.
FakeOidIndex	Norāda, vai izveidot mākslīgu unikālu indeksu OID laukam.
RowVersioning	Norāda, vai iespējot lietotnēm noteikt, vai datus ir modificējuši citi lietotāji laikā, kamēr mēģināt atjaunināt rindu. Ņemiet vērā, ka šis parametrs var paātrināt atjaunināšanas procesu, jo, lai atjauninātu rindu, WHERE klauzulā nav jānorāda katra atsevišķā kolonna.
ShowSystemTables	Norāda, vai datubāzes draiverim sistēmas tabulas SQLTables rezultātos jāuztver kā parastas tabulas; noderīgi piekļūstamībai, nodrošinot sistēmas tabulu redzamību.
Fetch	Norāda, vai draiverim automātiski izmantot declare cursor/fetch SELECT priekšrakstu apstrādei un uzturēt 100 rindu kešatmiņu.
BoolsAsChar	Kontrolē Būla tipu kartēšanu. Ja iestatīts uz "Yes", Būla vērtības tiek kartētas uz SQL_CHAR; pretējā gadījumā tās tiek kartētas uz SQL_BIT.
SSLmode	Norāda savienojuma SSL režīmu.
ConnSettings	Papildu iestatījumi, kas savienojuma izveides laikā tiek nosūtīti aizmugursistēmai.

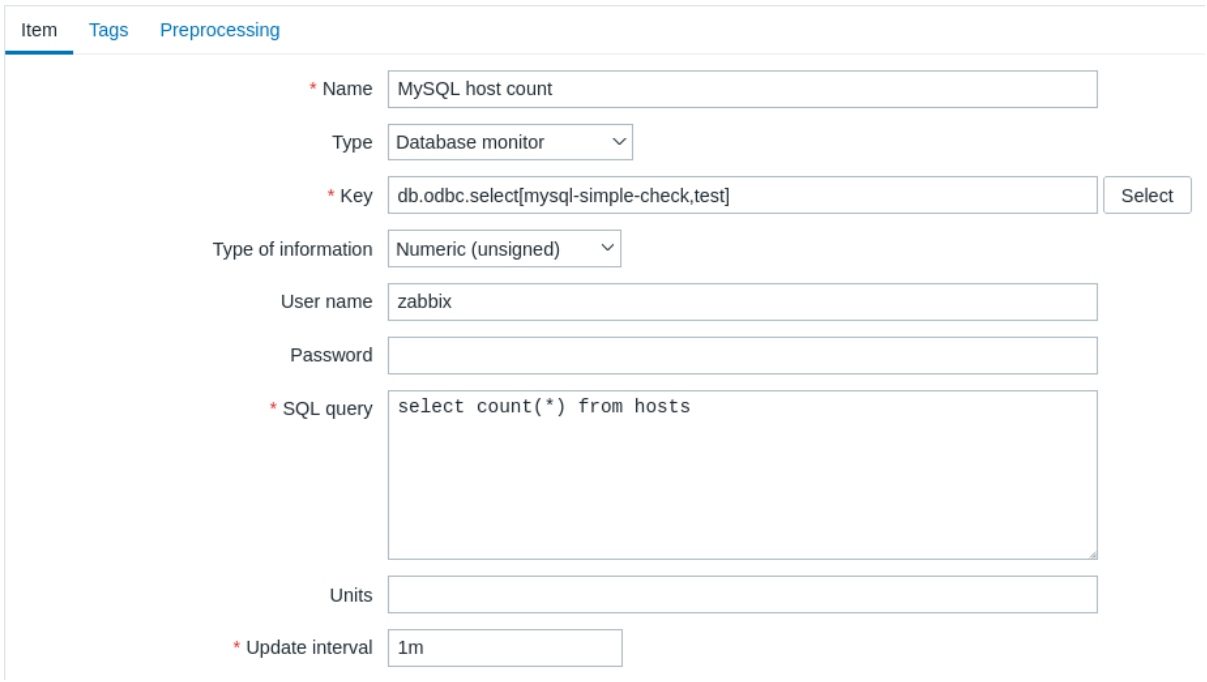
ODBC savienojuma testēšana

Lai pārbaudītu, vai ODBC savienojums darbojas veiksmīgi, varat izmantot `isql` utilītu (iekļauta `unixODBC` pakotnē):

```
isql test
+---------------------------------------+
| Connected!                            |
|                                       |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
|                                       |
+---------------------------------------+
```

Vienuma konfigurācija Zabbix lietotāja saskarnē

Konfigurējiet **datubāzes uzraudzības** vienumu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Datubāzes uzraudzības vienumiem ir jānorāda:

Type	Šeit atlasiet ”Database monitor”.
Key	Ievadiet vienu no atbalstītajām vienuma atslēgām: **db.odbc.select[]** - šis vienums atgriež vienu vērtību (SQL vaicājuma rezultāta pirmās rindas pirmo kolonnu); **db.odbc.get[]** - šis vienums atgriež vairākas rindas/kolonnas JSON formātā; **db.odbc.discovery[]** - šis vienums atgriež zema līmeņa atklāšanas datus.
User name	Ievadiet datubāzes lietotājvārdu (līdz 255 rakstzīmēm). Šis parametrs nav obligāts, ja datubāzes lietotājvārds ir norādīts `odbc.ini` failā. Ja tiek izmantota savienojuma virkne un lauks *User name* nav tukšs, tad tas tiek pievienots savienojuma virknei kā `UID=<user>`.
Password	Ievadiet datubāzes lietotāja paroli (līdz 255 rakstzīmēm). Šis parametrs nav obligāts, ja parole ir norādīta `odbc.ini` failā. Ja tiek izmantota savienojuma virkne un lauks *Password* nav tukšs, tad tā tiek pievienota savienojuma virknei kā `PWD=<password>`. Šajā laukā tiek atbalstītas speciālās rakstzīmes. Parole tiks pievienota savienojuma virknei aiz lietotājvārda, piemēram, `UID=<username>;PWD=P?;)*word`. Lai pārbaudītu iegūto virkni, varat izpildīt šādu komandu: `isql -v -k 'Driver=libmaodbc.so;Database=zabbix;UID=zabbix;PWD=P?;)*word'`
SQL query	Ievadiet SQL vaicājumu. Ņemiet vērā, ka ar `db.odbc.select[]` vaicājumam ir jāatgriež tikai viena vērtība.
Type of information	Šeit atlasiet informācijas tipu, ko atgriezīs vaicājums. Ja informācijas tips ir atlasīts nepareizi, vienums kļūs neatbalstīts.

Svarīgas piezīmes

- Datubāzes uzraudzības vienumi kļūs neatbalstīti, ja servera vai starpniekservera konfigurācijā netiks palaists neviens *odbc poller* process. Lai aktivizētu ODBC pollerus, iestatiet parametru `StartODBCPollers` Zabbix serveris konfigurācijas failā vai, ja pārbaudes veic starpniekserveris, Zabbix starpniekserveris konfigurācijas failā.
- Parametra *Timeout* vērtība vienuma konfigurācijas formā tiek izmantota kā ODBC pieteikšanās noildze un vaicājuma izpildes noildze. Ņemiet vērā, ka šie noildzes iestatījumi var tikt ignorēti, ja instalētais ODBC draiveris tos neatbalsta.
- SQL komandai jāatgriež rezultātu kopa, tāpat kā jebkuram vaicājumam, kurā izmantots priekšraksts `select`. Vaicājuma sintakse būs atkarīga no RDBMS, kas to apstrādās. Pieprasījuma sintaksei uz glabāto procedūru jāsākas ar atslēgvārdu `call`.

Vienuma atslēgas detaļas

Parametri bez leņķiekavām ir obligāti. Parametri, kas apzīmēti ar leņķiekavām **< >**, ir neobligāti.

db.odbc.select[<unikāls īss apraksts>,<dsn>,<savienojuma virkne>]

 Atgriež vienu vērtību, proti, SQL vaicājuma rezultāta pirmās rindas pirmo kolonnu.
 Atgrieztā vērtība: atkarībā no SQL vaicājuma.

Parametri:

- **unikāls īss apraksts** - unikāls īss apraksts, lai identificētu vienumu (izmantošanai trigeros u.c.);
- **dsn** - datu avota nosaukums (kā norādīts `odbc.ini`);
- **savienojuma virkne** - savienojuma virkne (var saturēt draiverim specifiskus argumentus).

Komentāri:

- Lai gan `dsn` un `savienojuma virkne` ir neobligāti parametri, ir nepieciešams norādīt vismaz vienu no tiem; ja ir definēti abi, `dsn` tiks ignorēts.
- Ja vaicājums atgriež vairāk nekā vienu kolonnu, tiek nolasīta tikai pirmā kolonna. Ja vaicājums atgriež vairāk nekā vienu rindu, tiek nolasīta tikai pirmā rinda.

db.odbc.get[<unikāls īss apraksts>,<dsn>,<savienojuma virkne>]

 Pārveido SQL vaicājuma rezultātu par JSON masīvu.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **unikāls īss apraksts** — unikāls īss apraksts, lai identificētu vienumu (izmantošanai trigeros u.c.);
- **dsn** — datu avota nosaukums (kā norādīts `odbc.ini`);
- **savienojuma virkne** — savienojuma virkne (var saturēt draiverim specifiskus argumentus).

Komentāri:

- Lai gan `dsn` un `savienojuma virkne` ir neobligāti parametri, ir nepieciešams norādīt vismaz vienu no tiem; ja ir definēti abi, `dsn` tiks ignorēts.
- Var tikt atgrieztas vairākas rindas/kolonnas JSON formātā. Šo vienumu var izmantot kā galveno vienumu, kas apkopo visus datus vienā sistēmas izsaukumā, savukārt atkarīgajos vienumos var izmantot JSONPath priekšapstrādi, lai iegūtu atsevišķas vērtības. Papildinformāciju skatiet atgrieztā formāta piemērā, kas izmantots zema līmeņa atklāšanā.

Piemērs:

```
# Savienojums MySQL ODBC draiverim 5:
db.odbc.get[MySQL example,,"Driver=/usr/local/lib/libmyodbc5a.so;Database=master;Server=127.0.0.1;Port=330
```

db.odbc.discovery[<unikāls īss apraksts>,<dsn>,<savienojuma virkne>]

 Pārveido SQL vaicājuma rezultātu par JSON masīvu, ko izmanto zema līmeņa atklāšanā. Kolonnu nosaukumi no vaicājuma rezultāta tiek pārvērsti par zema līmeņa atklāšanas makrosu nosaukumiem, kas ir sasaistīti ar atklāto lauku vērtībām. Šos makrosus var izmantot, veidojot vienumu, trigeru u.c. prototipus.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **unikāls īss apraksts** - unikāls īss apraksts vienuma identificēšanai (izmantošanai trigeros u.c.);
- **dsn** - datu avota nosaukums (kā norādīts `odbc.ini`);
- **savienojuma virkne** - savienojuma virkne (var saturēt draiverim specifiskus argumentus).

Komentāri:

- Lai gan `dsn` un `savienojuma virkne` ir neobligāti parametri, ir nepieciešams norādīt vismaz vienu no tiem; ja ir definēti abi, `dsn` tiks ignorēts.

Kļūdu ziņojumi

ODBC kļūdu ziņojumi ir strukturēti laukos, lai sniegtu detalizētu informāciju. Piemēram, kļūdas ziņojums var izskatīties šādi:

```
Cannot execute ODBC query: [SQL_ERROR]:[42601][7][ERROR: syntax error at or near ";"; Error while executin
```

- "`Cannot execute ODBC query`" - Zabbix ziņojums
- "`[SQL_ERROR]`" - ODBC atgriešanas kods
- "`[42601]`" - SQLState
- "`[7]`" - vietējās kļūdas kods
- "`[ERROR: syntax error at or near ";"; Error while executing the query]`" - vietējās kļūdas ziņojums

Ņemiet vērā, ka kļūdas ziņojuma garums ir ierobežots līdz 2048 baitiem, tāpēc ziņojums var tikt saīsināts. Ja ir vairāk nekā viens ODBC diagnostikas ieraksts, Zabbix mēģina tos savienot vienā ziņojumā (atdalot ar |) tik tālu, cik to pieļauj garuma ierobežojums.

15 Atkarīgie vienumi

Pārskats

Ir situācijas, kad viens vienums vienlaikus apkopo vairākus metriku rādītājus vai pat ir lietderīgāk saistītos metriku rādītājus ievākt vienlaikus, piemēram:

- Atsevišķu kodolu CPU noslodze
- Ienākošā/izejošā/kopējā tīkla plūsma

Lai nodrošinātu metriku lielapjoma ievākšanu un vienlaicīgu izmantošanu vairākos saistītos vienumos, Zabbix atbalsta atkarīgos vienumus. Atkarīgie vienumi ir atkarīgi no galvenā vienuma, kas to datus ievāc vienlaikus vienā vaicājumā. Jauna vērtība galvenajam vienumam automātiski aizpilda atkarīgo vienumu vērtības. Atkarīgajiem vienumiem nevar būt atšķirīgs atjaunināšanas intervāls no galvenā vienuma.

Zabbix priekšapstrādes opcijas var izmantot, lai no galvenā vienuma datiem iegūtu to daļu, kas nepieciešama atkarīgajam vienumam.

Priekšapstrādi pārvalda `preprocessing manager` process kopā ar darba pavedieniem, kas izpilda priekšapstrādes soļus. Visas vērtības ar priekšapstrādi, kas saņemtas no dažādiem datu ievācējiem, pirms pievienošanas vēstures kešatmiņai iziet caur priekšapstrādes pārvaldnieku. Saziņai starp datu ievācējiem (polleriem, trapperiem u.c.) un priekšapstrādes procesu tiek izmantota uz ligzdām balstīta IPC saziņa.

Zabbix serveris vai Zabbix starpniekserveris (ja hosts tiek uzraudzīts ar proxy) veic priekšapstrādes soļus un apstrādā atkarīgos vienumus.

Par galvenajiem vienumiem var iestatīt jebkura tipa vienumus, pat atkarīgos vienumus. Var izmantot papildu atkarīgo vienumu līmeņus, lai no esoša atkarīgā vienuma vērtības iegūtu mazākas daļas.

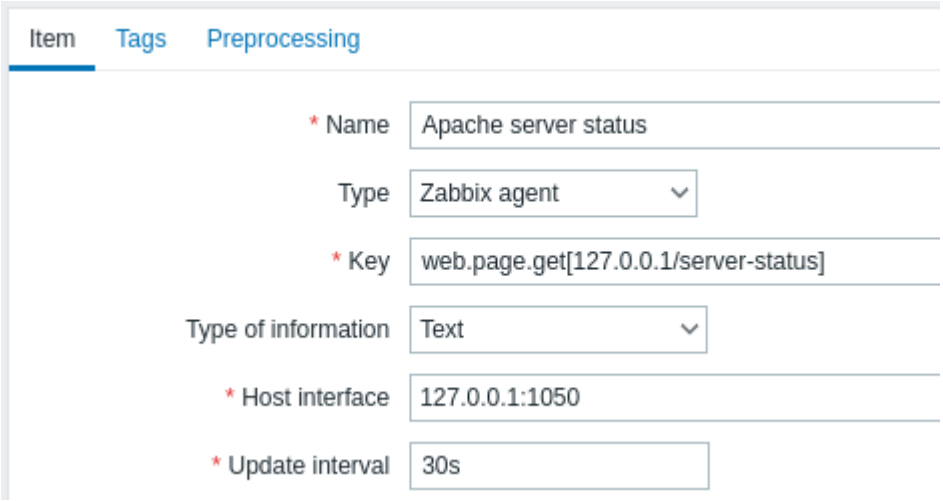
Ierobežojumi

- Ir atļautas tikai atkarības viena un tā paša hosta (veidnes) ietvaros
- Vienuma prototips var būt atkarīgs no cita vienuma prototipa vai parasta vienuma no tā paša hosta
- Atkarīgais vienums hostā ar galveno vienumu no veidnes netiks eksportēts uz XML

Vienuma konfigurācija

Atkarīgais vienums ir atkarīgs no tā galvenā vienuma datiem. Tāpēc vispirms ir jākonfigurē (vai jāpastāv) **galvenajam vienumam**:

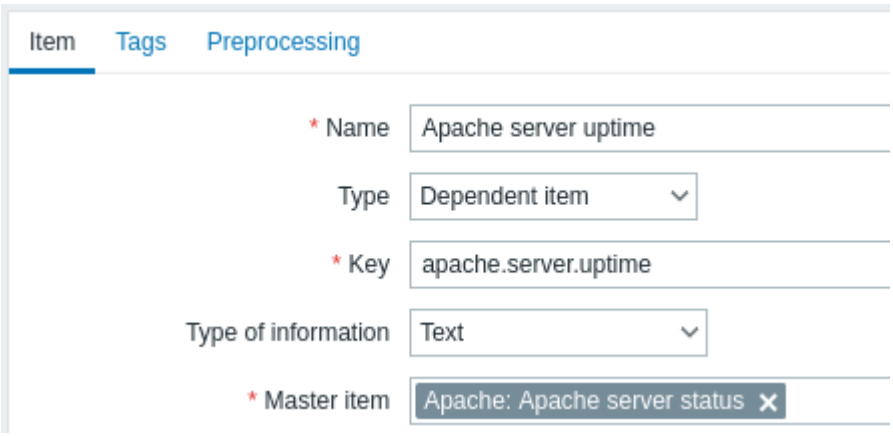
- Dodieties uz: *Datu ievākšana* → *Hosts*
- Noklikšķiniet uz *Items* attiecīgā hosta rindā
- Noklikšķiniet uz *Create item*
- Veidlapā ievadiet vienuma parametrus



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Noklikšķiniet uz *Add*, lai saglabātu galveno vienumu.

Pēc tam varat konfigurēt **atkarīgo vienumu**.

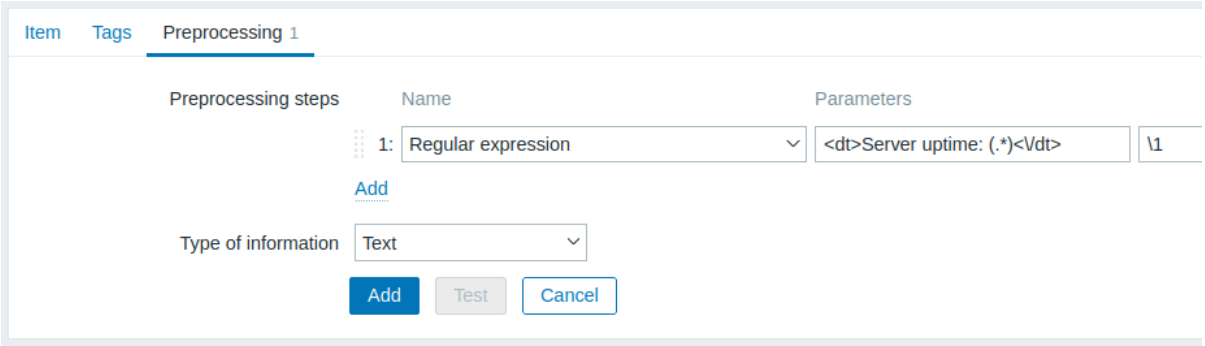


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuros atkarīgajiem vienumiem nepieciešama specifiska informācija, ir:

Type	Šeit atlasiet **Dependent item**.
Key	Ievadiet atslēgu, kas tiks izmantota vienuma atpazīšanai.
Master item	Atlasiet galveno vienumu. Galvenā vienuma vērtība tiks izmantota, lai aizpildītu atkarīgā vienuma vērtību.
Type of information	Atlasiet informācijas tipu, kas atbildīs saglabājamo datu formātam.

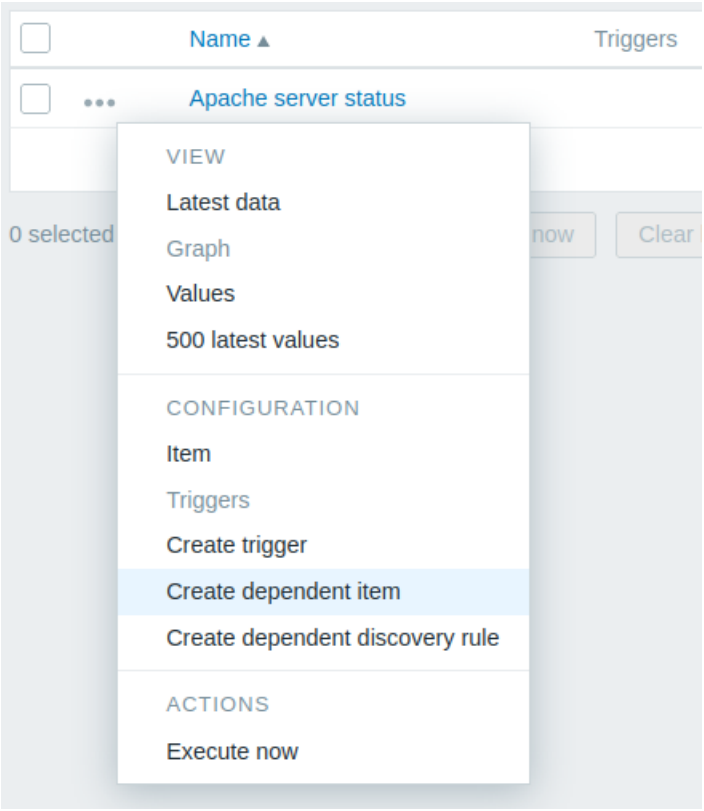
Varat izmantot vienuma vērtības priekšapstrādi, lai izvilktu nepieciešamo galvenā vienuma vērtības daļu.



Bez priekšapstrādes atkarīgā vienuma vērtība būs tieši tāda pati kā galvenā vienuma vērtība.

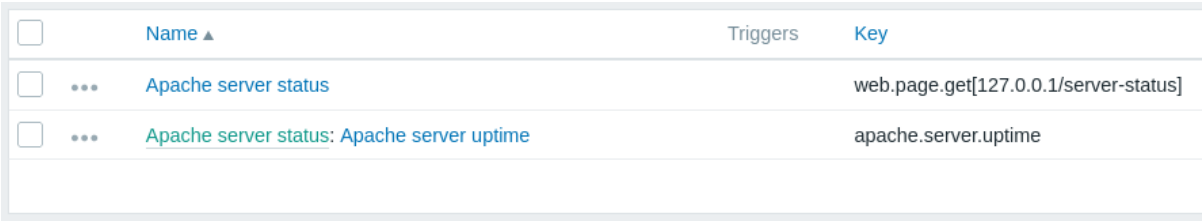
Noklikšķiniet uz *Add*, lai saglabātu atkarīgo vienumu.

Saīsni ātrākai atkarīgā vienuma izveidei var atvērt, noklikšķinot uz pogas ••• vienumu sarakstā un atlasot *Create dependent item*.



Attēlojums

Vienumu sarakstā atkarīgie vienumi tiek attēloti ar to galvenā vienuma nosaukumu kā prefiksu.



Ja galvenais vienums tiek dzēsts, tiek dzēsti arī visi no tā atkarīgie vienumi.

16 HTTP aģents

Pārskats

Šis vienuma tips ļauj aptaujāt datus, izmantojot HTTP/HTTPS protokolu. Ir iespējama arī datu saņemšana pasīvajā režīmā, izmantojot utilītu Zabbix sender vai Zabbix sender protocol (datu nosūtīšanai uz Zabbix serveri vai starpniekserveri), vai izmantojot API metodi `history.push` (datu nosūtīšanai uz Zabbix serveri).

HTTP vienumu pārbaudes izpilda Zabbix serveris. Tomēr, ja hostus uzrauga Zabbix starpniekserveris, HTTP vienumu pārbaudes izpilda starpniekserveris.

HTTP vienumu pārbaudēm nav nepieciešams, lai uz uzraugāmā hosta darbotos kāds aģents.

HTTP aģents atbalsta gan HTTP, gan HTTPS. Zabbix pēc izvēles sekos pāradresācijām (skatiet tālāk opciju *Sekot pāradresācijām*). Maksimālais pāradresāciju skaits ir stingri noteikts uz 10 (izmantojot cURL opciju CURLOPT_MAXREDIRS).

Attention:
Zabbix serverim/starpniekserverim sākotnēji jābūt konfigurētam ar cURL (libcurl) atbalstu.

HTTP pārbaudes tiek izpildītas asinhroni — nav nepieciešams saņemt atbildi uz vienu pieprasījumu, pirms tiek sāktas citas pārbaudes. Arī DNS atrisināšana notiek asinhroni.

Asinhrono pārbaužu maksimālais vienlaicīgums ir 1000 (noteikts ar MaxConcurrentChecksPerPoller).

Asinhrono HTTP aģenta aptaujātāju skaitu nosaka parametrs StartHTTPAgentPollers.

cURL pastāvīgo savienojumu iespēja ir pievienota HTTP aģenta pārbaudēm, sākot ar Zabbix 7.0.

Konfigurācija

Lai konfigurētu HTTP vienumu:

- Dodieties uz: *Datu ievākšana* → *Hosts*
- Noklikšķiniet uz *Items* attiecīgā hosta rindā
- Noklikšķiniet uz *Create item*
- Veidlapā ievadiet vienuma parametrus

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:

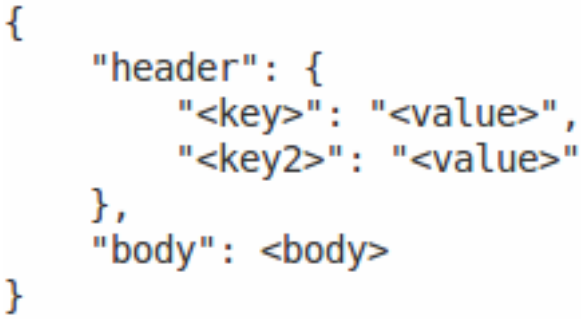
Enabled: ☑

Add | Test | Cancel

Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem HTTP vienumiem nepieciešama specifiska informācija, ir:

Parameter	Description
Type	Šeit atlasiet **HTTP aģents**.
Key	Ievadiet unikālu vienuma atslēgu.
URL	URL, ar kuru izveidot savienojumu un iegūt datus. Piemēram: https://www.example.com http://www.example.com/download Domēna vārdus var norādīt Unicode rakstzīmēs. Izpildot HTTP pārbaudi, tie automātiski tiek pārveidoti uz ASCII, izmantojot punycode. Pogu *Parse* var izmantot, lai no URL atdalītu neobligātos vaicājuma laukus (piemēram, ?name=Admin&password=mypassword), pārvietojot atribūtus un vērtības uz *Query fields* automātiskai URL kodēšanai. Ierobežots līdz 2048 rakstzīmēm. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_URL.
Query fields	Mainīgie URL vajadzībām (skatīt iepriekš). Norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski URL kodētas. Vērtības no makrosiem vispirms tiek atrisinātas un pēc tam automātiski URL kodētas. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_URL.
Request type	Atlasiet pieprasījuma metodes tipu: *GET*, *POST*, *PUT* vai *HEAD*
Request body type	Atlasiet pieprasījuma pamatteksta tipu: **Raw data** - pielāgots HTTP pieprasījuma pamatteksts, makrosi tiek aizvietoti, bet kodēšana netiek veikta **JSON data** - HTTP pieprasījuma pamatteksts JSON formātā. Makrosus var izmantot kā virkni, skaitli, true un false; makrosiem, kas izmantoti kā virknes, jābūt ietvertiem dubultpēdiņās. Vērtības no makrosiem vispirms tiek atrisinātas un pēc tam automātiski atsoļotas. Ja galvenēs nav norādīts ”Content-Type”, pēc noklusējuma tiks izmantots ”Content-Type: application/json” **XML data** - HTTP pieprasījuma pamatteksts XML formātā. Makrosus var izmantot kā teksta mezglu, atribūtu vai CDATA sadaļu. Vērtības no makrosiem vispirms tiek atrisinātas un pēc tam automātiski atsoļotas teksta mezglā un atribūtā. Ja galvenēs nav norādīts ”Content-Type”, pēc noklusējuma tiks izmantots ”Content-Type: application/xml”
Request body	Ievadiet pieprasījuma pamattekstu. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi.
Headers	Pielāgotas HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu. Norādītas kā atribūtu un vērtību pāri. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_HTTPHEADER.
Required status codes	Sagaidāmo HTTP statusa kodu saraksts. Ja Zabbix saņem kodu, kas nav šajā sarakstā, vienums kļūs neatbalstīts. Ja lauks ir tukšs, pārbaude netiek veikta. Piemēram: 200,201,210-299 Atbalstītie makrosi sarakstā: lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas izmanto cURL opciju CURLINFO_RESPONSE_CODE.
Follow redirects	Atzīmējiet izvēles rūtiņu, lai sekotu HTTP pāradresācijām. Tas iestata cURL opciju CURLOPT_FOLLOWLOCATION.
Retrieve mode	Atlasiet atbildes daļu, kas jāiegūst: **Body** - tikai pamatteksts **Headers** - tikai galvenes **Body and headers** - pamatteksts un galvenes

Parameter	Description
Convert to JSON	Galvenes tiek saglabātas kā atribūtu un vērtību pāri zem atslēgas ”header”, ja *Retrieve mode* nav iestatīts uz *Body*. Ja tiek konstatēts ’Content-Type: application/json’, tad pamatteksts tiek saglabāts kā objekts, pretējā gadījumā tas tiek saglabāts kā virkne, piemēram: 
HTTP proxy	Varat norādīt izmantojamo HTTP starpniekserveri formātā `[protocol://][username[:password]@]proxy.example.com[:port]`. Neobligāto prefiksu `protocol://` var izmantot, lai norādītu alternatīvus starpniekservera protokolus (piemēram, https, socks4, socks5; skatiet dokumentāciju; protokola prefiksa atbalsts tika pievienots cURL 7.21.7 versijā). Ja protokols nav norādīts, starpniekserveris tiks uzskatīts par HTTP starpniekserveri. Ja norādīsiet nepareizu protokolu, savienojums neizdosies un vienums kļūs neatbalstīts. Pēc noklusējuma tiks izmantots ports 1080. Ja norādīts, starpniekserveris pārrakstīs ar starpniekserveri saistītos vides mainīgos, piemēram, http_proxy, HTTPS_PROXY. Ja nav norādīts, starpniekserveris nepārrakstīs ar starpniekserveri saistītos vides mainīgos. Ievadītā vērtība tiek nodota ”kā ir”, bez papildu pārbaudes. *Ņemiet vērā*, ka ar HTTP starpniekserveri tiek atbalstīta tikai vienkāršā autentifikācija. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_PROXY.
HTTP authentication	Atlasiet autentifikācijas opciju: **None** - autentifikācija netiek izmantota; **Basic** - tiek izmantota pamata autentifikācija; **NTLM** - tiek izmantota NTLM (Windows NT LAN Manager) autentifikācija; **Kerberos** - tiek izmantota Kerberos autentifikācija (skatiet arī: Kerberos konfigurēšana ar Zabbix); **Digest** - tiek izmantota Digest autentifikācija. Tas iestata cURL opciju CURLOPT_HTTPAUTH.
User name	Ievadiet lietotājvārdu (līdz 255 rakstzīmēm). Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz Basic, NTLM, Kerberos vai Digest. Tiek atbalstīti lietotāja makrosi un zema līmeņa atklāšanas makrosi.
Password	Ievadiet lietotāja paroli (līdz 255 rakstzīmēm). Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz Basic, NTLM, Kerberos vai Digest. Tiek atbalstīti lietotāja makrosi un zema līmeņa atklāšanas makrosi.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiks automātiski ņemts no sistēmas mēroga sertifikācijas iestādes (CA) atrašanās vietas. CA failu atrašanās vietu varat pārrakstīt, izmantojot Zabbix serveris vai starpniekserveris konfigurācijas parametru SSLCALocation. Tas iestata cURL opciju CURLOPT_SSL_VERIFYPEER.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta Common Name lauks vai Subject Alternate Name lauks atbilst. Tas iestata cURL opciju CURLOPT_SSL_VERIFYHOST.
SSL certificate file	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai. Sertifikāta failam jābūt PEM[1] formātā. Ja sertifikāta fails satur arī privāto atslēgu, atstājiet lauku SSL key file tukšu. Ja atslēga ir šifrēta, norādiet paroli laukā SSL key password. Katalogs, kurā atrodas šis fails, tiek norādīts ar Zabbix serveris vai starpniekserveris konfigurācijas parametru SSLCertLocation. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_SSLCERT.

Parameter	Description
SSL key file	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai. Privātās atslēgas failam jābūt PEM[1] formātā. Katalogs, kurā atrodas šis fails, tiek norādīts ar Zabbix serveris vai starpniekserveris konfigurācijas parametru SSLKeyLocation. Atbalstītie makrosi: {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_SSLKEY.
SSL key password	SSL privātās atslēgas faila parole. Atbalstītie makrosi: lietotāja makrosi, zema līmeņa atklāšanas makrosi. Tas iestata cURL opciju CURLOPT_KEYPASSWD.
Timeout	Zabbix netērēs vairāk par iestatīto laiku URL apstrādei (1-600 sekundes). Faktiski šis parametrs nosaka maksimālo laiku savienojuma izveidei ar URL un maksimālo laiku HTTP pieprasījuma izpildei. Tāpēc Zabbix vienai pārbaudei netērēs vairāk par 2 x *Timeout* sekundēm. Tas iestata cURL opciju CURLOPT_TIMEOUT. Papildinformāciju par parametru *Timeout* skatiet sadaļā vispārīgie vienuma atribūti.
Enable trapping	Ja šī izvēles rūtiņa ir atzīmēta, vienums darbosies arī kā trapper vienums un pieņems datus, kas nosūtīti uz Zabbix serveris vai starpniekserveris, izmantojot utilītu Zabbix sender vai Zabbix sender protocol, vai arī pieņems datus, kas nosūtīti uz Zabbix serveris, izmantojot API metodi `history.push`. Papildinformāciju par datu nosūtīšanu skatiet: Trapper vienumi.
Allowed hosts	Redzams tikai tad, ja ir atzīmēta izvēles rūtiņa *Enable trapping*. Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Ja norādīts, ienākošie savienojumi tiks pieņemti tikai no šeit uzskaitītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka "IPv4-compatible IPv6 addresses" (prefikss 0000::/96) tiek atbalstītas, bet ir novecojušas saskaņā ar RFC4291. Piemērs: 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} Šajā laukā ir atļautas atstarpes un lietotāja makrosi. Šajā laukā ir atļauti hosta makrosi: {HOST.HOST}, {HOST.NAME}, {HOST.IP}, {HOST.DNS}, {HOST.CONN}.

Note:
Ja lauks *HTTP proxy* ir atstāts tukšs, cits veids, kā izmantot HTTP starpniekserveri, ir iestatīt ar starpniekserveri saistītos vides mainīgos.
HTTP gadījumā — iestatiet vides mainīgo `http_proxy` Zabbix serveris lietotājam. Piemēram:
 `http_proxy=http://proxy_ip:proxy_port`.
HTTPS gadījumā — iestatiet vides mainīgo `HTTPS_PROXY`. Piemēram:
 `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Plašāka informācija ir pieejama, izpildot čaulas komandu: *# man curl*.

Attention:
[1] Zabbix atbalsta sertifikātu un privātās atslēgas failus tikai PEM formātā. Ja jūsu sertifikāta un privātās atslēgas dati ir PKCS #12 formāta failā (parasti ar paplašinājumu *.p12 vai *.pfx), varat no tā ģenerēt PEM failu, izmantojot šādas komandas:

```
openssl pkcs12 -in ssl-cert.p12 -clcerts -nokeys -out ssl-cert.pem
openssl pkcs12 -in ssl-cert.p12 -nocerts -nodes  -out ssl-cert.key
```

Piemēri

Piemērs 1

Sūtiet vienkāršus GET pieprasījumus, lai iegūtu datus no tādiem pakalpojumiem kā Elasticsearch:

- Izveidojiet GET vienumu ar URL: `localhost:9200/?pretty`
- Pievērsiet uzmanību atbildei:

```
{
  "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"
}
```

- Tagad izgūstiet versijas numuru, izmantojot JSONPath priekšapstrādes soli: `$.version.number`

Piemērs 2

Sūtiet vienkāršus POST pieprasījumus, lai iegūtu datus no tādiem pakalpojumiem kā Elasticsearch:

- Izveidojiet POST vienumu ar URL: `http://localhost:9200/str/values/_search?scroll=10s`
- Konfigurējiet šādu POST pamattekstu, lai iegūtu procesora noslodzi (1 min vidējā vērtība uz kodolu)

```
{
    "query": {
        "bool": {
            "must": [{
                "match": {
                    "itemid": 28275
                }
            }],
            "filter": [{
                "range": {
                    "clock": {
                        "gt": 1517565836,
                        "lte": 1517566137
                    }
                }
            }]
        }
    }
}
```

- Saņemts:

```
{
    "_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,
```

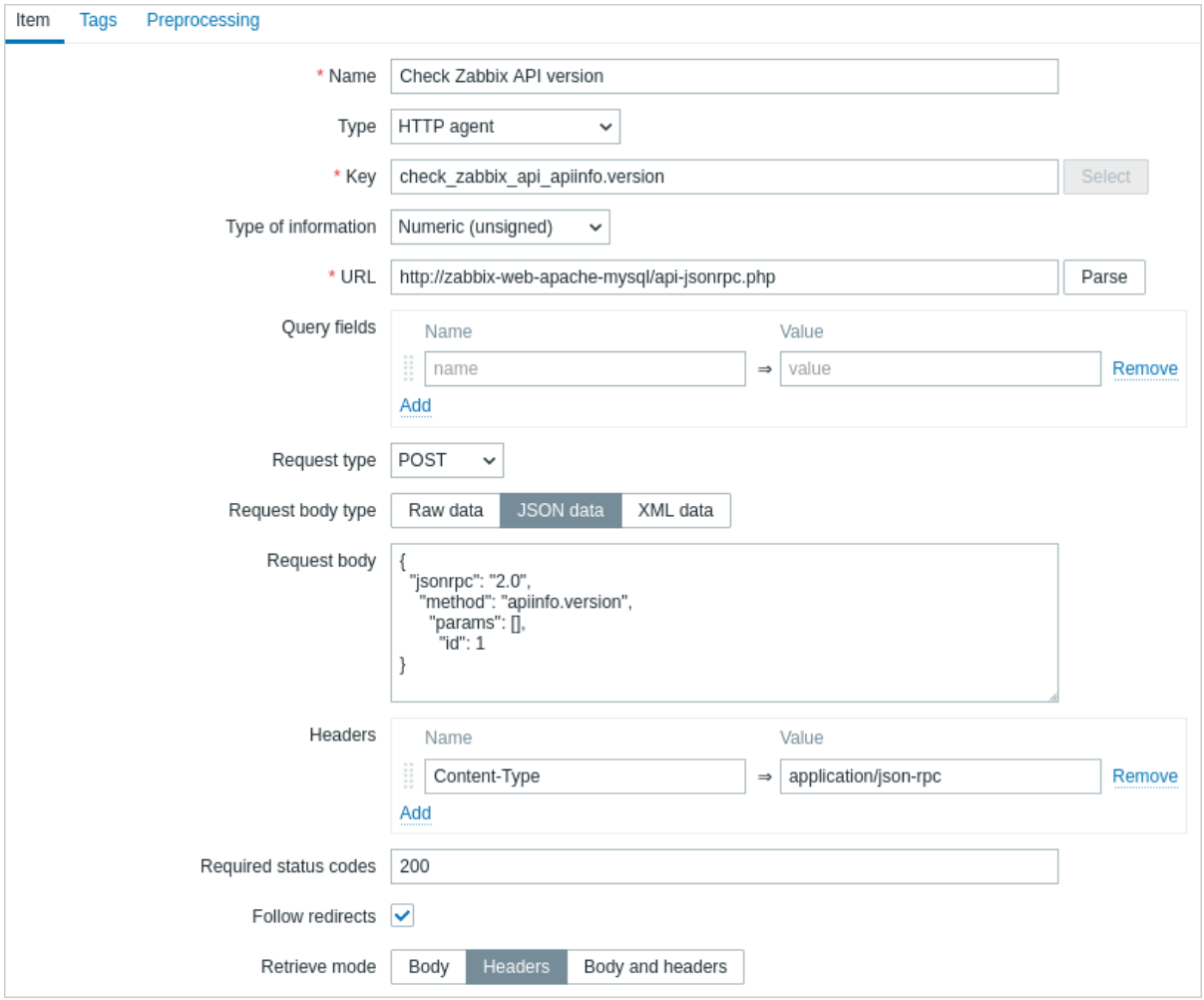
```
                    "ttl": 604800
                }
            }]
        }
    }
```

- Tagad izmantojiet JSONPath priekšapstrādes soli, lai iegūtu vienuma vērtību: `$.hits.hits[0]._source.value`

Piemērs 3

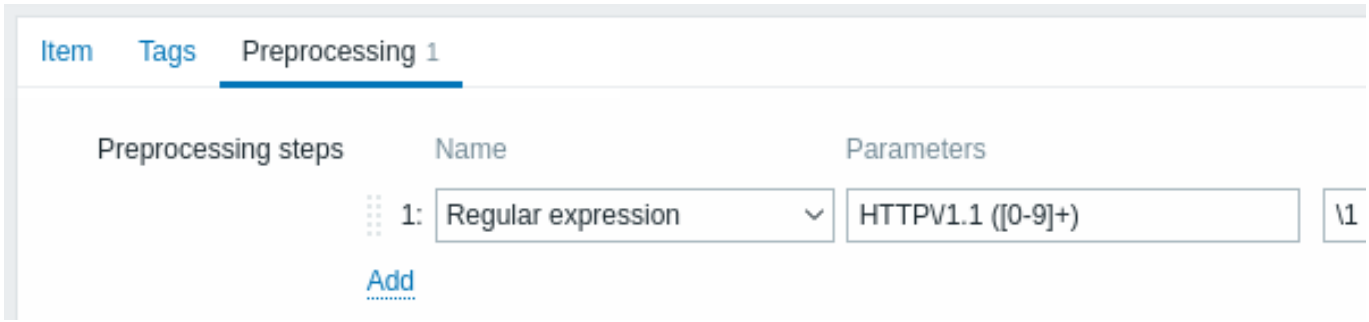
Pārbaude, vai Zabbix API darbojas, izmantojot apiinfo.version.

- Vienuma konfigurācija:

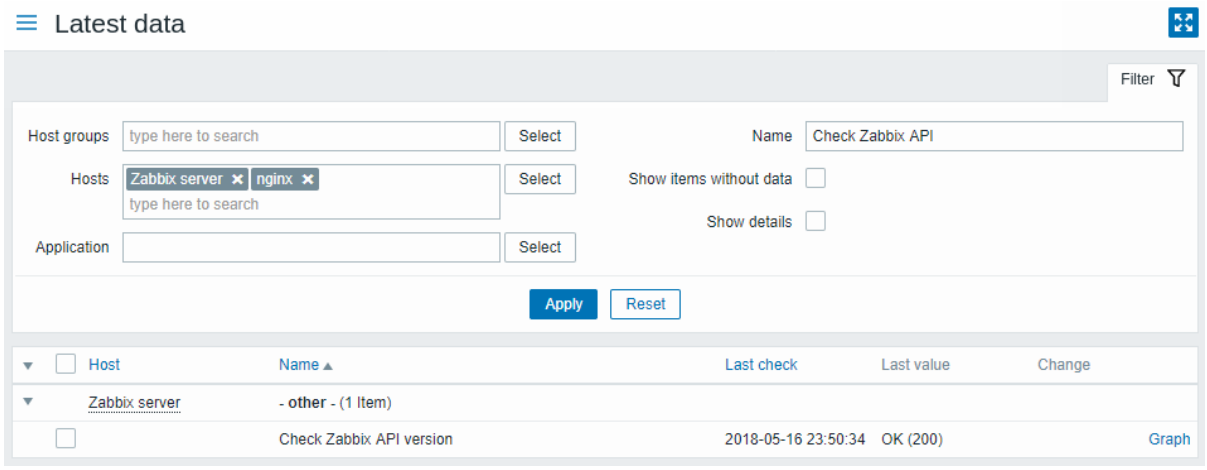


Ņemiet vērā POST metodes izmantošanu ar JSON datiem, pieprasījuma galveņu iestatīšanu un norādi atgriezt tikai galvenes:

- Vienuma vērtības priekšapstrāde ar regulāro izteiksmi, lai iegūtu HTTP kodu:



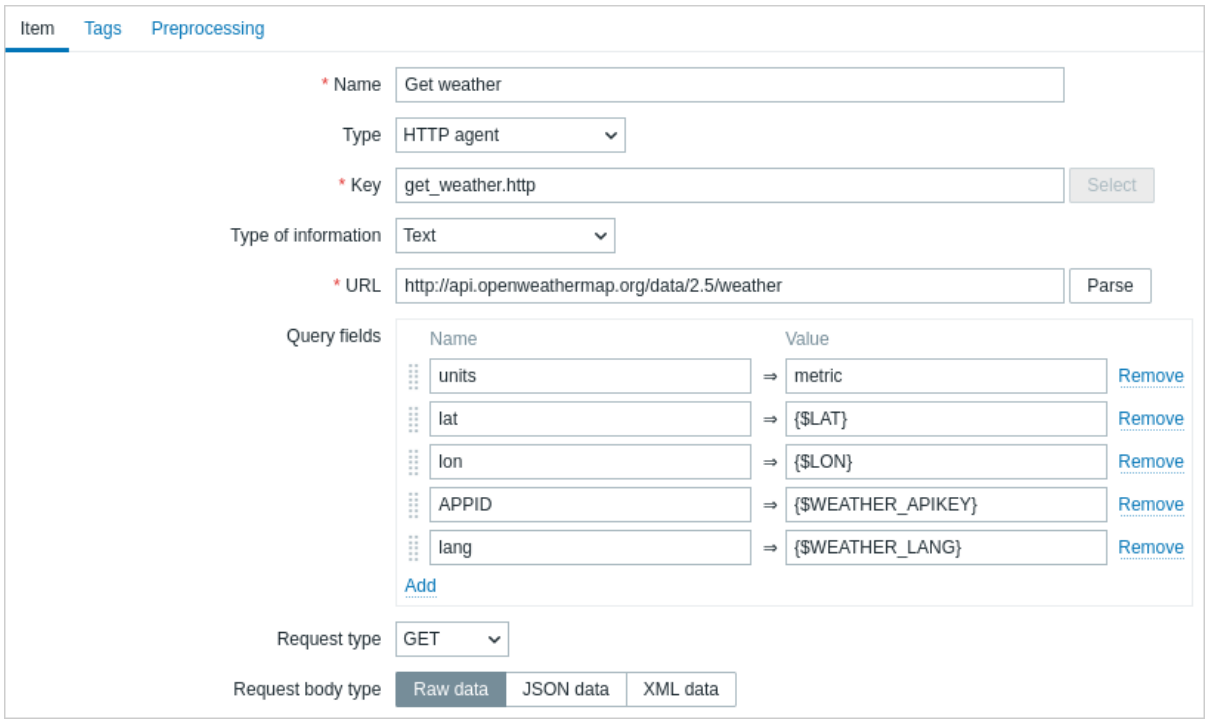
- Rezultāta pārbaude sadaļā *Jaunākie dati*:



Piemērs 4

Laikapstākļu informācijas iegūšana, pieslēdzoties Openweathermap publiskajam servisam.

- Konfigurējiet galveno vienumu lielapjoma datu vākšanai vienā JSON formātā:



Ņemiet vērā makrosu izmantošanu vaicājuma laukos. Skatiet Openweathermap API, lai uzzinātu, kā tos aizpildīt.

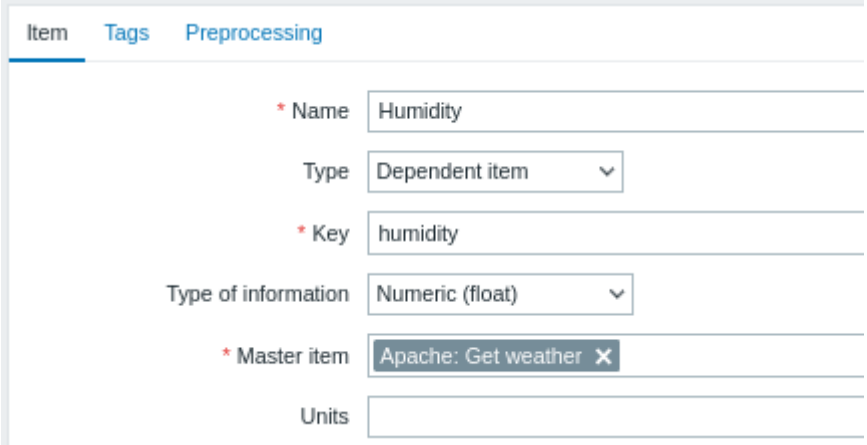
HTTP aģenta atbildē atgriezta JSON paraugs:

```
{
    "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
    }
}
```

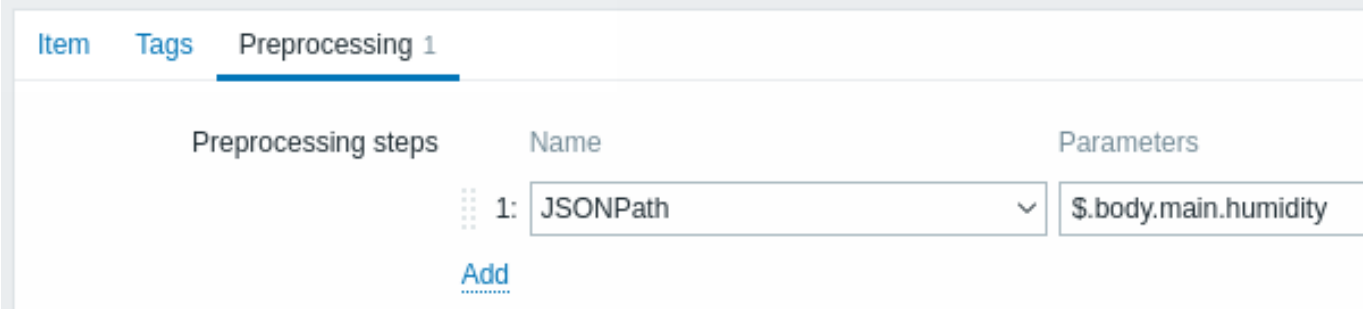

Nākamais uzdevums ir konfigurēt atkarīgos vienumus, kas iegūst datus no JSON.

- Konfigurējiet atkarīgā vienuma paraugu mitrumam:

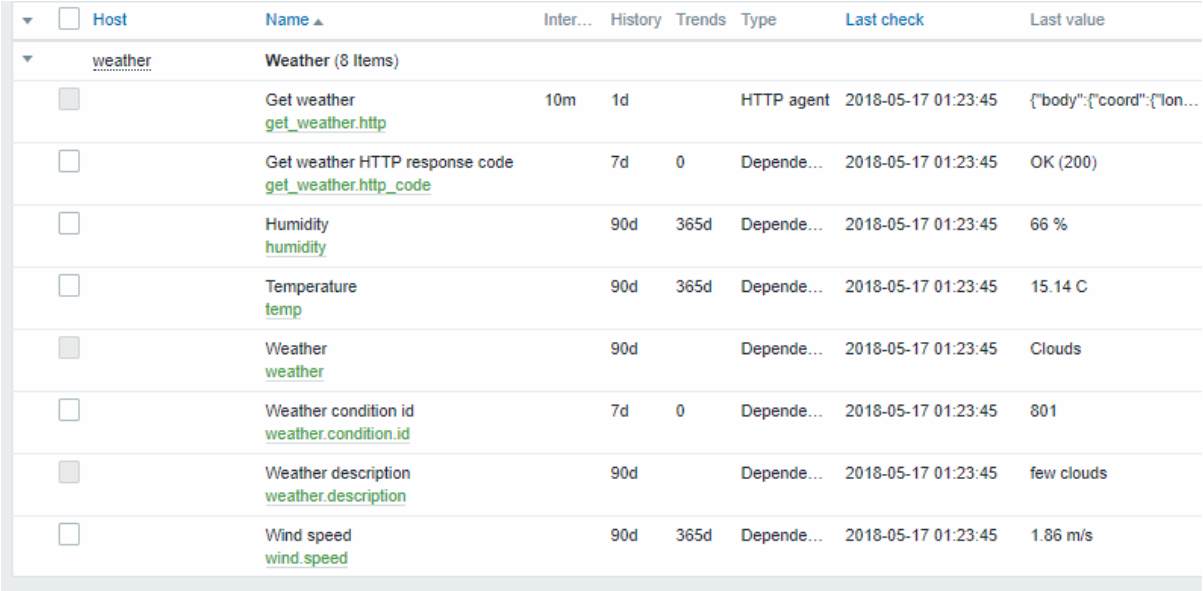


Citi laikapstākļu rādītāji, piemēram, 'Temperature', tiek pievienoti tādā pašā veidā.

- Atkarīgā vienuma vērtības priekšapstrādes paraugs ar JSONPath:



- Pārbaudiet laikapstākļu datu rezultātu sadaļā *Jaunākie dati*:

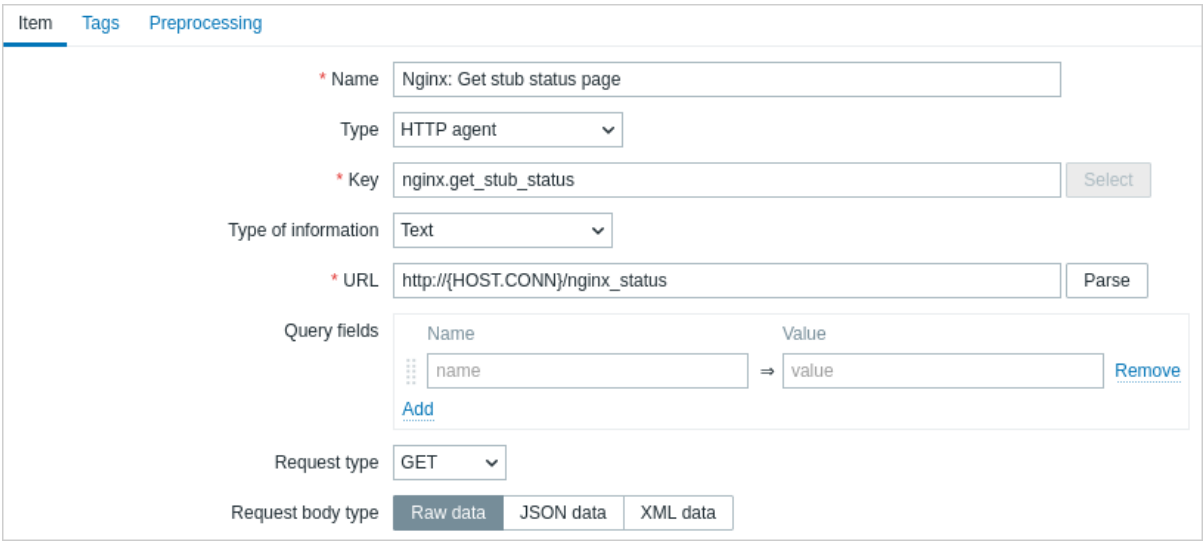


Host	Name ▲	Inter...	History	Trends	Type	Last check	Last value
weather	**Weather** (8 Items)						
	Get weather get_weather.http	10m	1d		HTTP agent	2018-05-17 01:23:45	{"body":{"coord":{"lon...
	Get weather HTTP response code get_weather.http_code		7d	0	Depende...	2018-05-17 01:23:45	OK (200)
	Humidity 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

Piemērs 5

Savienojuma izveide ar Nginx statusa lapu un tās metriku iegūšana paketē.

- Konfigurējiet Nginx, ievērojot oficiālo ceļvedi.
- Konfigurējiet galveno vienumu pakešveida datu vākšanai:

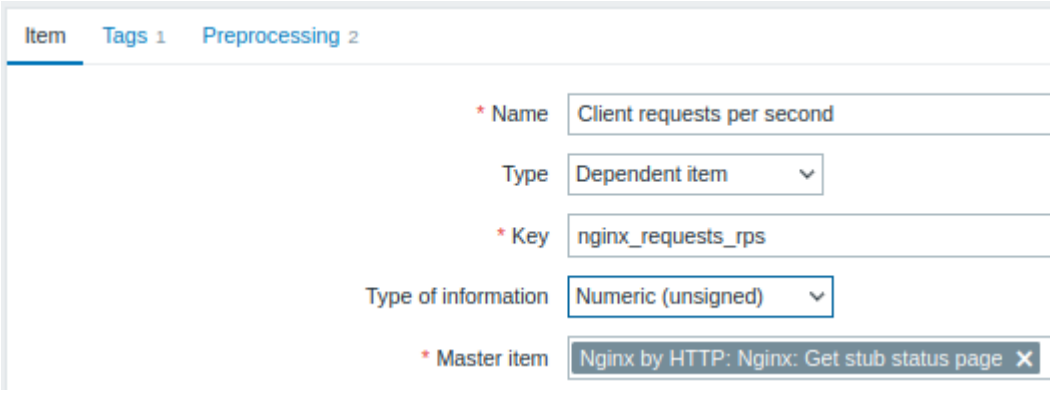


Nginx stub status parauga izvade:

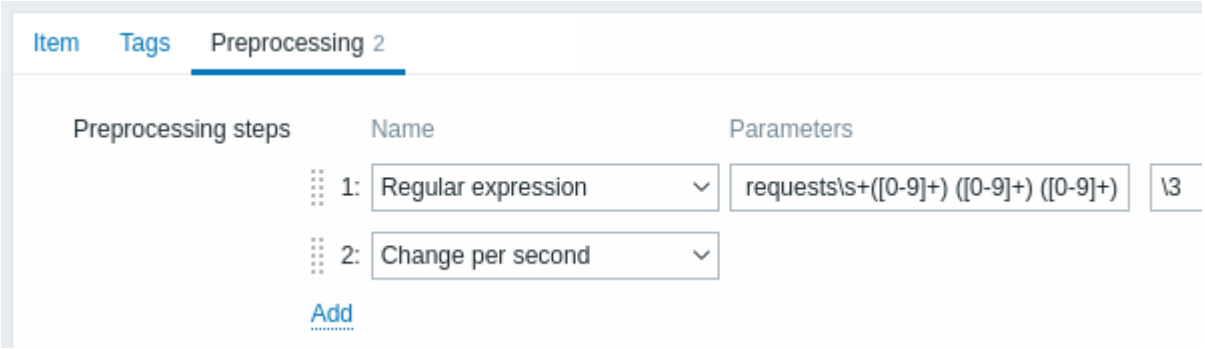
```
Active connections: 1 Active connections:
server accepts handled requests
 52 52 52
Reading: 0 Writing: 1 Waiting: 0
```

Nākamais uzdevums ir konfigurēt atkarīgos vienumus, kas izgūst datus.

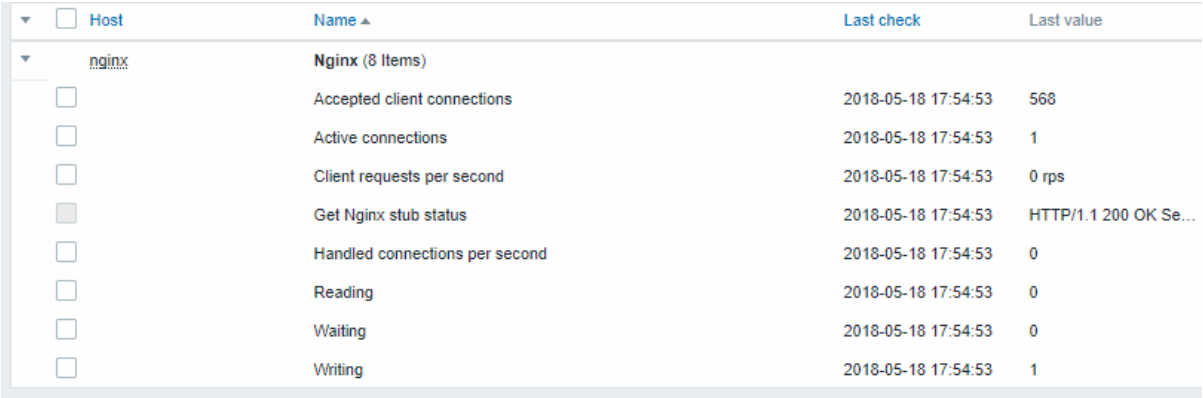
- Konfigurējiet atkarīgā vienuma paraugu pieprasījumiem sekundē:



- Atkarīgā vienuma vērtības pirmapstrādes paraugs ar regulāro izteiksmi `server accepts handled requests\s+([0-9]+) ([0-9]+) ([0-9]+)`:



- Pārbaudiet pilno rezultātu no stub moduļa sadaļā *Jaunākie dati*:



17 Prometheus pārbaudes

Pārskats

Zabbix var vaicāt metriku, kas ir pieejama Prometheus rindu formātā.

Lai sāktu vākt Prometheus datus, ir nepieciešami divi soļi:

- HTTP galvenais vienums, kas norāda uz atbilstošo datu galapunktu, piemēram, `https://<prometheus host>/metrics`
- atkarīgie vienumi, kas izmanto Prometheus priekšapstrādes opciju, lai vaicātu nepieciešamos datus no metrikām, ko savācis galvenais vienums

Ir pieejamas divas Prometheus datu priekšapstrādes opcijas:

- *Prometheus pattern* - tiek izmantots parastos vienumos, lai vaicātu Prometheus datus
- *Prometheus to JSON* - tiek izmantots parastos vienumos un zema līmeņa atklāšanai. Šajā gadījumā vaicātie Prometheus dati tiek atgriezti JSON formātā.

Lielapjoma apstrāde

Lielapjoma apstrāde tiek atbalstīta atkarīgajiem vienumiem. Lai iespējotu kešošanu un indeksēšanu, priekšapstrādei *Prometheus pattern* jābūt **pirmajam** priekšapstrādes solim. Ja *Prometheus pattern* ir pirmais priekšapstrādes solis, tad parsētie Prometheus dati tiek kešoti un indeksēti pēc pirmā `<label>==<value>` nosacījuma priekšapstrādes solī *Prometheus pattern*. Šis kešs tiek

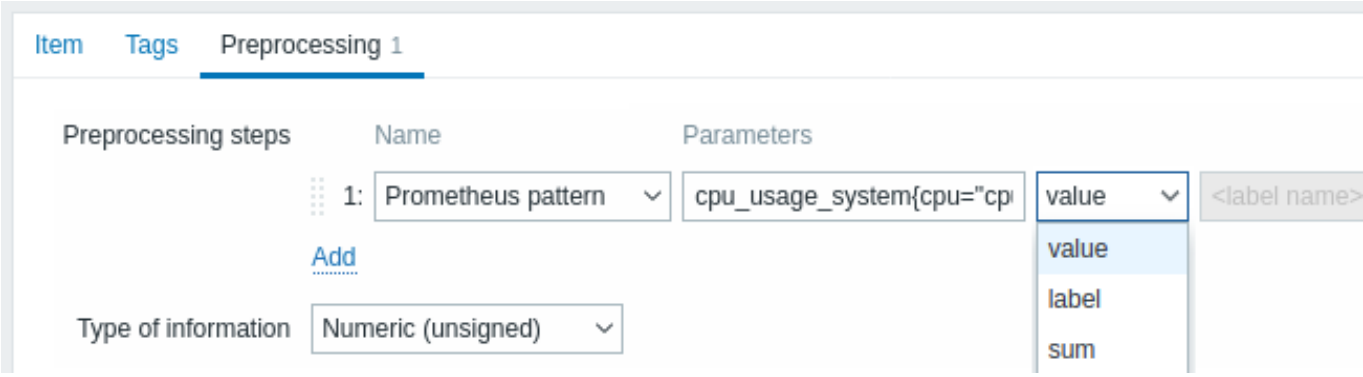
atkārtoti izmantots, apstrādājot citus atkarīgos vienumus šajā paketē. Optimālai veiktspējai pirmajai etiķetei vajadzētu būt tai, kurai ir visvairāk atšķirīgu vērtību.

Ja pirms pirmā soļa ir jāveic cita priekšapstrāde, tā būtu jāpārvieto vai nu uz galveno vienumu, vai uz jaunu atkarīgo vienumu, kas tiktu izmantots kā galvenais vienums atkarīgajiem vienumiem.

Konfigurācija

Ja HTTP master vienums jau ir konfigurēts, jums jāizveido atkarīgais vienums, kas izmanto Prometheus priekšapstrādes soli:

- Konfigurācijas formā ievadiet atkarīgā vienuma vispārīgos parametrus
- Atveriet cilni **Priekšapstrāde**
- Atlasiet Prometheus priekšapstrādes opciju (*Prometheus pattern* vai *Prometheus to JSON*)



Tālāk norādītie parametri ir specifiski priekšapstrādes opcijai *Prometheus pattern*:

Parametrs	Apraksts	Piemēri
Pattern	Lai definētu nepieciešamo datu šablonu, varat izmantot vaicājumu valodu, kas ir līdzīga Prometheus vaicājumu valodai (skatiet salīdzinājuma tabulu), piemēram: <metric name> - atlasīt pēc metrikas nosaukuma {__name__="<metric name>"} - atlasīt pēc metrikas nosaukuma {__name__=~"<regex>"} - atlasīt pēc metrikas nosaukuma, kas atbilst regulārajai izteiksmei {<label name>="<label value>",...} - atlasīt pēc etiķetes nosaukuma {<label name>=~"<regex>",...} - atlasīt pēc etiķetes nosaukuma, kas atbilst regulārajai izteiksmei {__name__=~".*"}==<value> - atlasīt pēc metrikas vērtības Vai iepriekš minētā kombināciju: <metric name>{<label1 name>="<label1 value>",<label2 name>=~"<regex>",...}==<value> Etiķetes vērtība var būt jebkura UTF-8 rakstzīmju secība, taču slīpsvītras, dubultpēdiņas un rindas pārneses rakstzīmes ir jāekranē attiecīgi kā \\, \" un \n; citas rakstzīmes nav jāekranē.	*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*

Parametrs	Apraksts	Piemēri
Result processing	Norādiet, vai atgriezt vērtību, etiķeti vai lietot atbilstošo funkciju (ja šablons atbilst vairākām rindām un rezultāts ir jāagregē): **value** - atgriezt metrikas vērtību (kļūda, ja atbilst vairākas rindas) **label** - atgriezt etiķetes vērtību, kas norādīta laukā *Label* (kļūda, ja atbilst vairākas metrikas) **sum** - atgriezt vērtību summu **min** - atgriezt minimālo vērtību **max** - atgriezt maksimālo vērtību **avg** - atgriezt vidējo vērtību **count** - atgriezt vērtību skaitu Šis lauks ir pieejams tikai opcijai *Prometheus pattern*.	Skatiet arī tālāk sniegtos parametru lietošanas piemērus.
Output	Definējiet etiķetes nosaukumu (neobligāti). Šajā gadījumā tiks atgriezta vērtība, kas atbilst etiķetes nosaukumam. Šis lauks ir pieejams tikai opcijai *Prometheus pattern*, ja laukā *Result processing* ir atlasīts 'Label'.	

Parametru lietošanas piemēri

1. Visizplatītākais lietošanas gadījums ir atgriezt **vērtību**. Lai atgrieztu /var/db vērtību no:

   `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`

izmantojiet šādus parametrus:

- *Pattern* - `node_disk_usage_bytes{path="/var/db"}`
- *Result processing* - atlasiet 'value'

2. Jūs varētu interesēt arī visu `node_disk_usage_bytes` parametru **vidējā** vērtība:

- *Pattern* - `node_disk_usage_bytes`
- *Result processing* - atlasiet 'avg'

3. Lai gan Prometheus atbalsta tikai skaitliskus datus, ir izplatīti izmantot apiešanas risinājumu, kas ļauj atgriezt arī atbilstošo teksta aprakstu. To var panākt ar filtru un norādot etiķeti. Tātad, lai atgrieztu etiķetes 'color' vērtību no

   `elasticsearch_cluster_health_status{cluster="elasticsearch",color="green"} 1<br> elasticsearch_cluster 0<br> elasticsearch_cluster_health_status{cluster="elasticsearch",color="yellow"} 0`

izmantojiet šādus parametrus:

- *Pattern* - `elasticsearch_cluster_health_status {cluster="elasticsearch"} == 1`
- *Result processing* - atlasiet 'label'
- *Label* - norādiet 'color'

Filtrs (balstīts uz skaitlisko vērtību '1') atbildīs attiecīgajai rindai, savukārt etiķete atgriezīs veselības stāvokļa aprakstu (pašlaik 'green', bet potenciāli arī 'red' vai 'yellow').

Prometheus uz JSON

Datus no Prometheus var izmantot zema līmeņa atklāšanai. Šajā gadījumā ir nepieciešami dati JSON formātā, un priekšapstrādes opcija *Prometheus uz JSON* atgriezīs tieši to.

Plašāku informāciju skatiet sadaļā Atklāšana, izmantojot Prometheus datus.

Vaicājumu valodu salīdzinājums

Tālāk esošajā tabulā ir uzskaitītas atšķirības un līdzības starp PromQL un Zabbix Prometheus priekšapstrādes vaicājumu valodu.

PromQL tūlītējā vektora selektors	Zabbix Prometheus priekšapstrādes vaicājumu valoda
Atšķirības	

PromQL tūlītējā vektora selektors		Zabbix Prometheus priekšapstrādes vaicājumu valoda
Vaicājuma mērķis	Prometheus serveris	Vienkāršs teksts Prometheus eksponēšanas formātā
Atgriež	Tūlītējs vektors	Metrikas vai etiķetes vērtība (Prometheus raksts) Metriku masīvs vienai vērtībai JSON formātā (Prometheus uz JSON)
Etiķešu atbilstības operatori	**=**, **!=**, **=~**, **!~**	**=**, **!=**, **=~**, **!~**
Regulārā izteiksme, ko izmanto etiķešu vai metriku nosaukumu atbilstības noteikšanai	RE2	PCRE
Salīdzināšanas operatori	Skatiet sarakstu	Vērtību filtrēšanai tiek atbalstīts tikai **==** (vienāds)
Līdzības		
Atlase pēc metrikas nosaukuma, kas ir vienāds ar virkni	<metric name> vai {__name__=”<metric name>”}	<metric name> vai {__name__=”<metric name>”}
Atlase pēc metrikas nosaukuma, kas atbilst regulārajai izteiksmei	{__name__=~”<regex>”}	{__name__=~”<regex>”}
Atlase pēc <label name> vērtības, kas ir vienāda ar virkni	{<label name>=”<label value>”,...}	{<label name>=”<label value>”,...}
Atlase pēc <label name> vērtības, kas atbilst regulārajai izteiksmei	{<label name>=~”<regex>”,...}	{<label name>=~”<regex>”,...}
Atlase pēc vērtības, kas ir vienāda ar virkni	{__name__=~”.*”} == <value>	{__name__=~”.*”} == <value>

18 Skripta vienumi

Pārskats

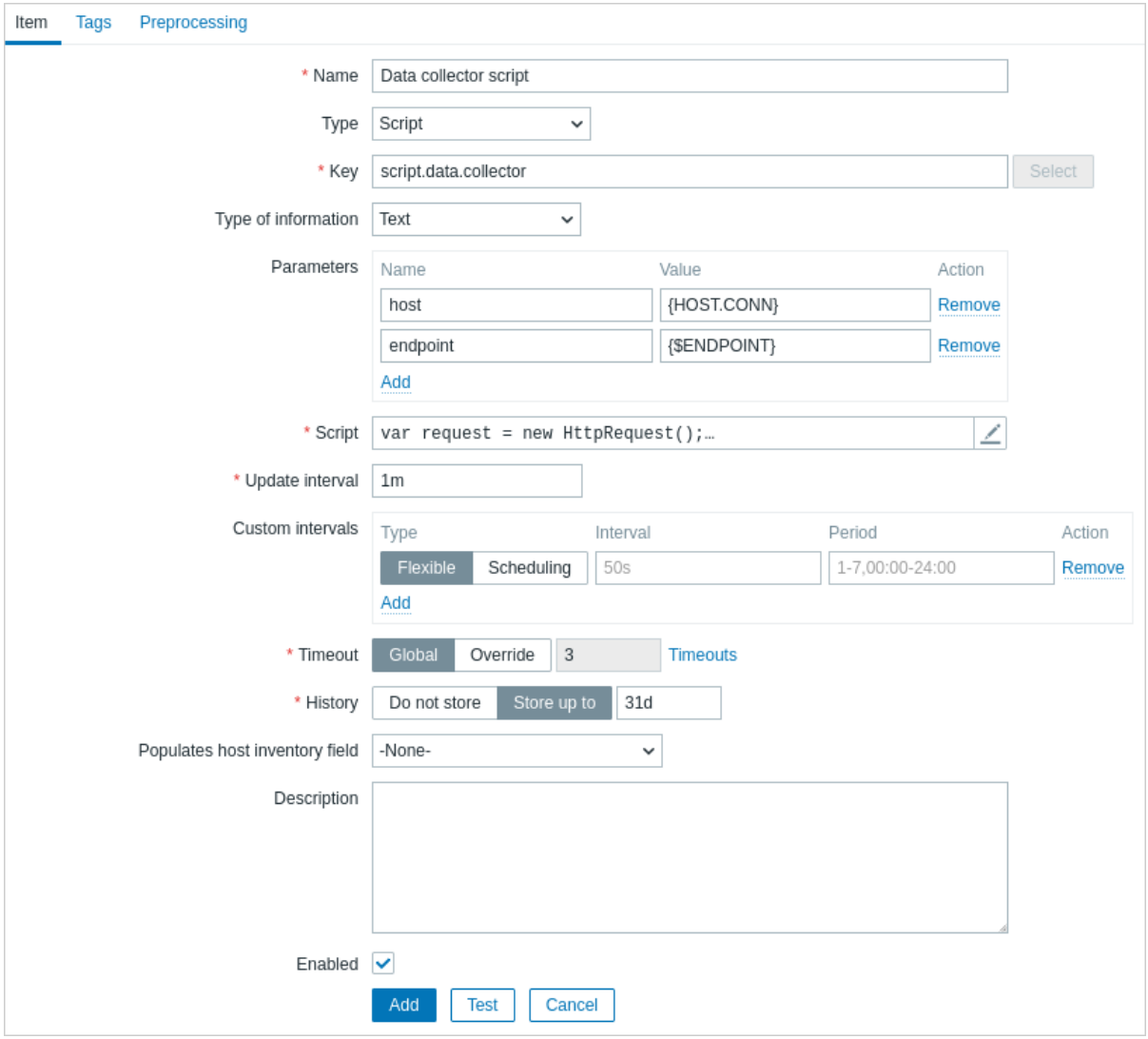
Skripta vienumus var izmantot datu vākšanai, izpildot lietotāja definētu JavaScript kodu ar iespēju iegūt datus, izmantojot HTTP/HTTPS. Papildus skriptam var norādīt neobligātu parametru sarakstu (nosaukuma un vērtības pārus) un noildzi.

Šis vienuma tips var būt noderīgs datu vākšanas scenārijos, kuros nepieciešami vairāki soļi vai sarežģīta loģika. Piemēram, skripta vienumu var konfigurēt tā, lai tas veiktu HTTP izsaukumu, pēc tam kaut kādā veidā apstrādātu pirmajā solī saņemtos datus un nodotu pārveidoto vērtību otrajam HTTP izsaukumam.

Skripta vienumus apstrādā Zabbix servera vai starpniekservera aptaujātāji.

Konfigurācija

Laukā *Type* vienuma konfigurācijas veidlapā atlasiet Script, pēc tam aizpildiet obligātos laukus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuros Script vienumiem ir jānorāda specifiska informācija, ir šādi:

Lauks	Apraksts
Key	Ievadiet unikālu atslēgu, kas tiks izmantota vienuma identificēšanai.
Parameters	Norādiet mainīgos, kas tiks nodoti skriptam kā atribūta un vērtības pāri. Tiek atbalstīti lietotāja makrosi. Lai redzētu, kuri iebūvētie makrosi tiek atbalstīti, meklējiet "Script-type item" tabulā supported macro.
Script	Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam. Šim kodam jānodrošina loģika metrikas vērtības atgriešanai. Kodam ir piekļuve visiem parametriem un visiem papildu JavaScript objektiem, ko pievieno Zabbix. Skatiet arī: JavaScript Guide.
Timeout	JavaScript izpildes noildze (1-600s; tās pārsniegšana atgriezīs kļūdu). Ņemiet vērā, ka atkarībā no skripta var būt nepieciešams ilgāks laiks, līdz noildze nostrādā. Papildinformāciju par parametru *Timeout* skatiet sadaļā general item attributes.

Piemēri

Vienkārša datu vākšana

Savāciet *https://www.example.com/release_notes* saturu:

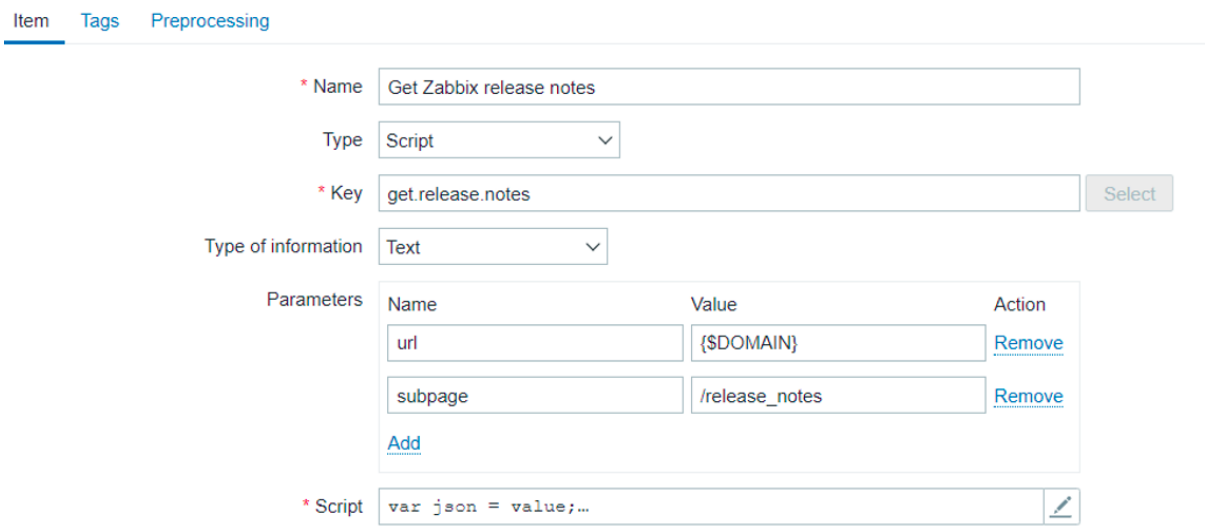
- Izveidojiet vienumu ar tipu "Script".
- Laukā *Script* ievadiet:

```
var request = new HttpRequest();
return request.get("https://www.example.com/release_notes");
```

Datu vākšana ar parametriem

Savāciet konkrētas lapas saturu un izmantojiet parametrus:

- Izveidojiet vienumu ar tipu ”Script” un diviem parametriem:
  - **url : {$DOMAIN}** (lietotāja makro {$DOMAIN} ir jābūt definētam, vēlams hosta līmenī)
  - **subpage : /release_notes**



- Laukā *Script* ievadiet:

```
var obj = JSON.parse(value);
var url = obj.url;
var subpage = obj.subpage;
var request = new HttpRequest();
return request.get(url + subpage);
```

Vairāki HTTP pieprasījumi

Savāciet saturu gan no *https://www.example.com*, gan no *https://www.example.com/release_notes*:

- Izveidojiet vienumu ar tipu ”Script”.
- Laukā *Script* ievadiet:

```
var request = new HttpRequest();
return request.get("https://www.example.com") + request.get("https://www.example.com/release_notes");
```

Žurnalēšana

Pievienojiet Zabbix servera žurnālam ierakstu ”Log test” un saņemiet atpakaļ vienuma vērtību ”1”:

- Izveidojiet vienumu ar tipu ”Script”.
- Laukā *Script* ievadiet:

```
Zabbix.log(3, 'Log test');
return 1;
```

19 Pārlūka vienumi

Pārskats

Pārlūka vienumi ļauj uzraudzīt sarežģītas tīmekļa vietnes un tīmekļa lietojumprogrammas, izmantojot pārlūku.

> **Attention:**
> Pārlūka vienumu atbalsts pašlaik ir eksperimentāls.

Pārlūka vienumi apkopo datus, izpildot lietotāja definētu JavaScript kodu un iegūstot datus, izmantojot HTTP/HTTPS. Šis vienums var simulēt tādas ar pārlūku saistītas darbības kā klikšķināšana, teksta ievade, pārvietošanās pa tīmekļa lapām un citas lietotāja mijiedarbības ar tīmekļa vietnēm vai tīmekļa lietojumprogrammām.

Papildus skriptam var norādīt neobligātu parametru sarakstu (nosaukuma un vērtības pārus) un noildzi.

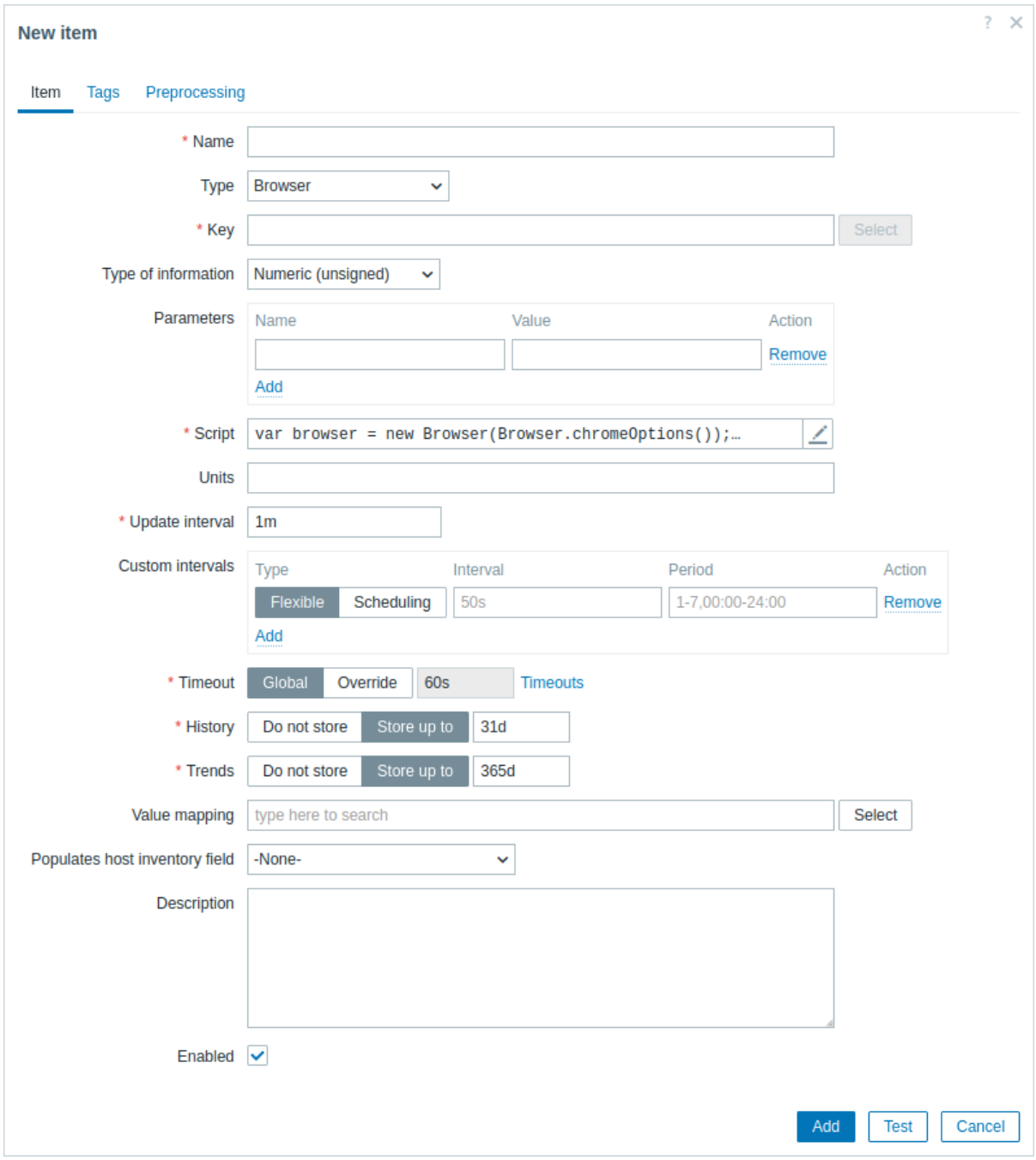
Attention:
Vienums daļēji ievieš W3C WebDriver standartu, izmantojot vai nu Selenium Server, vai parastu WebDriver (piemēram, ChromeDriver) kā tīmekļa testēšanas galapunktu. Lai vienums darbotos, iestatiet galapunktu Zabbix serveris/starpniekserveris konfigurācijas faila parametrā `WebDriverURL` (ja izmantojat ChromeDriver, skatiet Drošības apsvērumi). Labākai veiktspējai apsveriet iespēju izmantot atsevišķu serveri tīmekļa testēšanas videi.

Pārlūka vienumu pārbaudes izpilda un apstrādā Zabbix servera vai starpniekservera pārlūka aptaujātāji. Ja nepieciešams, varat pielāgot iepriekš atzaroto pārlūka aptaujātāju instanču skaitu Zabbix serveris/starpniekserveris konfigurācijas faila parametrā `StartBrowserPollers`.

Sarežģītu tīmekļa vietņu un tīmekļa lietojumprogrammu uzraudzībai ir pieejama veidne Website by Browser kā gatava veidne.

Konfigurācija

Laukā *Type* vienuma konfigurācijas formā atlasiet Browser un pēc tam aizpildiet obligātos laukus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lauki, kuriem Browser vienumiem ir nepieciešama specifiska informācija, ir šādi:

Lauks	Apraksts
Key	Ievadiet unikālu atslēgu, kas tiks izmantota vienuma identificēšanai.
Parameters	Norādiet mainīgos, kas tiks nodoti skriptam kā atribūta un vērtības pāri. Tiek atbalstīti lietotāja makrosi. Lai redzētu, kuri iebūvētie makrosi tiek atbalstīti, meklējiet "Browser-type item" atbalstīto makrosu tabulā.
Script	Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam. Šim kodam jānodrošina loģika metrikas vērtības atgriešanai. Kodam ir piekļuve visiem parametriem, visiem Zabbix pievienotajiem papildu JavaScript objektiem un Browser vienuma JavaScript objektiem. Skatiet arī: JavaScript Guide.
Timeout	JavaScript izpildes noildze (1-600s; tās pārsniegšana atgriezīs kļūdu). Ņemiet vērā, ka atkarībā no skripta var būt nepieciešams ilgāks laiks, līdz noildze nostrādā. Papildinformāciju par parametru *Timeout* skatiet sadaļā vispārīgie vienuma atribūti.

Piemēri

Lai skatītu piemēru, kā iestatīt Zabbix vietņu uzraudzībai, izmantojot veidni *Website by Browser*, skatiet Vietņu uzraudzība ar Browser vienumiem.

Noklusējuma skripts

Šis skripts:

1. Inicializē pārlūka sesiju.
2. Pāriet uz norādīto URL.
3. Apkopo veiktspējas ierakstus un sesijas statistiku un atgriež tos kā JSON virkni.

Laukā *Skripts* ievadiet:

```
var browser = new Browser(Browser.chromeOptions());

try {
    browser.navigate("http://example.com");
    browser.collectPerfEntries();
}
finally {
    return JSON.stringify(browser.getResult());
}
```

Inicializēt pārlūku ar pielāgotām iespējām

Šis skripts:

1. Inicializē pārlūka sesiju pieejamajam pārlūkam, pamatojoties uz pirmo atbilstošo pārlūku skriptā norādītajā secībā.
2. Definē pārlūka iespējas, tostarp lapas ielādes stratēģiju un katram pārlūkam specifiskas opcijas, piemēram, bezgalvas režīmu Chrome, Firefox un Microsoft Edge pārlūkiem.

Laukā *Script* ievadiet:

```
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"
            }
        ]
    }
});
```

Inicializēt pārlūku ar GUI

Pēc noklusējuma pārlūka sesijas (izņemot Safari) tiek inicializētas bezgalvas režīmā, kas nozīmē, ka pārlūka grafiskā lietotāja saskarne (GUI) netiek parādīta.

Tālāk redzamais skripts inicializē pārlūka sesiju ar iespējotu GUI.

Ņemiet vērā, ka, ja WebDriver nevar atrast pārlūka bināro failu, varat norādīt ceļu manuāli.

```
var opts = Browser.chromeOptions();
opts.capabilities.alwaysMatch['goog:chromeOptions'].args = [];

// Lai inicializētu Firefox sesiju ar GUI, noņemiet komentāru no tālāk norādītajām rindām:
// var opts = Browser.firefoxOptions();
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].binary = 'usr/bin/firefox';
// opts.capabilities.alwaysMatch['moz:firefoxOptions'].args = [];

// Lai inicializētu Microsoft Edge sesiju ar GUI, noņemiet komentāru no tālāk norādītajām rindām:
// 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:**
> Ja jūsu testi tiek izpildīti attālā serverī vai konteinerā, varat izmantot Virtual Network Computing (VNC) klientu, lai izveidotu savienojumu ar iekārtas VNC serveri. Tas ļauj attālināti skatīt pārlūka GUI un ar to mijiedarboties.

Uzņemt ekrānuzņēmumus

Tālāk redzamais skripts:

1. Inicializē pārlūka sesiju.
2. Iestata pārlūka skata apgabala izmēru, lai noteiktu ekrānuzņēmuma izmēru (norādīts kā parametri, skatiet zemāk).
3. Atver URL (norādīts kā parametrs, skatiet zemāk).
4. Apkopo sesijas statistiku, uzņem ekrānuzņēmumu un pievieno to apkopotajai statistikai.
5. Apstrādā kļūdas, uztverot kļūdu ziņojumus un ekrānuzņēmumu.
6. Atgriež apkopotos rezultātus kā JSON virkni.

Skripts izmanto arī parametrus no vienuma konfigurācijas formas:

- webURL - http://example.com
- width - 1920
- height - 1080

Laukā *Script* ievadiet:

```
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);
}
```

Pārbaudīt Zabbix pieteikšanos

Šis skripts:

1. Inicializē pārlūka sesiju.
2. Atver lapu (norādīta kā parametrs, skatiet zemāk).
3. Ievada lietotājvārdu un paroli (norādīti kā parametri, skatiet zemāk).
4. Atrod un noklikšķina uz pieteikšanās pogas.
5. Atrod un noklikšķina uz atteikšanās pogas.
6. Apkopo veiktspējas datus pirms pieteikšanās un pēc tās, kā arī pēc atteikšanās.
7. Apstrādā kļūdas, iegūstot kļūdu ziņojumus un ekrānuzņēmumu.
8. Atgriež apkopotos rezultātus kā JSON virkni.

Skripts izmanto arī parametrus no vienuma konfigurācijas formas:

- webURL - http://{HOST.CONN}/index.php
- username - {$USERNAME}
- password - {$PASSWORD}

Laukā *Script* ievadiet:

```
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);
}
```

Saišu atrašana

Šis skripts:

1. Inicializē pārlūka sesiju.
2. Definē funkciju dublējošos elementu noņemšanai no masīva (skatiet 5. soli).
3. Pāriet uz lapu (norādīta kā parametri, skatiet zemāk).
4. Atrod saites lapā.
5. Noņem dublējošās saites, lai nodrošinātu to unikalitāti.
6. Izgūst tikai tās saites, kas sākas ar "http".
7. Formatē izgūtās saites noteiktā struktūrā.
8. Apstrādā kļūdas, uztverot kļūdu ziņojumus un ekrānuzņēmumu.
9. Atgriež savāktos rezultātus kā JSON virkni.

Skripts izmanto arī parametrus no vienuma konfigurācijas formas:

- scheme - {$WEBSITE.SCHEME}
- domain - {$WEBSITE.DOMAIN}
- path - {$WEBSITE.PATH}

Laukā *Script* ievadiet:

```
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 Vēsture un tendences

Pārskats

Vēsture un tendences ir divi veidi, kā Zabbix glabā savāktos datus.

Kamēr vēsture saglabā katru savākto vērtību, tendences saglabā vidējo informāciju pa stundām, un tāpēc tās patērē mazāk resursu.

Vēstures saglabāšana

Varat iestatīt, cik dienas tiks saglabāta vēsture:

- vienuma rekvizītu formā
- veicot vienumu masveida atjaunināšanu
- iestatot housekeeper uzdevumus

Visi vecāki dati tiks noņemti ar housekeeper palīdzību.

Vispārīgs un stingrs ieteikums ir glabāt vēsturi pēc iespējas mazāku dienu skaitu, lai tādējādi nepārslogotu datubāzi ar lielu vēsturisko vērtību apjomu.

Tā vietā, lai glabātu garu vēsturi, varat ilgāk saglabāt tendenču datus. Piemēram, varat glabāt vēsturi 14 dienas un tendences 5 gadus.

Labu priekšstatu par to, cik daudz vietas nepieciešams vēstures un tendenču datiem, var iegūt, skatot datubāzes izmēra lapu.

Saglabājot īsāku vēsturi, jūs joprojām varēsiet pārskatīt vecākus datus grafikos, jo grafiki vecāku datu attēlošanai izmantos tendenču vērtības.

> **Attention:**
> Ja vēsture ir iestatīta uz '0', vienums atjauninās tikai atkarīgos vienumus un inventāru. Netiks novērtētas nekādas trigeru funkcijas, jo trigeru novērtēšana balstās tikai uz vēstures datiem.

Note:
Kā alternatīvu veidu vēstures saglabāšanai apsveriet vēstures eksportēšanas funkcionalitātes izmantošanu ielādējamos moduļos.

Tendenču glabāšana

Tendences ir iebūvēts vēsturisko datu samazināšanas mehānisms, kas skaitliskajiem datu tipiem katrai stundai saglabā minimālo, maksimālo, vidējo vērtību un kopējo vērtību skaitu.

Varat iestatīt, cik dienas tendences tiks glabātas:

- vienuma rekvizītu formā
- veicot vienumu masveida atjaunināšanu
- konfigurējot Housekeeper uzdevumus

Tendences parasti var glabāt daudz ilgāk nekā vēsturi. Visi vecākie dati tiks noņemti ar housekeeper palīdzību.

Zabbix serveris darbības laikā uzkrāj tendenču datus tendenču kešatmiņā, datiem ienākot sistēmā. Serveris ieraksta katra vienuma **iepriekšējās stundas** tendences datubāzē (kur tās var atrast lietotāja saskarne) šādās situācijās:

- serveris saņem vienuma pirmo pašreizējās stundas vērtību
- līdz pašreizējās stundas beigām ir atlikušas 5 vai mazāk minūtes, un joprojām nav saņemtas vienuma pašreizējās stundas vērtības
- serveris tiek apturēts

Lai grafikā redzētu tendences, jāgaida vismaz līdz nākamās stundas sākumam (ja vienums tiek atjaunināts bieži) un ne ilgāk kā līdz nākamās stundas beigām (ja vienums tiek atjaunināts reti), tas ir, maksimums 2 stundas.

Kad serveris ieraksta tendenču kešatmiņu un datubāzē šai stundai jau ir tendences (piemēram, serveris ir pārstartēts stundas vidū), serverim vienkāršu ievietošanas vaicājumu vietā jāizmanto atjaunināšanas vaicājumi. Tāpēc lielākās instalācijās, ja nepieciešama pārstartēšana, ir vēlams apturēt serveri vienas stundas beigās un palaist to nākamās stundas sākumā, lai izvairītos no tendenču datu pārklāšanās.

Vēstures tabulas nekādā veidā nepiedalās tendenču ģenerēšanā.

Attention:
Ja tendencēm ir iestatīta vērtība '0', Zabbix serveris vispār neaprēķina un nesaglabā tendences.

Note:
Tendences tiek aprēķinātas un saglabātas ar tādu pašu datu tipu kā sākotnējās vērtības. Tādēļ unsigned datu tipa vērtību vidējo vērtību aprēķini tiek noapaļoti, un jo mazāks ir vērtību intervāls, jo mazāk precīzs būs rezultāts. Piemēram, ja vienumam ir vērtības 0 un 1, vidējā vērtība būs 0, nevis 0.5.
Turklāt servera pārstartēšana var izraisīt unsigned datu tipa vidējo vērtību aprēķinu precizitātes zudumu pašreizējai stundai.

5 Lietotāja parametri

Pārskats

Dažkārt var būt nepieciešams palaist aģenta pārbaudi, kas nav iepriekš definēta Zabbix. Šādos gadījumos noder lietotāja parametri.

Jūs varat uzrakstīt komandu, kas iegūst nepieciešamos datus, un iekļaut to lietotāja parametrā aģenta konfigurācijas failā (konfigurācijas parametrs 'UserParameter').

Lietotāja parametram ir šāda sintakse:

```
UserParameter=<key>,<command>
```

Kā redzams, lietotāja parametrs satur arī atslēgu. Atslēga būs nepieciešama, konfigurējot vienumu. Ievadiet paša izvēlētu atslēgu, uz kuru būs viegli atsaukties (tai jābūt unikālai hosta ietvaros).

Restartējiet aģentu vai izmantojiet aģenta izpildlaika vadības opciju, lai ielādētu jauno parametru, piemēram:

```
zabbix_agentd -R userparameter_reload
```

Pēc tam, konfigurējot vienumu, ievadiet atslēgu, lai atsauktos uz komandu no lietotāja parametra, kuru vēlaties izpildīt.

Lietotāja parametri ir komandas, ko izpilda Zabbix aģents. Ņemiet vērā, ka pirms vienuma vērtības priekšapstrādes soļiem var tikt atgriezti līdz 16 MB datu.

UNIX operētājsistēmās kā komandrindas interpretators tiek izmantots **/bin/sh**. Lietotāja parametri ievēro aģenta pārbaudes noildzi; ja tiek sasniegta noildze, atzarotais lietotāja parametra process tiek pārtraukts.

Skatiet arī:

- Soli pa solim pamācību par lietotāja parametru izmantošanu
- Komandu izpilde

Vienkāršu lietotāja parametru piemēri

Vienkārša komanda:

```
UserParameter=ping,echo 1
```

Aģents vienmēr atgriezīs '1' vienumam ar atslēgu 'ping'.

Sarežģītāks piemērs:

```
UserParameter=mysql.ping,mysqladmin -uroot ping | grep -c alive
```

Aģents atgriezīs '1', ja MySQL serveris darbojas, pretējā gadījumā — '0'.

Elastīgi lietotāja parametri

Elastīgi lietotāja parametri pieņem parametrus ar atslēgu. Šādā veidā elastīgs lietotāja parametrs var būt par pamatu vairāku vienumu izveidei.

Elastīgiem lietotāja parametriem ir šāda sintakse:

```
UserParameter=key[*],command
```

Parametrs	Apraksts
Atslēga	Unikāla vienuma atslēga. [*] norāda, ka šī atslēga pieņem parametrus iekavās. Parametri tiek norādīti, konfigurējot vienumu.
Komanda	Komanda, kas tiks izpildīta, lai noteiktu atslēgas vērtību. *Tikai elastīgiem lietotāja parametriem*: Komandā var izmantot pozicionālās atsauces $1...$9, lai norādītu uz attiecīgo parametru vienuma atslēgā. Zabbix analizē parametrus, kas ietverti vienuma atslēgas [ ], un attiecīgi aizstāj $1,...,$9 komandā. $0 tiks aizstāts ar sākotnējo komandu (pirms $0,...,$9 izvēršanas), kas jāizpilda. Pozicionālās atsauces tiek interpretētas neatkarīgi no tā, vai tās ir ietvertas dubultajās (") vai vienkāršajās (') pēdiņās. Lai izmantotu pozicionālās atsauces bez izmaiņām, norādiet dubultu dolāra zīmi — piemēram, awk '{print $$2}'. Šajā gadījumā $$2, izpildot komandu, faktiski pārvērtīsies par $2.

Attention:
Pozicionālās atsauces ar $ zīmi Zabbix aģents meklē un aizstāj tikai elastīgajiem lietotāja parametriem. Vienkāršajiem lietotāja parametriem šāda atsauču apstrāde tiek izlaista, un tādēļ $ zīmes ekrānēšana nav nepieciešama.

Attention:
Noteiktas rakstzīmes lietotāja parametros pēc noklusējuma nav atļautas. Pilnu rakstzīmju sarakstu skatiet UnsafeUserParameters.

Piemērs 1

Kaut kas ļoti vienkāršs:

```
UserParameter=ping[*],echo $1
```

Mēs varam definēt neierobežotu skaitu vienumu uzraudzībai, kuriem visiem ir formāts ping[kaut kas].

- ping[0] - vienmēr atgriezīs '0'
- ping[aaa] - vienmēr atgriezīs 'aaa'

Piemērs 2

Pievienosim vairāk jēgas!

```
UserParameter=mysql.ping[*],mysqladmin -u$1 -p$2 ping | grep -c alive
```

Šo parametru var izmantot MySQL datubāzes pieejamības uzraudzībai. Mēs varam nodot lietotājvārdu un paroli:

```
mysql.ping[zabbix,our_password]
```

Piemērs 3

Cik rindu failā atbilst regulārai izteiksmei?

```
UserParameter=wc[*],grep -c "$2" $1
```

Šo parametru var izmantot, lai aprēķinātu rindu skaitu failā.

```
wc[/etc/passwd,root]
wc[/etc/services,zabbix]
```

Komandas rezultāts

Komandas atgrieztā vērtība ir standarta izvade kopā ar standarta kļūdu, ko ģenerē komanda.

> **Attention:**
> Vienums, kas atgriež tekstu (rakstzīmju, žurnāla vai teksta tipa informāciju), nekļūs neatbalstīts standarta kļūdas izvades gadījumā.

Atgrieztā vērtība ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas); piemērojami arī datu bāzes ierobežojumi.

Lietotāja parametri, kas atgriež tekstu (rakstzīmju, žurnāla vai teksta tipa informāciju), var atgriezt arī atstarpi. Nederīga rezultāta gadījumā vienums kļūs neatbalstīts.

1 Zabbix aģentu paplašināšana

Šajā pamācībā ir sniegtas soli pa solim instrukcijas par to, kā paplašināt Zabbix aģenta funkcionalitāti, izmantojot lietotāja parametru.

1. solis

Uzrakstiet skriptu vai komandrindu, lai iegūtu nepieciešamo parametru.

Piemēram, mēs varam uzrakstīt šādu komandu, lai iegūtu MySQL servera izpildīto vaicājumu kopējo skaitu:

```
mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

Izpildot šo komandu, tā atgriež SQL vaicājumu kopējo skaitu.

2. solis

Pievienojiet komandu failam zabbix_agentd.conf:

```
UserParameter=mysql.questions,mysqladmin -uroot status | cut -f4 -d":" | cut -f1 -d"S"
```

mysql.questions ir unikāls identifikators. Tas var būt jebkurš derīgs atslēgas identifikators, piemēram, *queries*.

Pārbaudiet šo parametru, izmantojot Zabbix aģents ar karodziņu "-t" (tomēr, ja tas tiek darbināts root lietotāja režīmā, ņemiet vērā, ka aģentam var būt atšķirīgas atļaujas, kad tas tiek palaists kā dēmons):

```
zabbix_agentd -t mysql.questions
```

3. solis

Pārlādējiet lietotāja parametrus no konfigurācijas faila, izpildot:

```
zabbix_agentd -R userparameter_reload
```

Varat arī restartēt aģentu, nevis izmantot izpildlaika vadības komandu.

Pārbaudiet parametru, izmantojot utilītu zabbix_get.

4. solis

Pievienojiet uzraudzītajam hostam jaunu vienumu ar atslēgu Key=mysql.questions. Vienuma tipam jābūt vai nu Zabbix aģents, vai Zabbix aģents (aktīvs).

Ņemiet vērā, ka atgriezto vērtību tipam Zabbix serverī jābūt iestatītam pareizi. Pretējā gadījumā Zabbix tās nepieņems.

6 Windows veiktspējas skaitītāji

Pārskats

Jūs varat efektīvi uzraudzīt Windows veiktspējas skaitītājus, izmantojot atslēgu *perf_counter[]*.

Piemēram:

```
perf_counter["\Processor(0)\Interrupts/sec"]
```

vai

```
perf_counter["\Processor(0)\Interrupts/sec", 10]
```

Plašāku informāciju par šīs atslēgas vai tās tikai angļu valodā pieejamā ekvivalenta `perf_counter_en` izmantošanu skatiet sadaļā Windows specifiskās vienumu atslēgas.

Lai iegūtu pilnu uzraudzībai pieejamo veiktspējas skaitītāju sarakstu, varat palaist:

```
typeperf -qx
```

Varat arī izmantot zema līmeņa atklāšanu, lai atklātu vairākas Windows veiktspējas skaitītāju objektu instances un automatizētu perf_counter vienumu izveidi vairāku instanču objektiem.

Skaitliskā attēlošana

Windows uztur objektu un veiktspējas skaitītāju nosaukumu skaitliskās attēlošanas formas (indeksus). Zabbix atbalsta šīs skaitliskās attēlošanas formas kā parametrus `perf_counter`, `perf_counter_en` vienumu atslēgām un `PerfCounter`, `PerfCounterEn` konfigurācijas parametros.

Tomēr nav ieteicams tās izmantot, ja vien nevarat garantēt, ka jūsu skaitliskie indeksi konkrētos hostos atbilst pareizajām virknēm. Ja jums ir jāizveido pārnesami vienumi, kas darbojas dažādos hostos ar dažādām lokalizētām Windows versijām, varat izmantot `perf_counter_en` atslēgu vai `PerfCounterEn` konfigurācijas parametru, kas ļauj izmantot nosaukumus angļu valodā neatkarīgi no sistēmas lokalizācijas.

Lai noskaidrotu skaitliskos ekvivalentus, palaidiet **regedit**, pēc tam atrodiet *Counter* sadaļā *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009*.

Reģistra ieraksts satur šādu informāciju:

```
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
....
```

Šeit varat atrast atbilstošos skaitļus katrai veiktspējas skaitītāja virknes daļai, piemēram, '\System\% Processor Time':

```
System → 2
% Processor Time → 6
```

Pēc tam varat izmantot šos skaitļus, lai attēlotu ceļu skaitliskā formā:

```
\2\6
```

Veiktspējas skaitītāju parametri

Jūs varat izvietot dažus PerfCounter parametrus Windows veiktspējas skaitītāju uzraudzībai.

Piemēram, varat pievienot šos ierakstus Zabbix aģents konfigurācijas failam:

```
PerfCounter=UserPerfCounter1,"\Memory\Page Reads/sec",30
or
PerfCounter=UserPerfCounter2,"\4\24",30
```

Ja šādi parametri ir iestatīti, pēc tam varat vienkārši izmantot *UserPerfCounter1* vai *UserPerfCounter2* kā atslēgas attiecīgo vienumu izveidei.

Atcerieties pēc izmaiņu veikšanas konfigurācijas failā restartēt Zabbix aģents.

7 Masveida atjaunināšana

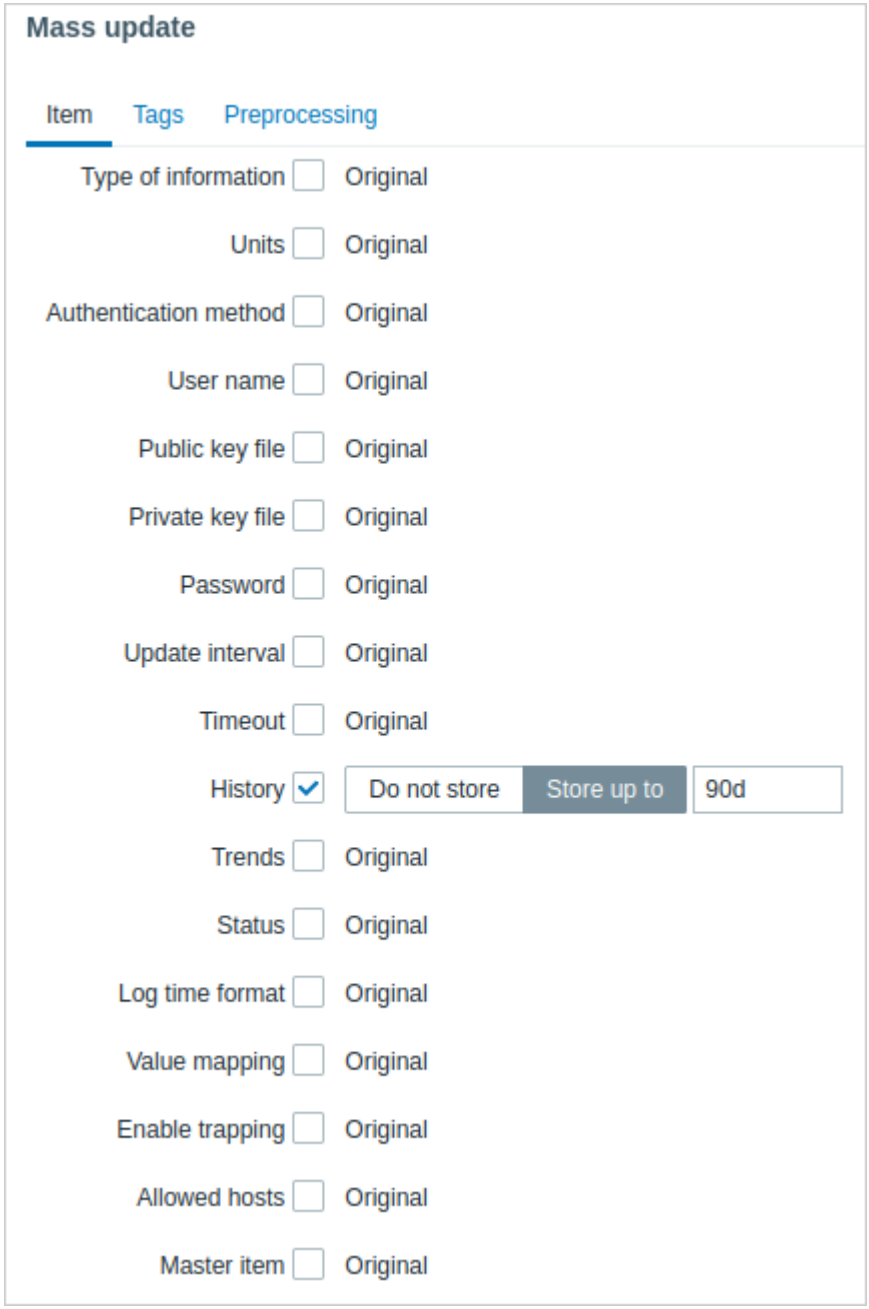
Pārskats

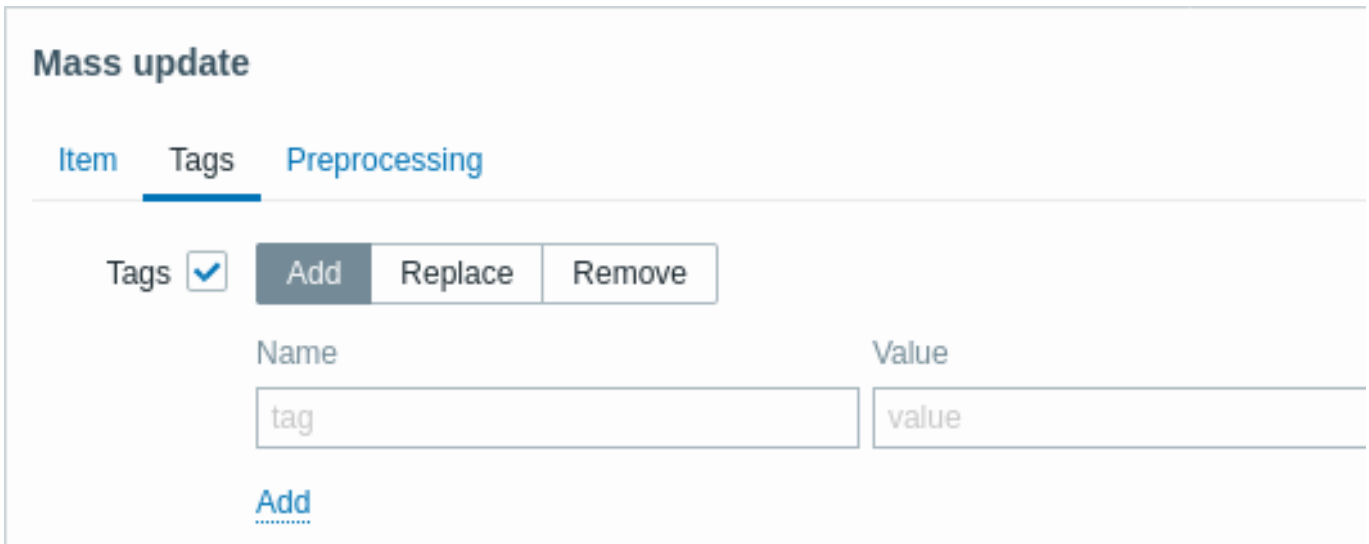
Dažkārt var būt nepieciešams vienlaikus mainīt kādu atribūtu vairākiem vienumiem. Tā vietā, lai atvērtu katru atsevišķo vienumu rediģēšanai, šim nolūkam varat izmantot masveida atjaunināšanas funkciju.

Masveida atjaunināšanas izmantošana

Lai masveidā atjauninātu dažus vienumus, rīkojieties šādi:

- Sarakstā atzīmējiet atjaunināmo vienumu izvēles rūtiņas
- Zem saraksta noklikšķiniet uz *Mass update*
- Pārejiet uz cilni ar nepieciešamajiem atribūtiem (*Vienums*, *Tags* vai *Preprocessing*)
- Atzīmējiet atjaunināmo atribūtu izvēles rūtiņas
- Ievadiet atribūtiem jaunās vērtības



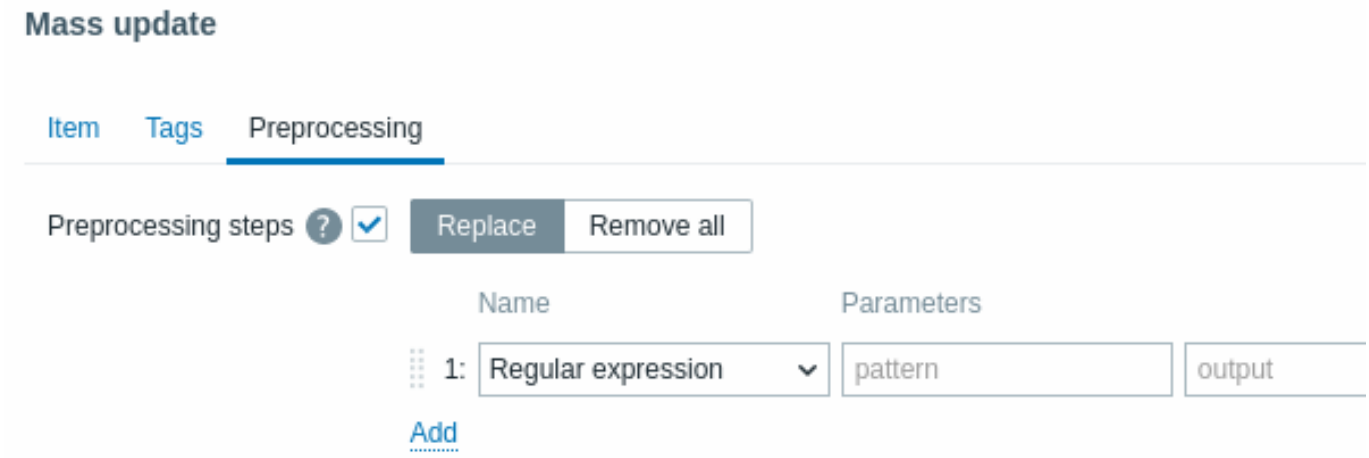


Opcija *Tags* ļauj:

- *Add* - pievienot norādītās birkas vienumiem (birkas ar vienādu nosaukumu, bet atšķirīgām vērtībām netiek uzskatītas par ”dublikātiem”, un tās var pievienot tam pašam vienumam).
- *Replace* - noņemt norādītās birkas un pievienot birkas ar jaunām vērtībām

- *Remove* - noņemt norādītās birkas no vienumiem

Birkās tiek atbalstīti lietotāja makrosi, {INVENTORY.*} makrosi, {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} makrosi.



Atzīmējiet izvēles rūtiņu blakus *Preprocessing steps*, lai aktivizētu priekšapstrādes soļu masveida atjaunināšanu.

Preprocessing masveida atjaunināšana ļauj:

- *Replace* - aizstāt esošos vienumu priekšapstrādes soļus ar tālāk norādītajiem priekšapstrādes soļiem
- *Remove all* - noņemt visus priekšapstrādes soļus no vienumiem

Kad pabeigts, noklikšķiniet uz *Update*.

8 Vērtību kartēšana

Pārskats

Vērtību kartēšana ļauj konfigurēt saņemto vērtību lietotājam draudzīgāku attēlojumu, izmantojot kartēšanu starp skaitliskām/teksta vērtībām un teksta attēlojumiem.

Piemēram, ja vienuma vērtība ir "0" vai "1", vērtību kartēšanu var izmantot, lai šīs vērtības attēlotu lietotājam saprotamākā veidā:

- 0 → Nav pieejams
- 1 → Pieejams

Vērtību kartēšanu datu dublējumkopiju tipiem var konfigurēt šādi:

- F → Pilna
- D → Diferenciālā
- I → Inkrementālā

Sprieguma vērtību diapazonus var konfigurēt šādi:

- <=209 → Zems
- 210-230 → Labi
- >=231 → Augsts

Vērtību kartēšana tiek izmantota Zabbix lietotāja saskarnē un paziņojumos, ko nosūta mediju tipi.

Attention:
Saņemtās vērtības aizstāšana ar konfigurēto attēlojumu tiek veikta gan Zabbix lietotāja saskarnē, gan serverī; tomēr serveris aizstāšanu veic tikai šādos gadījumos:

- kad tiek aizpildīts hosta inventārs;
- izvēršot atbalstītos makrosus {ITEM.VALUE}, {ITEM.LASTVALUE}, {EVENT.OPDATA} un {EVENT.CAUSE.OPDATA}.

Vērtību kartēšana tiek iestatīta veidnēs vai hostos. Pēc konfigurēšanas tā ir pieejama visiem vienumiem attiecīgajā veidnē vai hostā. Kad konfigurējat vienumus, parametrā *Value mapping* norādiet iepriekš konfigurētas vērtību kartēšanas nosaukumu.

Note:
Vērtību karšu mantošana netiek atbalstīta — hosti un veidnes nemanto vērtību kartēšanu no piesaistītajām veidnēm. Veidnes vienumi hostā turpinās izmantot veidnē konfigurēto vērtību kartēšanu.

Note:
Vērtību kartēšanu var izmantot ar vienumiem, kuriem informācijas tips ir *Numeric (unsigned)*, *Numeric (float)* un *Character*.

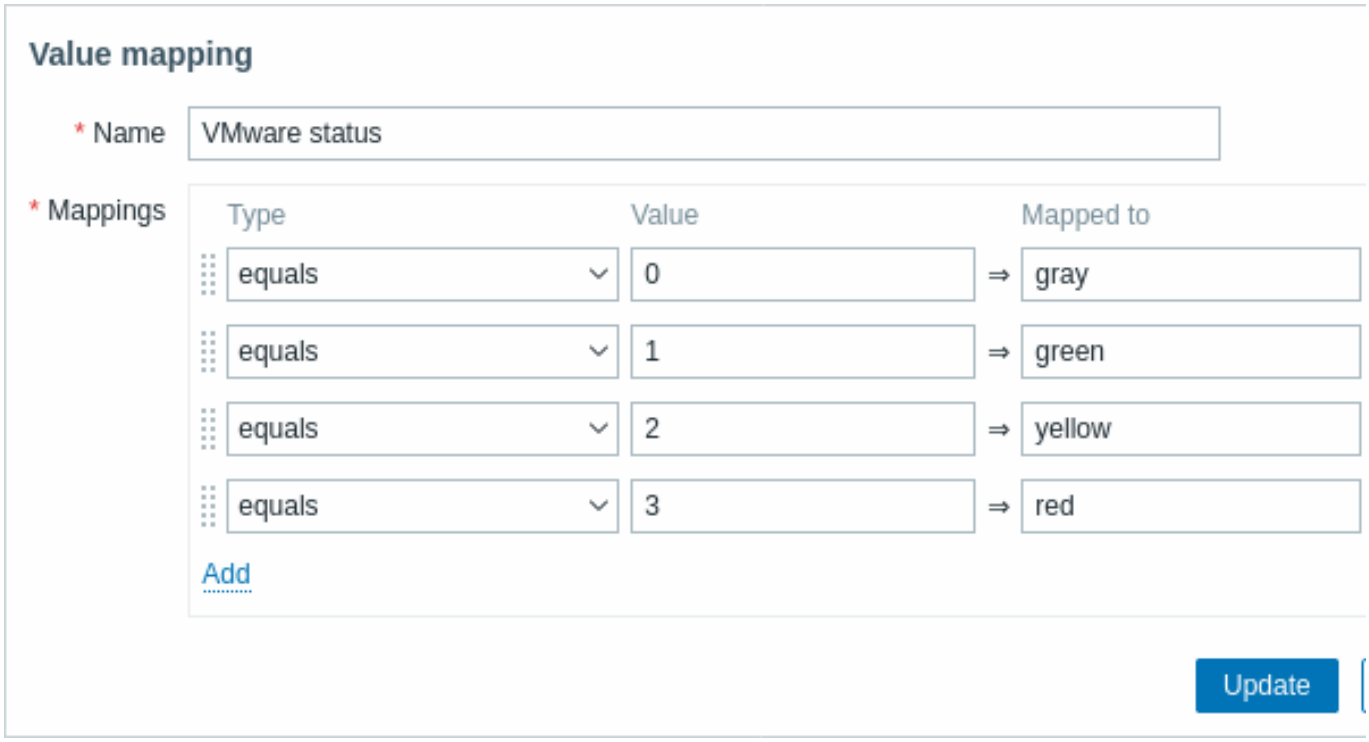
Vērtību kartēšana tiek eksportēta/importēta kopā ar attiecīgo veidni vai hostu. To var arī masveidā atjaunināt, izmantojot hosta un veidnes masveida atjaunināšanas formas.

Konfigurācija

Lai konfigurētu vērtību kartēšanu, izpildiet šīs darbības:

1. Atveriet hosta vai veidnes konfigurācijas formu.

2. Cilnē *Vērtību kartēšana* noklikšķiniet uz *Pievienot*, lai pievienotu jaunu vērtību kartēšanu, vai noklikšķiniet uz esošas kartēšanas nosaukuma, lai to rediģētu.



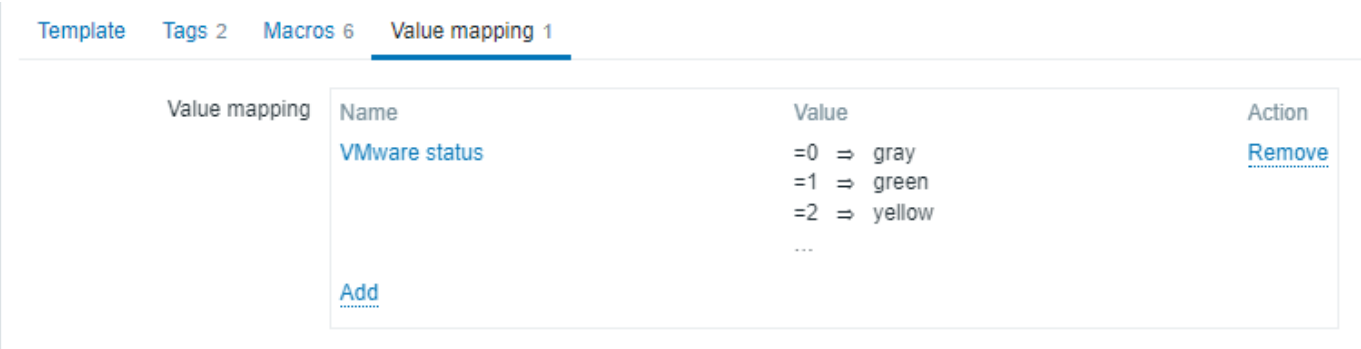
Vērtību kartēšanas parametri:

Parametrs	Apraksts
Nosaukums	Unikāls nosaukums vērtību kartēšanas kopai.
Kartējumi	Atsevišķi noteikumi skaitlisku/teksta vērtību kartēšanai uz teksta attēlojumiem. Kartēšana tiek piemērota noteikumu secībā, ko var mainīt, velkot.
Tips	Kartēšanas tips: **vienāds ar** - tiks kartētas vienādas vērtības; **ir lielāks vai vienāds ar** - tiks kartētas vienādas vai lielākas vērtības; **ir mazāks vai vienāds ar** - tiks kartētas vienādas vai mazākas vērtības; **diapazonā** - tiks kartētas vērtības diapazonā; diapazonu izsaka kā <number1>-<number2> vai <number>; tiek atbalstīti vairāki diapazoni (piemēram, 1-10,101-110,201); **regexp** - tiks kartētas vērtības, kas atbilst regulārajai izteiksmei (globālās regulārās izteiksmes netiek atbalstītas); **noklusējums** - tiks kartētas visas pārējās vērtības, izņemot tās, kurām ir noteikta specifiska kartēšana. Diapazonu kartēšanai tiek atbalstīti tikai skaitlisko vērtību tipi (*ir lielāks vai vienāds ar*, *ir mazāks vai vienāds ar*, *diapazonā*).
Vērtība	Ienākošā vērtība (atkarībā no kartēšanas tipa var saturēt diapazonu vai regulāro izteiksmi).

Parametrs	Apraksts
Kartēts uz	Ienākošās vērtības teksta attēlojums (līdz 64 rakstzīmēm).

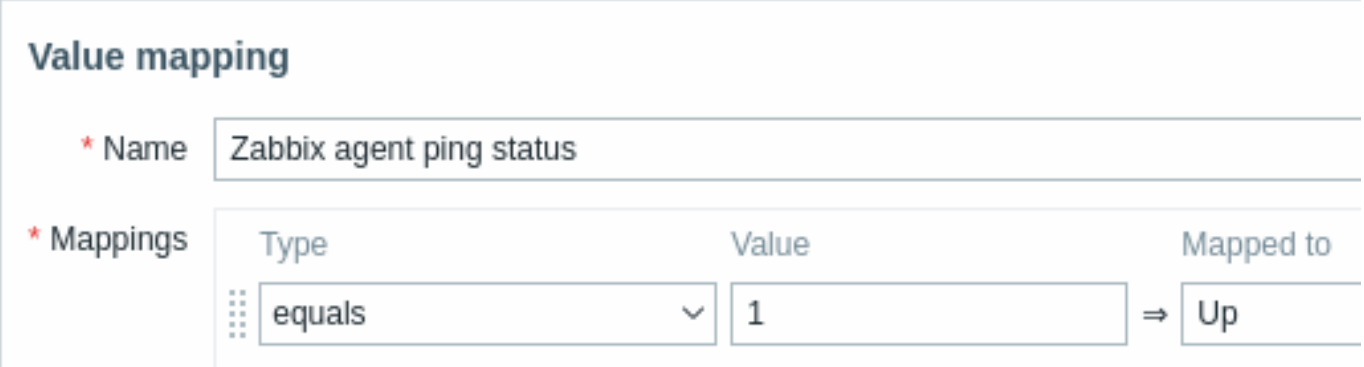
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Skatot vērtību kartēšanu sarakstā, ir redzami tikai pirmie trīs kartējumi, bet trīs punkti norāda, ka pastāv vēl citi kartējumi.



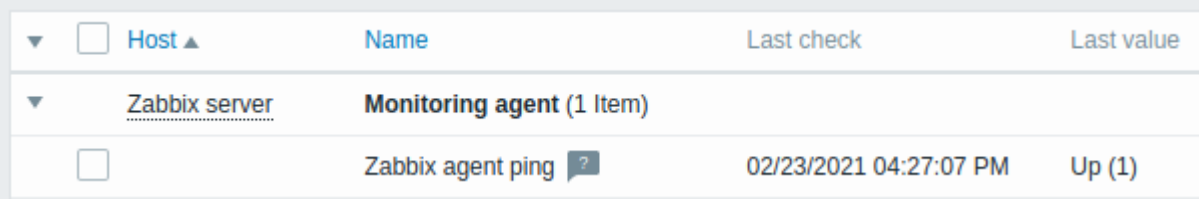
Vērtību kartēšanas piemērs

Viens no iepriekš definētajiem aģenta vienumiem *Zabbix agent ping* izmanto veidnes līmeņa vērtību kartēšanu "Zabbix agent ping status", lai attēlotu savas vērtības.



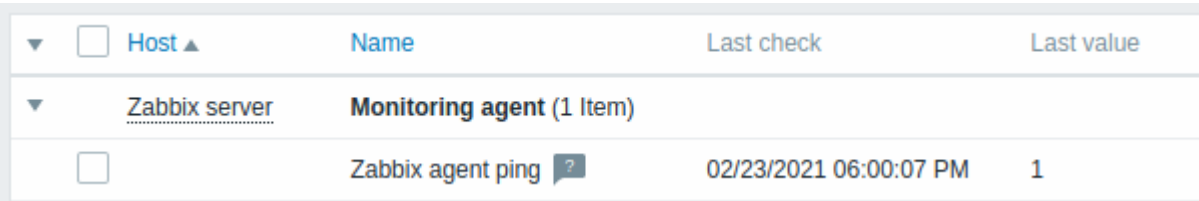
Vienuma konfigurācijas formā laukā *Value mapping* varat atrast atsauci uz šo vērtību kartēšanu:

Šī kartēšana tiek izmantota sadaļā *Monitoring* → *Latest data*, lai attēlotu "Up" (ar neapstrādāto vērtību iekavās).



Note:
Sadaļā *Latest data* attēlotās vērtības tiek saīsinātas līdz 20 simboliem. Ja tiek izmantota vērtību kartēšana, šis saīsinājums netiek piemērots kartētajai vērtībai, bet tikai neapstrādātajai vērtībai (attēlotai iekavās).

Bez iepriekš definētas vērtību kartēšanas jūs redzētu tikai "1", ko varētu būt grūti saprast.



9 Rinda

Pārskats

Rinda attēlo vienumus, kas gaida atsvaidzināšanu. Rinda ir tikai datu **loģisks** attēlojums. Zabbix nav IPC rindas vai cita rindas mehānisma.

Rindā ir iekļauti arī vienumi, kurus uzrauga starpniekserveri — tie tiks uzskaitīti kā rindā esoši starpniekservera vēstures datu atjaunināšanas periodā.

Rindā tiek attēloti tikai vienumi ar ieplānotiem atsvaidzināšanas laikiem. Tas nozīmē, ka no rindas ir izslēgti šādi vienumu tipi:

- aktīvie Zabbix aģenta log, logrt un event log vienumi
- SNMP trap vienumi
- trapper vienumi
- tīmekļa uzraudzības vienumi
- atkarīgie vienumi

Rindā parādītā statistika ir labs Zabbix servera veiktspējas rādītājs.

Rinda tiek iegūta tieši no Zabbix servera, izmantojot JSON protokolu. Informācija ir pieejama tikai tad, ja Zabbix serveris darbojas.

Attention:
Vienumi netiek uzskaitīti rindā, ja vienuma saskarne kļūst nepieejama savienojuma problēmu dēļ vai ja aģents nedarbojas pareizi.

Rindas lasīšana

Lai nolasītu rindu, dodieties uz *Administrēšana* > *Rinda* > *Rindas pārskats*.

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

Attēlā redzamā situācija kopumā ir "ok", tāpēc varam pieņemt, ka serveris darbojas labi.

Rinda parāda dažus vienumus, kas gaida līdz 30 sekundēm. Būtu lieliski zināt, kuri tieši vienumi tie ir.

Lai to izdarītu, virsraksta nolaižamajā izvēlnē atlasiet *Rindas detaļas*. Tagad jūs varat redzēt šo aizkavēto vienumu sarakstu.

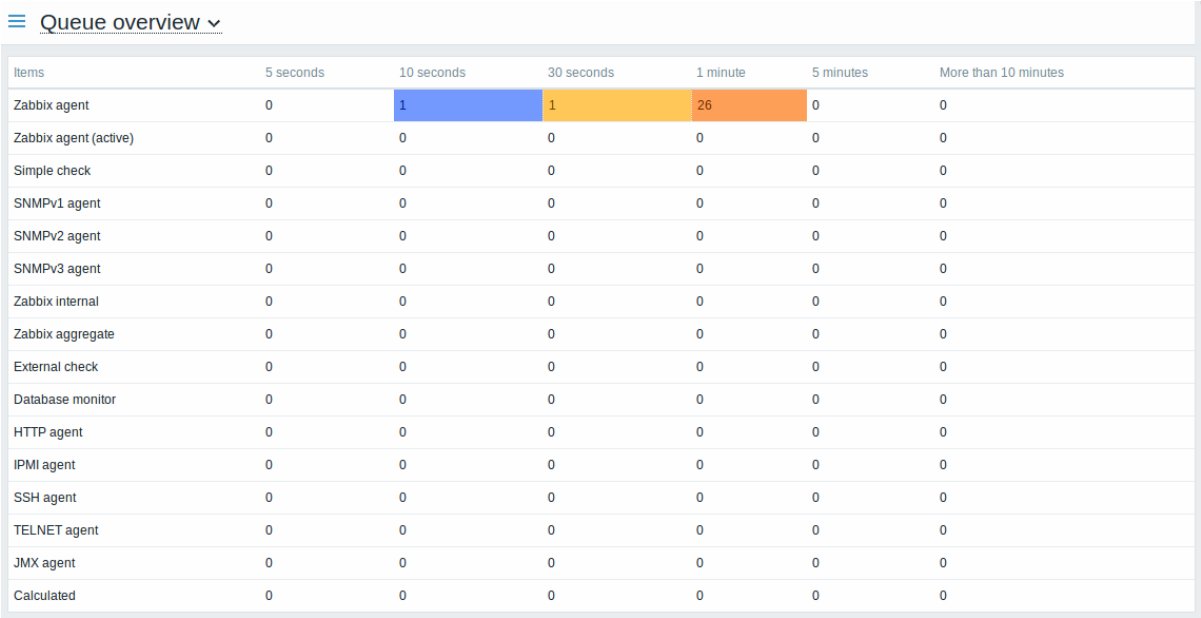
Queue details

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

Ja ir pieejama šī detalizētā informācija, var būt iespējams noskaidrot, kāpēc šie vienumi varētu būt aizkavēti.

Ja ir aizkavēts viens vai divi vienumi, iespējams, nav iemesla satraukumam. Tie var tikt atjaunināti pēc sekundes. Tomēr, ja redzat, ka daudzi vienumi tiek aizkavēti pārāk ilgi, iespējams, pastāv nopietnāka problēma.

Skatiet arī: Laika joslu saskaņošana, izmantojot plānošanas intervālus.



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

Rindas vienums

Īpašu iekšējo vienumu **zabbix[queue,<from>,<to>]** var izmantot, lai uzraudzītu rindas stāvokli Zabbix. Tas atgriezīs to vienumu skaitu, kuru apstrāde kavējas par norādīto laika daudzumu. Plašāku informāciju skatiet sadaļā Iekšējie vienumi.

10 Vērtību kešatmiņa

Pārskats

Lai padarītu trigeru izteiksmju, aprēķināto vienumu un dažu makrosu aprēķināšanu daudz ātrāku, Zabbix serveris atbalsta vērtību kešatmiņas opciju.

Šo atmiņā esošo kešatmiņu var izmantot vēsturisko datu piekļuvei, nevis veikt tiešus SQL izsaukumus uz datubāzi. Ja vēsturiskās vērtības kešatmiņā nav pieejamas, trūkstošās vērtības tiek pieprasītas no datubāzes un kešatmiņa tiek attiecīgi atjaunināta.

Vienumu vērtības paliek vērtību kešatmiņā, līdz notiek kāds no šiem gadījumiem:

- vienums tiek dzēsts (kešatmiņā saglabātās vērtības tiek dzēstas pēc nākamās konfigurācijas sinhronizācijas);
- vienuma vērtība ir ārpus trigera/aprēķinātā vienuma izteiksmē norādītā laika vai skaita diapazona (kešatmiņā saglabātā vērtība tiek noņemta, kad tiek saņemta jauna vērtība);
- trigera/aprēķinātā vienuma izteiksmē norādītais laika vai skaita diapazons tiek mainīts tā, ka aprēķinam ir nepieciešams mazāk datu (nevajadzīgās kešatmiņā saglabātās vērtības tiek noņemtas pēc 24 stundām).

Note:
Vērtību kešatmiņas statusu var novērot, izmantojot servera runtime control opciju `diaginfo` (vai `diaginfo=valuecache`) un pārbaudot sadaļu ar vērtību kešatmiņas diagnostikas informāciju. Tas var būt noderīgi, lai noteiktu nepareizi konfigurētus trigerus vai aprēķinātos vienumus.

Lai iespējotu vērtību kešatmiņas funkcionalitāti, Zabbix servera konfigurācijas failā tiek atbalstīts neobligāts parametrs **ValueCacheSize**.

Vērtību kešatmiņas uzraudzībai tiek atbalstīti divi iekšējie vienumi: **zabbix[vcache,buffer,<mode>]** un **zabbix[vcache,cache,<paramete**
Plašāku informāciju skatiet iekšējos vienumos.

11 Izpildīt tagad

Pārskats

Opcija *Izpildīt tagad* ļauj nekavējoties izpildīt pasīvās pārbaudes.

Vienumu vērtību apkopošana Zabbix sistēmā notiek saskaņā ar konfigurētajiem atjaunināšanas intervāliem. Dažiem vienumiem (piemēram, zema līmeņa atklāšanas kārtulām) ir gari intervāli, un jums var būt nepieciešams nekavējoties iegūt jaunu vērtību, piemēram, lai noteiktu izmaiņas atklājamajos resursos.

Opcija *Izpildīt tagad* tiek atbalstīta šādiem vienumu tipiem:

- Zabbix aģents (pasīvs)
- Vienkāršā pārbaude
- SNMP aģents (v1/v2/v3)
- Zabbix iekšējais
- Ārējā pārbaude
- Datubāzes monitors
- HTTP aģents
- IPMI aģents
- SSH aģents
- TELNET aģents
- JMX aģents
- Aprēķinātais
- Atkarīgais vienums
- Skripts
- Pārlūks

Ja vienuma tips ir ”Atkarīgais vienums”, tad arī tā galvenajam vienumam jābūt vienam no iepriekš minētajiem tipiem.

Konfigurācija

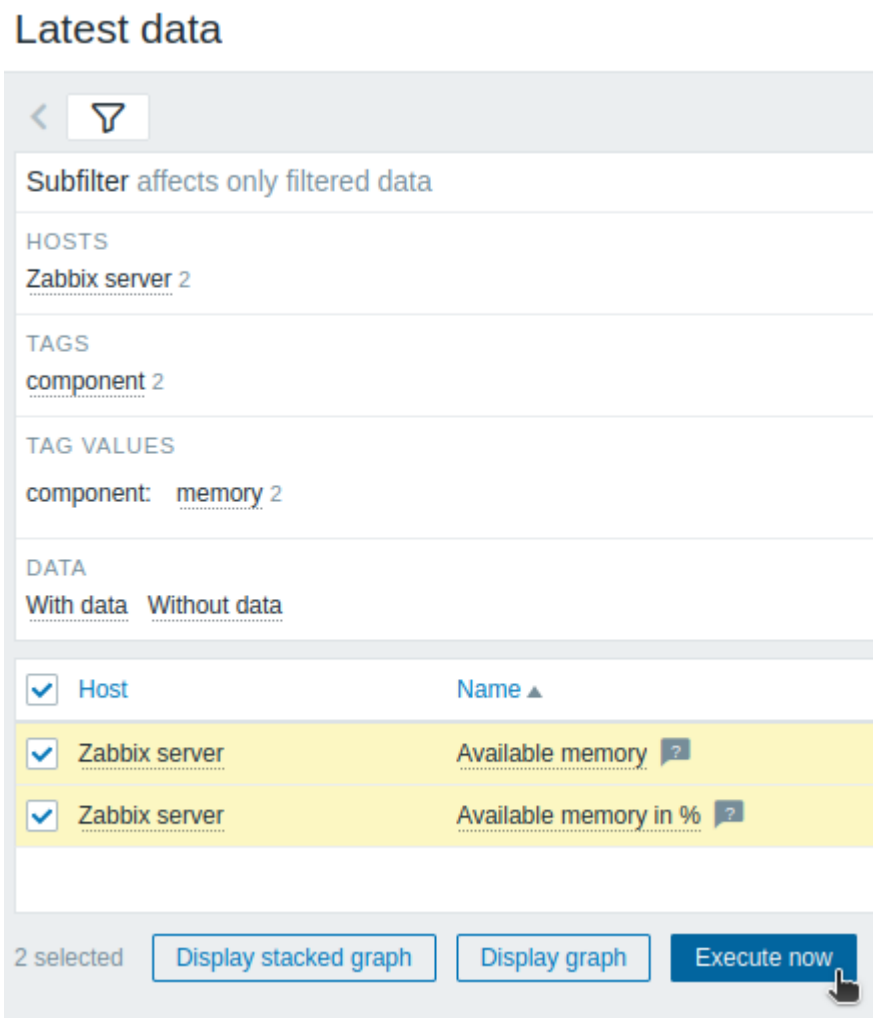
Jūs varat izpildīt pasīvās pārbaudes no:

- *Uzraudzība* > *Jaunākie dati*
- *Datu ievākšana* > *Hosti* > *Vienumi* vai *Atklāšana*
- vienuma izvēlnes
- Rediģējot vienumus vai zema līmeņa atklāšanas kārtulas (to konfigurācijas veidlapās)

Attention:
Pārbaudei jāeksistē konfigurācijas kešatmiņā, lai to varētu izpildīt; skatiet arī CacheUpdateFrequency. Kešatmiņa **netiek atjaunināta** pirms pārbaudes izpildes, tāpēc nesenās izmaiņas vienuma vai zema līmeņa atklāšanas kārtulas konfigurācijā netiks ņemtas vērā. Lai testētu vienumus vai ZLA kārtulas, kas tiek veidotas vai ir tikko izveidotas, tā vietā varat izmantot opciju Test.

Piemēram, lai izpildītu pasīvo pārbaudi sadaļā *Uzraudzība* > *Jaunākie dati*:

1. Atlasiet vienumus; pārbaudi varat izpildīt vairākiem vienumiem vienlaikus.
2. Noklikšķiniet uz *Izpildīt tagad*.



Note:
Sadaļā *Jaunākie dati* lietotāji var *Izpildīt tagad* vienumus hostos, kuriem viņiem ir atļauja *lasīt-rakstīt* vai atļauja *lasīt* un viņu lomai iespējota darbība *Invoke "Execute now" on read-only hosts* darbība.

12 Aģenta pārbaužu ierobežošana

Pārskats

Jūs varat kontrolēt, kuras vienumu atslēgas Zabbix aģents vai agent 2 drīkst vai nedrīkst izmantot, izpildot vienumu pārbaudes, attālās komandas vai skriptus.

Lai to izdarītu, izmantojiet šos aģenta konfigurācijas parametrus, lai definētu atļaušanas/aizliegšanas noteikumus:

- `AllowKey=<pattern>`
- `DenyKey=<pattern>`

`<pattern>` jāietver viena vienuma atslēga un tas atbalsta aizstājējzīmes (`*`). Aizstājējzīme atbilst jebkuram jebkādu rakstzīmju skaitam tās pozīcijā, un to var izmantot, lai saskaņotu vienumu atslēgas vai parametrus (piemēram, `vfs.file.*[*]`).

Attention:
Lai uzlabotu drošību, ieteicams aizstājējzīmju vietā izmantot precīzas vienumu atslēgas. Sīkāku informāciju skatiet sadaļā Atļaušanas/aizliegšanas noteikumu drošība.

Atšķirībā no citiem aģenta konfigurācijas parametriem, varat norādīt neierobežotu skaitu `AllowKey` vai `DenyKey` parametru.

Svarīgas piezīmes

- Visi `system.run` vienumi pēc noklusējuma ir atspējoti (pat ja `DenyKey` ir tukšs), it kā `DenyKey=system.run[*]` būtu iestatīts kā pēdējais noteikums. Tādēļ varat atļaut konkrētus `system.run` vienumus, skaidri neaizliedzot citus `system.run` vienumus.

- Vienumam, kas norādīts AllowKey, jābūt norādītam arī DenyKey (izņemot system.run vienumus); pretējā gadījumā Zabbix aģents **nestartēsies**.
- Ja iespējams, izmantojiet AllowKey, lai atļautu tikai nepieciešamos vienumus un aizliegtu visu pārējo. Dažas atslēgas var tikt ļaunprātīgi izmantotas, lai nolasītu neparedzētus failus, izmantojot ceļa traversēšanu (piemēram, vfs.file.contents["../../../../etc/passwd"]), un jaunās Zabbix aģenta versijās var tikt ieviestas atslēgas, kuras neaptver jūsu DenyKey noteikumi.
- AllowKey un DenyKey konfigurācija neietekmē HostnameItem, HostMetadataItem vai HostInterfaceItem aģenta parametrus.
- Aizliegtie vienumi kļūst neatbalstīti bez jebkādām norādēm vai kļūdu ziņojumiem; piemēram:
  - Zabbix aģenta --print (-p) komandrindas parametrs nerādīs aizliegto vienumu atslēgas.
  - Zabbix aģenta --test (-t) komandrindas parametrs aizliegtajām vienumu atslēgām atgriezīs "Unsupported item key.".
  - Kad žurnalēšana ir aktivizēta (LogRemoteCommands=1), Zabbix aģenta žurnālfailā netiks reģistrētas aizliegtās attālinātās komandas.

Atļaut/aizliegt noteikumu secība

Varat norādīt neierobežotu skaitu AllowKey vai DenyKey noteikumu, taču to secībai ir nozīme.

- Noteikumi tiek izvērtēti pa vienam, no augšas uz leju.
- Kad vienuma atslēga atbilst noteikumam, tā tiek vai nu atļauta, vai aizliegta, un noteikumu izvērtēšana tiek pārtraukta.

Piemēram, izvērtējot vfs.file.contents[/etc/passwd], noteikumi tiek apstrādāti šādi:

```
AllowKey=vfs.file.contents[/tmp/app.log]    # Vienuma atslēgas raksts neatbilst, aģents pāriet pie nākamā
AllowKey=vfs.file.contents[/etc/passwd]     # Vienuma atslēgas raksts atbilst; aģents atļauj vienuma pārba
DenyKey=vfs.file.*[*]                       # Aģents ignorē noteikumu, jo izvērtēšana ir pārtraukta.
```

Šāda noteikumu secība aizliegs vienuma pārbaudi:

```
DenyKey=vfs.file.*[*]                       # Vienuma atslēgas raksts atbilst; aģents aizliedz vienuma pār
AllowKey=vfs.file.contents[/etc/passwd]     # Aģents ignorē noteikumu, jo izvērtēšana ir pārtraukta.
AllowKey=vfs.file.contents[/tmp/app.log]    # Aģents ignorē noteikumu, jo izvērtēšana ir pārtraukta.
```

Piemēri

Tālāk sniegtie piemēri parāda bieži izmantotus AllowKey un DenyKey konfigurācijas modeļus.

Atļaut konkrētas pārbaudes un komandas

Atļaut tikai divas vfs.file vienumu pārbaudes un divas system.run komandas:

```
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:
Iestatīt DenyKey=system.run[*] nav nepieciešams, jo visas pārējās system.run komandas pēc noklusējuma ir aizliegtas.

Skriptu atļaušana

Atļaujiet Zabbix aģentam izpildīt skriptus hostos, izmantojot visas pieejamās metodes:

- Globālie skripti, kurus var izpildīt lietotāja saskarnē vai izmantojot API (šī metode vienmēr izmanto system.run[myscript.sh] atslēgu)
- Attālinātās komandas no darbību operācijām (šī metode vienmēr izmanto system.run[myscript.sh,nowait] atslēgu)
- system.run Zabbix aģenta vienumi ar skriptu, piemēram:
  - system.run[myscript.sh]
  - system.run[myscript.sh,wait]
  - system.run[myscript.sh,nowait]

```
AllowKey=system.run[myscript.sh,*]
```

Lai kontrolētu wait/nowait parametru, jāiestata cits noteikums. Piemēram, varat atļaut tikai system.run[myscript.sh,wait] vienumus, tādējādi izslēdzot citas metodes:

```
AllowKey=system.run[myscript.sh,wait]
```

Allow/deny noteikumu drošināšana

Šis piemērs parāda, kā padarīt drošākus pārāk visatļaujošus AllowKey vai DenyKey noteikumus.

Apsveriet šādus noteikumus:

```
AllowKey=system.run["C:\Program^ Files\Zabbix^ Agent^ 2\scripts\test.bat*"]
DenyKey=vfs.file.*
DenyKey=system.cpu.load[*]
```

> **Note:**
> Sistēmā Windows atstarpes ceļā ir jāekranē, izmantojot jumtiņa simbolu (^).

Šie noteikumi satur aizstājējzīmi (*), ko var izmantot ļaunprātīgi:

- Skriptu test.bat var izpildīt ar jebkādiem argumentiem, tostarp neparedzētiem.
- Šablons vfs.file.* atbilst tikai vienumu atslēgām bez parametriem; tomēr visiem vfs.file vienumiem ir nepieciešami parametri.
- Šablons system.cpu.load[*] atbilst tikai vienumu atslēgām ar parametriem; tomēr system.cpu.load vienumiem parametri nav nepieciešami.

Lai padarītu šos noteikumus drošākus, skaidri atļaujiet test.bat izpildi tikai ar konkrētiem argumentiem un aizliedziet pareizos vienumu atslēgu šablonus; piemēram:

```
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[*]
```

Varat pārbaudīt noteikumus, izpildot šādas komandas, kas atgriezīs 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]
```

Šablonu piemēri

Tālāk esošajā tabulā parādīts, kā tiek saskaņoti vienumu atslēgu šabloni:

- Atslēga atbilst šablonam tikai tad, ja tā atbilst **visiem** nosacījumiem kolonnā *Atbilst*.
- Parametriem jābūt pilnībā ietvertiem kvadrātiekavās (piemēram, vfs.file.contents[* un vfs.file.contents*utf8] ir nederīgi šabloni).

Šablons	Atbilst	Piemēri
*	Jebkura atslēga ar vai bez parametriem	
vfs.file.*	Atslēga sākas ar vfs.file. Nav parametru	Atbilst: vfs.file.size vfs.file.contents Neatbilst: vfs.file.contents[] vfs.file.size[/var/log/app.log]
vfs.*.contents	Atslēga sākas ar vfs. Atslēga beidzas ar .contents Nav parametru	Atbilst: vfs..contents vfs.mount.point.file.contents Neatbilst: vfs.contents vfs.file.contents[]

Šablons	Atbilst	Piemēri

`vfs.file.*[*]`	Atslēga sākas ar `vfs.file.`
	Jebkuri vai tukši parametri
		Atbilst
		`vfs.file.get.custom[]`
		`vfs.file.size[/var/log/app.log, utf8]`
		
		Neatbilst:
		`vfs.file.get.custom`

`vfs.file.contents`	Atslēga ir `vfs.file.contents`
	Nav parametru
		Atbilst:
		`vfs.file.contents`
		
		Neatbilst:
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[]`	Atslēga ir `vfs.file.contents[]`
	Tukši parametri
		Atbilst:
		`vfs.file.contents[]`
		
		Neatbilst:
		`vfs.file.contents`

`vfs.file.contents[*]`	Atslēga ir `vfs.file.contents`
	Jebkuri vai tukši parametri
		Atbilst:
		`vfs.file.contents[/path/to/file]`
		
		Neatbilst:
		`vfs.file.contents`

`vfs.file.contents[/etc/passwd,*]`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs ir `/etc/passwd`
	Jebkurš vai tukšs otrais parametrs
		Atbilst:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Neatbilst:
		`vfs.file.contents[]`
		`vfs.file.contents[/etc/passwd]`

`vfs.file.contents[*passwd*]`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs satur `passwd`
	Nav otrā parametra
		Atbilst:
		`vfs.file.contents[/etc/passwd]`
		
		Neatbilst:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`

`vfs.file.contents[*passwd*,*]`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs satur `passwd`
	Jebkurš vai tukšs otrais parametrs
		Atbilst:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Neatbilst:
		`vfs.file.contents[/etc/passwd]`
		`vfs.file.contents[/tmp/test]`

`vfs.file.contents[/etc/passwd,utf8]`	Atslēga ir `vfs.file.contents`
	Pirmais parametrs ir `/etc/passwd`
	Otrais parametrs ir `utf8`
		Atbilst:
		`vfs.file.contents[/etc/passwd,utf8]`
		
		Neatbilst:
		`vfs.file.contents[/etc/passwd,]`
		`vfs.file.contents[/etc/passwd,utf16]`

3 Problēmu noteikšana ar trigeriem

Problēmu noteikšana Zabbix sistēmā ir balstīta uz **trigeriem**.

Trigeris apraksta problēmas nosacījumu, ļaujot Zabbix reaģēt, ja problēma rodas.

Pilna problēmu noteikšanas darbplūsma ir šāda:

1. Pārliecinieties, ka jums ir vienums, kas vāc datus.
2. Definējiet šim vienumam trigeri. Bez trigera Zabbix tikai vāc datus, bet nekad uz tiem nereaģē.

3. Novērojiet noteiktās problēmas:
    - Sadaļā *Monitoring* > *Problems*
    - *Problems* logrīkā
4. Saņemiet paziņojumus, izpildiet attālās komandas (ja tās ir definētas).

Ņemiet vērā, ka oficiālajās Zabbix veidnēs ir iepriekš definēti trigeri. Skatiet pamācību par veidnes izmantošanu, lai uzraudzītu Linux.

Kas ir ”trigeris”?

Trigeris apraksta *problēmas nosacījumu*, izmantojot trigera izteiksmi.

Problēmas nosacījums, kad tas reāli iestājas, ir brīdis, kad trigeris ”nostrādā” (Zabbix tiek izveidota problēma).

Problēmas nosacījums var būt balstīts uz jaunāko vērtību, vidējo vērtību, atklātu virkni un daudzām citām funkcijām. Trigerus nevar izveidot vienumiem, kas atgriež vērtības ar bināro vai JSON datu tipu.

Vienkārša trigera izteiksme aprēķina funkciju (piemēram, `max()` maksimālajai vērtībai) datiem, kas norādīti iekavās (parasti datu vienums un laika logs), un pēc tam salīdzina šo aprēķinu ar slieksni.

Piemēram:

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Šis trigeris nostrādā, ja brīvā diska vieta, kas mērīta 5 minūšu laikā, pastāvīgi ir bijusi zem 10 GB.

Trigerim var būt šādi stāvokļi:

Trigera stāvoklis	Apraksts	Zabbix
OK	Trigera izteiksme ir novērtēta kā ”0” (vai FALSE)	Problēma nav izveidota Problēma ir atrisināta (ja tā pastāvēja)
Problem	Trigera izteiksme ir novērtēta kā ”1” (vai TRUE)	Problēma ir izveidota

Dažreiz trigera izteiksmes novērtēšana kāda iemesla dēļ nav iespējama. Skatiet Nezināms izteiksmes stāvoklis.

Citi resursi:

- Trigera piemēri
- Trigera konfigurēšana
- Labākā prakse

Aprēķina biežums

Trigeris tiek pārrēķināts katru reizi, kad Zabbix serveris saņem **jaunu vērtību**, kas ir daļa no izteiksmes. Kad tiek saņemta jauna vērtība, katra funkcija, kas ir iekļauta izteiksmē, tiek pārrēķināta (ne tikai tā, kura saņēma jauno vērtību).

Turklāt trigeris tiek pārrēķināts ik pēc 30 sekundēm (ar vēstures sinhronizētāja palīdzību), ja izteiksme satur kādu no šiem elementiem:

- Datuma un laika funkcijas
- nodata() funkciju

Trigeri, kas satur tikai tendenču funkcijas, tiek novērtēti vienu reizi katrā mazākajā izteiksmē norādītajā laika periodā. Lai gan daudzas trigeru funkcijas izmanto vienuma vērtību vēstures datus, tendenču funkcijas izmanto tendenču datus.

1 Trigeru piemēri

Šajā lapā ir apkopoti trigeru piemēri.

Piemēri ir sakārtoti pēc lietošanas gadījuma:

- Nepieejams hosts
- Nepieejams starpniekserveris
- Nepieejams SMTP klasteris
- Nesasniedzams hosts
- Negaidīta pārstartēšana
- Izmaiņas svarīgos failos
- Izmaiņas DNS vaicājumā

- Neatbilstoša programmatūra dažādos hostos
- Nesinhronizēti pulksteņi
- Novecojis aģents
- Nestabila saskarne
- Liela ienākošā datplūsma
- Maz brīvas diska vietas
- Maz brīvas diska vietas (dinamisks slieksnis)
- Liela CPU noslodze
- CPU noslodzes salīdzinājums
- Ilgtermiņa CPU noslodzes salīdzinājums
- Virkne, kas satur speciālās rakstzīmes

Noteikt nepieejamu hostu

```
max(/host/zabbix[host,agent,available],5m)=0
```

Šis trigeris nostrādā, ja Zabbix aģents hostā ir bijis nepieejams 5 minūtes.
 Funkcija: max
 Datu vienums: zabbix[host,agent,ava

Alternatīva:

```
nodata(/host/agent.ping,5m)=1
```

Šis trigeris nostrādā, ja no Zabbix aģenta 5 minūšu laikā nav saņemti dati.
 Funkcija: nodata
 Datu vienums: agent.ping

Noteikt nepieejamu starpniekserveri

```
fuzzytime(/host/zabbix[proxy,{$PROXY_NAME},lastaccess],1m)=0
```

Šis trigeris nostrādā, ja Zabbix starpniekservera dati atpaliek no Zabbix servera laika par 1 minūti.
 Funkcija: fuzzytime
 Datu vienums: zabbix[proxy,{$PROXY_NAME},lastaccess]

Noteikt nepieejamu SMTP klasteri

```
last(/smtp1.example.com/net.tcp.service[smtp])=0 and last(/smtp2.example.com/net.tcp.service[smtp])=0
```

Šis trigeris nostrādā, ja abi SMTP serveri nav pieejami.
 Funkcija: last
 Datu vienums: net.tcp.service

Noteikt nesasniedzamu hostu

```
count(/host/icmpping,30m,,"0")>5
```

Šis trigeris nostrādā, ja hosts pēdējo 30 minūšu laikā ar ping nav sasniedzams vairāk nekā 5 reizes.
 Funkcija: count
 Datu vienums: icmpping

Noteikt negaidītu pārstartēšanu

```
change(/host/system.uptime)<0
```

Šis trigeris nostrādā, ja tiek konstatēta negatīva sistēmas darbības laika vērtības izmaiņa (norādot uz pārstartēšanu).
 Funkcija: change
 Datu vienums: system.uptime

Noteikt izmaiņas svarīgos failos

```
last(/host/vfs.file.cksum[/etc/passwd],#1)<>last(/host/vfs.file.cksum[/etc/passwd],#2)
```

Šis trigeris nostrādā, ja /etc/passwd ir mainīts. Izteiksme ir patiesa, ja iepriekšējā /etc/passwd kontrolsumma atšķiras no jaunākās. Līdzīgas izteiksmes var būt noderīgas, lai uzraudzītu izmaiņas svarīgos failos, piemēram, /etc/passwd, /etc/inetd.conf, /kernel u.c.
 Funkcija: last
 Datu vienums: vfs.file.cksum

DNS vaicājuma izmaiņu noteikšana

```
last(/Zabbix server/net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1])<>"{$WEBSITE
```

Pievērsiet uzmanību pēdiņām ap otro operandu.

Šis trigeris nostrādā, ja vaicājuma rezultāts nav vienāds ar to, ko tas parasti atgriež:

```
example.com             MX      0 mail.example.com
```

Funkcija: last
 Datu vienums: net.dns.record[192.0.2.0,{$WEBSITE_NAME},{$DNS_RESOURCE_RECORD_TYPE},2,1], ar definētajiem makrosiem:

```
{$WEBSITE_NAME} = example.com
{$DNS_RESOURCE_RECORD_TYPE} = MX
```

Noteikt neatbilstošu programmatūru dažādos hostos

```
last(/host/vfs.file.contents[/etc/os-release])<>last(/host2/vfs.file.contents[/etc/os-release])
```

Šis trigeris nostrādā, ja Ubuntu versija dažādos hostos atšķiras. Ņemiet vērā, ka šeit operandi ir funkcijas, kas atgriež virknes.
 Funkcija: `last`
 Datu vienums: `vfs.file.contents`

Noteikt nesinhronizētus pulksteņus

```
fuzzytime(/host/system.localtime,10s)=0
```

Trigeris nostrādā, ja klienta lokālais laiks un Zabbix servera laiks atšķiras par vairāk nekā 10 sekundēm.
 Funkcija: `fuzzytime`
 Datu vienums: `system.localtime`

Ņemiet vērā, ka `system.localtime` Zabbix aģentam jākonfigurē kā pasīvā pārbaude; Zabbix aģentā 2 to var konfigurēt kā aktīvo pārbaudi.

Noteikt novecojušu aģentu

```
find(/host/agent.version,,"like","beta")=1
```

Šis trigeris nostrādā, ja Zabbix aģentam ir beta versija. Zabbix aģents ir jājaunina.
 Funkcija: `find`
 Datu vienums: `agent.version`

Noteikt interfeisu ar nestabilu darbību

```
changecount(/host/vfs.file.contents["/sys/class/net/eth0/operstate"],1h)>5
```

Šis trigeris nostrādā, ja eth0 darbības stāvoklis (up/down/unknown) stundas laikā ir mainījies vairāk nekā 5 reizes.
 Funkcija: `changecount`
 Datu vienums: `vfs.file.contents`

Noteikt lielu ienākošo datplūsmu

```
min(/host/net.if.in[eth0,bytes],5m)>100K
```

Šis trigeris nostrādā, ja saņemto baitu skaits uz eth0 pēdējo piecu minūšu laikā vienmēr bija lielāks par 100 kilobaitiem. Iespējams, kāds lejupielādē lielu failu.
 Funkcija: `min`
 Datu vienums: `net.if.in[eth0,bytes]`

Noteikt mazu brīvās diska vietas apjomu

```
max(/host/vfs.fs.size[/,free],5m)<10G
```

Trigeris nostrādā, ja brīvās diska vietas apjoms pastāvīgi (5 minūtes) ir mazāks par 10 GB.

Varat arī definēt atkopšanas izteiksmi:

```
min(/host/vfs.fs.size[/,free],10m)>40G
```

Problēma tiek **atrisināta** tikai tad, kad brīvās diska vietas apjoms pastāvīgi (10 minūtes) pārsniedz 40 GB.
 Funkcija: `min`
 Datu vienums: `vfs.fs.size`

Noteikt mazu diska vietas apjomu (dinamisks slieksnis)

```
last(/template/hrStorageFree[{#SNMPVALUE}])<last(/template/hrStorageSize[{#SNMPVALUE}])*0.1
```

Trigeris nostrādā, ja brīvā krātuves vieta (piešķires vienībās) samazinās zem 10 procentiem. Ievērojiet, ka tiek izmantota cita vienuma vērtība, lai iegūtu adaptīvu trigeris slieksni, kas ir piemērojams atklātai dažāda izmēra krātuvei. Funkcija: `last`

Noteikt augstu CPU noslodzi

```
last(/host/system.cpu.load[all,avg1])>5
```

Šis trigeris nostrādā, ja vidējā procesora noslodze vienu minūti ir bijusi virs 5.

Variācijas:

```
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
```

Šādi trigeri analizē 5 minūšu datus un nostrādā tikai tad, ja CPU noslodze nekad nav zemāka par 2. Papildus šie trigeri nostrādā:

- tikai naktī (00:00-06:00)
- jebkurā laikā, izņemot 2 stundas nedēļas maiņas laikā (svētdiena, 23:00 - pirmdiena, 01:00)

```
(last(/host/system.cpu.load[all,avg1])>5) + (last(/host2/system.cpu.load[all,avg1])>5) + (last(/host3/syst
```

Šis trigeris nostrādā, ja procesora noslodze ir pārāk augsta vismaz divos no trim hostiem.
 Funkcijas: `last`, `min`, `dayofweek`, `time`
 Datu vienums: `system.cpu.load`

Salīdzināt CPU slodzi

```
avg(/Zabbix server/system.cpu.load,1h)/avg(/Zabbix server/system.cpu.load,1h:now-1d)>2
```

Trigeris nostrādā, ja vidējā slodze šodien pārsniedz vidējo slodzi tajā pašā stundā vakar (izmantojot laika nobīdi `now-1d`) vairāk nekā divas reizes.
 Funkcija: `avg`
 Datu vienums: `system.cpu.load`

Salīdzināt ilgtermiņa CPU slodzes

```
trendavg(/host/system.cpu.load,1M:now/M)>1.1*trendavg(/host/system.cpu.load,1M:now/M-1M)
```

Šis trigeris nostrādā, ja CPU slodze hostā pagājušajā mēnesī palielinājās par vairāk nekā 10%
 Funkcija: `trendavg`
 Datu vienums: `system.cpu.load`

Varat arī izmantot lauku Notikuma nosaukums trigera konfigurācijā, lai izveidotu jēgpilnu brīdinājuma ziņojumu, piemēram, lai saņemtu kaut ko līdzīgu

```
"Exchange servera slodze jūlijā palielinājās par 24% (0.69), salīdzinot ar jūniju (0.56)"
```

notikuma nosaukums jādefinē šādi:

```
Load of {HOST.HOST} server increased by {{?100*trendavg(//system.cpu.load,1M:now/M)/trendavg(//system.cpu.
```

Šāda veida problēmai ir lietderīgi arī atļaut manuālu aizvēršanu trigera konfigurācijā.

Noteikt virkni, kas satur speciālās rakstzīmes

```
last(/host/vfs.file.contents[/tmp/hello])={$HELLO_MACRO}
```

Trigeris nostrādā, ja `/tmp/hello` saturs ir vienāds ar virkni, kas definēta `{$HELLO_MACRO}`:

```
{$HELLO_MACRO} = \" //hello ?\"
```

Alternatīvi, varat salīdzināt ar virkni tieši:

```
last(/Zabbix server/vfs.file.contents[/tmp/hello])="\\\" //hello ?\\\""
```

Ņemiet vērā, ka speciālās rakstzīmes (\\ un ") tiek atsoļotas, kad virkne tiek salīdzināta tieši.

Funkcija: `last`
 Datu vienums: `vfs.file.contents`

2 Trigera konfigurēšana

Šajā lapā ir aprakstīts, kā konfigurēt trigeri Zabbix lietotāja saskarnē.

Ņemiet vērā, ka oficiālajās Zabbix veidnēs ir iepriekš definēti trigeri. Skatiet ceļvedi par veidnes piemērošanu, lai uzraudzītu Linux. Ir iespējams arī veikt konfigurēto trigeru masveida atjaunināšanu.

Citi resursi:

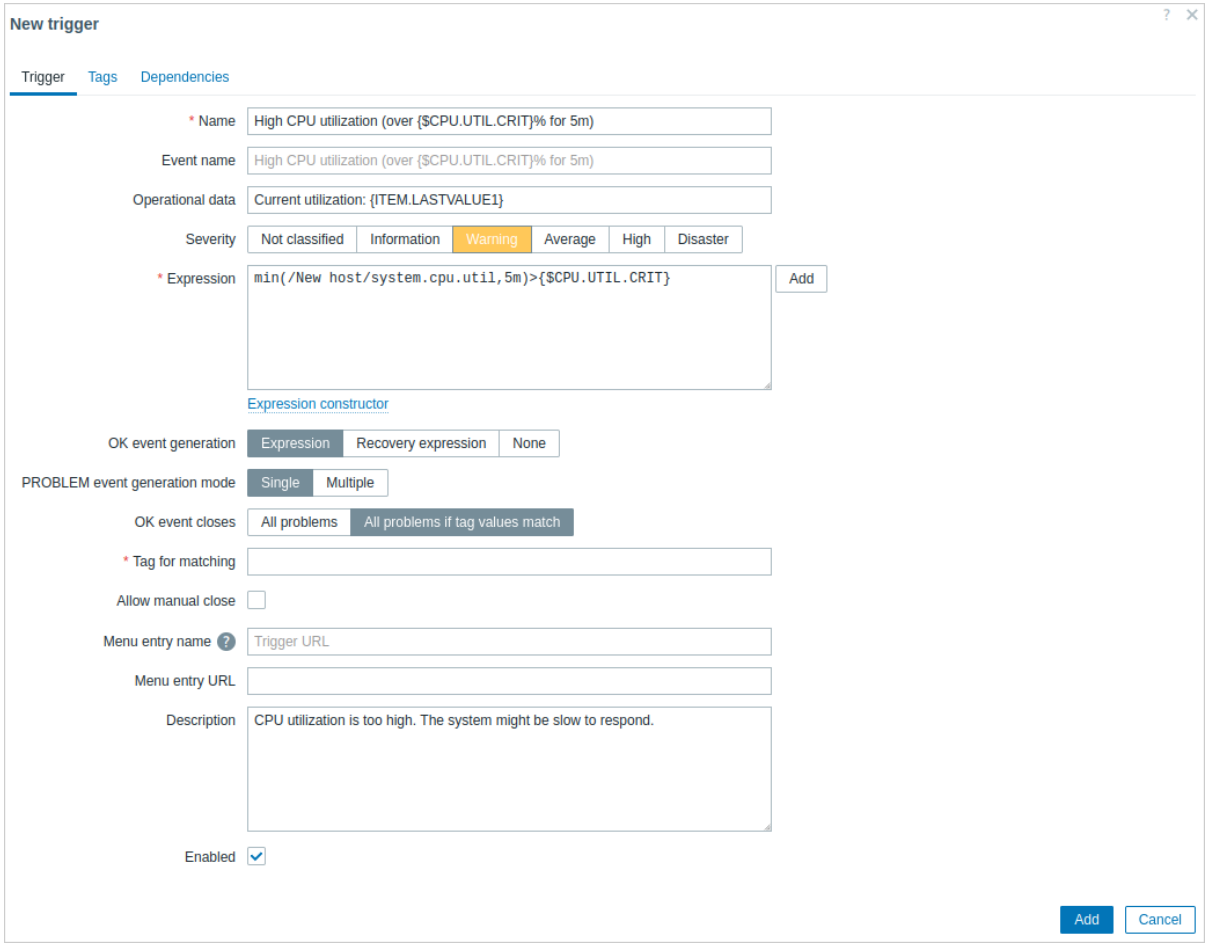
- Problēmu noteikšana ar trigeriem (vispārīgs ievads)
- Trigeru piemēri (pēc lietošanas gadījuma)
- Trigera izteiksme (sintakses informācija)
- Labākā prakse

Konfigurācija

Lai konfigurētu trigeri, rīkojieties šādi:

- Dodieties uz: *Datu ievākšana* > *Hosts*
- Hosts rindā noklikšķiniet uz *Trigeri*
- Labajā pusē noklikšķiniet uz *Izveidot trigeri* (vai uz trigera nosaukuma, lai rediģētu esošu trigeri)
- Veidlapā ievadiet trigera parametrus

Cilnē **Trigeris** ir ietverti visi būtiskie trigera atribūti.

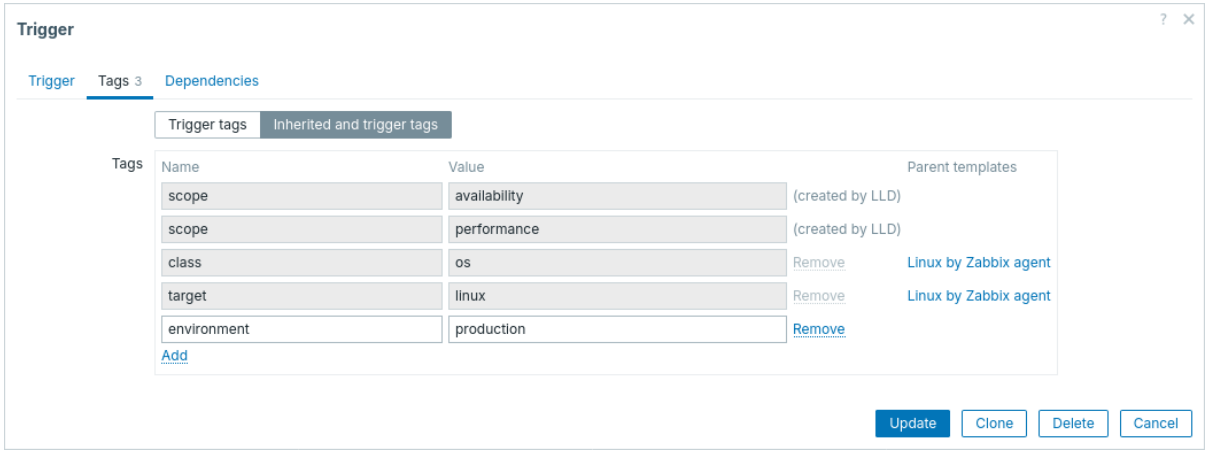


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Nosaukums	Trigera nosaukums. Atbalstītie makrosi ir: {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.*} un lietotāja makrosi {$MACRO}. Makrosus **$1, $2...$9** var izmantot, lai atsauktos uz izteiksmes pirmo, otro...devīto konstanti. *Piezīme*: makrosi $1-$9 tiks korekti atrisināti, ja tie atsaucas uz konstantēm salīdzinoši vienkāršās, tiešās izteiksmēs. Piemēram, nosaukums "Procesora slodze virs $1 uz {HOST.NAME}" automātiski mainīsies uz "Procesora slodze virs 5 uz New host", ja izteiksme ir last(/New host/system.cpu.load[percpu,avg1])>5.
Notikuma nosaukums	Ja definēts, šis nosaukums tiks izmantots problēmas notikuma nosaukuma izveidei. Pēc noklusējuma notikuma nosaukums ir tāds pats kā trigera nosaukums. Notikuma nosaukumu var izmantot, lai veidotu jēgpilnus brīdinājumus, kas satur problēmas datus (skatiet piemēru). Tiek atbalstīts tas pats makrosu kopums kā trigera nosaukumā, kā arī {TIME}, {TIMESTAMP} un izteiksmju makrosi {?EXPRESSION}.
Operatīvie dati	Operatīvie dati ļauj definēt patvaļīgas virknes kopā ar makrosiem. Makrosi tiks dinamiski atrisināti uz reāllaika datiem sadaļā *Uzraudzība* > *Problēmas*. Kamēr makrosi trigera nosaukumā (skatīt iepriekš) tiks atrisināti uz to vērtībām problēmas rašanās brīdī un kļūs par statiska problēmas nosaukuma pamatu, makrosi operatīvajos datos saglabā iespēju dinamiski attēlot pašu jaunāko informāciju. Ja trigera līmenī nav konfigurēti operatīvie dati, tiks attēlotas visu izteiksmē iekļauto vienumu jaunākās vērtības. Tiek atbalstīts tas pats makrosu kopums kā trigera nosaukumā.
Svarīgums	Iestatiet nepieciešamo trigera svarīgumu, noklikšķinot uz pogām.

Parametrs	Apraksts
Izteiksme	Loģiskā izteiksme, ko izmanto problēmas nosacījumu definēšanai. Tiek atbalstīti laika sufiksi un atmiņas lieluma sufiksi. Problēma tiek izveidota pēc tam, kad ir izpildīti visi izteiksmē iekļautie nosacījumi, t.i., izteiksmes rezultāts ir TRUE. Problēma tiks atrisināta, tiklīdz izteiksmes rezultāts būs FALSE, ja vien sadaļā *Atkopšanas izteiksme* nav norādīti papildu atkopšanas nosacījumi.
OK notikuma ģenerēšana	OK notikuma ģenerēšanas opcijas: **Izteiksme** - OK notikumi tiek ģenerēti, pamatojoties uz to pašu izteiksmi kā problēmas notikumi; **Atkopšanas izteiksme** - OK notikumi tiek ģenerēti, ja problēmas izteiksmes rezultāts ir FALSE un atkopšanas izteiksmes rezultāts ir TRUE; **Nav** - šajā gadījumā trigeris pats nekad neatgriezīsies OK stāvoklī.
Atkopšanas izteiksme	Loģiskā izteiksme (neobligāta), kas definē papildu nosacījumus, kuriem jābūt izpildītiem, pirms problēma tiek atrisināta, pēc tam kad sākotnējās problēmas izteiksmes rezultāts jau ir novērtēts kā FALSE. Atkopšanas izteiksme ir noderīga trigera histerēzei. **Nav** iespējams atrisināt problēmu tikai ar atkopšanas izteiksmi, ja problēmas izteiksmes rezultāts joprojām ir TRUE. Šis lauks ir pieejams tikai tad, ja īpašībai *OK notikuma ģenerēšana* ir atlasīta vērtība 'Atkopšanas izteiksme'.
PROBLEM notikuma ģenerēšanas režīms	Problēmas notikumu ģenerēšanas režīms: **Viens** - viens notikums tiek ģenerēts, kad trigeris pirmo reizi pāriet stāvoklī 'Problēma'; **Vairāki** - notikums tiek ģenerēts pie *katras* trigera novērtēšanas kā 'Problēma'.
OK notikums aizver	Izvēlieties, vai OK notikums aizver: **Visas problēmas** - visas šī trigera problēmas; **Visas problēmas, ja atbilst birku vērtības** - tikai tās trigera problēmas, kurām sakrīt notikuma birku vērtības.
Birka salīdzināšanai	Ievadiet notikuma birkas nosaukumu, ko izmantot notikumu korelācijai. Šis lauks tiek parādīts, ja īpašībai *OK notikums aizver* ir atlasīta vērtība 'Visas problēmas, ja atbilst birku vērtības', un šajā gadījumā tas ir obligāts.
Atļaut manuālu aizvēršanu	Atzīmējiet, lai atļautu šī trigera ģenerēto problēmas notikumu manuālu aizvēršanu. Manuāla aizvēršana ir iespējama, apstiprinot problēmas notikumus.
Izvēlnes ieraksta nosaukums	Ja nav tukšs, šeit ievadītais nosaukums (līdz 64 rakstzīmēm) tiek izmantots vairākās lietotāja saskarne vietās kā etiķete trigera URL, kas norādīts parametrā *Izvēlnes ieraksta URL*. Ja lauks ir tukšs, tiek izmantots noklusējuma nosaukums *Trigera URL*. Tiek atbalstīts tas pats makrosu kopums kā trigera nosaukumā, kā arī {EVENT.ID}, {HOST.ID} un {TRIGGER.ID}.
Izvēlnes ieraksta URL	Ja nav tukšs, šeit ievadītais URL (līdz 2048 rakstzīmēm) ir pieejams kā saite notikuma izvēlnē vairākās lietotāja saskarne vietās, piemēram, noklikšķinot uz problēmas nosaukuma sadaļā *Uzraudzība* > *Problēmas* vai paneļa logrīkā *Problēmas*. Tiek atbalstīts tas pats makrosu kopums kā trigera nosaukumā, kā arī {EVENT.ID}, {HOST.ID} un {TRIGGER.ID}. Piezīme: lietotāja makrosi ar slepenām vērtībām URL netiks atrisināti.
Apraksts	Teksta lauks, ko izmanto, lai sniegtu vairāk informācijas par šo trigeri. Var saturēt norādījumus konkrētas problēmas novēršanai, atbildīgā personāla kontaktinformāciju utt. Tiek atbalstīts tas pats makrosu kopums kā trigera nosaukumā.
Iespējots	Noņemot atzīmi no šīs izvēles rūtiņas, trigeris pēc vajadzības tiks atspējots. Atspējota trigera problēmas vairs netiek rādītas lietotāja saskarne, taču tās netiek dzēstas.

Cilne **Birkas** ļauj definēt trigera līmeņa birkas. Visas šī trigera problēmas tiks marķētas ar šeit ievadītajām vērtībām.



Papildus tam opcija *Mantotās un trigera birkas* ļauj skatīt veidnes līmenī definētās birkas, ja trigeris nāk no šīs veidnes. Ja ir vairākas veidnes ar vienādu birku, šīs birkas tiek parādītas vienu reizi, un veidņu nosaukumi tiek atdalīti ar komatiem. Trigeris "nemanto" un neparāda hosta līmeņa birkas.

Parametrs	Apraksts
Nosaukums/Vērtība	Iestatiet pielāgotas birkas, lai marķētu trigera notikumus. Birkas ir birkas nosaukuma un vērtības pāris. Varat izmantot tikai nosaukumu vai savienot to pārī ar vērtību. Trigerim var būt vairākas birkas ar vienādu nosaukumu, bet atšķirīgām vērtībām. Notikumu birkās tiek atbalstīti lietotāja makrosi, lietotāja makrosi ar kontekstu, zema līmeņa atklāšanas makrosi un makrosu funkcijas ar `{{ITEM.VALUE}}`, `{{ITEM.LASTVALUE}}`. Zema līmeņa atklāšanas makrosus var izmantot makrosa kontekstā. Trigera birku vērtībās tiek atbalstīts makross {TRIGGER.ID}. Tas var būt noderīgs, lai identificētu trigerus, kas izveidoti no trigeru prototipiem, un, piemēram, apspiestu problēmas no šiem trigeriem apkopes laikā. Ja izvērstās vērtības kopējais garums pārsniedz 255, tas tiks saīsināts līdz 255 rakstzīmēm. Skatiet visus notikumu birkām atbalstītos makrosus. Notikumu birkas var izmantot notikumu korelācijai, darbību nosacījumos, un tās būs redzamas arī sadaļā *Uzraudzība* > *Problēmas* vai logrīkā *Problēmas*.

Cilnē **Atkarības** ir ietvertas visas trigera atkarības.

Noklikšķiniet uz *Pievienot*, lai pievienotu jaunu atkarību.

Note:
Trigeri var konfigurēt arī, atverot esošu trigeri, nospiežot pogu *Klonēt* un pēc tam saglabājot to ar citu nosaukumu.

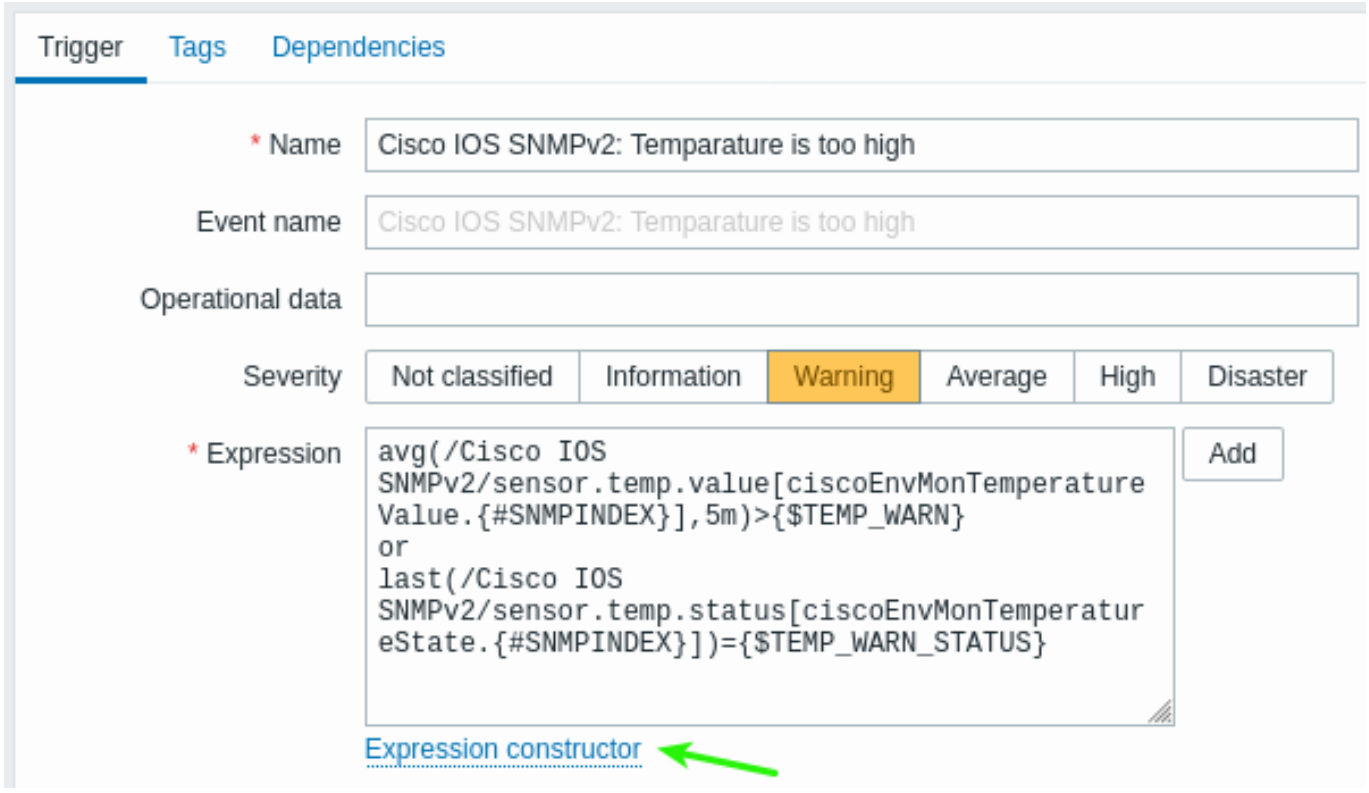
Trigera izteiksmes testēšana

Ir iespējams pārbaudīt, kāds būtu izteiksmes rezultāts atkarībā no saņemtās vērtības.

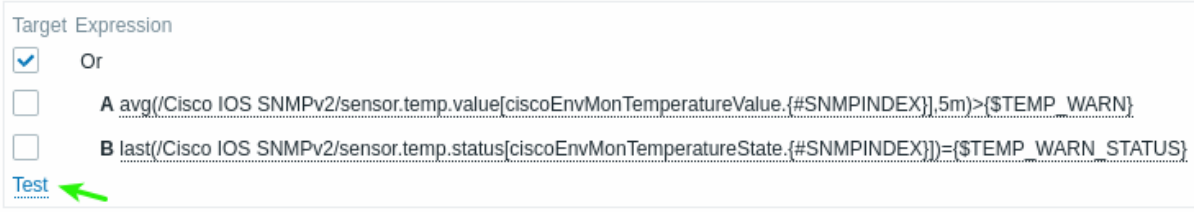
Kā piemērs ir izmantota šāda izteiksme no oficiālas veidnes:

```
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}
```

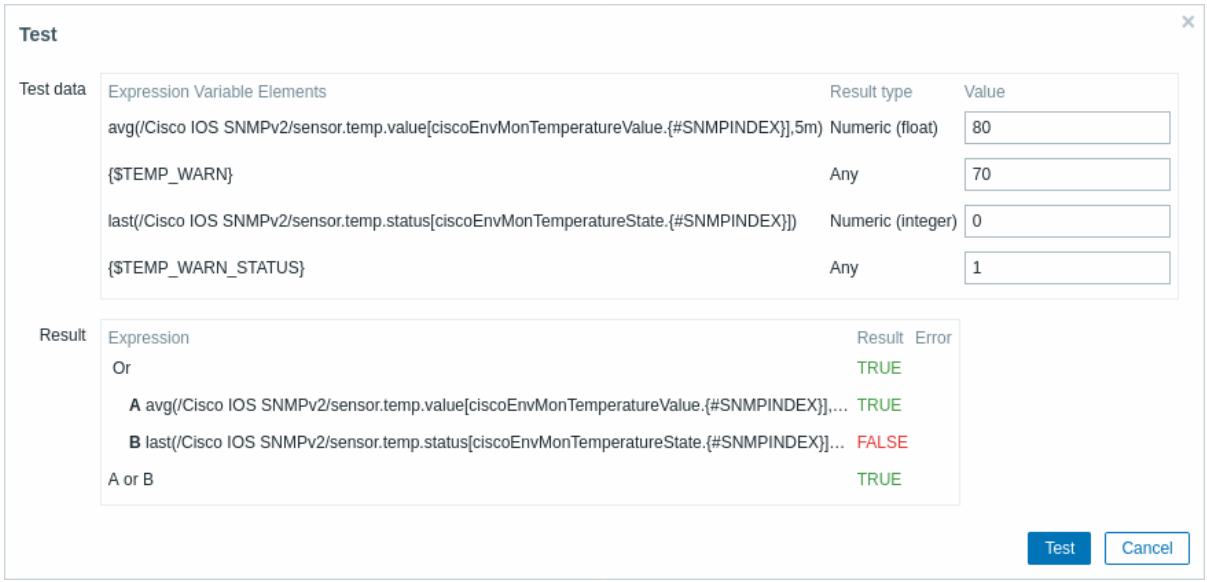
Lai pārbaudītu izteiksmi, zem izteiksmes lauka noklikšķiniet uz *Expression constructor*.



Sadaļā Expression constructor ir uzskaitītas visas atsevišķās izteiksmes. Lai atvērtu testēšanas logu, zem izteiksmju saraksta noklikšķiniet uz *Test*.



Testēšanas logā varat ievadīt paraugvērtības (’80’, ’70’, ’0’, ’1’ šajā piemērā) un pēc tam, noklikšķinot uz pogas *Test*, redzēt izteiksmes rezultātu.



Var redzēt gan atsevišķo izteiksmju, gan visas izteiksmes rezultātu.

”TRUE” nozīmē, ka norādītā izteiksme ir pareiza. Šajā konkrētajā A gadījumā ”80” ir lielāks par norādīto {$TEMP_WARN} vērtību, kas šajā piemērā ir ”70”. Kā paredzēts, tiek parādīts rezultāts ”TRUE”.

”FALSE” nozīmē, ka norādītā izteiksme nav pareiza. Šajā konkrētajā B gadījumā {$TEMP_WARN_STATUS} ”1” ir jābūt vienādam ar norādīto vērtību, kas šajā piemērā ir ”0”. Kā paredzēts, tiek parādīts rezultāts ”FALSE”.

Izvēlētais izteiksmes tips ir ”OR”. Ja vismaz viens no norādītajiem nosacījumiem (šajā gadījumā A vai B) ir TRUE, arī kopējais rezultāts būs TRUE. Tas nozīmē, ka pašreizējā vērtība pārsniedz brīdinājuma vērtību un ir radusies problēma.

3 Trigera izteiksme

Pārskats

Šajā lapā ir aprakstīta trigera izteiksmes sintakse un novērtēšanas detaļas.

Vienkāršas izteiksmes sintakse ir:

```
function(/host/item,time_period)<operator><constant>
```

Šajā izteiksmē pirmais operands (pa kreisi no operatora) ir funkcija ar tās parametriem iekavās (parasti datu vienums un laika periods).

Funkcija tiek izmantota, lai analizētu saņemtos datus norādītajā laika periodā, kā rezultātā tiek iegūta aprēķināta vērtība.

Pēc tam šī vērtība tiek salīdzināta ar otro operandu, izmantojot operatoru. Šajā piemērā otrais operands ir konstante, taču tā var būt arī cita funkcija.

Piemēram:

```
min(/Zabbix server/net.if.in[eth0,bytes],5m)>100K
```

Šis trigeris nostrādā, ja saņemto baitu skaits uz eth0 pēdējo piecu minūšu laikā vienmēr ir bijis lielāks par 100 kilobaitiem. Šajā gadījumā izteiksme ir patiesa un tiek izveidota problēma.

Trigeru izteiksmes ir ārkārtīgi elastīgas. Sarežģītākās izteiksmēs var kombinēt vairākas funkcijas, operatorus un konstantes.

Skatiet arī:

- Trigeru piemēri
- Problēmu noteikšana ar trigeriem (vispārīgs ievads)

Funkcijas

Funkcijas ļauj analizēt savāktās vērtības, piemēram, aprēķināt vidējo vai atrast noteiktu virkni.

Noklikšķiniet uz attiecīgās funkciju grupas, lai skatītu sīkāku informāciju.

Funkciju grupa	Funkcijas
Agregācijas funkcijas	avg, bucket_percentile, count, histogram_quantile, item_count, kurtosis, mad, max, min, skewness, stddevpop, stddevsamp, sum, sumofsquares, varpop, varsamp
Foreach funkcijas	avg_foreach, bucket_rate_foreach, count_foreach, exists_foreach, last_foreach, max_foreach, min_foreach, sum_foreach
Bitu operācijas funkcijas	bitand, bitlshift, bitnot, bitor, bitrshift, bitxor
Datuma un laika funkcijas	date, dayofmonth, dayofweek, now, time
Vēstures funkcijas	change, changecount, count, countunique, find, first, firstclock, fuzzytime, last, lastclock, logeventid, logseverity, logsource, logtimestamp, monodec, monoinc, nodata, percentile, rate
Tendenču funkcijas	baselinedev, baselinewma, trendavg, trendcount, trendmax, trendmin, trendstl, trendsum
Matemātiskās funkcijas	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
Operatoru funkcijas	between, in
Prognozējošās funkcijas	forecast, timeleft
Virkņu funkcijas	ascii, bitlength, bytelength, char, concat, insert, jsonpath, left, length, ltrim, mid, repeat, replace, right, rtrim, trim, xmlxpath

Ja vien nav norādīts citādi, šīs funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Foreach funkcijas tiek atbalstītas tikai agregācijas aprēķiniem.

Parasti funkcijas salīdzināšanai atgriež skaitliskas vērtības. Ja tiek atgrieztas virknes, salīdzināšana ir iespējama ar operatoriem **=** un **<>** (skatiet piemēru Neatbilstošas programmatūras noteikšana dažādos hostos).

Funkcijas parametri

Funkcijas parametri ļauj norādīt:

- vienuma atslēgu (kā `/hosts/key`) funkcijām, kas atsaucas uz hosta vienuma vēsturi
- laika periodu (un citus funkcijai specifiskus parametrus)
- citas izteiksmes

Vienuma atslēga

Norādītajam vienumam jābūt atbalstītā stāvoklī (izņemot funkciju **nodata()**, kas tiek aprēķināta arī neatbalstītiem vienumiem).

Hosta nosaukuma izlaišana pirmajā parametrā (t.i., kā `function(//key,parameter,...)`) tiek atbalstīta tikai noteiktos kontekstos:

- Aprēķināto vienumu formulā
- Izteiksmju makrosos, kurus var izmantot:
  - Laukā *Notikuma nosaukums*
  - Grafika nosaukumā
  - "Hosts" un "Trigeris" kartes elementos

Šajos kontekstos varat izmantot arī makrosu `{HOST.HOST}`. `{HOST.HOST<1-9>}` var izmantot lauka *Notikuma nosaukums* un kartes elementa "Trigeris" gadījumā, lai atsauktos uz konkrētu vienumu trigera izteiksmē. Ja hosta nosaukums šajos kontekstos ir izlaists vai aizstāts ar `{HOST.HOST}`, atsauce norāda uz pirmo vienumu trigera izteiksmē vai uz pirmo vienumu grafikā. Ārpus šiem atbalstītajiem kontekstiem hosta nosaukuma izlaišana trigeru izteiksmēs radīs kļūdu. Skatiet piemēru Salīdzināt ilgtermiņa CPU slodzes, lai redzētu dubultās slīpsvītras lietojuma ilustrāciju Notikuma nosaukuma makrosos.

Laika periods

Funkcijai specifiskie parametri tiek novietoti aiz vienuma atslēgas un ir atdalīti no vienuma atslēgas ar komatu.

Lielākā daļa skaitlisko funkciju pieņem **laika periodu** kā parametru. Tas ļauj norādīt mūs interesējošo intervālu. To var norādīt kā laika periodu (30s, 10m, 1h) vai kā vērtību diapazonu (#5 — pēdējām piecām vērtībām).

Lai norādītu laika periodu, var izmantot sekundes vai laika sufiksus. Ja parametram priekšā ir restīte, tam ir cita nozīme:

Expression	Description
sum(/host/key,**10m)**	Vērtību summa pēdējās 10 minūtēs.
sum(/host/key,**#10)**	Pēdējo desmit vērtību summa.

Parametriem ar restīti funkcijā **last** ir cita nozīme — tie apzīmē N-to iepriekšējo vērtību, tādēļ, ja dotās vērtības ir 30, 70, 20, 60, 50 (no jaunākās uz vecāko):

- `last(/host/key,#2)` atgriezīs '70'
- `last(/host/key,#5)` atgriezīs '50'

Laika periods tiek mērīts līdz "tagad" — kur "tagad" ir trigeris pēdējais pārrēķina laiks (skatiet Aprēķināšanas biežums); "tagad" nav serveris "tagad" laiks.

Laika periods nosaka vienu no šiem variantiem:

- Ņemt vērā visas vērtības starp "tagad-laika periods" un "tagad" (vai, ar laika nobīdi, starp "tagad-laika nobīde-laika periods" un "tagad-laika_nobīde")
- Ņemt vērā ne vairāk kā norādīto vērtību skaitu no pagātnes līdz "tagad"
  - Ja norādītajam laika periodam vai vērtību skaitam nav pieejama neviena vērtība, tad trigeris vai aprēķinātais vienums, kas izmanto šo funkciju, kļūst neatbalstīts

Ņemiet vērā:

- Ja trigerī tiek izmantota tikai viena funkcija (kas atsaucas uz datu vēsturi), "tagad" vienmēr ir pēdējā saņemtā vērtība. Piemēram, ja pēdējā vērtība tika saņemta pirms stundas, laika periods tiks uzskatīts par tādu, kas beidzas ar pēdējo vērtību pirms stundas.
- Jauns trigeris tiek aprēķināts, tiklīdz tiek saņemta pirmā vērtība (vēstures funkcijas); funkcijām date and time un nodata() tas tiks aprēķināts 30 sekunžu laikā. Tādējādi trigeris tiks aprēķināts pat tad, ja kopš trigeris izveides vēl nav pagājis iestatītais laika periods (piemēram, viena stunda). Trigeris tiks aprēķināts arī pēc pirmās vērtības saņemšanas, pat ja laika diapazons bija iestatīts, piemēram, uz pēdējām desmit vērtībām.

Laika nobīde

Kā funkcijas parametrs tiek atbalstīta neobligāta laika nobīde ar laiku vai vērtību skaitu. Šis parametrs ļauj atsaukties uz datiem no pagātnes laika perioda.

Laika nobīde sākas ar `now` — norādot pašreizējo laiku, un tai seko `+N<time unit>` vai `-N<time unit>` — lai pieskaitītu vai atņemtu N laika vienības.

Piemēram, `avg(/host/key,1h:now-1d)` atgriezīs vidējo vērtību par vienu stundu pirms vienas dienas.

Attention:
Laika nobīde, kas norādīta mēnešos (M) un gados (y), tiek atbalstīta tikai tendenču funkcijām. Citas funkcijas atbalsta sekundes (s), minūtes (m), stundas (h), dienas (d) un nedēļas (w).

Laika nobīde ar absolūtiem laika periodiem

Laika nobīdes parametrā tiek atbalstīti absolūti laika periodi, piemēram, no pusnakts līdz pusnaktij dienai, no pirmdienas līdz svētdienai nedēļai, no mēneša pirmās līdz pēdējai dienai mēnesim.

Laika nobīde absolūtiem laika periodiem sākas ar `now` — norādot pašreizējo laiku, un tai seko jebkurš skaits laika operāciju: `/<time unit>` — nosaka laika vienības sākumu un beigas, piemēram, no pusnakts līdz pusnaktij dienai, `+N<time unit>` vai `-N<time unit>` — lai pieskaitītu vai atņemtu N laika vienības.

Lūdzu, ņemiet vērā, ka laika nobīdes vērtība var būt lielāka vai vienāda ar 0, savukārt laika perioda minimālā vērtība ir 1.

Parameter	Description
1d:now/d	Vakar
1d:now/d+1d	Šodien
2d:now/d+1d	Pēdējās 2 dienas
1w:now/w	Pagājušā nedēļa
1w:now/w+1w	Šī nedēļa

Citas izteiksmes

Funkciju parametri var saturēt citas izteiksmes, kā parādīts šādā sintaksē:

```
min(min(/host/key,1h),min(/host2/key2,1h)*10)
```

Ņemiet vērā, ka citas izteiksmes nevar izmantot, ja funkcija atsaucas uz vienuma vēsturi. Piemēram, šāda sintakse **nav** atļauta:

~~`min(/host/key,#5*10)`~~

Lai gan citu trigeru izteiksmju izmantošana kā funkciju parametri ir ierobežota ar funkcijām, kas trigeros neizmanto vēsturi, šis ierobežojums neattiecas uz aprēķinātajiem vienumiem.

Operatori

Tālāk norādītie operatori tiek atbalstīti trigeriem **(dilstošā izpildes prioritātes secībā)**:

Prioritāte	Operators	Definīcija	**Piezīmes par nezināmām vērtībām**	Piespiedu operanda pārveidošana par float [1]
1	**-**	Unārais mīnuss	**-** Unknown → Unknown	Jā
2	**not**	Loģiskais NOT	**not** Unknown → Unknown	Jā
3	*	Reizināšana	0 * Unknown → Unknown (jā, Unknown, nevis 0 — lai nezaudētu Unknown aritmētiskajās operācijās) 1.2 * Unknown → Unknown	Jā
	/	Dalīšana	Unknown / 0 → kļūda Unknown / 1.2 → Unknown 0.0 / Unknown → Unknown	Jā
4	+	Aritmētiskā saskaitīšana	1.2 + Unknown → Unknown	Jā
	-	Aritmētiskā atņemšana	1.2 - Unknown → Unknown	Jā
5	<	Mazāks par. Operators ir definēts šādi: A<B ⇔ (A<B-0.000001)	1.2 < Unknown → Unknown	Jā

Prioritāte	Operators	Definīcija	**Piezīmes par nezināmām vērtībām**	Piespiedu operanda pārveidošana par float [1]
	<=	Mazāks vai vienāds ar. Operators ir definēts šādi: A<=B ⇔ (A≤B+0.000001)	Unknown **<=** Unknown → Unknown	Jā
	>	Lielāks par. Operators ir definēts šādi: A>B ⇔ (A>B+0.000001)		Jā
	>=	Lielāks vai vienāds ar. Operators ir definēts šādi: A>=B ⇔ (A≥B-0.000001)		Jā
6	**=**	Vienāds ar. Operators ir definēts šādi: A=B ⇔ (A≥B-0.000001) and (A≤B+0.000001)		Nē [1]
	<>	Nav vienāds ar. Operators ir definēts šādi: A<>B ⇔ (A<B-0.000001) or (A>B+0.000001)		Nē [1]
7	**and**	Loģiskais AND	0 **and** Unknown → 0 1 **and** Unknown → Unknown Unknown **and** Unknown → Unknown	Jā
8	**or**	Loģiskais OR	1 **or** Unknown → 1 0 **or** Unknown → Unknown Unknown **or** Unknown → Unknown	Jā

[1] Virknes operands joprojām tiek pārveidots par skaitlisku, ja:

- otrs operands ir skaitlisks
- operandam tiek izmantots operators, kas nav **=** vai **<>**

(Ja pārveidošana neizdodas, skaitliskais operands tiek pārveidots par virknes operandu un abi operandi tiek salīdzināti kā virknes.)

Operatori **not**, **and** un **or** ir reģistrjutīgi, un tiem jābūt rakstītiem ar mazajiem burtiem. Tiem arī jābūt atdalītiem ar atstarpēm vai iekavām.

Visiem operatoriem, izņemot unāro **-** un **not**, ir asociativitāte no kreisās uz labo pusi. Unārais **-** un **not** nav asociatīvi (tas nozīmē, ka **-(-1)** un **not (not 1)** jāizmanto **--1** un **not not 1** vietā).

Novērtēšanas rezultāts:

- operatori **<**, **<=**, **>**, **>=**, **=**, **<>** trigera izteiksmē dod rezultātu '1', ja norādītā attiecība ir patiesa, un '0', ja tā ir aplama. Ja vismaz viens operands ir Unknown, rezultāts ir Unknown;
- **and** zināmiem operandiem dod '1', ja abi tā operandi nav vienādi ar '0'; pretējā gadījumā tas dod '0'; nezināmiem operandiem **and** dod '0' tikai tad, ja viens operands ir vienāds ar '0'; pretējā gadījumā tas dod 'Unknown';
- **or** zināmiem operandiem dod '1', ja kāds no tā operandiem nav vienāds ar '0'; pretējā gadījumā tas dod '0'; nezināmiem operandiem **or** dod '1' tikai tad, ja viens operands nav vienāds ar '0'; pretējā gadījumā tas dod 'Unknown';
- loģiskās noliegšanas operatora **not** rezultāts zināmam operandam ir '0', ja tā operanda vērtība nav vienāda ar '0'; '1', ja tā operanda vērtība ir vienāda ar '0'. Nezināmam operandam **not** dod 'Unknown'.

Nezināms izteiksmes stāvoklis

Iespējams, ka trigera izteiksmē parādās nezināms operands, ja:

- tiek izmantots neatbalstīts vienums
- funkcijas novērtēšana atbalstītam vienumam beidzas ar kļūdu

Šādā gadījumā trigera izteiksme parasti tiek novērtēta kā `Unknown` (jo to nevar novērtēt)

Ir iespējams saņemt paziņojumus par nezināmiem trigeriem.

Izņēmumi

Neraugoties uz nezināmu operandu, trigeru izteiksmes dažos gadījumos var tikt novērtētas ar zināmu rezultātu (Problem/OK):

- Funkcija `nodata()` tiek novērtēta neatkarīgi no tā, vai atsauces vienums ir atbalstīts vai nav.
- Izteiksmes ar AND/OR var tikt novērtētas ar zināmu rezultātu divos gadījumos:
  - **1. gadījums**: "`1 or some_function(unsupported_item1) or some_function(unsupported_item2) or ...`" tiek novērtēta ar zināmu rezultātu ('1' vai "Problem"),
  - **2. gadījums**: "`0 and some_function(unsupported_item1) and some_function(unsupported_item2) and ...`" tiek novērtēta ar zināmu rezultātu ('0' vai "OK").
- Ja funkcijas novērtēšana atbalstītam vienumam beidzas ar kļūdu, funkcijas vērtība kļūst par `Unknown` un tā piedalās turpmākajā izteiksmes novērtēšanā kā nezināms operands.

Nezināmi operandi var "pazust" tikai loģiskajās izteiksmēs, kā aprakstīts iepriekš. Aritmētiskajās izteiksmēs nezināmi operandi vienmēr noved pie `Unknown` (izņemot dalīšanu ar 0).

Attention:
Nezināms izteiksmes stāvoklis nemaina trigera stāvokli (Problem/OK). Tātad, ja trigera stāvoklis bija "Problem" (skatiet 1. gadījumu), tas paliek problēmas stāvoklī pat tad, ja zināmā daļa tiek atrisināta ('1' kļūst par '0'), jo izteiksme tagad tiek novērtēta kā `Unknown`, un tas nemaina trigera stāvokli.

Ja trigera izteiksme ar vairākiem neatbalstītiem vienumiem tiek novērtēta kā `Unknown`, kļūdas ziņojums lietotāja saskarnē attiecas uz pēdējo novērtēto neatbalstīto vienumu.

Vērtību kešošana

Vērtības, kas nepieciešamas trigeru novērtēšanai, tiek kešotas Zabbix serverī. Tāpēc pēc servera restartēšanas trigeru novērtēšana kādu laiku rada lielāku datubāzes noslodzi.

Vērtību kešatmiņa netiek notīrīta, kad vienuma vēstures vērtības tiek noņemtas (vai nu manuāli, vai ar housekeeper palīdzību), tāpēc serveris izmantos kešotās vērtības, līdz tās kļūs vecākas par laika periodiem, kas definēti trigeru funkcijās, vai līdz serveris tiks restartēts.

Note:
Ja kešatmiņā nav nesenu datu un funkcijā nav definēts vaicājuma periods, Zabbix pēc noklusējuma vaicās datubāzi par vēsturiskajām vērtībām, meklējot datus līdz pat vienai nedēļai pagātnē.

1 Agregācijas funkcijas

Ja vien nav norādīts citādi, visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Agregācijas funkcijas var darboties ar:

- vienumu vēsturi, piemēram, `min(/host/key,1h)`

- foreach funkcijām kā vienīgo parametru, piemēram, `min(last_foreach(/*/key))` (tikai aprēķinātajos vienumos; nevar izmantot trigeros)

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
avg	Vienuma vidējā vērtība noteiktajā novērtēšanas periodā.
bucket_percentile	Aprēķina procentili no histogrammas spaiņiem.
count	Vērtību skaits masīvā, ko atgriež foreach funkcija.
histogram_quantile	Aprēķina φ-kvantili no histogrammas spaiņiem.
item_count	Konfigurācijā esošo vienumu skaits, kas atbilst filtra kritērijiem.
kurtosis	Varbūtības sadalījuma ”astainums” savāktajās vērtībās noteiktajā novērtēšanas periodā.
mad	Mediānā absolūtā novirze savāktajās vērtībās noteiktajā novērtēšanas periodā.
max	Vienuma augstākā vērtība noteiktajā novērtēšanas periodā.
min	Vienuma zemākā vērtība noteiktajā novērtēšanas periodā.
skewness	Varbūtības sadalījuma asimetrija savāktajās vērtībās noteiktajā novērtēšanas periodā.
stddevpop	Populācijas standartnovirze savāktajās vērtībās noteiktajā novērtēšanas periodā.
stddevsamp	Izlases standartnovirze savāktajās vērtībās noteiktajā novērtēšanas periodā.
sum	Savākto vērtību summa noteiktajā novērtēšanas periodā.
sumofsquares	Savākto vērtību kvadrātu summa noteiktajā novērtēšanas periodā.
varpop	Savākto vērtību populācijas dispersija noteiktajā novērtēšanas periodā.
varsamp	Savākto vērtību izlases dispersija noteiktajā novērtēšanas periodā.

Kopīgie parametri

- `/host/key` ir kopīgs obligāts pirmais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi
- `(sec|#num)<:time shift>` ir kopīgs otrais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi, kur:
  - **sec** - maksimālais novērtēšanas periods sekundēs (var izmantot laika sufiksus), vai
  - **#num** - maksimālais novērtēšanas diapazons pēdējās savāktajās vērtībās (ja pirms tā ir jaucējzīme)
  - **time shift** (neobligāts) ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Skatiet papildinformāciju par time shift norādīšanu.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Neobligātie funkciju parametri (vai parametru daļas) ir norādīti ar < >
- Katrai funkcijai specifiskie parametri ir aprakstīti pie attiecīgās funkcijas
- Parametri `/host/key` un `(sec|#num)<:time shift>` nekad nedrīkst būt pēdiņās

avg(/host/key,(sec|#num)<:time shift>)

Vienuma vidējā vērtība noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītās foreach funkcijas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: skatiet kopīgos parametrus.

Laika nobīde ir noderīga, ja nepieciešams salīdzināt pašreizējo vidējo vērtību ar vidējo vērtību kādu laiku iepriekš.

Piemēri:

```
avg(/host/key,1h) #vidējā vērtība pēdējā stundā līdz šim brīdim
avg(/host/key,1h:now-1d) #vidējā vērtība vienas stundas periodam no pirms 25 stundām līdz pirms 24 stundām
avg(/host/key,#5) #piecu jaunāko vērtību vidējā vērtība
avg(/host/key,#5:now-1d) #piecu jaunāko vērtību vidējā vērtība, izņemot vērtības, kas saņemtas pēdējo 24 s
avg(/host/proc.num,5m)>300 #trigeris, ja vidējais procesu skaits pēdējās 5 minūtēs ir bijis virs 300
```

bucket_percentile(item filter,time period,percentage)

Aprēķina procentili no histogrammas spaiņiem.

Parametri:

- **vienuma filtrs** - skatiet vienuma filtru;
- **laika periods** - skatiet laika periodu;
- **procentuālā vērtība** - procenti (0-100).

Komentāri:

- Atbalstīts tikai aprēķinātajos vienumos;
- Šī funkcija ir aizstājvārds `histogram_quantile(percentage/100, bucket_rate_foreach(item filter, time period, 1))`.

Piemēri:

```
bucket_percentile(/*/http_request_duration_seconds_bucket[*],5m,95) #calculate the 95th percentile respons
bucket_percentile(/*/apiserver_request_duration_seconds_bucket[*],10m,99) #calculate the 99th percentile A
```

count(func_foreach(item filter,<time period>),<operator>,<pattern>)

Vērtību skaits masīvā, ko atgriež foreach funkcija.
 Atbalstītās foreach funkcijas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri:

- **func_foreach** - foreach funkcija, kurai jāaprēķina atgriezto vērtību skaits. Sīkāku informāciju skatiet foreach funkcijās. Ņemiet vērā, ka count_foreach un bucket_rate_foreach atbalsta papildu parametrus.
- **item filter** - skatiet vienuma filtru;
- **time period** - skatiet laika periodu;
- **operator** (jābūt dubultpēdiņās). Atbalstītie `operatori`:
eq - vienāds
ne - nav vienāds
gt - lielāks
ge - lielāks vai vienāds
lt - mazāks
le - mazāks vai vienāds
like - atbilst, ja satur šablonu (reģistrjutīgs)
bitand - bitu AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
- **pattern** - nepieciešamais šablons (virknes argumentiem jābūt dubultpēdiņās); tiek atbalstīts, ja *operator* ir norādīts trešajā parametrā.

Komentāri:

- Izmantojot **count()** ar vēsturei piesaistītu foreach funkciju (max_foreach, avg_foreach u.c.), var rasties ietekme uz veiktspēju, savukārt **exists_foreach()** izmantošana, kas darbojas tikai ar konfigurācijas datiem, šādu efektu neradīs.
- Neobligātos parametrus *operator* vai *pattern* pēc komata nevar atstāt tukšus, tos var tikai pilnībā izlaist.
- Ja kā trešais parametrs tiek izmantots *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, kas atdalīti ar '/': **number_to_compare_with/mask**. count() aprēķina "bitu AND" no vērtības un *mask*, un salīdzina rezultātu ar *number_to_compare_with*. Ja "bitu AND" rezultāts ir vienāds ar *number_to_compare_with*, vērtība tiek ieskaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, var norādīt tikai *mask* (bez '/').
- Ja kā trešais parametrs tiek izmantots *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek pārmantota no globālās regulārās izteiksmes iestatījumiem. Regulārās izteiksmes salīdzināšanai peldošā komata vērtības vienmēr tiks attēlotas ar 4 zīmēm aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība starp decimālo (glabājas datubāzē) un bināro (izmanto Zabbix serveris) attēlojumu var ietekmēt 4. zīmi aiz komata.

Piemēri:

```
count(max_foreach(/*/net.if.in[*],1h)) #calculate the number of net.if.in items that received data in the
count(last_foreach(/*/vfs.fs.size[*,pused]),"gt",95) #calculate the number of file systems with over 95% c
```

histogram_quantile(quantile,bucket1,value1,bucket2,value2,...)

Aprēķina φ-kvantili no histogrammas spaiņiem.
 Atbalstītā foreach funkcija: *bucket_rate_foreach*.

Parametri:

- **quantile** — $0 \le \varphi \le 1$;
- **bucketN, valueN** — manuāli ievadīti parametru pāri (>=2) vai bucket_rate_foreach atbilde.

Komentāri:

- Atbalstīts tikai aprēķinātajos vienumos;
- Funkcionāli atbilst PromQL funkcijai 'histogram_quantile';
- Atgriež -1, ja pēdējā 'Infinity' spaiņa ("+*inf*") vērtības ir vienādas ar 0.

Piemēri:

```
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)

Konfigurācijā esošo vienumu skaits, kas atbilst filtra kritērijiem.
 Atbalstītais vērtības tips: *Integer*.

Parametrs:

- **vienuma filtrs** — vienumu atlases kritēriji; ļauj veikt atsauces pēc hostu grupas, hosta, vienuma atslēgas un tagiem. Tiek atbalstītas aizstājējzīmes. Plašāku informāciju skatiet vienuma filtrs.

Komentāri:

- Atbalstīts tikai aprēķinātajos vienumos;
- Darbojas kā funkcijas *count(exists_foreach(item_filter))* aizstājvārds.

Piemēri:

```
item_count(/*/agent.ping?[group="Host group 1"]) #aprēķina hostu skaitu ar vienumu agent.ping grupā "Host
```

kurtosis(/host/key,(sec|#num)<:time shift>)

Savākto vērtību varbūtības sadalījuma ”astainums” noteiktajā novērtēšanas periodā. Skatiet arī: Kurtosis.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach function: *last_foreach*.

Parametri: skatiet common parameters.

Piemēri:

```
kurtosis(/host/system.cpu.util[,user],5m)>5 #trigeris uz straujiem CPU noslodzes pīķiem 5 minūšu laikā
```

mad(/host/key,(sec|#num)<:time shift>)

Mediānā absolūtā novirze savāktajām vērtībām definētajā novērtēšanas periodā. Skatiet arī: Mediānā absolūtā novirze.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
last(/host/system.cpu.util[,user])>avg(/host/system.cpu.util[,user],1h)+3*mad(/host/system.cpu.util[,user]
```

max(/host/key,(sec|#num)<:time shift>)

Vienuma augstākā vērtība noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītās foreach funkcijas: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
max(/host/key,1h) - min(/host/key,1h) #aprēķina starpību starp maksimālo un minimālo vērtību pēdējās stund
max(last_foreach(/*/vfs.fs.size[*,pused]?[group="Linux servers"])) #atgriež augstāko diska lietojumu visos
```

min(/host/key,(sec|#num)<:time shift>)

Zemākā vienuma vērtība noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītās foreach functions: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
max(/host/key,1h) - min(/host/key,1h) #aprēķina starpību starp maksimālo un minimālo vērtību pēdējās stund
min(last_foreach(/*/vfs.file.contents["/sys/class/net/enp0s3/operstate"]?[group="Linux servers"])) #atgrie
```

skewness(/host/key,(sec|#num)<:time shift>)

Savākto vērtību varbūtības sadalījuma asimetrija noteiktajā novērtēšanas periodā. Skatiet arī: Skewness.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
skewness(/host/vm.memory.size[used],5m)<-20 #trigeris pēkšņiem negatīviem kritumiem, kas norāda uz avārijā
```

stddevpop(/host/key,(sec|#num)<:time shift>)

Populācijas standartnovirze savāktajās vērtībās noteiktajā novērtēšanas periodā. Skatiet arī: Standartnovirze.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
stddevpop(/host/system.cpu.util[,user],10m)>7 #trigeris, ja CPU lietojums stipri svārstās (populācijas sta
```

stddevsamp(/host/key,(sec|#num)<:time shift>)

Izlases standartnovirze savāktajām vērtībām noteiktajā novērtēšanas periodā. Skatiet arī: Standartnovirze.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Lai šī funkcija darbotos, ir nepieciešamas vismaz divas datu vērtības.

Piemēri:

```
stddevsamp(/host/system.cpu.util[,user],10m)>7 #trigeris, ja CPU lietojums stipri svārstās (izlases standa
```

sum(/host/key,(sec|#num)<:time shift>)

Savākto vērtību summa noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītās foreach functions: *avg_foreach*, *count_foreach*, *exists_foreach*, *last_foreach*, *max_foreach*, *min_foreach*, *sum_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
sum(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #the sum of latest values for incoming network
```

sumofsquares(/host/key,(sec|#num)<:time shift>)

Savākto vērtību kvadrātu summa noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītās foreach function: *last_foreach*.

Parametri: skatiet common parameters.

Piemēri:

```
sumofsquares(/host/calculated.net.if.rate,1m) #tīkla datplūsmas ātruma kvadrātu summa (ienākošā pret izejo
sumofsquares(last_foreach(/*/net.if.in[*]?[group="Linux servers"])) #jaunāko ienākošās tīkla datplūsmas vē
```

varpop(/host/key,(sec|#num)<:time shift>)

Savākto vērtību populācijas dispersija noteiktajā novērtēšanas periodā. Skatiet arī: Dispersija.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Piemēri:

```
varpop(/host/system.cpu.util[,user],10m)>50 #trigeris, ja CPU lietojums stipri svārstās (dispersija ir vir
```

varsamp(/host/key,(sec|#num)<:time shift>)

Savākto vērtību izlases dispersija noteiktajā novērtēšanas periodā. Skatiet arī: Dispersija.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atbalstītā foreach funkcija: *last_foreach*.

Parametri: skatiet kopīgos parametrus.

Lai šī funkcija darbotos, ir nepieciešamas vismaz divas datu vērtības.

Piemēri:

```
varsamp(/host/system.cpu.util[,user],10m)>50 #trigeris, ja CPU lietojums stipri svārstās (izlases dispersi
```

Skatīt visas atbalstītās funkcijas.

Foreach funkcijas

Pārskats

Foreach funkcijas tiek izmantotas apkoptajos aprēķinos, lai katram vienumam, kas ir atlasīts ar izmantoto **vienumu filtru**, atgrieztu vienu apkopotu vērtību. Tiek atgriezts vērtību masīvs.

Attention:
Foreach funkcijas tiek atbalstītas tikai aprēķinātajos vienumos kā daļa no apkopoto aprēķinu formulām. Tās nevar izsaukt parastās vienumu atslēgās vai trigeru izteiksmēs ārpus šī konteksta.

Piemēram, funkcija *avg_foreach* atgriezīs vērtību masīvu, kur katra vērtība ir atlasītā vienuma vēstures vērtību *vidējā* vērtība norādītajā laika intervālā.

item filter ir daļa no sintakses, ko izmanto foreach funkcijas. Vienumu filtrā tiek atbalstīta aizstājējzīmju izmantošana, tādējādi nepieciešamos vienumus var atlasīt ļoti elastīgi.

Atbalstītās funkcijas

Function	Description
avg_foreach	Atgriež katra vienuma vidējo vērtību.
bucket_rate_foreach	Atgriež pārus (groza augšējā robeža, ātruma vērtība), kas ir piemēroti izmantošanai funkcijā histogram_quantile(), kur "groza augšējā robeža" ir vienuma atslēgas parametra vērtība, kas definēta <parameter number> parametrā.
count_foreach	Atgriež vērtību skaitu katram vienumam.
exists_foreach	Atgriež '1' katram iespējotam vienumam.
last_foreach	Atgriež katra vienuma pēdējo vērtību.
max_foreach	Atgriež katra vienuma maksimālo vērtību.
min_foreach	Atgriež katra vienuma minimālo vērtību.
sum_foreach	Atgriež vērtību summu katram vienumam.

Funkcijas sintakse

Foreach funkcijas atbalsta divus kopīgus parametrus: `vienuma filtrs` (skatiet detalizētu informāciju zemāk) un `laika periods`:

```
foreach_function(item filter,time period)
```

Piemēram:

```
avg_foreach(/*/mysql.qps?[group="MySQL Servers"],5m)
```

atgriezīs katra 'mysql.qps' vienuma piecu minūšu vidējo vērtību MySQL serveru grupā.

Ņemiet vērā, ka dažas funkcijas atbalsta papildu parametrus.

Vienumu filtra sintakse

Vienumu filtrs:

```
/host/key[parameters]?[conditions]
```

sastāv no četrām daļām, kur:

- host - hosta nosaukums
- key - vienuma atslēga (bez parametriem)
- parameters - vienuma atslēgas parametri
- conditions - uz hostu grupu un/vai vienuma tagu balstīti nosacījumi (izteiksmes veidā)

Atstarpes ir atļautas tikai nosacījumu izteiksmē.

Aizstājējzīmju lietošana

- Aizstājējzīmi var izmantot, lai aizstātu hosta nosaukumu, vienuma atslēgu vai atsevišķu vienuma atslēgas parametru.
- Vai nu hostam, vai vienuma atslēgai jābūt norādītai bez aizstājējzīmes. Tātad `/host/*` un `/*/key` ir derīgi filtri, bet `/*/*` ir nederīgs.
- Aizstājējzīmi nevar izmantot hosta nosaukuma, vienuma atslēgas vai vienuma atslēgas parametra *daļai*.
- Aizstājējzīme neatbilst vairāk nekā vienam vienuma atslēgas parametram. Tāpēc aizstājējzīme jānorāda katram parametram atsevišķi (t.i., `key[abc,*,*]`).

Nosacījumu izteiksme

Nosacījumu izteiksme atbalsta:

- operandus:
  - group - hostu grupa
  - tag - vienuma tags

    - "<text>" - virknes konstante ar \ atsoļa rakstzīmi, lai atsoļotu " un \
- reģistrjutīgus virkņu salīdzināšanas operatorus: =, <>
- loģiskos operatorus: and, or, not
- grupēšanu ar iekavām: ( )

Virkņu konstanšu pēdiņas ir obligātas. Tiek atbalstīta tikai pilnas virknes salīdzināšana ar reģistrjutību.

Warning:
Norādot tagus filtrā (t.i. tag="tagname:value"), kols ":" tiek izmantots kā atdalītājs. Viss pēc tā tiek uzskatīts par taga vērtību. Tādēļ pašlaik netiek atbalstīta tāda taga nosaukuma norādīšana, kas satur ":".

Piemēri

Var izmantot sarežģītu filtru, atsaucoties uz vienuma atslēgu, hostu grupu un tagiem, kā parādīts piemēros:

Sintakses piemērs	Apraksts
/host/key[abc,*]	Atbilst līdzīgiem vienumiem šajā hostā.
/*/key	Atbilst vienam un tam pašam vienumam jebkurā hostā.
/*/key?[group="ABC" and tag="tagname:value"]	Atbilst vienam un tam pašam vienumam jebkurā hostā no ABC grupas ar tagiem 'tagname:value'.
/*/key[a,*,c]?[(group="ABC" and tag="Tag1") or (group="DEF" and (tag="Tag2" or tag="Tag3:value"))]	Atbilst līdzīgiem vienumiem jebkurā hostā no ABC vai DEF grupas ar attiecīgajiem tagiem.

Visiem norādītajiem vienumiem jāeksistē un jāvāc dati. Aprēķinos tiek iekļauti tikai iespējoti vienumi iespējotos hostos. Vienumi neatbalstītā stāvoklī netiek iekļauti.

Attention:
Ja atsauktā vienuma atslēga tiek mainīta, filtrs ir jāatjaunina manuāli.

Norādot vecākgrupas hostu grupu, tiek iekļauta vecākgrupa un visas ligzdotās hostu grupas ar to vienumiem.

Laika periods

Otrais parametrs ļauj norādīt laika periodu agregācijai. Laika periodu var izteikt tikai kā laiku; vērtību skaits (ar prefiksu **#**) netiek atbalstīts.

Ērtības labad šajā parametrā var izmantot atbalstītos vienību simbolus, piemēram, '5m' (piecas minūtes) '300s' (300 sekundes) vietā vai '1d' (viena diena) '86400' (86400 sekundes) vietā.

Funkcijai *last_foreach* laika periods ir neobligāts parametrs (atbalstīts kopš Zabbix 7.0), kuru var izlaist:

```
last_foreach(/*/key?[group="host group"])
```

Laika periods netiek atbalstīts ar funkciju *exists_foreach*.

Papildu parametri

bucket_rate_foreach

Trešo neobligāto parametru atbalsta funkcija *bucket_rate_foreach*:

```
bucket_rate_foreach(item filter,time period,<parameter number>)
```

kur <parameter number> ir "bucket" vērtības pozīcija vienuma atslēgā. Piemēram, ja "bucket" vērtība myItem[aaa,0.2] ir '0.2', tad tās pozīcija ir 2.

Noklusējuma vērtība parametram <parameter number> ir '1'.

count_foreach

Trešo un ceturto neobligāto parametru atbalsta funkcija *count_foreach*:

```
count_foreach(item filter,time period,<operator>,<pattern>)
```

Kur:

- **operator** ir nosacījuma operators vienuma vērtībām (jābūt dubultpēdiņās). Atbalstītie operators:
eq - vienāds
ne - nav vienāds
gt - lielāks
ge - lielāks vai vienāds
lt - mazāks
le - mazāks vai vienāds
like - atbilst, ja satur šablonu (reģistrjutīgs)
bitand - bitu AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta pattern
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta pattern

- **pattern** ir obligātais šablons (virknes argumentiem jābūt dubultpēdiņās); tiek atbalstīts, ja *operator* ir norādīts trešajā parametrā.

Komentāri:

- Neobligātos parametrus *operator* vai *pattern* nevar atstāt tukšus aiz komata, tos var tikai pilnībā izlaist.
- Ja kā trešais parametrs ir *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, kas atdalīti ar '/': **number_to_compare_with/mask**. count_foreach() aprēķina "bitwise AND" no vērtības un *mask* un salīdzina rezultātu ar *number_to_compare_with*. Ja "bitwise AND" rezultāts ir vienāds ar *number_to_compare_with*, vērtība tiek ieskaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, jānorāda tikai *mask* (bez '/').
- Ja kā trešais parametrs ir *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālās regulārās izteiksmes iestatījumiem. Regexp atbilstības vajadzībām peldošā komata vērtības vienmēr tiks attēlotas ar 4 zīmēm aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība starp decimālo (glabājas datubāzē) un bināro (izmanto Zabbix serveris) attēlojumu var ietekmēt 4. zīmi aiz komata.

Plašāku informāciju un foreach funkciju lietošanas piemērus skatiet sadaļā agregētie aprēķini.

Darbība atkarībā no pieejamības

Tālāk esošā tabula parāda, kā katra funkcija darbojas ierobežotas hosta/vienuma un vēstures datu pieejamības gadījumos.

Function	Disabled host	Unavailable host with data	Unavailable host without data	Disabled item	Unsupported item	Data retrieval error (SQL)
avg_foreach	ignorēt	atgriezt vidējo	ignorēt	ignorēt	ignorēt	ignorēt
bucket_rate_foreach	ignorēt	atgriezt bucket rate	ignorēt	ignorēt	ignorēt	ignorēt
count_foreach	ignorēt	atgriezt skaitu	0	ignorēt	ignorēt	ignorēt
exists_foreach	ignorēt	1	1	ignorēt	1	n/a
last_foreach	ignorēt	atgriezt pēdējo vērtību	ignorēt	ignorēt	ignorēt	ignorēt
max_foreach	ignorēt	atgriezt maksimumu	ignorēt	ignorēt	ignorēt	ignorēt
min_foreach	ignorēt	atgriezt minimumu	ignorēt	ignorēt	ignorēt	ignorēt
sum_foreach	ignorēt	atgriezt summu	ignorēt	ignorēt	ignorēt	ignorēt

Ja vienums tiek *ignorēts*, agregācijai nekas netiek pievienots.

2 Bitu funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
bitand	"Bitu UN" vērtība vienuma vērtībai un maskai.
bitlshift	Vienuma vērtības bitu nobīde pa kreisi.
bitnot	"Bitu NE" vērtība vienuma vērtībai.
bitor	"Bitu VAI" vērtība vienuma vērtībai un maskai.
bitrshift	Vienuma vērtības bitu nobīde pa labi.
bitxor	"Bitu izslēdzošais VAI" vērtība vienuma vērtībai un maskai.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu

- Izteiksmes tiek pieņemtas kā parametri
- Neobligātie funkciju parametri (vai parametru daļas) ir norādīti ar < >

bitand(value,mask)

”Bitu UN” vērtība starp vienuma vērtību un masku.
 Atbalstītie vērtību tipi: *Integer*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **mask** (obligāts) - 64 bitu neparakstīts vesels skaitlis (0 - 18446744073709551615).

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formātā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

Piemēri:

```
bitand(last(/host/key),12)=8 or bitand(last(/host/key),12)=4 #3. vai 4. bits ir iestatīts, bet ne abi vien
bitand(last(/host/key),20)=16 #3. bits nav iestatīts un 5. bits ir iestatīts
```

bitlshift(value,bits to shift)

Vienuma vērtības bitu nobīde pa kreisi.
 Atbalstītie vērtību tipi: *Integer*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **bits to shift** (obligāts) - nobīdāmo bitu skaits.

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formātā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

bitnot(value)

Vienuma vērtības ”bitu loģiskā NOT” vērtība.
 Atbalstītie vērtību tipi: *Integer*.

Parametrs:

- **value** - pārbaudāmā vērtība.

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formātā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

bitor(value,mask)

Vienuma vērtības un maskas ”bitu VAI” vērtība.
 Atbalstītie vērtību tipi: *Integer*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **mask** (obligāts) - 64 bitu bezzīmes vesels skaitlis (0 - 18446744073709551615).

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formātā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

bitrshift(value,bits to shift)

Vienuma vērtības bitu nobīde pa labi.
 Atbalstītie vērtību tipi: *Integer*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **bits to shift** (obligāts) - pārbīdāmo bitu skaits.

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formātā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

bitxor(value,mask)

Vienuma vērtības un maskas ”bitu izslēdzošais VAI” vērtība.
 Atbalstītie vērtību tipi: *Integer*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **mask** (obligāts) - 64 bitu vesels skaitlis bez zīmes (0 - 18446744073709551615).

Lai gan salīdzināšana tiek veikta bitu līmenī, visas vērtības ir jānorāda un tiek atgrieztas decimālajā formā. Piemēram, 3. bita pārbaude tiek veikta, salīdzinot ar 4, nevis 100.

Skatiet visas atbalstītās funkcijas.

3 Datuma un laika funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Attention:
Datuma un laika funkcijas nevar izmantot izteiksmē pašas par sevi; izteiksmē jābūt iekļautai vismaz vienai funkcijai no citas grupas, kas atsaucas uz hosta vienumu (izņemot funkciju nodata()). Detalizētu informāciju par to, kā datuma un laika funkcijas darbojas izteiksmēs, skatiet sadaļā Aprēķina laiks.

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
date	Pašreizējais datums YYYYMMDD formātā.
dayofmonth	Mēneša diena diapazonā no 1 līdz 31.
dayofweek	Nedēļas diena diapazonā no 1 līdz 7.
now	Sekunžu skaits kopš Epoch (1970. gada 1. janvāris, 00:00:00 UTC).
time	Pašreizējais laiks HHMMSS formātā.

Funkcijas informācija

date

Pašreizējais datums YYYYMMDD formātā.

Piemērs:

`date()<20220101`

dayofmonth

Mēneša diena diapazonā no 1 līdz 31.

Piemērs:

`dayofmonth()=1`

dayofweek

Nedēļas diena diapazonā no 1 līdz 7 (Pirmd. - 1, Svētd. - 7).

Piemērs (tikai darba dienas):

`dayofweek()<6`

Piemērs (tikai nedēļas nogale):

`dayofweek()>5`

now

Sekunžu skaits kopš Unix laikmeta sākuma (00:00:00 UTC, 1970. gada 1. janvāris).

Piemērs:

`now()<1640998800`

laiks

Pašreizējais laiks HHMMSS formātā.

Piemērs (tikai nakts laikā, 00:00-06:00):

`time()<060000`

Skatiet visas atbalstītās funkcijas.

4 Vēstures funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Funkcija	Apraksts
change	Atšķirības apjoms starp iepriekšējo un jaunāko vērtību.
changecount	Izmaiņu skaits starp blakus esošām vērtībām noteiktajā novērtēšanas periodā.
count	Vērtību skaits noteiktajā novērtēšanas periodā.
countunique	Unikālo vērtību skaits noteiktajā novērtēšanas periodā.
find	Atrod vērtības atbilstību noteiktajā novērtēšanas periodā.
first	Pirmā (vecākā) vērtība noteiktajā novērtēšanas periodā.
firstclock	Pirmās (vecākās) vērtības laikspiedols noteiktajā novērtēšanas periodā.
fuzzytime	Pārbauda, cik ļoti pasīvā aģenta laiks atšķiras no Zabbix servera/starpniekservera laika.
last	Jaunākā vērtība.
lastclock	N-tās jaunākās vērtības laikspiedols noteiktajā novērtēšanas periodā.
logeventid	Pārbauda, vai pēdējā žurnāla ieraksta notikuma ID atbilst regulārajai izteiksmei.
logseverity	Pēdējā žurnāla ieraksta nopietnības līmenis.
logsource	Pārbauda, vai pēdējā žurnāla ieraksta avots atbilst regulārajai izteiksmei.
logtimestamp	N-tās jaunākās žurnāla vienuma vērtības žurnāla ziņojuma laikspiedols.
monodec	Pārbauda, vai ir notikusi monotona vērtību samazināšanās.
monoinc	Pārbauda, vai ir notikusi monotona vērtību palielināšanās.
nodata	Pārbauda, vai nav saņemti dati.
percentile	Perioda P-tā procentile, kur P (procenti) ir norādīts ar trešo parametru.
rate	Vidējais pieauguma ātrums sekundē monotoni pieaugošam skaitītājam noteiktajā laika periodā.

Bieži izmantotie parametri

- `/host/key` ir bieži izmantots obligāts pirmais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi
- `(sec|#num)<:time shift>` ir bieži izmantots otrais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi, kur:
  - **sec** - maksimālais novērtēšanas periods sekundēs (var izmantot laika sufiksus), vai
  - **#num** - maksimālais novērtēšanas diapazons pēdējās savāktajās vērtībās (ja pirms tā ir jaucējzīme)
  - **time shift** (neobligāts) ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Skatiet vairāk informācijas par time shift norādīšanu.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Neobligātie funkciju parametri (vai parametru daļas) ir norādīti ar < >
- Katras funkcijas specifiskie parametri ir aprakstīti pie attiecīgās funkcijas
- Parametri `/hosts/key` un `(sec|#num)<:time shift>` nekad nedrīkst būt pēdiņās

change(/host/key)

Atšķirības apjoms starp iepriekšējo un jaunāko vērtību.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.
 Virkņu gadījumā atgriež: 0 - vērtības ir vienādas; 1 - vērtības atšķiras.

Parametri: skatiet kopīgos parametrus.

Komentāri:

- Tiks aprēķināta skaitliskā starpība, kā redzams šajos ienākošo piemērvērtību piemēros ('iepriekšējā' un 'jaunākā' vērtība = starpība):
'1' un '5' = +4
'3' un '1' = −2
'0' un '-2.5' = −2.5.
- Trigeros noder pēkšņu lēcienu (vai kritumu), skaitītāju atiestatīšanas un skaitlisku stāvokļu pāreju noteikšanai.
- Skatiet arī: abs salīdzinājumam.

Piemēri:

```
change(/host/system.uptime)<0 #system uptime change has been negative since the last value (indicating a r
change(/host/system.cpu.load[all,avg1])>2 #CPU load (for one minute) has jumped by more than 2 since the l
change(/host/vfs.fs.size[/,free])<-1G #free disk space has dropped by more than 1 GB between checks
```

changecount(/host/key,(sec|#num)<:time shift>,<mode>)

Izmaiņu skaits starp blakus esošām vērtībām noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **mode** (jābūt dubultpēdiņās) — iespējamās vērtības: *all* — skaitīt visas izmaiņas (noklusējums); *dec* — skaitīt samazinājumus; *inc* — skaitīt palielinājumus

Komentāri:

- Ne-skaitlisku vērtību tipiem parametrs *mode* tiek ignorēts.
- Trigeros noderīgs biežu stāvokļa izmaiņu uztveršanai (kas norāda uz nestabilitāti).

Piemēri:

```
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>)

Vērtību skaits noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **operator** (jābūt dubultpēdiņās) Atbalstītie `operators`:
eq - vienāds (noklusējums integer, float)
ne - nav vienāds
gt - lielāks
ge - lielāks vai vienāds
lt - mazāks
le - mazāks vai vienāds
like (noklusējums string, text, log) - atbilst, ja satur šablonu (reģistrjutīgs)
bitand - bitu AND
regexp - reģistrjutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
iregexp - reģistrnejutīga atbilstība regulārajai izteiksmei, kas norādīta `pattern`
- **pattern** - nepieciešamais šablons (string argumentiem jābūt dubultpēdiņās)

Komentāri:

- Float vienumi tiek salīdzināti ar precizitāti 2.22e-16.
- *like* netiek atbalstīts kā operators integer vērtībām.
- *like* un *bitand* netiek atbalstīti kā operatori float vērtībām.
- String, text un log vērtībām tiek atbalstīti tikai operatori *eq*, *ne*, *like*, *regexp* un *iregexp*.
- Ja kā operators tiek izmantots *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, atdalītus ar '/': **number_to_compare_with/mask**. count() aprēķina "bitu AND" no vērtības un *mask* un salīdzina rezultātu ar *number_to_compare_with*. Ja "bitu AND" rezultāts ir vienāds ar *number_to_compare_with*, vērtība tiek ieskaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, jānorāda tikai *mask* (bez '/').
- Ja kā operators tiek izmantots *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālās regulārās izteiksmes iestatījumiem. Regexp salīdzināšanas vajadzībām float vērtības vienmēr tiks attēlotas ar 4 cipariem aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība starp decimālo (glabājas datubāzē) un bināro (izmanto Zabbix serveris) attēlojumu var ietekmēt 4. ciparu aiz komata.

Piemēri:

```
count(/host/icmpping,30m,,"0")>5 #ping nav izdevies vairāk nekā 5 reizes 30 minūtēs
count(/host/key,10m,"like","error") #vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas satur 'error'
count(/host/key,10m,,12) #vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas ir vienādas ar '12'
count(/host/key,10m,"gt",12) #vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas ir lielākas par '12'
count(/host/key,#10,"gt",12) #vērtību skaits pēdējās 10 vērtībās līdz šim brīdim, kas ir lielākas par '12'
count(/host/key,10m:now-1d,"gt",12) #vērtību skaits starp periodu pirms 24 stundām un pirms 24 stundām mīn
count(/host/key,10m,"bitand","6/7") #vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kurām 3 vismazāk n
count(/host/key,10m:now-1d) #vērtību skaits starp periodu pirms 24 stundām un pirms 24 stundām mīnus 10 mi
```

countunique(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Unikālo vērtību skaits definētajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus

- **operator** (jābūt dubultpēdiņās). Atbalstītie `operatori`:
eq - vienāds (noklusējums integer, float)
ne - nav vienāds
gt - lielāks par
ge - lielāks vai vienāds
lt - mazāks par
le - mazāks vai vienāds
like (noklusējums string, text, log) - atbilst, ja satur šablonu (reģistrjutīgs)
bitand - bitu UN
regexp - reģistrjutīga atbilstība `pattern` norādītajai regulārajai izteiksmei
iregexp - reģistrnejutīga atbilstība `pattern` norādītajai regulārajai izteiksmei
- **pattern** - nepieciešamais šablons (virknes argumentiem jābūt dubultpēdiņās)

Komentāri:

- Float vienumi tiek salīdzināti ar precizitāti 2.22e-16.
- *like* netiek atbalstīts kā operators integer vērtībām.
- *like* un *bitand* netiek atbalstīti kā operatori float vērtībām.
- String, text un log vērtībām tiek atbalstīti tikai operatori *eq*, *ne*, *like*, *regexp* un *iregexp*.
- Ja kā operators tiek izmantots *bitand*, ceturto `pattern` parametru var norādīt kā divus skaitļus, atdalītus ar '/': **number_to_compare_with/mask**. countunique() aprēķina "bitu UN" no vērtības un *mask* un salīdzina rezultātu ar *number_to_compare_with*. Ja "bitu UN" rezultāts ir vienāds ar *number_to_compare_with*, vērtība tiek ieskaitīta.
Ja *number_to_compare_with* un *mask* ir vienādi, var norādīt tikai *mask* (bez '/').
- Ja kā operators tiek izmantots *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek mantota no globālās regulārās izteiksmes iestatījumiem. Regexp salīdzināšanas vajadzībām float vērtības vienmēr tiks attēlotas ar 4 cipariem aiz '.'. Ņemiet vērā arī to, ka lieliem skaitļiem atšķirība starp decimālo (glabājas datubāzē) un bināro (izmanto Zabbix serveris) attēlojumu var ietekmēt 4. ciparu aiz komata.

Piemēri:

```
countunique(/host/key,10m) #unikālo vērtību skaits pēdējās 10 minūtēs līdz šim brīdim
countunique(/host/key,10m,"like","error") #unikālo vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas
countunique(/host/key,10m,,12) #unikālo vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas ir vienādas
countunique(/host/key,10m,"gt",12) #unikālo vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kas ir liel
countunique(/host/key,#10,"gt",12) #unikālo vērtību skaits pēdējās 10 vērtībās līdz šim brīdim, kas ir lie
countunique(/host/key,10m:now-1d,"gt",12) #unikālo vērtību skaits starp periodu pirms 24 stundām un 10 min
countunique(/host/key,10m,"bitand","6/7") #unikālo vērtību skaits pēdējās 10 minūtēs līdz šim brīdim, kurā
countunique(/host/key,10m:now-1d) #unikālo vērtību skaits starp periodu pirms 24 stundām un 10 minūtēm līd
```

find(/host/key,(sec|#num)<:time shift>,<operator>,<pattern>)

Atrod vērtības atbilstību definētajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.
 Atgriež: 1 - atrasts; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus
- **sec** vai **#num** (neobligāts) - pēc noklusējuma tiek izmantota jaunākā vērtība, ja nav norādīts
- **operator** (jābūt dubultpēdiņās) Atbalstītie `operatori`:
eq - vienāds (noklusējums integer, float)
ne - nav vienāds
gt - lielāks
ge - lielāks vai vienāds
lt - mazāks
le - mazāks vai vienāds
like (noklusējums string, text, log) - atbilst, ja satur `pattern` norādīto virkni (reģistrjutīgs)
bitand - bitu UN
regexp - reģistrjutīga atbilstība `pattern` norādītajai regulārajai izteiksmei
iregexp - reģistrnejutīga atbilstība `pattern` norādītajai regulārajai izteiksmei
- **pattern** - nepieciešamais raksts (virknes argumentiem jābūt dubultpēdiņās); Perl Compatible Regular Expression (PCRE) regulārā izteiksme, ja `operator` ir *regexp*, *iregexp*

Komentāri:

- Ja tiek apstrādāta vairāk nekā viena vērtība, tiek atgriezts '1', ja ir vismaz viena atbilstoša vērtība.
- *like* netiek atbalstīts kā operators integer vērtībām.
- *like* un *bitand* netiek atbalstīti kā operatori float vērtībām.
- String, text un log vērtībām tiek atbalstīti tikai operatori *eq*, *ne*, *like*, *regexp* un *iregexp*.
- Ja kā operators tiek izmantots *regexp* vai *iregexp*, ceturtais `pattern` parametrs var būt parasta vai globāla (sākas ar '@') regulārā izteiksme. Globālo regulāro izteiksmju gadījumā reģistrjutība tiek pārmantota no globālās regulārās izteiksmes iestatījumiem.

Piemēri:

```
find(/host/key,10m,"like","error") #atrod vērtību, kas satur 'error', pēdējo 10 minūšu laikā līdz šim brīd
find(/host/agent.version,,"like","beta")=1 #Zabbix aģents izmanto beta versiju, tas ir jāatjaunina
find(/host/log[/var/log/nginx/access.log],,"regexp"," 500 ")=1 #ir atrasta iekšējā tīmekļa servera kļūda
```

first(/host/key,sec<:time shift>)

Pirmā (vecākā) vērtība definētajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus

Skatiet arī last().

Piemērs:

```
first(/host/key,1h) #retrieve the oldest value within the last hour until now
```

firstclock(/host/key,sec<:laika nobīde>)

Vecākās vērtības laikspiedols definētajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus.

Funkcija neizdodas ar kļūdu, ja norādītajā periodā nav savākti dati.

Skatiet arī lastclock().

Piemēri:

```
firstclock(/host/key,1h) #iegūt vecākās vērtības laikspiedolu pēdējās stundas laikā
firstclock(/host/key,1h:now-24h) #iegūt vecākās vērtības laikspiedolu pēdējās stundas laikā pirms dienas
```

fuzzytime(/host/key,sec)

Pārbauda, cik lielā mērā pasīvā aģenta laiks atšķiras no Zabbix servera/starpniekservera laika.
 Atbalstītie vērtību tipi: *Float*, *Integer*.
 Atgriež: 1 - atšķirība starp pasīvā vienuma vērtību (kā laika zīmogu) un Zabbix servera/starpniekservera laika zīmogu (vērtības ievākšanas laiku) ir mazāka vai vienāda ar *sec* sekundēm; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus.

Komentāri:

- Parasti izmanto kopā ar vienumu 'system.localtime', lai pārbaudītu, vai lokālais laiks ir sinhronizēts ar Zabbix servera lokālo laiku. *Ņemiet vērā*, ka 'system.localtime' Zabbix aģentam ir jākonfigurē kā pasīvā pārbaude; Zabbix aģentā 2 to var konfigurēt arī kā aktīvo pārbaudi.
- Var izmantot arī ar atslēgu `vfs.file.time[/path/file,modify]`, lai pārbaudītu, ka fails ilgu laiku nav ticis atjaunināts.
- Šo funkciju nav ieteicams izmantot sarežģītās trigeru izteiksmēs (kur iesaistīti vairāki vienumi), jo tā var radīt negaidītus rezultātus (laika starpība tiks mērīta ar jaunāko metriku), piemēram, `fuzzytime(/Host/system.localtime,60s)=0 or last(/Host/trap)<>0`.

Piemēri:

```
fuzzytime(/host/system.localtime,5m)=0 #klienta lokālais laiks atšķiras no Zabbix servera/starpniekservera
fuzzytime(/host/system.localtime,5m)=0 and nodata(/host/system.localtime,10m)=0 #klienta lokālais laiks at
```

last(/host/key,<#num<:time shift»)

Pēdējā vērtība.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus
- **#num** (neobligāts) - N-tā pēdējā vērtība

Komentāri:

- Ņemiet vērā, ka ar jaucējzīmi apzīmēts laika periods (#N) šeit darbojas citādi nekā daudzās citās funkcijās. Piemēram: `last(/host/key)` vienmēr ir vienāds ar `last(/host/key,#1)`; `last(/host/key,#3)` - trešā pēdējā vērtība (*nevis* pēdējās trīs vērtības).
- Zabbix negarantē precīzu vērtību secību, ja vēsturē vienas sekundes laikā pastāv vairāk nekā divas vērtības.
- Skatiet arī first().

Piemēri:

```
last(/host/key) #iegūt pēdējo vērtību
last(/host/key,#2) #iegūt iepriekšējo vērtību
last(/host/key,#1)<>last(/host/key,#2) #pēdējās divas vērtības atšķiras
```

lastclock(/host/key,<#num<:time shift»)

N-tās jaunākās vērtības laikspiedols noteiktajā novērtēšanas periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*, *String*, *Text*, *Log*.

Parametri:

- Skatiet kopīgos parametrus;
- **#num** (neobligāts) — N-tā jaunākā vērtība.

Funkcija neizdodas ar kļūdu, ja norādītajā periodā nav savākti dati vai nav savākta N-tā vērtība.

Skatiet arī firstclock().

Piemēri:

```
lastclock(/host/key) #iegūt jaunākās vērtības laikspiedolu
lastclock(/host/key,#2) #iegūt 2. jaunākās vērtības laikspiedolu
now()-lastclock(/host/heartbeat.item)>300 #kopš heartbeat vienuma atjaunināšanas ir pagājušas vairāk nekā
lastclock(/host/system.cpu.load[all,avg1])-lastclock(/host/system.cpu.load[all,avg1],#2)>300 #atjaunināšan
((now()-lastclock(/host/system.cpu.load[all,avg1])<120 and last(/host/system.cpu.load[all,avg1])>5)) #CPU
```

logeventid(/host/key,<#num<:time shift»,<pattern>)

Pārbauda, vai pēdējā žurnāla ieraksta notikuma ID atbilst regulārajai izteiksmei.
 Atbalstītie vērtību tipi: *Log*.
 Atgriež: 0 - neatbilst; 1 - atbilst.

Parametri:

- Skatiet kopīgos parametrus
- **#num** (neobligāts) - N-tā jaunākā vērtība
- **pattern** (neobligāts) - regulārā izteiksme, kas apraksta nepieciešamo šablonu, Perl Compatible Regular Expression (PCRE) stilā (virknes argumentiem jābūt dubultpēdiņās)

Piemēri:

```
logeventid(/host/eventlog[Security],,"4625")=1 #atrasts žurnāla ieraksts ar ID, kas atbilst "4625" (neveik
logeventid(/host/eventlog[System],,"6008|41")=1 #atrasts žurnāla ieraksts ar ID, kas atbilst "6008" vai "4
```

logseverity(/host/key,<#num<:time shift»)

Pēdējā žurnāla ieraksta žurnāla smaguma pakāpe.
 Atbalstītie vērtību tipi: *Log*.
 Atgriež: 0 - noklusējuma smaguma pakāpe; N - smaguma pakāpe (vesels skaitlis, noderīgs Windows notikumu žurnāliem: 1 - Informācija, 2 - Brīdinājums, 4 - Kļūda, 7 - Neveiksmīga auditēšana, 8 - Veiksmīga auditēšana, 9 - Kritiska, 10 - Detalizēta).

Parametri:

- Skatiet kopīgos parametrus
- **#num** (neobligāts) - N-tā jaunākā vērtība

Zabbix ņem žurnāla smaguma pakāpi no Windows notikumu žurnāla lauka **Information**.

Piemēri:

```
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>)

Pārbauda, vai pēdējā žurnāla ieraksta žurnāla avots atbilst regulārajai izteiksmei.
 Atbalstītie vērtību tipi: *Log*.
 Atgriež: 0 - neatbilst; 1 - atbilst.

Parametri:

- Skatiet kopīgos parametrus
- **#num** (neobligāts) - N-tā jaunākā vērtība
- **pattern** (neobligāts) - regulārā izteiksme, kas apraksta nepieciešamo šablonu, Perl Compatible Regular Expression (PCRE) stilā (virknes argumentiem jābūt dubultpēdiņās)

Parasti izmanto Windows notikumu žurnāliem.

Piemēri:

```
logsource(/host/eventlog[Application],,"MSSQLSERVER")=1 #a log entry with source matching "MSSQLSERVER" fo
logsource(/host/eventlog[System],,"Service Control Manager")=1 #a log entry with source matching "Service
logsource(/host/eventlog[System],,"Service Control Manager")=1 and logeventid(/host/eventlog[System],,"703
```

logtimestamp(/host/key,<#num<:time shift»)

N-tās jaunākās žurnāla vienuma vērtības žurnāla ziņojuma laikspiedols.
 Atbalstītie vērtību tipi: *Log*.

Parametri:

- Skatiet kopīgos parametrus;
- **#num** (neobligāts) — N-tā jaunākā vērtība.

Komentāri:

- Laika nobīdes aprēķins ir balstīts uz vienuma vērtības pulksteni, nevis uz žurnāla ziņojuma laikspiedolu;
- Funkcija neizdodas ar kļūdu, ja:
  - tiek saņemts vienums, kas nav žurnāla tipa;
  - norādītajā periodā nav savākti dati;
  - nav savākta N-tā vērtība;
  - žurnāla ziņojums nesatur laikspiedola vērtību.

Piemēri:

```
logtimestamp(/host/key) #iegūt jaunākā žurnāla ziņojuma laikspiedolu
logtimestamp(/host/key,#2) #iegūt 2. jaunākā žurnāla ziņojuma laikspiedolu
logtimestamp(/host/key,#3:now-1d) #iegūt 3. jaunākā žurnāla ziņojuma laikspiedolu pirms dienas
```

monodec(/host/key,(sec|#num)<:time shift>,<mode>)

Pārbauda, vai ir noticis monotons vērtību samazinājums.
 Atbalstītie vērtību tipi: *Integer*.
 Atgriež: 1 - ja visi elementi laika periodā nepārtraukti samazinās; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus
- **mode** (jābūt dubultpēdiņās) - *weak* (katra vērtība ir mazāka vai tāda pati kā iepriekšējā; pēc noklusējuma) vai *strict* (katra vērtība ir samazinājusies)

Piemēri:

```
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 #nginx procesu skaits 10 minūšu laikā ir monotoni samazināji
```

monoinc(/host/key,(sec|#num)<:time shift>,<mode>)

Pārbauda, vai ir noticis monotons vērtību pieaugums.
 Atbalstītie vērtību tipi: *Integer*.
 Atgriež: 1 - ja visi elementi laika periodā nepārtraukti palielinās; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus
- **mode** (jābūt dubultpēdiņās) - *weak* (katra vērtība ir lielāka vai tāda pati kā iepriekšējā; pēc noklusējuma) vai *strict* (katra vērtība ir palielinājusies)

Piemēri:

```
monoinc(/host/system.localtime,#3,"strict")=0 #the system local time has not been increasing consistently
monoinc(/host/vfs.dir.count[/mnt/data/logs],24h,"weak")=0 #trigger if the file count has stagnated over 24
```

nodata(/host/key,sec,<mode>)

Pārbauda, vai nav saņemti dati.
 Atbalstītie vērtību tipi: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Atgriež: 1 - ja noteiktajā laika periodā dati nav saņemti; 0 - pretējā gadījumā.

Parametri:

- Skatiet kopīgos parametrus
- **sec** - periods nedrīkst būt mazāks par 30 sekundēm, jo vēstures sinhronizētāja process šo funkciju aprēķina tikai ik pēc 30 sekundēm; `nodata(/host/key,0)` nav atļauts
- **mode** - ja iestatīts uz *strict* (dubultpēdiņās), šī funkcija nebūs jutīga pret starpniekserveris pieejamību (sīkāku informāciju skatiet komentāros)

Komentāri:

- pēc noklusējuma starpniekserveris uzraudzītie 'nodata' trigeri ir jutīgi pret starpniekserveris pieejamību — ja starpniekserveris kļūst nepieejams, 'nodata' trigeri neaktivizēsies uzreiz pēc savienojuma atjaunošanas, bet izlaidīs datus par aizkavēto periodu. Ņemiet vērā, ka pasīvajiem starpniekserveriem apspiešana tiek aktivizēta, ja savienojums tiek atjaunots vairāk nekā pēc 15 sekundēm un ne mazāk kā pēc 2 sekundēm. Aktīvajiem starpniekserveriem apspiešana tiek aktivizēta, ja

savienojums tiek atjaunots vairāk nekā pēc 15 sekundēm. Lai izslēgtu jutību pret starpniekserveris pieejamību, izmantojiet trešo parametru, piemēram: `nodata(/host/key,5m,"strict")`; šajā gadījumā funkcija aktivizēsies, tiklīdz būs pagājis novērtēšanas periods (piecas minūtes) bez datiem.

- Šī funkcija parādīs kļūdu, ja 1. parametra periodā:
- nav datu un Zabbix serveris tika restartēts
- nav datu un apkope tika pabeigta
- nav datu un vienums tika pievienots vai atkārtoti iespējots.
- Kļūdas tiek parādītas *Info* kolonnā trigera konfigurācijā.
- Šī funkcija var nedarboties pareizi, ja starp Zabbix serveris, starpniekserveris un aģents ir laika atšķirības. Skatiet arī: Laika sinhronizācijas prasība.

Piemērs:

```
nodata(/host/agent.ping,5m)=1 #trigeris, ja 5 minūtes nav saņemti dati no Zabbix aģents
```

percentile(/host/key,(sec|#num)<:time shift>,percentage)

P-tā procentile periodā, kur P (procenti) ir norādīti ar trešo parametru.
 Atbalstītie vērtību tipi: *Float*, *Integer*.

Parametri:

- Skatiet kopīgos parametrus
- **percentage** — peldošā komata skaitlis no 0 līdz 100 (ieskaitot), ar ne vairāk kā 4 cipariem aiz komata

Piemēri:

```
percentile(/host/net.if.in[eth0,bytes],1h,95)>1000000 #95. procentile tīkla ienākošajai plūsmai (baiti/sek
percentile(/host/system.cpu.util,5m,95)>80 #95. procentile CPU noslodzes user-time procentuālajai vērtībai
percentile(/host/icmppingsec[192.168.0.2],15m,95)>0.15 #lielākā daļa latentuma mērījumu ir zem 150 ms, bet
percentile(/host/net.if.in[eth0,bytes],1h,50) #aprēķināt 50. procentili (mediānas vērtību) ienākošajai tīk
(percentile(/host/net.if.in[eth0,bytes],1h,50)+percentile(/host/net.if.in[eth0,bytes],1h,51))/2 #aprēķināt
```

rate(/host/key,sec<:time shift>)

Monotoni pieaugoša skaitītāja vidējais pieaugums sekundē noteiktajā laika periodā.
 Atbalstītie vērtību tipi: *Float*, *Integer*.

Parametri:

- Skatiet kopīgos parametrus

Funkcionāli atbilst PromQL funkcijai 'rate'.

Piemēri:

```
rate(/host/key,30s) #ja monotons pieaugums 30 sekunžu laikā ir 20, šī funkcija atgriež 0.67.
rate(/host/net.if.in[eth0,bytes],5m)>500000 #ienākošās saskarnes datplūsmas ātrums uz eth0 pēdējo 5 minūšu
rate(/host/app.requests.count,1m)>100 #pieprasījumu skaitītājs pēdējās minūtes laikā ir pieaudzis par >100
```

Skatiet visas atbalstītās funkcijas.

5 Trendu funkcijas

Trendu funkcijas, atšķirībā no vēstures funkcijām, aprēķiniem izmanto trendu datus.

Trendi glabā stundas apkopotās vērtības. Trendu funkcijas izmanto šos stundas vidējos rādītājus, un tādējādi ir noderīgas ilgtermiņa analīzei.

Trendu funkciju rezultāti tiek kešoti, tāpēc vairāki vienas un tās pašas funkcijas izsaukumi ar vienādiem parametriem informāciju no datubāzes iegūst tikai vienu reizi. Trendu funkciju kešatmiņu kontrolē servera parametrs TrendFunctionCacheSize.

Trigeri, kas atsaucas **tikai** uz trendu funkcijām, tiek novērtēti vienu reizi mazākajā laika periodā izteiksmē. Piemēram, tāds trigeris kā

```
trendavg(/host/key,1d:now/d) > 1 or trendavg(/host/key2,1w:now/w) > 2
```

tiks novērtēts vienu reizi dienā. Ja trigeris satur gan trendu, gan vēstures (vai datuma un laika un/vai nodata()) funkcijas, tas tiek aprēķināts saskaņā ar parastajiem principiem.

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
baselinedev	Atgriež noviržu skaitu (pēc stddevpop algoritma) starp pēdējo datu periodu un tiem pašiem datu periodiem iepriekšējās sezonās.
baselinewma	Aprēķina bāzlīniju, vidējojot datus no tā paša laika intervāla vairākos vienādos laika periodos (”sezonās”), izmantojot svērto slīdošā vidējā algoritmu.
trendavg	Trendu vērtību vidējā vērtība noteiktajā laika periodā.
trendcount	Veiksmīgi iegūto vēstures vērtību skaits, kas izmantotas trenda vērtības aprēķināšanai noteiktajā laika periodā.
trendmax	Maksimālā trendu vērtība noteiktajā laika periodā.
trendmin	Minimālā trendu vērtība noteiktajā laika periodā.
trendstl	Atgriež anomāliju īpatsvaru noteikšanas periodā — decimālvērtību no 0 līdz 1, kas ir `((anomāliju vērtību skaits)/(kopējais vērtību skaits))`.
trendsum	Trendu vērtību summa noteiktajā laika periodā.

Bieži izmantotie parametri

- `/host/key` ir bieži izmantots obligātais pirmais parametrs
- `time period:time shift` ir bieži izmantots otrais parametrs, kur:
  - **time period** - laika periods (minimums ’1h’), definēts kā <N><time unit>, kur `N` - laika vienību skaits, `time unit` - h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads).
  - **time shift** - laika perioda nobīde (skatiet funkciju piemērus)

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Neobligātie funkciju parametri (vai parametru daļas) ir norādīti ar < >
- Katras funkcijas specifiskie parametri ir aprakstīti pie attiecīgās funkcijas
- Parametri `/hosts/key` un `time period:time shift` nekad nedrīkst būt pēdiņās

baselinedev(/host/key,data period:time shift,season unit,num seasons)

Atgriež noviržu skaitu (pēc stddevpop algoritma) starp pēdējo datu periodu un tiem pašiem datu periodiem iepriekšējās sezonās.

Parametri:

- Skatiet kopīgos parametrus
- **data period** - datu apkopošanas periods sezonas ietvaros, definēts kā <N><time unit>, kur:
`N` - laika vienību skaits
`time unit` - h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads), jābūt vienādam ar sezonu vai mazākam par to
- **season unit** - kalendārā vienība, kas definē vienu sezonu (h, d, w, M, y); nevar būt mazāka par data period
- **num seasons** - novērtējamo sezonu skaits

Piemēri:

```
baselinedev(/host/key,1d:now/d,"M",6) #aprēķina populācijas standartnoviržu skaitu starp iepriekšējo dienu
baselinedev(/host/key,1h:now/h,"d",10)  #aprēķina populācijas standartnoviržu skaitu starp iepriekšējo stu
```

baselinewma(/host/key,data period:time shift,season unit,num seasons)

Aprēķina bāzlīniju, vidējojot datus no viena un tā paša laika intervāla vairākos vienādos laika periodos (”sezonās”), izmantojot svērto slīdošā vidējā algoritmu.

Parametri:

- Skatiet kopīgos parametrus
- **data period** - datu apkopošanas periods sezonas ietvaros, definēts kā <N><time unit>, kur:
`N` - laika vienību skaits
`time unit` - h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads), jābūt vienādam ar sezonu vai mazākam par to
Time shift - laika perioda nobīde, nosaka datu apkopošanas laika intervāla beigas sezonās (skatiet piemērus)
- **season unit** - kalendārā vienība, kas definē vienu sezonu (h, d, w, M, y); nevar būt mazāka par data period
- **num seasons** - novērtējamo sezonu skaits

Piemēri:

```
baselinewma(/host/key,1h:now/h,"d",3) #bāzlīnijas aprēķināšana no tās pašas diennakts stundas pēdējās 3 pi
baselinewma(/host/key,2h:now/h,"d",3) #bāzlīnijas aprēķināšana no tā paša divu stundu intervāla pēdējās 3
baselinewma(/host/key,1d:now/d,"M",4) #bāzlīnijas aprēķināšana no tās pašas mēneša dienas kā "vakar" 4 mēn
```

trendavg(/host/key,time period:time shift)

Tendenču vērtību vidējā vērtība noteiktajā laika periodā.

Parametri:

- Skatiet kopīgos parametrus

Piemēri:

```
trendavg(/host/key,1h:now/h) #vidējā vērtība iepriekšējai stundai (piem., 12:00-13:00)
trendavg(/host/key,1h:now/h-1h) #vidējā vērtība pirms divām stundām (11:00-12:00)
trendavg(/host/key,1h:now/h-2h) #vidējā vērtība pirms trim stundām (10:00-11:00)
trendavg(/host/key,1M:now/M-1y) #vidējā vērtība iepriekšējam mēnesim pirms gada
```

trendcount(/host/key,time period:time shift)

Veiksmīgi izgūto vēstures vērtību skaits, kas izmantots, lai aprēķinātu trenda vērtību noteiktajā laika periodā.

Parametri:

- Skatiet kopīgos parametrus

Piemēri:

```
trendcount(/host/key,1h:now/h) #vērtību skaits iepriekšējai stundai (piem., 12:00-13:00)
trendcount(/host/key,1h:now/h-1h) #vērtību skaits pirms divām stundām (11:00-12:00)
trendcount(/host/key,1h:now/h-2h) #vērtību skaits pirms trim stundām (10:00-11:00)
trendcount(/host/key,1M:now/M-1y) #vērtību skaits iepriekšējam mēnesim pirms gada
```

trendmax(/host/key,time period:time shift)

Maksimālā vērtība tendenču vērtībās norādītajā laika periodā.

Parametri:

- Skatiet kopīgos parametrus

Piemēri:

```
trendmax(/host/key,1h:now/h) #maksimālā vērtība iepriekšējai stundai (piemēram, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → aprēķināt starpību starp maksimālo un minimā
trendmax(/host/key,1h:now/h-1h) #maksimālā vērtība pirms divām stundām (11:00-12:00)
trendmax(/host/key,1h:now/h-2h) #maksimālā vērtība pirms trim stundām (10:00-11:00)
trendmax(/host/key,1M:now/M-1y) #maksimālā vērtība iepriekšējam mēnesim pirms gada
```

trendmin(/host/key,time period:time shift)

Minimums trenda vērtībās noteiktajā laika periodā.

Parametri:

- Skatiet kopīgos parametrus

Piemēri:

```
trendmin(/host/key,1h:now/h) #minimums iepriekšējai stundai (piemēram, 12:00-13:00)
trendmax(/host/key,1h:now/h) - trendmin(/host/key,1h:now/h) → aprēķināt starpību starp maksimālo un minimā
trendmin(/host/key,1h:now/h-1h) #minimums pirms divām stundām (11:00-12:00)
trendmin(/host/key,1h:now/h-2h) #minimums pirms trim stundām (10:00-11:00)
trendmin(/host/key,1M:now/M-1y) #minimums iepriekšējam mēnesim pirms gada
```

trendstl(/host/key,eval period:time shift,detection period,season,<deviations>,<devalg>,<s window>)

Atgriež anomāliju biežumu noteikšanas periodā — decimālvērtību no 0 līdz 1, kas ir `((anomāliju vērtību skaits)/(kopējais vērtību skaits))`.

Parametri:

- Skatiet kopīgos parametrus

- **eval period** — laika periods, kas jāsadala komponentēs (minimums '1h'), definēts kā <N><time unit>, kur
`N` — laika vienību skaits
`time unit` — h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads)
- **detection period** — laika periods pirms eval period beigām, kuram tiek aprēķinātas anomālijas (minimums '1h', nevar būt garāks par eval period), definēts kā <N><time unit>, kur
`N` — laika vienību skaits
`time unit` — h (stunda), d (diena), w (nedēļa)
- **season** — īsākais laika periods, kurā ir sagaidāms atkārtojošs modelis ("sezona") (minimums '2h', nevar būt garāks par eval period, ierakstu skaitam eval period jābūt lielākam par divkāršu iegūtās frekvences vērtību (season/h)), definēts kā <N><time unit>, kur
`N` — laika vienību skaits
`time unit` — h (stunda), d (diena), w (nedēļa)
- **deviations** — noviržu skaits (aprēķināts ar devalg), ko uzskatīt par anomāliju (var būt decimāldaļa), (jābūt lielākam vai vienādam ar 1, noklusējuma vērtība ir 3)
- **devalg** (jābūt dubultpēdiņās) — novirzes algoritms; var būt *stddevpop*, *stddevsamp* vai *mad* (noklusējums)
- **s window** — loess loga diapazons (lagos) sezonālās komponentes iegūšanai (noklusējuma vērtība ir 10 * ierakstu skaits eval period + 1)

Piemēri:

```
trendstl(/host/key,100h:now/h,10h,2h) #analizē pēdējās 100 trendu datu stundas, nosaka anomāliju biežumu š
trendstl(/host/key,100h:now/h-10h,100h,2h,2.1,"mad") #analizē 100 stundu trendu datu periodu līdz 10 stund
trendstl(/host/key,100d:now/d-1d,10d,1d,4,,10) #analizē 100 dienu trendu datus līdz vienai dienai pagātnē,
trendstl(/host/key,1M:now/M-1y,1d,2h,,"stddevsamp") #analizē iepriekšējo mēnesi pirms gada, nosaka anomāli
```

trendsum(/host/key,time period:time shift)

Tendenču vērtību summa noteiktajā laika periodā.

Parametri:

- Skatiet kopīgos parametrus

Piemēri:

```
trendsum(/host/key,1h:now/h) #summa iepriekšējai stundai (piemēram, 12:00-13:00)
trendsum(/host/key,1h:now/h-1h) #summa periodam pirms divām stundām (11:00-12:00)
trendsum(/host/key,1h:now/h-2h) #summa periodam pirms trim stundām (10:00-11:00)
trendsum(/host/key,1M:now/M-1y) #summa iepriekšējam mēnesim pirms gada
```

Skatiet visas atbalstītās funkcijas.

6 Matemātiskās funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Matemātiskās funkcijas tiek atbalstītas float un integer vērtību tipiem, ja vien nav norādīts citādi.

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
abs	Vērtības absolūtā vērtība.
acos	Vērtības arkuskosinuss kā leņķis, izteikts radiānos.
asin	Vērtības arkussinuss kā leņķis, izteikts radiānos.
atan	Vērtības arkustangenss kā leņķis, izteikts radiānos.
atan2	Ordinātas (vērtības) un abscisas koordinātu arkustangenss, kas norādīts kā leņķis, izteikts radiānos.
avg	Atsaukto vienumu vērtību vidējā vērtība.
cbrt	Vērtības kuba sakne.
ceil	Noapaļo vērtību uz augšu līdz tuvākajam lielākajam vai vienādajam veselajam skaitlim.
cos	Vērtības kosinuss, kur vērtība ir leņķis, kas izteikts radiānos.
cosh	Vērtības hiperboliskais kosinuss.
cot	Vērtības kotangenss, kur vērtība ir leņķis, kas izteikts radiānos.
degrees	Pārveido vērtību no radiāniem uz grādiem.
e	Eilera skaitlis (2.718281828459045).
exp	Eilera skaitlis vērtības pakāpē.
expm1	Eilera skaitlis vērtības pakāpē mīnus 1.

Function	Description
floor	Noapaļo vērtību uz leju līdz tuvākajam mazākajam vai vienādajam veselajam skaitlim.
log	Naturālais logaritms.
log10	Decimāllogaritms.
max	Atsaukto vienumu vērtību lielākā vērtība.
min	Atsaukto vienumu vērtību mazākā vērtība.
mod	Dalīšanas atlikums.
pi	Pi konstante (3.14159265358979).
power	Vērtības pakāpe.
radians	Pārveido vērtību no grādiem uz radiāniem.
rand	Atgriež nejaušu vesela skaitļa vērtību.
round	Noapaļo vērtību līdz noteiktam zīmju skaitam aiz komata.
signum	Atgriež '-1', ja vērtība ir negatīva, '0', ja vērtība ir nulle, '1', ja vērtība ir pozitīva.
sin	Vērtības sinuss, kur vērtība ir leņķis, kas izteikts radiānos.
sinh	Vērtības hiperboliskais sinuss, kur vērtība ir leņķis, kas izteikts radiānos.
sqrt	Vērtības kvadrātsakne.
sum	Atsaukto vienumu vērtību summa.
tan	Vērtības tangenss.
truncate	Saīsina vērtību līdz noteiktam zīmju skaitam aiz komata.

Funkciju detaļas

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Izteiksmes tiek pieņemtas kā parametri
- Neobligātie funkciju parametri (vai parametru daļas) tiek norādīti ar < >

abs(value)

Vērtības absolūtā vērtība (no 0).

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, gan '3', gan '-3' absolūtā vērtība būs '3'.

Piemērs:

```
abs(last(/host/key))>10
```

acos(value)

Vērtības arkuskosinuss kā leņķis, izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Vērtībai jābūt no -1 līdz 1. Piemēram, vērtības '0.5' arkuskosinuss būs '2.0943951'.

Piemērs:

```
acos(last(/host/key))
```

asin(value)

Vērtības arkussinuss kā leņķis, izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Vērtībai jābūt no -1 līdz 1. Piemēram, vērtības '0.5' arkussinuss būs '-0.523598776'.

Piemērs:

```
asin(last(/host/key))
```

atan(value)

Vērtības arkustangenss kā leņķis, izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ arkustangenss būs ’0.785398163’.

Piemērs:

```
atan(last(/host/key))
```

atan2(value,abscissa)

Ordinātas (value) un abscisas koordinātu arkustangens, kas norādīts kā leņķis radiānos.

Parametri:

- **value** - pārbaudāmā vērtība;
- **abscissa** - abscisas vērtība.

Piemēram, ordinātas un abscisas koordinātu arkustangens vērtībai ’1’ būs ’2.21429744’.

Piemērs:

```
atan2(last(/host/key),2)
```

avg(<value1>,<value2>,...)

Atsauktā vienuma vērtību vidējā vērtība.

Parametrs:

- **valueX** - vērtība, ko atgriež cita funkcija, kas darbojas ar vienuma vēsturi.

Piemērs:

```
avg(avg(/host/key,1h),avg(/host2/key2,1h))
```

cbrt(value)

Vērtības kubsakne.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, skaitļa ’64’ kubsakne būs ’4’, bet skaitļa ’63’ kubsakne būs ’3.97905721’.

Piemērs:

```
cbrt(last(/host/key))
```

ceil(value)

Noapaļo vērtību uz augšu līdz tuvākajam lielākajam vai vienādam veselam skaitlim.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, ’2.4’ tiks noapaļots uz augšu līdz ’3’. Skatiet arī floor().

Piemērs:

```
ceil(last(/host/key))
```

cos(value)

Vērtības kosinuss, kur vērtība ir leņķis, kas izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ kosinuss būs ’0.54030230586’.

Piemērs:

```
cos(last(/host/key))
```

cosh(value)

Vērtības hiperboliskais kosinuss. Atgriež vērtību kā reālu skaitli, nevis zinātniskajā pierakstā.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ hiperboliskais kosinuss būs ’1.54308063482’.

Piemērs:

```
cosh(last(/host/key))
```

cot(value)

Vērtības kotangenss, kur vērtība ir leņķis, kas izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ kotangenss būs ’0.54030230586’.

Piemērs:

```
cot(last(/host/key))
```

degrees(value)

Pārveido vērtību no radiāniem uz grādiem.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtība ’1’, pārveidota grādos, būs ’57.2957795’.

Piemērs:

```
degrees(last(/host/key))
```

e

Eilera skaitlis (2.718281828459045).

Piemērs:

```
e()
```

exp(value)

Eilera skaitlis vērtības pakāpē.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, Eilera skaitlis vērtības ’2’ pakāpē būs ’7.38905609893065’.

Piemērs:

```
exp(last(/host/key))
```

expm1(value)

Eilera skaitlis pakāpē ar vērtību mīnus 1.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, Eilera skaitlis pakāpē ar vērtību ’2’ mīnus 1 būs ’6.38905609893065’.

Piemērs:

```
expm1(last(/host/key))
```

floor(value)

Noapaļo vērtību uz leju līdz tuvākajam mazākajam vai vienādam veselam skaitlim.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, ’2.6’ tiks noapaļots uz leju līdz ’2’. Skatiet arī ceil().

Piemērs:

```
floor(last(/host/key))
```

log(value)

Naturālais logaritms.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’2’ naturālais logaritms būs ’0.69314718055994529’.

Piemērs:

```
log(last(/host/key))
```

log10(value)

Decimālais logaritms.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’5’ decimālais logaritms būs ’0.69897000433’.

Piemērs:

```
log10(last(/host/key))
```

max(<value1>,<value2>,...)

Atsauktā vienuma vērtību augstākā vērtība.

Parametrs:

- **valueX** - vērtība, ko atgriež cita funkcija, kas darbojas ar vienuma vēsturi.

Piemērs:

```
max(avg(/host/key,1h),avg(/host2/key2,1h))
```

min(<vērtība1>,<vērtība2>,...)

Zemākā vērtība no norādītā vienuma vērtībām.

Parametrs:

- **vērtībaX** - vērtība, ko atgriež cita funkcija, kas darbojas ar vienuma vēsturi.

Piemērs:

```
min(avg(/host/key,1h),avg(/host2/key2,1h))
```

mod(value,denominator)

Dalīšanas atlikums.

Parametri:

- **value** - pārbaudāmā vērtība;
- **denominator** - dalītājs.

Piemēram, vērtības ’5’ dalīšanas atlikums ar dalītāju ’2’ būs ’1’.

Piemērs:

```
mod(last(/host/key),2)
```

pi

Konstante Pi (3.14159265358979).

Piemērs:

```
pi()
```

power(value,power value)

Vērtības pakāpe.

Parametri:

- **value** - pārbaudāmā vērtība;
- **power value** - izmantojamā N-tā pakāpe.

Piemēram, vērtības ’2’ 3. pakāpe būs ’8’.

Piemērs:

```
power(last(/host/key),3)
```

radians(value)

Pārveido vērtību no grādiem uz radiāniem.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtība ’1’, pārveidota radiānos, būs ’0.0174532925’.

Piemērs:

```
radians(last(/host/key))
```

rand

Atgriež nejaušu vesela skaitļa vērtību. Pseido-nejauši ģenerēts skaitlis, izmantojot laiku kā sākumvērtību (pietiekami matemātiskiem nolūkiem, bet ne kriptogrāfijai).

Piemērs:

```
rand()
```

round(value,decimal places)

Noapaļo vērtību līdz norādītajam zīmju skaitam aiz komata.

Parametri:

- **value** - pārbaudāmā vērtība;
- **decimal places** - norādiet zīmju skaitu aiz komata noapaļošanai (iespējama arī vērtība 0).

Piemēram, vērtība ’2.5482’, noapaļota līdz 2 zīmēm aiz komata, būs ’2.55’.

Piemērs:

```
round(last(/host/key),2)
```

signum(value)

Atgriež ’-1’, ja vērtība ir negatīva, ’0’, ja vērtība ir nulle, ’1’, ja vērtība ir pozitīva.

Parametrs:

- **value** - pārbaudāmā vērtība.

Piemērs:

```
signum(last(/host/key))
```

sin(value)

Vērtības sinuss, kur vērtība ir leņķis, kas izteikts radiānos.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ sinuss būs ’0.8414709848’.

Piemērs:

```
sin(last(/host/key))
```

sinh(value)

Vērtības hiperboliskais sinuss.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ hiperboliskais sinuss būs ’1.17520119364’.

Piemērs:

```
sinh(last(/host/key))
```

sqrt(value)

Vērtības kvadrātsakne.
 Šī funkcija neizdosies ar negatīvu vērtību.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’3.5’ kvadrātsakne būs ’1.87082869339’.

Piemērs:

```
sqrt(last(/host/key))
```

sum(<value1>,<value2>,...)

Atsaukto vienumu vērtību summa.

Parametrs:

- **valueX** - vērtība, ko atgriež cita funkcija, kas darbojas ar vienuma vēsturi.

Piemērs:

```
sum(avg(/host/key,1h),avg(/host2/key2,1h))
```

tan(value)

Vērtības tangenss.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, vērtības ’1’ tangenss būs ’1.55740772465’.

Piemērs:

```
tan(last(/host/key))
```

truncate(value,decimal places)

Saīsina vērtību līdz norādītajam zīmju skaitam aiz komata.

Parametri:

- **value** - pārbaudāmā vērtība;
- **decimal places** - norādiet zīmju skaitu aiz komata saīsināšanai (iespējams arī 0).

Piemēram, vērtība ’2.5482’, saīsināta līdz 2 zīmēm aiz komata, būs ’2.54’.

Piemērs:

```
truncate(last(/host/key),2)
```

Skatiet visas atbalstītās funkcijas.

7 Operatoru funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Funkcija	Apraksts
between	Pārbaudīt, vai vērtība ietilpst norādītajā diapazonā.
in	Pārbaudīt, vai vērtība ir vienāda ar vismaz vienu no uzskaitītajām vērtībām.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Izteiksmes tiek pieņemtas kā parametri

between(value,min,max)

Pārbauda, vai vērtība ietilpst norādītajā diapazonā.
 Atbalstītie vērtību tipi: *Integer*, *Float*.
 Atgriež: 1 - diapazonā; 0 - pretējā gadījumā.

Parametri:

- **value** - pārbaudāmā vērtība;
- **min** - minimālā vērtība;
- **max** - maksimālā vērtība.

Piemērs:

```
between(last(/host/key),1,10)=1 #trigeris, ja vērtība ir no 1 līdz 10
```

in(value,value1,value2,...valueN)

Pārbauda, vai vērtība ir vienāda ar vismaz vienu no uzskaitītajām vērtībām.
 Atbalstītie vērtību tipi: *Integer*, *Float*, *Character*, *Text*, *Log*.
 Atgriež: 1 - ja ir vienāda; 0 - pretējā gadījumā.

Parametri:

- **value** - pārbaudāmā vērtība;
- **valueX** - uzskaitītās vērtības (virknes vērtībām jābūt dubultpēdiņās).

Vērtība tiek salīdzināta ar uzskaitītajām vērtībām kā skaitļi, ja visas šīs vērtības var pārveidot skaitliskā formā; pretējā gadījumā tās tiek salīdzinātas kā virknes.

Piemērs:

```
in(last(/host/key),5,10)=1 #trigeris, ja pēdējā vērtība ir vienāda ar 5 vai 10
in("text",last(/host/key),last(/host/key,#2))=1 #trigeris, ja "text" ir vienāds ar jebkuru no pēdējām 2 vē
```

Skatiet visas atbalstītās funkcijas.

8 Prognozējošās funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Funkcija	Apraksts
forecast	Vienuma nākotnes vērtība, maksimums, minimums, delta vai vidējā vērtība.
timeleft	Laiks sekundēs, kas nepieciešams, lai vienums sasniegtu norādīto slieksni.

Kopīgie parametri

- `/host/key` ir kopīgs obligāts pirmais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi
- `(sec|#num)<:time shift>` ir kopīgs otrais parametrs funkcijām, kas atsaucas uz hosta vienuma vēsturi, kur:
  - **sec** - maksimālais novērtēšanas periods sekundēs (var izmantot laika sufiksus), vai
  - **#num** - maksimālais novērtēšanas diapazons pēdējās savāktajās vērtībās (ja pirms tā ir restīte)
  - **time shift** (neobligāts) ļauj pārvietot novērtēšanas punktu atpakaļ laikā. Skatiet sīkāku informāciju par time shift norādīšanu.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Neobligātie funkciju parametri (vai parametru daļas) tiek norādīti ar < >
- Katras funkcijas specifiskie parametri ir aprakstīti pie attiecīgās funkcijas

- Parametri `/host/key` un `(sec|#num)<:time shift>` nekad nedrīkst būt pēdiņās

forecast(/host/key,(sec|#num)<:time shift>,time,<fit>,<mode>)

Vienuma nākotnes vērtība, maksimums, minimums, delta vai vidējā vērtība.
 Atbalstītie vērtību tipi: *Float*, *Integer*.

Parametri:

- Skatiet kopīgos parametrus;
- **time** - prognozēšanas horizonts sekundēs (var izmantot laika sufiksus); tiek atbalstītas arī negatīvas vērtības;
- **fit** (neobligāts; jāliek dubultpēdiņās) - funkcija, ko izmanto vēsturisko datu pielāgošanai. Atbalstītās pielāgošanas funkcijas:
linear - lineāra funkcija (noklusējums)
polynomialN - N pakāpes polinoms (1 <= N <= 6)
exponential - eksponenciāla funkcija
logarithmic - logaritmiska funkcija
power - pakāpes funkcija
Ņemiet vērā, ka *polynomial1* ir ekvivalents *linear*;
- **mode** (neobligāts; jāliek dubultpēdiņās) - pieprasītā izvade. Atbalstītie režīmi:
value - vērtība (noklusējums)
max - maksimums
min - minimums
delta - *max-min*
avg - vidējā vērtība
Ņemiet vērā, ka *value* novērtē vienuma vērtību brīdī `now + time`; *max*, *min*, *delta* un *avg* analizē vienuma vērtības novērtējumu intervālā starp `now` un `now + time`.

Komentāri:

- Ja atgriežamā vērtība ir lielāka par 1.7976931348623158E+308 vai mazāka par -1.7976931348623158E+308, atgrieztā vērtība tiek attiecīgi ierobežota līdz 1.7976931348623158E+308 vai -1.7976931348623158E+308;
- Kļūst neatbalstīts tikai nepareizas izmantošanas gadījumā izteiksmē (nepareizs vienuma tips, nederīgi parametri), pretējā gadījumā kļūdu gadījumā atgriež -1;
- Skatiet arī papildu informāciju par prognozējošajām trigeru funkcijām.

Piemēri:

```
forecast(/host/key,#10,1h) #prognozē vienuma vērtību pēc vienas stundas, pamatojoties uz pēdējām 10 vērtīb
forecast(/host/key,1h,30m) #prognozē vienuma vērtību pēc 30 minūtēm, pamatojoties uz pēdējās stundas datie
forecast(/host/key,1h:now-1d,12h) #prognozē vienuma vērtību pēc 12 stundām, pamatojoties uz vienas stundas
forecast(/host/key,1h,10m,"exponential") #prognozē vienuma vērtību pēc 10 minūtēm, pamatojoties uz pēdējās
forecast(/host/key,1h,2h,"polynomial3","max") #prognozē maksimālo vērtību, ko vienums var sasniegt nākamo
forecast(/host/key,#2,-20m) #novērtē vienuma vērtību pirms 20 minūtēm, pamatojoties uz pēdējām divām vērtī
```

timeleft(/host/key,(sec|#num)<:time shift>,threshold,<fit>)

Laiks sekundēs, kas nepieciešams, lai vienums sasniegtu norādīto slieksni.
 Atbalstītie vērtību tipi: *Float*, *Integer*.

Parametri:

- Skatiet kopīgos parametrus;
- **threshold** — sasniedzamā vērtība (var izmantot vienību sufiksus);
- **fit** (neobligāts; jāliek dubultpēdiņās) — skatiet forecast().

Komentāri:

- Ja atgriežamā vērtība ir lielāka par 1.7976931348623158E+308, atgrieztā vērtība tiek ierobežota līdz 1.7976931348623158E+308;
- Atgriež 1.7976931348623158E+308, ja slieksni nevar sasniegt;
- Kļūst neatbalstīts tikai tad, ja tas tiek nepareizi izmantots izteiksmē (nepareizs vienuma tips, nederīgi parametri), pretējā gadījumā kļūdu gadījumā atgriež -1;
- Skatiet arī papildu informāciju par prognozējošajām trigeru funkcijām.

Piemēri:

```
timeleft(/host/key,#10,0) #laiks līdz vienuma vērtība sasniedz nulli, pamatojoties uz pēdējām 10 vērtībām
timeleft(/host/key,1h,100) #laiks līdz vienuma vērtība sasniedz 100, pamatojoties uz pēdējās stundas datie
timeleft(/host/key,1h:now-1d,100) #laiks līdz vienuma vērtība sasniedz 100, pamatojoties uz vienas stundas
timeleft(/host/key,1h,200,"polynomial2") #laiks līdz vienuma vērtība sasniedz 200, pamatojoties uz pēdējās
```

Skatīt visas atbalstītās funkcijas.

Prognozējošās trigeru funkcijas

Pārskats

Dažreiz ir pazīmes, kas liecina par gaidāmu problēmu. Šīs pazīmes var pamanīt, lai jau iepriekš varētu veikt darbības, kas novērstu problēmu vai vismaz samazinātu tās ietekmi.

Zabbix piedāvā rīkus, lai prognozētu uzraudzītās sistēmas turpmāko uzvedību, pamatojoties uz vēsturiskajiem datiem. Šie rīki ir realizēti, izmantojot prognozējošās trigeru funkcijas.

Funkcijas

Pirms trigeris iestatīšanas ir nepieciešams noteikt, kas ir problēmas stāvoklis un cik daudz laika ir nepieciešams, lai rīkotos. Pēc tam ir divi veidi, kā iestatīt trigeris, kas signalizē par iespējamu nevēlamu situāciju. Pirmais: trigeris jānostrādā, kad ir sagaidāms, ka sistēma nonāks problēmas stāvoklī pēc "laika rīcībai". Otrais: trigeris jānostrādā, kad sistēma sasniegs problēmas stāvokli mazāk nekā "laikā rīcībai". Atbilstošās trigeris funkcijas, ko izmantot, ir **forecast** un **timeleft**. Ņemiet vērā, ka pamatā esošā statistiskā analīze abām funkcijām būtībā ir identiska. Jūs varat iestatīt trigeris jebkurā no šiem veidiem pēc savas izvēles, iegūstot līdzīgus rezultātus.

Parametri

Abas funkcijas izmanto gandrīz vienu un to pašu parametru kopu. Atsaucei izmantojiet atbalstīto funkciju sarakstu.

Laika intervāls

Vispirms jums jānorāda vēsturiskais periods, kuru Zabbix vajadzētu analizēt, lai izveidotu prognozi. To var izdarīt ierastajā veidā, izmantojot parametru `time period` un neobligātu laika nobīdi, tāpat kā tas tiek darīts ar funkcijām **avg**, **count**, **delta**, **max**, **min** un **sum**.

Prognozēšanas horizonts

(tikai **forecast**)
 Parametrs `time` norāda, cik tālu nākotnē Zabbix vajadzētu ekstrapolēt atkarības, ko tas atrod vēsturiskajos datos. Neatkarīgi no tā, vai izmantojat `time_shift` vai ne, `time` vienmēr tiek skaitīts, sākot no pašreizējā brīža.

Sasniedzamais slieksnis

(tikai **timeleft**)
 Parametrs `threshold` norāda vērtību, kuru analizētajam vienumam jāsasniedz, neatkarīgi no tā, vai no augšas vai no apakšas. Kad ir noteikta f(t) (skatiet zemāk), jāatrisina vienādojums f(t) = `threshold` un jāatgriež sakne, kas ir tuvāk pašreizējam brīdim un atrodas pa labi no tā, vai 1.7976931348623158E+308, ja šādas saknes nav.

Note:
Kad vienuma vērtības tuvojas slieksnim un pēc tam to šķērso, **timeleft** pieņem, ka šis krustpunkts jau ir pagātnē, un tāpēc pārslēdzas uz nākamo krustpunktu ar `threshold` līmeni, ja tāds ir. Ieteicamā prakse ir izmantot prognozes kā papildinājumu parastajai problēmu diagnostikai, nevis kā tās aizstājēju.[1]

Aproksimācijas funkcijas

Noklusējuma `fit` ir *lineārā* funkcija. Taču, ja jūsu uzraudzītā sistēma ir sarežģītāka, jums ir pieejamas vairākas izvēles iespējas.

`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	$x = a*t^b$

Režīmi

(tikai **forecast**)
 Katru reizi, kad tiek novērtēta trigera funkcija, tā iegūst datus no norādītā vēstures perioda un pielāgo datiem norādīto funkciju. Tādēļ, ja dati nedaudz atšķiras, arī pielāgotā funkcija nedaudz atšķirsies. Ja mēs vienkārši aprēķinām pielāgotās funkcijas vērtību noteiktā brīdī nākotnē, jūs neko neuzzināsiet par to, kā analizētais vienums, visticamāk, uzvedīsies laika posmā no šī brīža līdz šim nākotnes brīdim. Dažām `fit` opcijām (piemēram, *polynomial*) vienkārša vērtība nākotnē var būt maldinoša.

`mode`	**forecast** rezultāts
value	f(now + `time`)
max	$max_{now <= t <= now + time}$ f(t)
min	$min_{now <= t <= now + time}$ f(t)
delta	*max - min*
avg	f(t) vidējā vērtība (now <= t <= now + `time`) saskaņā ar definīciju

Detalizēta informācija

Lai izvairītos no aprēķiniem ar ļoti lieliem skaitļiem, mēs uzskatām pirmās vērtības laikspiedolu norādītajā periodā plus 1 ns par jaunu nulles laiku (pašreizējais epoch laiks ir 10^9 kārtā, epoch kvadrāts ir 10^{18}, dubultā precizitāte ir aptuveni 10^{-16}). 1 ns tiek pievienota, lai nodrošinātu visas pozitīvās laika vērtības *logaritmiskajām* un *pakāpes* aproksimācijām, kurās tiek aprēķināts

log(t). Laika nobīde neietekmē *lineārās*, *polinomiālās*, *eksponenciālās* aproksimācijas (izņemot to, ka aprēķini kļūst vienkāršāki un precīzāki), bet maina *logaritmisko* un *pakāpes* funkciju formu.

Iespējamās kļūdas

Funkcijas atgriež -1 šādās situācijās:

- norādītais novērtēšanas periods nesatur datus;
- matemātiskās operācijas rezultāts nav definēts[3];
- skaitliskas sarežģītības (diemžēl dažām ievades datu kopām dubultprecizitātes peldošā komata formāta diapazons un precizitāte kļūst nepietiekami)[4].

Note:
Brīdinājumi vai kļūdas netiek norādītas, ja izvēlētā aproksimācija vāji apraksta sniegtos datus vai arī precīzai prognozēšanai datu ir vienkārši pārāk maz.

Piemēri un kļūdu apstrāde

Lai saņemtu brīdinājumu, kad jūsu hostā drīz beigsies brīvā diska vieta, varat izmantot šādu trigera izteiksmi:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h
```

Tomēr kļūdas kods -1 var ietekmēt rezultātu un pārslēgt jūsu trigeri problēmas stāvoklī. Kopumā tas ir labi, jo saņemat brīdinājumu, ka jūsu prognozes nedarbojas korekti un tās vajadzētu rūpīgāk pārbaudīt, lai noskaidrotu iemeslu. Taču dažreiz tas ir slikti, jo -1 var vienkārši nozīmēt, ka pēdējās stundas laikā netika iegūti dati par hosta brīvo diska vietu. Ja saņemat pārāk daudz kļūdaini pozitīvu brīdinājumu, apsveriet iespēju izmantot sarežģītāku trigera izteiksmi [5]:

```
timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1
```

Situācija ir nedaudz sarežģītāka ar **forecast**. Pirmkārt, -1 var vai nevar pārslēgt trigeri problēmas stāvoklī atkarībā no tā, vai jums ir izteiksme, piemēram, `forecast(/host/item,(...))<...`, vai izteiksme, piemēram, `forecast(/host/item,(...))>...`

Turklāt -1 var būt derīga prognoze, ja vienuma vērtībai ir normāli būt negatīvai. Taču šādas situācijas iespējamība reālajā pasaulē ir niecīga (skatiet, kā darbojas operators =). Tāpēc pievienojiet `... or forecast(/host/item,(...))=-1` vai `... and forecast(/host/item,(...))<>-1`, ja attiecīgi vēlaties vai nevēlaties uzskatīt -1 par problēmu.

Piezīmes

[1] Piemēram, vienkāršs trigeris, piemēram, `timeleft(/host/item,1h,X) < 1h`, var pāriet problēmas stāvoklī, kad vienuma vērtība tuvojas X, un pēc tam pēkšņi atjaunoties, tiklīdz tiek sasniegta vērtība X. Ja problēma ir tāda, ka vienuma vērtība ir zemāka par X, izmantojiet: `last(/host/item) < X or timeleft(/host/item,1h,X) < 1h` Ja problēma ir tāda, ka vienuma vērtība ir augstāka par X, izmantojiet: `last(/host/item) > X or timeleft(/host/item,1h,X) < 1h`

[2] Polinoma pakāpe var būt no 1 līdz 6, *polynomial1* ir ekvivalents *linear*. Tomēr augstākas pakāpes polinomus izmantojiet piesardzīgi. Ja novērtēšanas periods satur mazāk punktu, nekā nepieciešams polinoma koeficientu noteikšanai, polinoma pakāpe tiks samazināta (piemēram, tiek pieprasīts *polynomial5*, bet ir tikai 4 punkti, tāpēc tiks pielāgots *polynomial3*).

[3] Piemēram, *exponential* vai *power* funkciju pielāgošana ietver vienuma vērtību log() aprēķināšanu. Ja dati satur nulles vai negatīvus skaitļus, tiks saņemta kļūda, jo log() ir definēts tikai pozitīvām vērtībām.

[4] *linear*, *exponential*, *logarithmic* un *power* pielāgojumiem visus nepieciešamos aprēķinus var uzrakstīt tieši. *polynomial* gadījumā bez papildu darbībām var aprēķināt tikai *value*. *avg* aprēķināšana ietver polinoma pirmatvasinājuma (analītisku) aprēķināšanu. *max*, *min* un *delta* aprēķināšana ietver polinoma atvasinājuma (analītisku) aprēķināšanu un tā sakņu atrašanu (skaitliski). f(t) = 0 atrisināšana ietver polinoma sakņu atrašanu (skaitliski).

[5] Taču šajā gadījumā -1 var izraisīt jūsu trigera atjaunošanos no problēmas stāvokļa. Lai būtu pilnībā aizsargāts, izmantojiet: `timeleft(/host/vfs.fs.size[/,free],1h,0)<1h and ({TRIGGER.VALUE}=0 and timeleft(/host/vfs.fs.size[/,free],1h,0)<>-1 or {TRIGGER.VALUE}=1)`

9 Virkņu funkcijas

Visas šeit uzskaitītās funkcijas tiek atbalstītas:

- Trigeru izteiksmēs
- Aprēķinātajos vienumos
- Izteiksmju makrosos

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Function	Description
ascii	Vērtības kreisāk esošās rakstzīmes ASCII kods.
bitlength	Vērtības garums bitos.
bytelength	Vērtības garums baitos.
char	Atgriež rakstzīmi, interpretējot vērtību kā ASCII kodu.
concat	Virkne, kas iegūta, savienojot atsauktās vienumu vērtības vai konstantās vērtības.
insert	Ievieto norādītās rakstzīmes vai atstarpes rakstzīmju virknē, sākot no norādītās pozīcijas virknē.
jsonpath	Atgriež JSONPath rezultātu.
left	Atgriež vērtības kreisāk esošās rakstzīmes.
length	Vērtības garums rakstzīmēs.
ltrim	Noņem norādītās rakstzīmes no virknes sākuma.
mid	Atgriež N rakstzīmju apakšvirkni, sākot no rakstzīmes pozīcijas, kas norādīta ar 'start'.
repeat	Atkārto virkni.
replace	Atrod šablonu vērtībā un aizstāj to ar aizvietotāju.
right	Atgriež vērtības labāk esošās rakstzīmes.
rtrim	Noņem norādītās rakstzīmes no virknes beigām.
trim	Noņem norādītās rakstzīmes no virknes sākuma un beigām.
xmlxpath	Atgriež XML XPath rezultātu.

Funkciju informācija

Dažas vispārīgas piezīmes par funkciju parametriem:

- Funkciju parametrus atdala ar komatu
- Izteiksmes tiek pieņemtas kā parametri
- Virknes parametriem jābūt dubultpēdiņās; pretējā gadījumā tie var tikt nepareizi interpretēti
- Neobligātie funkciju parametri (vai parametru daļas) tiek norādīti ar < >

ascii(value)

Vērtības kreisākās rakstzīmes ASCII kods.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemēram, tāda vērtība kā 'Abc' atgriezīs '65' ('A' ASCII kodu).

Piemērs:

```
ascii(last(/host/key))
```

bitlength(value)

Vērtības garums bitos.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*, *Integer*.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemērs:

```
bitlength(last(/host/key))
```

bytelength(value)

Vērtības garums baitos.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*, *Integer*.

Parametrs:

- **value** - pārbaudāmā vērtība

Piemērs:

```
bytelength(last(/host/key))
```

char(value)

Atgriež rakstzīmi, interpretējot vērtību kā ASCII kodu.
 Atbalstītie vērtību tipi: *Integer*.

Parametrs:

- **value** - pārbaudāmā vērtība

Vērtībai jābūt diapazonā no 0 līdz 255. Piemēram, vērtība '65' (interpretēta kā ASCII kods) atgriezīs 'A'.

Piemērs:

```
char(last(/host/key))
```

concat(<value1>,<value2>,...)

Virkne, kas iegūta, savienojot norādīto vienumu vērtības vai konstantās vērtības.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*, *Float*, *Integer*.

Parametrs:

- **valueX** - vērtība, ko atgriež kāda no vēstures funkcijām, vai konstanta vērtība (virkne, vesels skaitlis vai peldošā komata skaitlis). Jāsatur vismaz divi parametri.

Piemēram, vērtība 'Zab', kas savienota ar 'bix' (konstanta virkne), atgriezīs 'Zabbix'.

Piemēri:

```
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)

Ievieto norādītās rakstzīmes vai atstarpes rakstzīmju virknē, sākot no norādītās pozīcijas virknē.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **start** - sākuma pozīcija;
- **length** - aizstājamās pozīcijas;
- **replacement** - aizstāšanas virkne.

Piemēram, vērtība 'Zabbbix' tiks aizstāta ar 'Zabbix', ja 'bb' (sākuma pozīcija 3, aizstājamās pozīcijas 2) tiek aizstāts ar 'b'.

Piemērs:

```
insert(last(/host/key),3,2,"b")
```

jsonpath(value,path,<default>)

Atgriež JSONPath rezultātu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **path** - ceļš (jānorāda pēdiņās);
- **default** - neobligātā rezerves vērtība, ja JSONPath vaicājums neatgriež datus. Ņemiet vērā, ka citu kļūdu gadījumā tiek atgriezta kļūme (piemēram, "unsupported construct").

Piemērs:

```
jsonpath(last(/host/proc.get[zabbix_agentd,,,summary]),"$..size")
```

left(value,count)

Atgriež vērtības kreisākās rakstzīmes.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **count** - atgriežamo rakstzīmju skaits.

Piemēram, no 'Zabbix' var atgriezt 'Zab', norādot 3 atgriežamās kreisākās rakstzīmes. Skatiet arī right().

Piemērs:

```
left(last(/host/key),3) #atgriež trīs kreisākās rakstzīmes
```

length(value)

Vērtības garums rakstzīmēs.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametrs:

- **value** - pārbaudāmā vērtība.

Piemēri:

```
length(last(/host/key)) #jaunākās vērtības garums
length(last(/host/key,#3)) #trešās jaunākās vērtības garums
length(last(/host/key,#1:now-1d)) #jaunākās vērtības garums pirms vienas dienas
```

ltrim(value,<chars>)

Noņem norādītās rakstzīmes no virknes sākuma.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **chars** (neobligāts) - norādiet noņemamās rakstzīmes.

Pēc noklusējuma no sākuma tiek noņemtas atstarpes (ja nav norādītas neobligātās rakstzīmes). Skatiet arī: rtrim(), trim().

Piemēri:

```
ltrim(last(/host/key)) #noņem atstarpes no virknes sākuma
ltrim(last(/host/key),"Z") #noņem visas 'Z' no virknes sākuma
ltrim(last(/host/key)," Z") #noņem visas atstarpes un 'Z' no virknes sākuma
```

mid(value,start,length)

Atgriež apakšvirkni ar N rakstzīmēm, sākot no rakstzīmes pozīcijas, kas norādīta ar 'start'.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **start** - apakšvirknes sākuma pozīcija;
- **length** - pozīciju skaits, ko atgriezt apakšvirknē.

Piemēram, ir iespējams atgriezt 'abbi' no vērtības 'Zabbix', ja sākuma pozīcija ir 2 un atgriežamo pozīciju skaits ir 4.

Piemērs:

```
mid(last(/host/key),2,4)="abbi"
```

repeat(value,count)

Atkārto virkni.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **count** - atkārtojumu skaits.

Piemērs:

```
repeat(last(/host/key),2) #atkārto vērtību divas reizes
```

replace(value,pattern,replacement)

Atrodiet šablonu vērtībā un aizstājiet to ar aizvietojumu. Tiks aizstātas visas šablona parādīšanās reizes.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **pattern** - atrodamais šablons;
- **replacement** - virkne, ar ko aizstāt šablonu.

Piemērs:

```
replace(last(/host/key),"ibb","abb") #aizstāt visus 'ibb' ar 'abb'
```

right(value,count)

Atgriež vērtības labējās rakstzīmes.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **count** - atgriežamo rakstzīmju skaits.

Piemēram, no 'Zabbix' var atgriezt 'bix', norādot atgriezt 3 labējās rakstzīmes. Skatiet arī left().

Piemērs:

```
right(last(/host/key),3) #atgriež trīs labējās rakstzīmes
```

rtrim(value,<chars>)

Noņem norādītās rakstzīmes no virknes beigām.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **chars** (neobligāts) - norādiet noņemamās rakstzīmes.

Pēc noklusējuma tiek noņemtas atstarpes virknes labajā pusē (ja nav norādītas neobligātās rakstzīmes). Skatiet arī: ltrim(), trim().

Piemēri:

```
rtrim(last(/host/key)) #noņem atstarpes no virknes beigām
rtrim(last(/host/key),"x") #noņem jebkuru 'x' no virknes beigām
rtrim(last(/host/key),"x ") #noņem jebkuru 'x' un atstarpi no virknes beigām
```

trim(value,<chars>)

Noņem norādītās rakstzīmes no virknes sākuma un beigām.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **chars** (neobligāts) - norādiet noņemamās rakstzīmes.

Pēc noklusējuma atstarpes tiek noņemtas no abām pusēm (ja nav norādītas neobligātās rakstzīmes). Skatiet arī: ltrim(), rtrim().

Piemēri:

```
trim(last(/host/key)) #noņem atstarpes no virknes sākuma un beigām
trim(last(/host/key),"_") #noņem '_' no virknes sākuma un beigām
```

xmlxpath(value,path,<default>)

Atgriež XML XPath rezultātu.
 Atbalstītie vērtību tipi: *String*, *Text*, *Log*.

Parametri:

- **value** - pārbaudāmā vērtība;
- **path** - ceļš (jānorāda pēdiņās);
- **default** - neobligātā rezerves vērtība, ja XML XPath vaicājums atgriež tukšu mezglu kopu. Tā netiks atgriezta, ja tukšais rezultāts nav mezglu kopa (t.i., tukša virkne). Citu kļūdu gadījumā tiek atgriezta kļūme (piemēram, ”invalid expression”).

Piemērs:

```
xmlxpath(last(/host/xml_result),"/response/error/status")
```

Skatīt visas atbalstītās funkcijas.

4 Trigeru labākā prakse

Trigeri ir jaudīgs rīks, taču tie var arī radīt nevēlamu brīdinājumu troksni. Lai redzētu vairāk reālu signālu un mazāk trokšņa, ievērojiet šos ieteikumus:

1. Samaziniet trigeru jutību. Tā vietā, lai brīdinātu par jaunāko vērtību (pārāk augsta/pārāk zema), analizējiet vidējo vērtību ilgākā periodā, izmantojot tādas funkcijas kā **avg**, **min** un **max**.
2. Apsveriet iespēju izmantot funkciju **percentile** trigeros (iestatiet to uz 95% vai 5%), ja vēlaties izvairīties no brīdinājumiem nejaušu pīķu un kritumu dēļ.
3. Izmantojiet histērēzi, lai izvairītos no trigeru stāvokļa svārstībām — biežām trigera stāvokļa izmaiņām (OK ↔ Problem). Problēmas stāvokļa nepārtrauktību var definēt, pievienojot atkopšanas izteiksmi (atsevišķu nosacījumu problēmas atrisināšanai).
4. Izmantojiet trigeru atkarības, lai izvairītos no brīdinājumiem, kas nav saistīti ar pamatcēloni.
5. Izmantojiet trigera nopietnību, lai brīdinātu tikai par nopietnākajām problēmām.
6. Definējiet apkopes logus.

Histerēze

Dažkārt starp problēmas un atkopšanas stāvokļiem ir nepieciešams intervāls, nevis vienkārša robežvērtība. Piemēram, ja vēlamies definēt trigeri, kas ziņo par problēmu, kad servera telpas temperatūra pārsniedz 20°C, un vēlamies, lai tas paliktu problēmas stāvoklī, līdz temperatūra nokrītas zem 15°C, ar vienkāršu trigera robežvērtību 20°C nepietiks.

Tā vietā vispirms ir jādefinē trigera izteiksme problēmas notikumam (temperatūra virs 20°C). Pēc tam ir jādefinē papildu atkopšanas nosacījums (temperatūra zem 15°C). Tas tiek izdarīts, definējot *Atkopšanas izteiksmi*, konfigurējot trigeri.

Šajā gadījumā problēmas atkopšana notiek divos soļos:

- Vispirms problēmas izteiksmei (temperatūra virs 20°C) jānovērtējas kā FALSE
- Pēc tam atkopšanas izteiksmei (temperatūra zem 15°C) jānovērtējas kā TRUE

Atkopšanas izteiksme tiek novērtēta tikai pēc tam, kad problēmas notikums ir atrisināts. Tas vien, ka atkopšanas izteiksme ir TRUE, neatrisina problēmu, ja problēmas izteiksme joprojām ir TRUE!

Piemērs

Temperatūra servera telpā ir pārāk augsta.

Problēmas izteiksme:

```
last(/server/temp)>20
```

Atkopšanas izteiksme:

```
last(/server/temp)<=15
```

Note:
Nav lietderīgi izmantot makrosu {TRIGGER.VALUE} atkopšanas izteiksmē, jo šī izteiksme tiek novērtēta tikai tad, kad trigeris ir "Problēma" stāvoklī. Līdz ar to, novērtējot izteiksmi, {TRIGGER.VALUE} vienmēr tiks atrisināts kā "1" (kas norāda uz "Problēma" stāvokli).

Trigera nopietnība

Trigera nopietnību var izmantot, lai izvairītos no brīdinājumu trokšņa mazāk nopietnu problēmu gadījumā.

Skatiet visas labākās prakses.

Pārskats

Nopietnība atspoguļo trigera relatīvo svarīgumu.

Nopietnības piešķiršana ir noderīga ne tikai vizuālajam attēlojumam, bet arī, lai izvairītos no paziņojumiem par nenozīmīgām problēmām.

Trigeru nopietnības tiek izmantotas šādiem mērķiem:

- Vizuāls attēlojums (atsevišķa krāsa katrai izveidotajai problēmai, pamatojoties uz sākotnējo trigera nopietnību)
- Paziņojumu ierobežošana, pamatojoties uz nopietnību (skatiet darbību nosacījumus)
- Lietotāja datu nesēji — atšķirīgi datu nesēji (paziņojumu kanāli) dažādām nopietnībām (piemēram, SMS trigeriem ar *Augstu* un *Katastrofas* nopietnību, bet e-pasts trigeriem ar citām nopietnībām)
- Audio globālajos trauksmes signālos — atšķirīgs audio dažādām nopietnībām

Pēc noklusējuma tiek atbalstītas šādas trigeru nopietnības:

Nopietnība	Krāsa	Apraksts
Nav klasificēts	Pelēka	Var izmantot gadījumos, kad notikuma nopietnības līmenis nav zināms, nav noteikts, neietilpst regulārās uzraudzības tvērumā utt., piemēram, sākotnējās konfigurēšanas laikā, kā vietturi turpmākam novērtējumam vai kā daļu no integrācijas procesa.
Informācija	Gaiši zila	Var izmantot informatīviem notikumiem, kuriem nav nepieciešama tūlītēja uzmanība, bet kuri joprojām var sniegt vērtīgu ieskatu.
Brīdinājums	Dzeltena	Var izmantot, lai norādītu uz iespējamu problēmu, kurai var būt nepieciešama izmeklēšana vai rīcība, bet kas nav kritiska.
Vidēja	Oranža	Var izmantot, lai norādītu uz būtisku problēmu, kas būtu jānovērš salīdzinoši drīz, lai nepieļautu turpmākas problēmas.
Augsta	Gaiši sarkana	Var izmantot, lai norādītu uz kritiskām problēmām, kurām nepieciešama tūlītēja uzmanība, lai izvairītos no būtiskiem traucējumiem.
Katastrofa	Sarkana	Var izmantot, lai norādītu uz smagu incidentu, kam nepieciešama tūlītēja rīcība, lai, piemēram, novērstu sistēmas darbības pārtraukumus vai datu zudumu.

Trigeru nopietnību nosaukumus un krāsas var **pielāgot** sadaļā *Administrēšana* > *Vispārīgi* > *Trigeru attēlošanas opcijas*. Krāsas ir kopīgas visām lietotāja saskarne tēmas.

Pielāgotu nopietnības līmeņu nosaukumu tulkošana

> **Attention:**
> Ja tiek izmantoti Zabbix lietotāja saskarnes tulkojumi, pielāgotie nopietnības līmeņu nosaukumi pēc noklusējuma aizstās iztulkotos nosaukumus.

Noklusējuma trigeru nopietnības līmeņu nosaukumi ir pieejami tulkošanai visās lokalizācijās.
Ja nopietnības līmeņa nosaukums tiek mainīts, visās lokalizācijās tiek izmantots pielāgots nosaukums, un ir nepieciešams papildu manuāls tulkojums.

Pielāgota nopietnības līmeņa nosaukuma tulkošanas procedūra:

- iestatiet nepieciešamo pielāgoto nopietnības līmeņa nosaukumu, piemēram, 'Important'
- rediģējiet <frontend_dir>/locale/<required_locale>/LC_MESSAGES/frontend.po
- pievienojiet 2 rindas:

```
msgid "Important"
msgstr "<tulkojuma virkne>"
```

un saglabājiet failu.

- izveidojiet .mo failus, kā aprakstīts <frontend_dir>/locale/README

Šeit **msgid** ir jāsakrīt ar jauno pielāgoto nopietnības līmeņa nosaukumu, bet **msgstr** jābūt tā tulkojumam konkrētajā valodā.

Šī procedūra jāveic pēc katras nopietnības līmeņa nosaukuma maiņas.

Trigeru atkarības

Trigeru atkarības var izmantot, lai izvairītos no brīdinājumiem, kas nav saistīti ar pamatcēloni.

Skatiet visas labākās prakses.

Pārskats

Dažkārt viena hosta pieejamība ir atkarīga no cita. Serveris, kas atrodas aiz maršrutētāja, kļūs nesasniedzams, ja maršrutētājs pārstās darboties. Ja abiem ir konfigurēti trigeri, jūs varat saņemt paziņojumus par diviem nedarbojošiem hostiem, lai gan patiesībā vainīgs bija tikai maršrutētājs.

Šādā situācijā var noderēt atkarība starp hostiem. Ja atkarība ir iestatīta, atkarīgo objektu paziņojumi var tikt aizturēti un nosūtīts tikai paziņojums par pamatproblēmu.

Lai gan Zabbix tieši neatbalsta atkarības starp hostiem, tās var definēt ar citu, elastīgāku metodi — trigeru atkarībām. Trigerim var būt viens vai vairāki citi trigeri, no kuriem tas ir atkarīgs.

Tātad mūsu vienkāršajā piemērā mēs atveram servera trigeru konfigurācijas formu un iestatām, ka tas ir atkarīgs no attiecīgā maršrutētāja trigera. Ar šādu atkarību servera trigeris nemainīs savu stāvokli, kamēr trigeris, no kura tas ir atkarīgs, atrodas stāvoklī 'PROBLEM' — un tādējādi netiks veiktas nekādas atkarīgās darbības un netiks nosūtīti paziņojumi.

Ja gan serveris, gan maršrutētājs nedarbojas un ir iestatīta atkarība starp trigeri "server down" un trigeri "router down", Zabbix neizpildīs darbības atkarīgajam trigerim.

Kamēr vecāktrigeris atrodas stāvoklī PROBLEM, tā atkarīgie trigeri var ziņot vērtības, kurām nevar uzticēties. Tādēļ atkarīgie trigeri netiks pārvērtēti, kamēr vecāktrigeris (iepriekš minētajā piemērā maršrutētājs):

- atgriežas no stāvokļa 'PROBLEM' uz 'OK';
- maina savu stāvokli no 'PROBLEM' uz 'UNKNOWN';
- tiek aizvērts manuāli, ar korelācijas palīdzību vai izmantojot date and time un/vai nodata() funkcijas;
- tiek atrisināts ar tāda vienuma vērtību, kas nav iesaistīts atkarīgajā trigerī;
- tiek atspējots, tam ir atspējots vienums vai atspējots vienuma hosts

Visos iepriekš minētajos gadījumos atkarīgais trigeris (serveris) tiks pārvērtēts tikai tad, kad tam tiks saņemta jauna metrika. Tas nozīmē, ka atkarīgais trigeris var netikt atjaunināts nekavējoties.

Papildus:

- Trigera atkarību var pievienot no jebkura hosta trigera uz jebkuru citu hosta trigeri, ja vien tas neizraisa ciklisku atkarību.

- Trigera atkarību var pievienot no vienas veidnes uz citu. Ja kāds trigeris no veidnes A ir atkarīgs no kāda trigera no veidnes B, veidni A var piesaistīt hostam (vai citai veidnei) tikai kopā ar veidni B, bet veidni B var piesaistīt hostam (vai citai veidnei) arī atsevišķi.
- Trigera atkarību var pievienot no veidnes trigera uz hosta trigeri. Šajā gadījumā, piesaistot šādu veidni hostam, tiks izveidots hosta trigeris, kas ir atkarīgs no tās pašas trigera veidnes, no kuras trigeris bija atkarīgs. Tas ļauj, piemēram, izveidot veidni, kurā daži trigeri ir atkarīgi no maršrutētāja (hosta) trigeriem. Visi hosti, kas piesaistīti šai veidnei, būs atkarīgi no šī konkrētā maršrutētāja.
- Trigera atkarību nevar pievienot no hosta trigera uz veidnes trigeri.
- Trigera atkarību var pievienot no trigera prototipa uz citu trigera prototipu (viena un tā paša zema līmeņa atklāšanas noteikuma ietvaros) vai reālu trigeri. Trigera prototips nevar būt atkarīgs no trigera prototipa no cita LLD noteikuma vai no trigera, kas izveidots no trigera prototipa. Hosta trigera prototips nevar būt atkarīgs no trigera no veidnes.

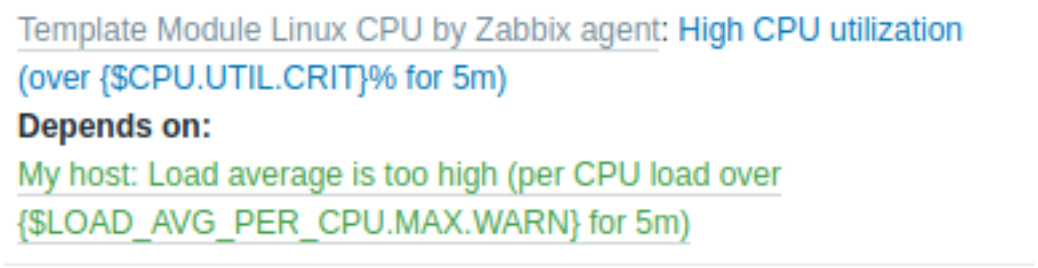
Konfigurācija

Lai definētu atkarību:

1. Atveriet cilni *Atkarības* trigera konfigurācijas formā.
2. Sadaļā *Atkarības* noklikšķiniet uz *Pievienot* un atlasiet vienu vai vairākus trigerus, no kuriem šis trigeris būs atkarīgs.

3. Noklikšķiniet uz *Atjaunināt*.

Tagad trigeru sarakstā ir redzama norāde par tā atkarību.



Vairāku atkarību piemērs

Piemēram, hosts atrodas aiz Router2, bet Router2 atrodas aiz Router1.

```
Zabbix - Router1 - Router2 - Host
```

Ja Router1 nedarbojas, tad ir acīmredzams, ka arī hosts un Router2 nav sasniedzami, tomēr saņemt trīs paziņojumus par to, ka hosts, Router1 un Router2 visi nedarbojas, ir pārmērīgi.

Tāpēc šajā gadījumā mēs definējam divas atkarības:

```
trigeris 'Host is down' ir atkarīgs no trigera 'Router2 is down'
trigeris 'Router2 is down' ir atkarīgs no trigera 'Router1 is down'
```

Pirms mainīt trigera 'Host is down' statusu, Zabbix pārbaudīs atbilstošās trigeru atkarības. Ja tādas tiks atrastas un viens no šiem trigeriem būs `Problem` stāvoklī, tad trigera stāvoklis netiks mainīts, darbības netiks izpildītas un paziņojumi netiks nosūtīti.

Zabbix šo pārbaudi veic rekursīvi. Ja Router1 vai Router2 nav sasniedzams, hosta trigeris netiks atjaunināts.

4 Notikumi

Pārskats

Zabbix ģenerē vairāku veidu notikumus:

- trigeru notikumi — ikreiz, kad trigeris maina savu statusu (*OK→PROBLEM→OK*)
- pakalpojumu notikumi — ikreiz, kad pakalpojums maina savu statusu (*OK→PROBLEM→OK*)

- atklāšanas notikumi — kad tiek noteikti hosti vai pakalpojumi
- automātiskās reģistrācijas notikumi — kad aktīvie aģenti tiek automātiski reģistrēti serverī
- iekšējie notikumi — kad vienums/zema līmeņa atklāšanas kārtula kļūst neatbalstīta vai trigeris pāriet nezināmā stāvoklī

Notikumiem tiek pievienotas laika atzīmes, un tie var kalpot par pamatu darbībām, piemēram, paziņojumu e-pasta nosūtīšanai u.c.

Lai lietotāja saskarnē skatītu notikumu detalizētu informāciju, dodieties uz *Monitoring* → *Problems*. Tur varat noklikšķināt uz notikuma datuma un laika, lai skatītu notikuma detalizētu informāciju.

Papildu informācija ir pieejama šeit:

- trigeru notikumi
- citi notikumu avoti

1 Trigera notikumu ģenerēšana

Pārskats

Trigera stāvokļa maiņa ir visbiežākais un vissvarīgākais notikumu avots. Katru reizi, kad trigeris maina savu stāvokli, tiek ģenerēts notikums. Notikums satur informāciju par trigera stāvokļa maiņu — kad tā notika un kāds ir jaunais stāvoklis.

Trigeri izveido divu veidu notikumus — `Problem` un `OK`.

Problēmu notikumi

Problēmas notikums tiek izveidots:

- kad trigera izteiksme tiek novērtēta kā TRUE, ja trigeris ir OK stāvoklī;
- katru reizi, kad trigera izteiksme tiek novērtēta kā TRUE, ja trigerim ir iespējota vairāku problēmu notikumu ģenerēšana.

OK notikumi

OK notikums aizver saistīto problēmas notikumu(-us), un to var izveidot 3 komponenti:

- trigeri — pamatojoties uz iestatījumiem ’OK event generation’ un ’OK event closes’;
- notikumu korelācija
- uzdevumu pārvaldnieks — kad notikums tiek manuāli aizvērts

Trigeri

Trigeriem ir iestatījums ’OK event generation’, kas nosaka, kā tiek ģenerēti OK notikumi:

- *Expression* - OK notikums tiek ģenerēts trigerim problēmas stāvoklī, kad tā izteiksme tiek novērtēta kā FALSE. Šis ir vienkāršākais iestatījums, kas ir iespējots pēc noklusējuma.
- *Recovery expression* - OK notikums tiek ģenerēts trigerim problēmas stāvoklī, kad tā izteiksme tiek novērtēta kā FALSE un atkopšanas izteiksme tiek novērtēta kā TRUE. To var izmantot, ja trigera atkopšanas kritēriji atšķiras no problēmas kritērijiem.
- *None* - OK notikums nekad netiek ģenerēts. To var izmantot kopā ar vairāku problēmu notikumu ģenerēšanu, lai vienkārši nosūtītu paziņojumu, kad kaut kas notiek.

Papildus tam trigeriem ir iestatījums ’OK event closes’, kas nosaka, kuri problēmu notikumi tiek aizvērti:

- *All problems* - OK notikums aizvērs visas atvērtās problēmas, ko izveidojis trigeris
- *All problems if tag values match* - OK notikums aizvērs atvērtās problēmas, ko izveidojis trigeris un kurām ir vismaz viena atbilstoša birkas vērtība. Birka tiek definēta trigera iestatījumā ’Tag for matching’. Ja nav problēmu notikumu, ko aizvērt, tad OK notikums netiek ģenerēts. To bieži sauc par notikumu korelāciju trigera līmenī.

Notikumu korelācija

Notikumu korelācija (saukta arī par globālo notikumu korelāciju) ir veids, kā iestatīt pielāgotus notikumu aizvēršanas noteikumus (kā rezultātā tiek ģenerēts OK notikums).

Noteikumi nosaka, kā jaunie problēmu notikumi tiek sapāroti ar esošajiem problēmu notikumiem, un ļauj aizvērt jauno notikumu vai atrastos notikumus, ģenerējot atbilstošus OK notikumus.

Tomēr notikumu korelācija ir jākonfigurē ļoti uzmanīgi, jo tā var negatīvi ietekmēt notikumu apstrādes veiktspēju vai, nepareizi konfigurēta, aizvērt vairāk notikumu, nekā paredzēts (sliktākajā gadījumā var tikt aizvērti pat visi problēmu notikumi). Daži konfigurācijas ieteikumi:

1. vienmēr samaziniet korelācijas tvērumu, iestatot unikālu birku kontroles notikumam (notikumam, kas tiek sapārots ar vecajiem notikumiem), un izmantojiet korelācijas nosacījumu ’jaunā notikuma birka’
2. neaizmirstiet pievienot nosacījumu, kas balstīts uz veco notikumu, izmantojot darbību ’aizvērt veco notikumu’, pretējā gadījumā var tikt aizvērtas visas esošās problēmas
3. izvairieties no bieži lietotu birku nosaukumu izmantošanas, ko lieto dažādas korelācijas konfigurācijas

Uzdevumu pārvaldnieks

Ja trigerim ir iespējots iestatījums “Allow manual close”, tad ir iespējams manuāli aizvērt trigerā ģenerētos problēmu notikumus. Tas tiek darīts lietotāja saskarnē, atjauninot problēmu. Notikums netiek aizvērts tieši — tā vietā tiek izveidots “close event” uzdevums, ko drīzumā apstrādā uzdevumu pārvaldnieks. Uzdevumu pārvaldnieks ģenerēs atbilstošu OK notikumu, un problēmas notikums tiks aizvērts.

2 Citi notikumu avoti

Servisa notikumi

Servisa notikumi tiek ģenerēti tikai tad, ja šo notikumu servisa darbības ir iespējotas. Šajā gadījumā katra servisa stāvokļa maiņa izveido jaunu notikumu:

- Problēmas notikums — kad servisa stāvoklis tiek mainīts no `OK` uz `Problem`
- OK notikums — kad servisa stāvoklis tiek mainīts no `Problem` uz `OK`

Notikums satur informāciju par servisa stāvokļa maiņu — kad tā notika un kāds ir jaunais stāvoklis.

Atklāšanas notikumi

Zabbix periodiski skenē IP diapazonus, kas definēti tīkla atklāšanas noteikumos. Pārbaudes biežums ir konfigurējams katram noteikumam atsevišķi. Tiklīdz tiek atklāts hosts vai pakalpojums, tiek ģenerēts atklāšanas notikums (vai vairāki notikumi).

Zabbix ģenerē šādus notikumus:

Notikums	Kad tiek ģenerēts
Pakalpojums darbojas	Katru reizi, kad Zabbix nosaka aktīvu pakalpojumu.
Pakalpojums nedarbojas	Katru reizi, kad Zabbix nevar noteikt pakalpojumu.
Hosts darbojas	Ja IP adresei vismaz viens no pakalpojumiem ir pieejams.
Hosts nedarbojas	Ja neviens no pakalpojumiem neatbild.
Pakalpojums atklāts	Ja pakalpojums atkal ir pieejams pēc dīkstāves vai tiek atklāts pirmo reizi.
Pakalpojums zaudēts	Ja pakalpojums tiek zaudēts pēc tam, kad tas bija pieejams.
Hosts atklāts	Ja hosts atkal ir pieejams pēc dīkstāves vai tiek atklāts pirmo reizi.
Hosts zaudēts	Ja hosts tiek zaudēts pēc tam, kad tas bija pieejams.

Aktīvā aģenta automātiskās reģistrācijas notikumi

Aktīvā aģenta automātiskā reģistrācija izveido notikumus Zabbix.

Ja tas ir konfigurēts, aktīvā aģenta automātiskās reģistrācijas notikums tiek izveidots, kad iepriekš nezināms aktīvais aģents pieprasa pārbaudes vai ja ir mainījušies hosta metadati. Serveris pievieno jaunu automātiski reģistrētu hostu, izmantojot saņemto aģenta IP adresi un portu.

Plašāku informāciju skatiet lapā aktīvā aģenta automātiskā reģistrācija.

Iekšējie notikumi

Iekšējie notikumi notiek, kad:

- vienums maina stāvokli no ’normal’ uz ’unsupported’
- vienums maina stāvokli no ’unsupported’ uz ’normal’
- zema līmeņa atklāšanas kārtula maina stāvokli no ’normal’ uz ’unsupported’
- zema līmeņa atklāšanas kārtula maina stāvokli no ’unsupported’ uz ’normal’
- trigeris maina stāvokli no ’normal’ uz ’unknown’
- trigeris maina stāvokli no ’unknown’ uz ’normal’

Iekšējo notikumu ieviešanas mērķis ir ļaut lietotājiem saņemt paziņojumus, kad notiek jebkurš iekšējais notikums, piemēram, ja vienums kļūst neatbalstīts un pārtrauc datu vākšanu.

Iekšējie notikumi tiek izveidoti tikai tad, ja šiem notikumiem ir iespējotas iekšējās darbības. Lai apturētu iekšējo notikumu ģenerēšanu (piemēram, vienumiem kļūstot neatbalstītiem), atspējojiet visas darbības iekšējiem notikumiem sadaļā Brīdinājumi → Darbības → Iekšējās darbības.

Note:
Ja iekšējās darbības ir atspējotas, kamēr objekts atrodas 'unsupported' stāvoklī, šim objektam joprojām tiks izveidots atkopšanas notikums.
Ja iekšējās darbības ir iespējotas, kamēr objekts atrodas 'unsupported' stāvoklī, šim objektam tiks izveidots atkopšanas notikums, pat ja objektam nav izveidots 'problem event'.

Skatiet arī: Paziņojuma saņemšana par neatbalstītiem vienumiem

3 Manuāla problēmu aizvēršana

Pārskats

Lai gan parasti problēmu notikumi tiek atrisināti automātiski, kad trigeris pāriet no stāvokļa **Problem** uz **OK**, var būt gadījumi, kad ir grūti noteikt, vai problēma ir atrisināta, izmantojot trigera izteiksmi. Šādos gadījumos problēma ir jāatrisina manuāli.

Piemēram, *syslog* var ziņot, ka daži kodola parametri ir jāpielāgo optimālai veiktspējai. Šajā gadījumā par problēmu tiek ziņots Linux administratoriem, viņi to novērš un pēc tam manuāli aizver problēmu.

Problēmas var aizvērt manuāli tikai tiem trigeriem, kuriem ir iespējota opcija *Allow manual close*.

Kad problēma tiek "manuāli aizvērta", Zabbix ģenerē jaunu iekšējo uzdevumu Zabbix serveris. Pēc tam process *task manager* izpilda šo uzdevumu un ģenerē OK notikumu, tādējādi aizverot problēmas notikumu.

Ģenerētais *OK* notikums ietver pilnu šim notikumam atrisināto notikuma birku kopu (ieskaitot no veidnēm, hostiem un trigeriem mantotās birkas). Šīs birkas ir pieejamas paziņojumos un makrosos, piemēram, `{EVENT.RECOVERY.TAGS}` un `{EVENT.RECOVERY.TAGSJSON}`.

Manuāli aizvērta problēma nenozīmē, ka pamatā esošais trigeris nekad vairs nepāries *Problem* stāvoklī. Trigera izteiksme tiek pārvērtēta, un tās rezultātā var atkal rasties problēma:

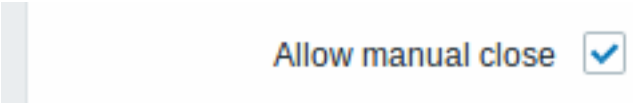
- Kad tiek saņemti jauni dati jebkuram vienumam, kas iekļauts trigera izteiksmē (ņemiet vērā, ka vērtības, kas atmestas ierobežošanas priekšapstrādes solī, netiek uzskatītas par saņemtām un neizraisīs trigera izteiksmes pārvērtēšanu);
- Kad izteiksmē tiek izmantotas date and time un/vai nodata() funkcijas.

Konfigurācija

Lai manuāli aizvērtu problēmu, ir nepieciešami divi soļi.

Trigera konfigurācija

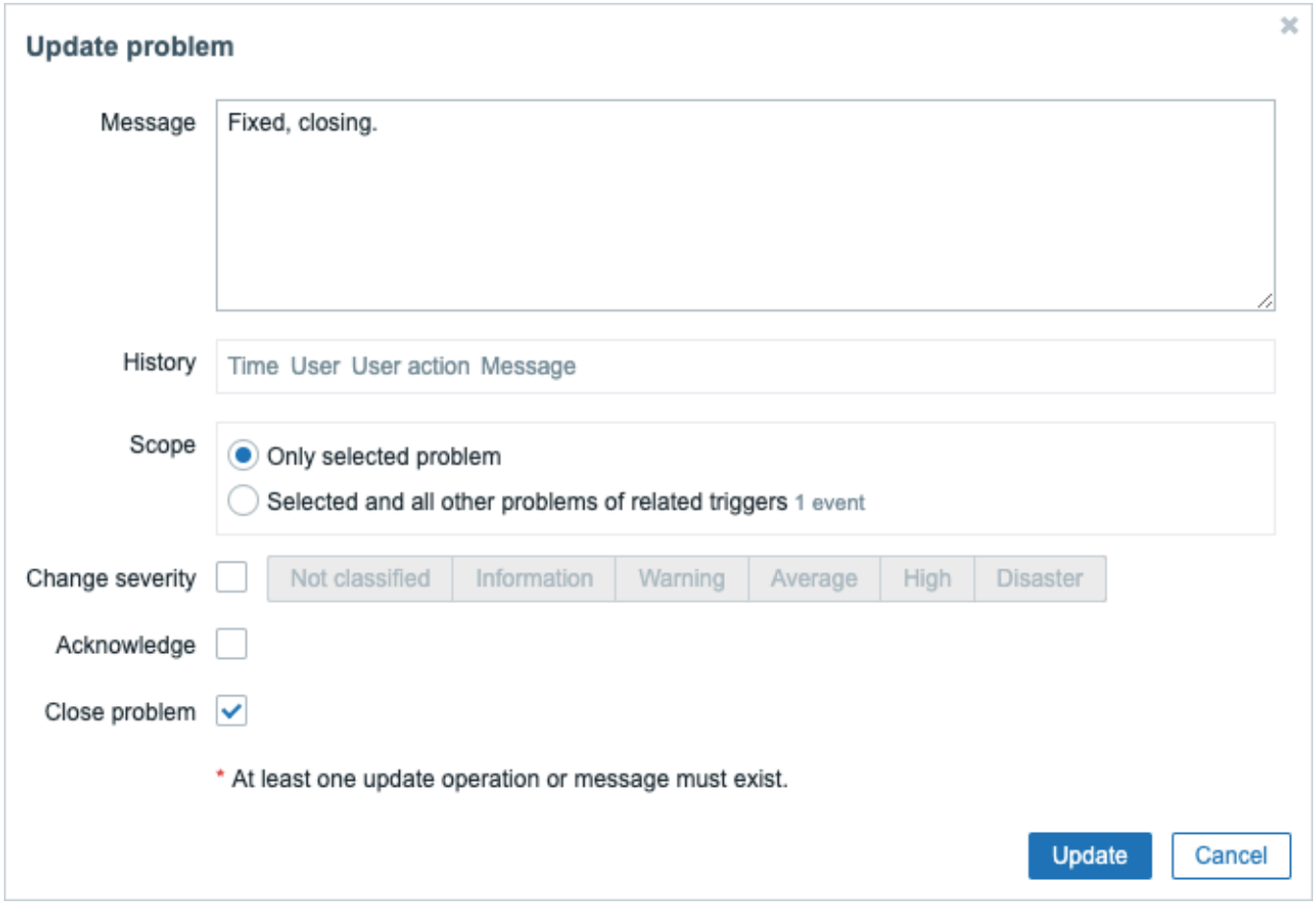
Trigera konfigurācijā iespējojiet opciju *Atļaut manuālu aizvēršanu*.



Problēmas atjaunināšanas logs

Ja trigerim ar karodziņu *Manual close* rodas problēma, varat atvērt šīs problēmas problēmas atjaunināšanas uznirstošo logu un aizvērt problēmu manuāli.

Lai aizvērtu problēmu, formā atzīmējiet opciju *Close problem* un noklikšķiniet uz *Update*.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Pieprasījumu apstrādā Zabbix serveris. Parasti problēmas aizvēršana aizņem dažas sekundes. Šī procesa laikā sadaļā *Monitoring* > *Problems* kā problēmas statuss tiek rādīts *CLOSING*.

Verifikācija

Var pārbaudīt, ka problēma ir aizvērta manuāli:

- notikuma detaļās, kas pieejamas sadaļā *Monitoring* > *Problems*;
- izmantojot `{EVENT.UPDATE.HISTORY}` makro paziņojumu ziņojumos, kas sniegs šo informāciju.

5 Notikumu korelācija

Pārskats

Notikumu korelācija ļauj korelēt problēmu notikumus ar to atrisinājumu ļoti precīzā un elastīgā veidā.

Notikumu korelāciju var definēt:

- trigera līmenī - vienu trigeri var izmantot, lai saistītu atsevišķas problēmas ar to risinājumu
- globāli - problēmas var korelēt ar to risinājumu no cita trigera/aptaujāšanas metodes, izmantojot globālās korelācijas kārtulas

1 Uz trigeriem balstīta notikumu korelācija

Pārskats

Uz trigeriem balstīta notikumu korelācija ļauj korelēt atsevišķas problēmas, par kurām ziņo viens trigeris.

Lai gan parasti OK notikums var aizvērt visus problēmu notikumus, ko izveidojis viens trigeris, ir gadījumi, kad nepieciešama detalizētāka pieeja. Piemēram, uzraugot žurnālfailus, iespējams, vēlēsieties noteikt konkrētas problēmas žurnālfailā un aizvērt tās atsevišķi, nevis visas kopā.

Tas attiecas uz trigeriem, kuriem parametram *PROBLEM event generation mode* ir iestatīta vērtība *Multiple*. Šādi trigeri parasti tiek izmantoti žurnālu uzraudzībai, slazdu apstrādei utt.

Zabbix ir iespējams saistīt problēmu notikumus, pamatojoties uz tagiem. Tagi tiek izmantoti, lai iegūtu vērtības un izveidotu problēmu notikumu identifikāciju. Izmantojot to, problēmas var arī aizvērt atsevišķi, pamatojoties uz atbilstošu tagu.

Citiem vārdiem sakot, viens un tas pats trigeris var izveidot atsevišķus notikumus, kurus identificē notikuma tags. Tādēļ problēmu notikumus var identificēt pa vienam un aizvērt atsevišķi, pamatojoties uz identifikāciju pēc notikuma taga.

Kā tas darbojas

Žurnālu uzraudzībā jūs varat sastapt rindas, kas ir līdzīgas šīm:

```
Line1: Pakalpojums 1 apstājās
Line2: Pakalpojums 2 apstājās
Line3: Pakalpojums 1 tika restartēts
Line4: Pakalpojums 2 tika restartēts
```

Notikumu korelācijas ideja ir spēt sasaistīt problēmas notikumu no Line1 ar atrisinājumu no Line3 un problēmas notikumu no Line2 ar atrisinājumu no Line4, un aizvērt šīs problēmas pa vienai:

```
Line1: Pakalpojums 1 apstājās
Line3: Pakalpojums 1 tika restartēts #problēma no Line 1 aizvērta

Line2: Pakalpojums 2 apstājās
Line4: Pakalpojums 2 tika restartēts #problēma no Line 2 aizvērta
```

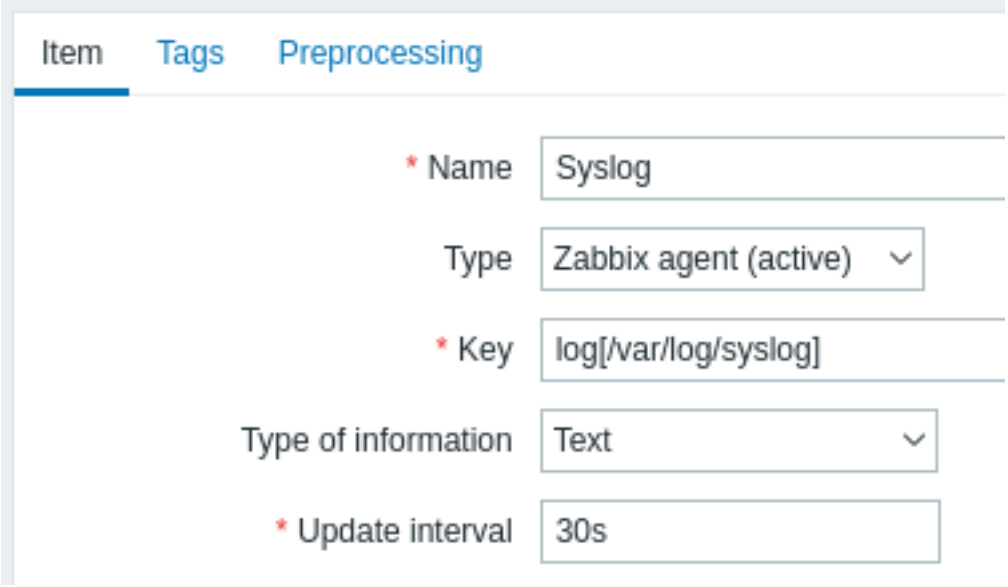
Lai to izdarītu, jums šie saistītie notikumi ir jāmarķē, piemēram, kā "Service 1" un "Service 2". To var izdarīt, piemērojot regulāro izteiksmi žurnāla rindai, lai iegūtu birkas vērtību. Pēc tam, kad notikumi tiek izveidoti, tiem attiecīgi tiek piešķirtas birkas "Service 1" un "Service 2", un problēmu var sasaistīt ar atrisinājumu.

Konfigurācija

Vienums

Lai sāktu, iespējams, vēlēsieties iestatīt vienumu, kas uzrauga žurnālfailu, piemēram:

```
log[/var/log/syslog]
```



Kad vienums ir iestatīts, uzgaidiet minūti, līdz tiek uztvertas konfigurācijas izmaiņas, un pēc tam dodieties uz Jaunākie dati, lai pārliecinātos, ka vienums ir sācis vākt datus.

Trigeris

Kad vienums darbojas, ir jākonfigurē trigeris. Ir svarīgi izlemt, kuri ieraksti žurnālfailā ir pelnījuši uzmanību. Piemēram, šāda trigera izteiksme meklēs virkni, piemēram, 'Stopping', lai signalizētu par iespējamām problēmām:

```
find(/My host/log[/var/log/syslog],,"regexp","Stopping")=1
```

Attention:
Lai nodrošinātu, ka katra rinda, kas satur virkni "Stopping", tiek uzskatīta par problēmu, trigera konfigurācijā iestatiet arī *Problem event generation mode* uz 'Multiple'.

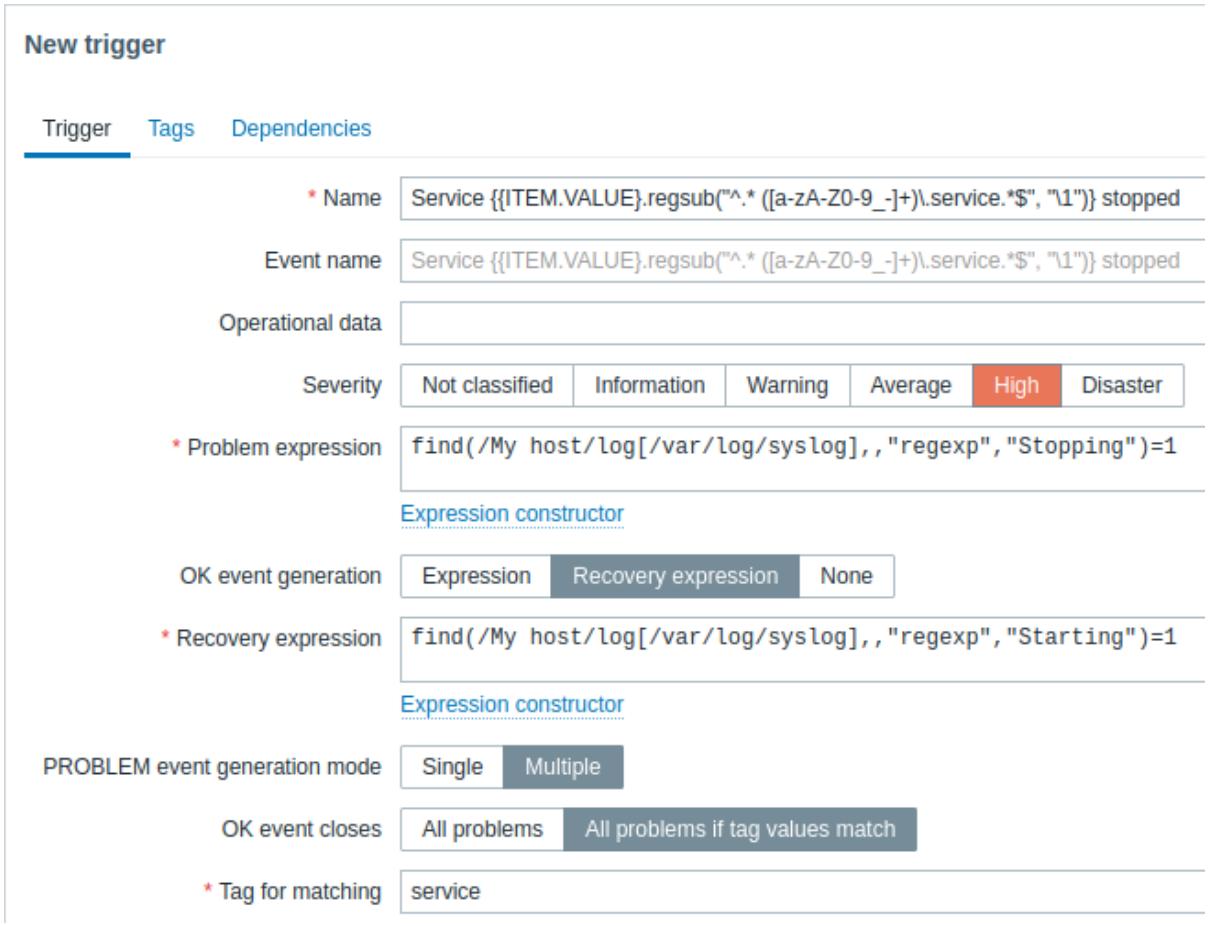
Pēc tam definējiet atkopšanas izteiksmi. Šī atkopšanas izteiksme atrisinās visas problēmas, ja tiks atrasta žurnāla rinda, kas satur virkni "Starting":

```
find(/My host/log[/var/log/syslog],,"regexp","Starting")=1
```

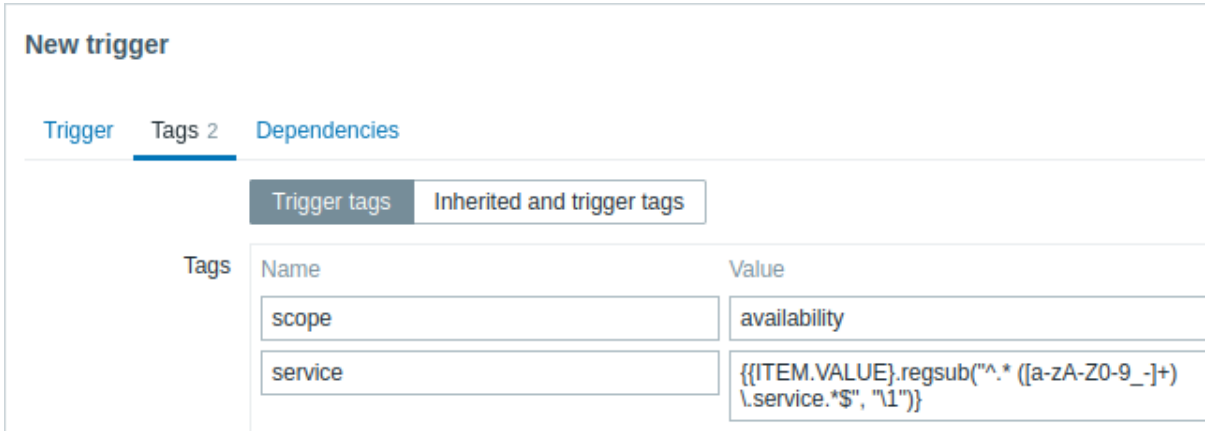
Tā kā mēs to nevēlamies, ir svarīgi kaut kā nodrošināt, lai tiktu aizvērtas atbilstošās pamatproblēmas, nevis vienkārši visas problēmas. Šeit var palīdzēt marķēšana.

Problēmas un to atrisinājumus var sasaistīt, norādot birku trigera konfigurācijā. Jāveic šādi iestatījumi:

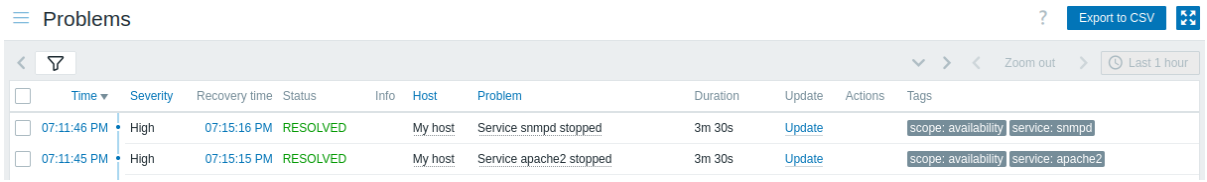
- *Problem event generation mode*: Multiple
- *OK event closes*: All problems if tag values match
- Ievadiet birkas nosaukumu notikumu saskaņošanai



- konfigurējiet birkas, lai izvilktu birku vērtības no žurnāla rindām



Ja konfigurācija ir veiksmīga, jūs varēsiet redzēt problēmu notikumus, kas marķēti pēc lietotnes un sasaistīti ar to atrisinājumu sadaļā *Monitoring* → *Problems*.



Warning:
Tā kā ir iespējama nepareiza konfigurācija, kad līdzīgas notikumu birkas var tikt izveidotas **nesaistītām** problēmām, lūdzu, iepazīstieties ar tālāk aprakstītajiem gadījumiem!

- Indeksētie makrosi vienmēr attiecas uz trigera konfigurācijas lauku *Expression*, nevis uz *Recovery expression*. Piemēram, atkopšanas notikumā **{ITEM.VALUE1}** tiks atrisināts kā pirmā vienuma jaunākā vērtība **problēmas izteiksmē** atkopšanas brīdī. Ja atkopšanas izteiksme ir balstīta uz citu vienumu un problēmas izteiksmes vienuma vērtība līdz atkopšanas brīdim mainās, notikumiem būs atšķirīgas birkas un tie netiks korelēti.

- Ja divas lietotnes raksta kļūdu un atkopšanas ziņojumus vienā un tajā pašā žurnālfailā, lietotājs var izlemt izmantot divas *service* birkas vienā un tajā pašā trigerī ar atšķirīgām birku vērtībām, izmantojot atsevišķas regulārās izteiksmes birku vērtībās, lai izvilktu, piemēram, servisa A un servisa B nosaukumus no makrosa {ITEM.VALUE} (piemēram, ja ziņojumu formāti atšķiras). Tomēr tas var nedarboties, kā plānots, ja regulārajām izteiksmēm nav atbilstības. Neatbilstošas regulārās izteiksmes radīs tukšas birku vērtības, un ar vienu tukšu birkas vērtību gan problēmas, gan OK notikumos pietiek, lai tās korelētu. Tāpēc servisa A atkopšanas ziņojums var nejauši aizvērt servisa B kļūdas ziņojumu.
- Faktiskās birkas un birku vērtības kļūst redzamas tikai tad, kad nostrādā trigeris. Ja izmantotā regulārā izteiksme nav derīga, tā klusi tiek aizstāta ar virkni *UNKNOWN*. Ja sākotnējais problēmas notikums ar birkas vērtību *UNKNOWN* tiek palaists garām, vēlāk var parādīties OK notikumi ar to pašu birkas vērtību *UNKNOWN*, kas var aizvērt problēmu notikumus, kurus tiem nevajadzētu aizvērt.
- Ja lietotājs izmanto makrosu {ITEM.VALUE} bez makrosu funkcijām kā birkas vērtību, tiek piemērots 255 rakstzīmju ierobežojums. Ja žurnāla ziņojumi ir gari un pirmās 255 rakstzīmes nav specifiskas, tas arī var radīt līdzīgas notikumu birkas nesaistītām problēmām.

2 Globālā notikumu korelācija

Pārskats

Globālā notikumu korelācija ļauj aptvert visus Zabbix uzraudzītos mērījumus un izveidot korelācijas.

Ir iespējams korelēt notikumus, ko izveidojuši pilnīgi atšķirīgi trigeri, un piemērot tiem visiem vienādas darbības. Izveidojot viedus korelācijas noteikumus, ir iespējams izvairīties no tūkstošiem atkārtotu paziņojumu un koncentrēties uz problēmas pamatcēloņiem!

Globālā notikumu korelācija ir jaudīgs mehānisms, kas ļauj atteikties no uz viena trigera balstītas problēmu un atrisināšanas loģikas. Līdz šim vienu problēmas notikumu izveidoja viens trigeris, un problēmas atrisināšanai mēs bijām atkarīgi no tā paša trigera. Mēs nevarējām atrisināt problēmu, ko izveidojis viens trigeris, ar citu trigeri. Taču ar uz notikumu tagiem balstītu notikumu korelāciju mēs to varam.

Piemēram, žurnāla trigeris var ziņot par lietotnes problēmām, kamēr aptaujas trigeris var ziņot, ka lietotne darbojas un ir pieejama. Izmantojot notikumu tagu priekšrocības, jūs varat piešķirt žurnāla trigerim tagu *status:down*, bet aptaujas trigerim — tagu *status:up*. Pēc tam globālās korelācijas noteikumā varat sasaistīt šos trigerus un piešķirt šai korelācijai atbilstošu darbību, piemēram, aizvērt vecos notikumus.

Citā pielietojumā globālā korelācija var identificēt līdzīgus trigerus un piemērot tiem vienādu darbību. Ko darīt, ja mēs vēlētos saņemt tikai vienu problēmas ziņojumu par katru tīkla porta problēmu? Nav nepieciešams ziņot par tiem visiem. Arī tas ir iespējams, izmantojot globālo notikumu korelāciju.

Globālā notikumu korelācija tiek konfigurēta **korelācijas noteikumos**. Korelācijas noteikums nosaka, kā jaunie problēmu notikumi tiek sasaistīti ar esošajiem problēmu notikumiem un ko darīt atbilstības gadījumā (aizvērt jauno notikumu, aizvērt atbilstošos vecos notikumus, ģenerējot atbilstošus OK notikumus). Ja problēma tiek aizvērta ar globālo korelāciju, tas tiek norādīts *Monitoring* > *Problems Info* kolonnā.

Globālās korelācijas noteikumu konfigurēšana ir pieejama tikai Super Admin līmeņa lietotājiem.

Attention:
Notikumu korelācija jākonfigurē ļoti rūpīgi, jo tā var negatīvi ietekmēt notikumu apstrādes veiktspēju vai nepareizas konfigurācijas gadījumā aizvērt vairāk notikumu, nekā bija paredzēts (sliktākajā gadījumā var tikt aizvērti pat visi problēmu notikumi).

Lai konfigurētu globālo korelāciju **droši**, ievērojiet šos svarīgos ieteikumus:

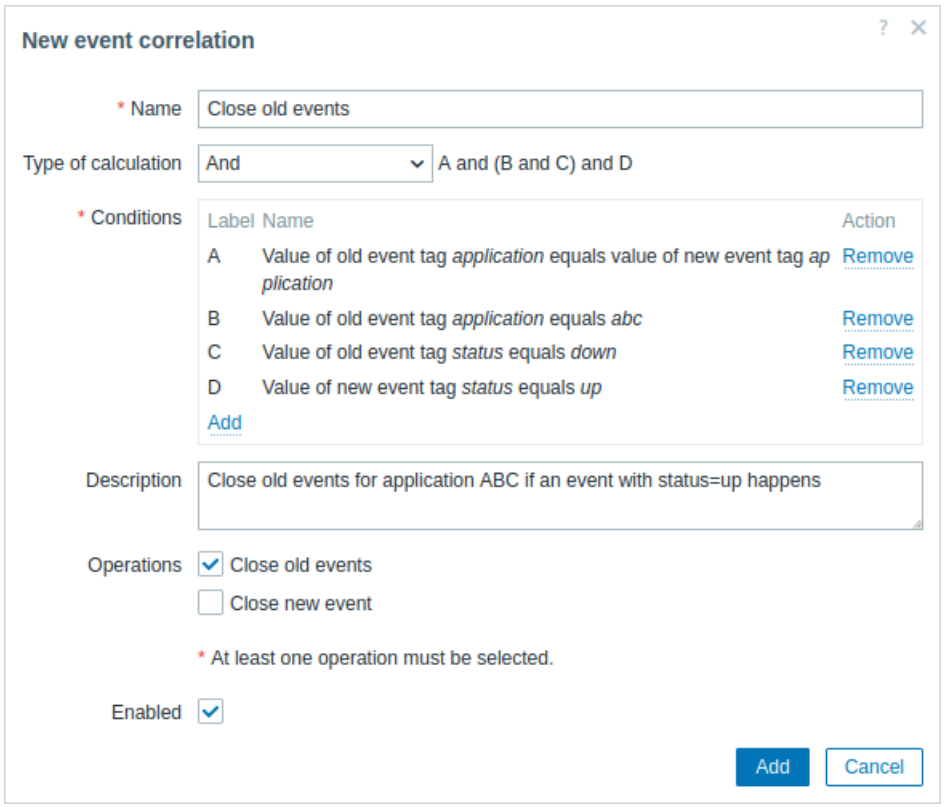
- Samaziniet korelācijas tvērumu. Vienmēr iestatiet unikālu tagu jaunajam notikumam, kas tiek sasaistīts ar vecajiem notikumiem, un izmantojiet korelācijas nosacījumu *New event tag name*.
- Pievienojiet skaidri definētu vecā notikuma nosacījumu, izmantojot **Close old events**. Vienmēr pievienojiet vismaz vienu *Old event* nosacījumu (piemēram, *Old event tag name*, *Old event tag value* vai *Event tag pair*), izvēloties **Close old events** — pretējā gadījumā noteikums var atbilst nesaistītiem esošajiem problēmu notikumiem un tos aizvērt (sliktākajā gadījumā — visas problēmas). Dodiet priekšroku **Event tag pair**, lai salīdzinātu izpildlaika vērtības (host:port, session id u.c.), un, ja iespējams, vēl vairāk sašauriniet atbilstību pēc hosta vai hostu grupas.
- Izvairieties no bieži lietotiem tagu nosaukumiem, kurus var izmantot dažādas korelācijas konfigurācijas.
- Ierobežojiet korelācijas noteikumu skaitu līdz tiem, kas jums patiešām ir nepieciešami.

Skatiet arī: zināmās problēmas.

Konfigurācija

Lai konfigurētu notikumu korelācijas noteikumus globāli:

- Dodieties uz *Datu ievākšana* > *Notikumu korelācija*
- Noklikšķiniet uz *Izveidot notikumu korelāciju* labajā pusē (vai uz korelācijas nosaukuma, lai rediģētu esošu noteikumu)
- Ievadiet korelācijas noteikuma parametrus formā

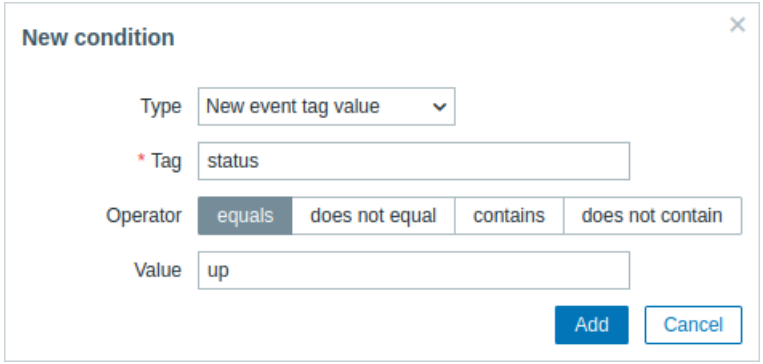


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Unikāls korelācijas noteikuma nosaukums.
Type of calculation	Ir pieejamas šādas nosacījumu aprēķināšanas opcijas: **And** - jāizpildās visiem nosacījumiem **Or** - pietiek, ja izpildās viens nosacījums **And/Or** - AND dažādiem nosacījumu tipiem un OR vienam un tam pašam nosacījumu tipam **Custom expression** - lietotāja definēta aprēķina formula darbības nosacījumu novērtēšanai. Tai jāietver visi nosacījumi (attēloti kā lielie burti A, B, C, ...) un tā var ietvert atstarpes, tabulācijas, iekavas ( ), **and** (reģistrjutīgs), **or** (reģistrjutīgs), **not** (reģistrjutīgs).
Conditions	Nosacījumu saraksts. Sīkāku informāciju par nosacījuma konfigurēšanu skatiet zemāk.
Description	Korelācijas noteikuma apraksts.

Parameter	Description
Operations	Atzīmējiet darbības izvēles rūtiņu, kas jāizpilda, kad notikums tiek korelēts. Ir pieejamas šādas darbības: **Close old events** - aizvērt vecos notikumus, kad notiek jauns notikums. Izmantojot darbību *Close old events*, vienmēr pievienojiet nosacījumu, kas balstīts uz veco notikumu, pretējā gadījumā var tikt aizvērtas visas esošās problēmas. **Close new event** - aizvērt jauno notikumu, kad tas notiek. **Warning!** Neatstājiet vecā/jaunā notikuma nosacījumus tukšus, izmantojot *Close old events*/*Close new event*. Ja atlasīsiet darbību *Close old events*, nepievienojot nosacījumu, kas atbilst vecajam notikumam, Zabbix var saskaņot visus esošos vecos notikumus un tos aizvērt. Vienmēr pievienojiet skaidri definētu vecā notikuma nosacījumu (piemēram, *Old event tag name* vai *Event tag pair*), izmantojot *Close old events*. Piemēram, noteikums, kas izmanto tikai *New event condition* un darbību *Close old events*, atbildīs visiem vecajiem notikumiem, kas atbilst (trūkstošajiem) vecā notikuma kritērijiem, — faktiski aizverot vecās problēmas.
Enabled	Ja atzīmēsiet šo izvēles rūtiņu, korelācijas noteikums tiks iespējots.

Lai konfigurētu jauna nosacījuma detaļas, blokā Conditions noklikšķiniet uz Add. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt nosacījuma detaļas.



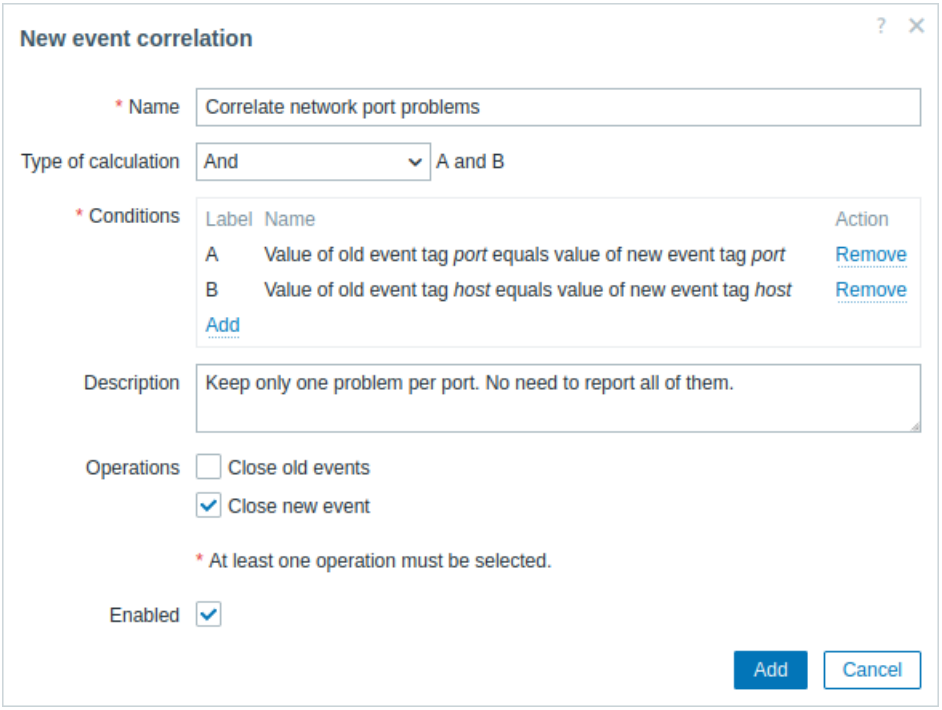
Parameter	Description
New condition	Atlasiet nosacījumu notikumu korelēšanai. *Ņemiet vērā*, ka, ja nav norādīts vecā notikuma nosacījums, var tikt saskaņoti un aizvērti visi vecie notikumi. Līdzīgi, ja nav norādīts jaunā notikuma nosacījums, var tikt saskaņoti un aizvērti visi jaunie notikumi. Ir pieejami šādi nosacījumi: **Old event tag name** - norādiet vecā notikuma birkas nosaukumu saskaņošanai. **New event tag name** - norādiet jaunā notikuma birkas nosaukumu saskaņošanai. **New event host group** - norādiet jaunā notikuma hostu grupu saskaņošanai. **Event tag pair** - norādiet jaunā notikuma birkas nosaukumu un vecā notikuma birkas nosaukumu saskaņošanai. Šajā gadījumā būs atbilstība, ja birku **vērtības** abos notikumos sakrīt. Birku *nosaukumiem* nav jāsakrīt. Šī opcija ir noderīga izpildlaika vērtību saskaņošanai, kuras konfigurēšanas laikā var nebūt zināmas (skatiet arī Piemērs). **Old event tag value** - norādiet vecā notikuma birkas nosaukumu un vērtību saskaņošanai, izmantojot šādus operatorus: *equals* - ir vecā notikuma birkas vērtība *does not equal* - nav vecā notikuma birkas vērtība *contains* - satur virkni vecā notikuma birkas vērtībā *does not contain* - nesatur virkni vecā notikuma birkas vērtībā **New event tag value** - norādiet jaunā notikuma birkas nosaukumu un vērtību saskaņošanai, izmantojot šādus operatorus: *equals* - ir jaunā notikuma birkas vērtība *does not equal* - nav jaunā notikuma birkas vērtība *contains* - satur virkni jaunā notikuma birkas vērtībā *does not contain* - nesatur virkni jaunā notikuma birkas vērtībā

Warning:
Tā kā ir iespējama nepareiza konfigurācija, ja līdzīgas notikumu birkas var tikt izveidotas **nesaistītām** problēmām, lūdzu, pārskatiet tālāk aprakstītos gadījumus!

- Faktiskās birkas un birku vērtības kļūst redzamas tikai tad, kad nostrādā trigeris. Ja izmantotā regulārā izteiksme nav derīga, tā tiek klusi aizstāta ar virkni *UNKNOWN*. Ja sākotnējais problēmas notikums ar birkas vērtību *UNKNOWN* tiek palaists garām, vēlāk var parādīties sekojoši OK notikumi ar to pašu birkas vērtību *UNKNOWN*, kas var aizvērt problēmu notikumus, kurus tiem nevajadzētu aizvērt.
- Ja lietotājs izmanto makrosu {ITEM.VALUE} bez makrosu funkcijām kā birkas vērtību, tiek piemērots 255 rakstzīmju ierobežojums. Ja žurnāla ziņojumi ir gari un pirmās 255 rakstzīmes nav specifiskas, tas arī var radīt līdzīgas notikumu birkas nesaistītām problēmām.

Piemērs

Apturiet atkārtotus problēmu notikumus no tā paša tīkla porta.



Šis globālais korelācijas noteikums korelēs problēmas, ja trigerim pastāv *host* un *port* birku vērtības un tās ir vienādas sākotnējā un jaunajā notikumā.

Darbība aizvērs jaunus problēmu notikumus tajā pašā tīkla portā, atstājot atvērtu tikai sākotnējo problēmu.

6 Marķēšana

Pārskats

Birkas sastāv no birkas nosaukuma un birkas vērtības. Marķējot entītijas, var izmantot tikai nosaukumu vai arī savienot to pārī ar vērtību (piemēram, `mysql`, `jira`, `target:mysql`, `service:jira` u.c.).

Birkas var definēt dažādām entītijām:

- Veidnes
- Hosti
- Vienumi
- Tīmekļa scenāriji
- Trigeri
- Pakalpojumi
- Veidņu vienumi un trigeri
- Hostu, vienumu un trigeru prototipi

Veidņu, *Hostu*, *Vienumu*, *Trigeru*, *Tīmekļa scenāriju* sarakstos un to prototipos kolonnā *Birkas* tiek parādītas gan parastās, gan mantotās birkas. Novietojot kursoru virs *mantotas* birkas vai noklikšķinot uz tās, tiek parādīts JavaScript uznirstošais paskaidro-

jums ar tekstu ”Mantotā birka”. Ja mantotā birka pastāv arī kā parastā birka, tiek rādīts cits paskaidrojuma teksts atkarībā no saraksta, kurā tā parādās (piemēram, ”Mantotā un veidnes birka” veidņu sarakstā vai ”Mantotā un hosta birka” hostu sarakstā). Citās lietotāja saskarnes vietās parastās un mantotās birkas tiek rādītas kopā (dublikāti tiek noņemti), bez ikonām un bez papildu paskaidrojuma teksta.

> **Note:**
> Skatiet oficiālās Zabbix vadlīnijas par vispārīgiem ieteikumiem birku definēšanai, kā arī konkrētas norādes par veidnēm, vienumiem, trigeriem un zemā līmeņa atklāšanas noteikumiem.

Birkām ir vairāki mērķi, jo īpaši notikumu marķēšana. Ja entītijas ir marķētas ar birkām, jebkurš jauns notikums, kas saistīts ar marķētu entītiju, mantos tās birkas. Piemēram:

- ar marķētām veidnēm — jebkura hosta problēma (ko izveidojuši trigeri no veidnes) mantos veidnes birkas.
- ar marķētiem hostiem — jebkura hosta problēma mantos hosta birkas.
- ar marķētiem vienumiem/tīmekļa scenārijiem — jebkura vienuma/tīmekļa scenārija problēma mantos vienuma/tīmekļa scenārija birkas.
- ar marķētiem trigeriem — jebkura problēma, ko izveidojis trigeris, mantos trigeru birkas.

Problēmas notikums manto visas birkas no visas entītiju ķēdes — veidnēm, hostiem, vienumiem/tīmekļa scenārijiem, trigeriem. Identiskas `tag:value` kombinācijas (pēc makrosu atrisināšanas) tiek apvienotas vienā, tādējādi novēršot dublēšanos.

Atkopšanas notikumi, kas ģenerēti ar manuālu aizvēršanu, ietver arī atrisinātās notikuma birkas, kas mantotas no veidnēm, hostiem, vienumiem/tīmekļa scenārijiem un trigeriem. Šīs birkas ir pieejamas paziņojumos un ar makrosiem, piemēram, `{EVENT.RECOVERY.TAGS}` un `{EVENT.RECOVERY.TAGSJSON}`.

Pielāgotās notikumu birkas nodrošina lielāku elastību. Piemēram:

- notikumu korelāciju var konfigurēt, pamatojoties uz notikumu birkām.
- darbību nosacījumus var konfigurēt, pamatojoties uz notikumu birkām.
- vienumu problēmas var grupēt, pamatojoties uz notikumu birkām.
- problēmu birkas var izmantot, lai piesaistītu problēmas pakalpojumiem.

Entītijas var marķēt ar vienādu birkas nosaukumu, bet atšķirīgām birkas vērtībām (piemēram, `component:memory` un `component:storage`). Līdzīgi entītijai var būt birka bez vērtības un tā pati birka ar vērtību (piemēram, `database` un `database:postgresql`). Šādas birkas netiek uzskatītas par dublikātiem.

Lietošanas gadījumi

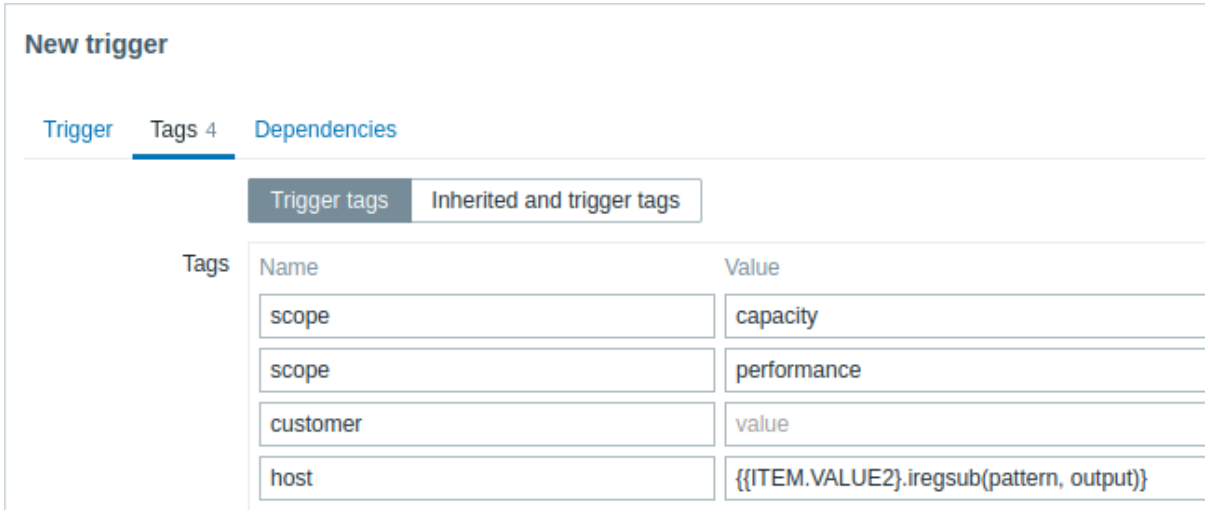
Daži bieži sastopami marķēšanas lietošanas gadījumi ir šādi:

1. Atzīmēt trigeru notikumus:
   - Definējiet trigera birku (piemēram, `scope:performance`).
   - Problēmām, kas izveidotas ar šo trigeri, būs trigera birka.
2. Atzīmēt no veidnes mantotās problēmas:
   - Definējiet veidnes birku (piemēram, `target:mysql`).
   - Problēmām, kas izveidotas ar trigeriem no šīs veidnes, būs veidnes birka.
3. Atzīmēt hosta problēmas:
   - Definējiet hosta birku (piemēram, `service:jira`).
   - Problēmām, kas izveidotas ar trigeriem no šī hosta, būs hosta birka.
4. Filtrēt saistītos vienumus:
   - Definējiet vienuma birku (piemēram, `component:cpu`).
   - Sadaļā *Monitoring* > *Latest data* vienumus var filtrēt pēc birkas `component:cpu`.
5. Izmantot no vienuma vērtības iegūto informāciju kā birkas vērtību:
   - Definējiet birku ar makrosu kā birkas vērtību (piemēram, `tag-name:{{ITEM.VALUE<N>}.regsub()}`).
   - Sadaļā *Monitoring* > *Problems* problēmām birkas vērtība tiks atrisināta uz datiem, kas iegūti no vienuma vērtības.
6. Identificēt problēmas žurnālfailā un aizvērt tās atsevišķi:
   - Definējiet trigera birku žurnāla uzraudzības vienuma trigerim, kas izmantos makrosu, lai iegūtu vērtības no vienuma vērtības (piemēram, `service:{{ITEM.VALUE<N>}.regsub()}`).
   - Trigera konfigurācijā iestatiet notikumu korelāciju:
     - iestatiet *PROBLEM event generation mode* uz ”Multiple”;
     - iestatiet *OK event closes* uz ”All problems if tag values match”;
     - iestatiet birku atbilstības noteikšanai.
   - Problēmām, kas izveidotas ar žurnāla vienuma trigeri, būs trigera birka, un tās tiks aizvērtas individuāli.
7. Filtrēt paziņojumus:
   - Definējiet trigeru birkas (piemēram, `scope:security` trigerim1 un `scope:availability` trigerim2).
   - Izmantojiet birku filtrēšanu darbību nosacījumos, lai saņemtu paziņojumus tikai par notikumiem, kas atbilst birku datiem.

8. Identificēt problēmas paziņojumos:
    - Definējiet trigeru birkas.
    - Izmantojiet {EVENT.TAGS} makrosu problēmas paziņojumā.
    - Problēmas paziņojumā būs iekļautas trigeru birkas, tādējādi atvieglojot noteikšanu, kurai lietojumprogrammai/pakalpojumam paziņojums pieder.
9. Vienkāršot konfigurācijas uzdevumus, izmantojot veidņu birkas:
    - Definējiet veidnes trigera birku.
    - Trigeriem, kas izveidoti no šī veidnes trigera, būs šī birka.
10. Izveidot trigerus ar birkām no zema līmeņa atklāšanas (LLD):
    - Definējiet trigera prototipa birku ar LLD makrosu birkas nosaukumā vai vērtībā (piemēram, `scope:{#FSNAME}`).
    - Trigeriem, kas izveidoti no trigera prototipa, būs šī birka.
11. Saskaņot pakalpojumus, izmantojot pakalpojumu birkas:
    - Definējiet pakalpojumu birkas.
    - Konfigurējiet pakalpojumu darbības pakalpojumiem ar atbilstošām birkām.
    - Papildus izmantojiet pakalpojumu birkas, lai saistītu pakalpojumu ar SLA SLA aprēķiniem.
12. Saistīt pakalpojumus ar problēmām, izmantojot pakalpojumu problēmu birkas:
    - Definējiet problēmas birku pakalpojuma konfigurācijā (piemēram, `target:mysql`).
    - Problēmas ar atbilstošu birku tiks automātiski korelētas ar pakalpojumu, un pakalpojuma statuss mainīsies atbilstoši konfigurētajiem pakalpojuma statusa aprēķināšanas noteikumiem.
13. Apspiest problēmas, kad hosts ir uzturēšanas režīmā:
    - Definējiet birkas uzturēšanas perioda konfigurācijā.
    - Problēmas ar definētajām birkām tiks apspiestas.
14. Piešķirt piekļuvi lietotāju grupām:
    - Definējiet birkas lietotāju grupas konfigurācijā.
    - Lietotāji lietotāju grupā varēs skatīt tikai problēmas ar definētajām birkām.

Konfigurācija

Tagus var definēt īpašā cilnē, piemēram, trigera konfigurācijā:



Makrosu atbalsts

Built-in un lietotāja makrosi tagos tiek atrisināti notikuma brīdī. Līdz notikuma iestāšanās brīdim šie makrosi Zabbix lietotāja saskarnē tiks rādīti neatrisināti.

Zema līmeņa atklāšanas makrosi tiek atrisināti atklāšanas procesa laikā.

Tālāk norādītos makrosus var izmantot trigeru tagu nosaukumos un vērtībās:

- {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}, un {HOST.ID} iebūvētie makrosi
- {INVENTORY.*} iebūvētie makrosi (lai atsauktos uz hosta inventāra vērtībām no viena vai vairākiem hostiem trigera izteiksmē)
- Lietotāja makrosi un lietotāja makrosi ar kontekstu (konteksts var ietvert zema līmeņa atklāšanas makrosus)
- Zema līmeņa atklāšanas makrosi (tikai trigera prototipa tagos)

Tālāk norādītos makrosus var izmantot veidnes, hosta un vienuma/tīmekļa scenārija tagu nosaukumos un vērtībās:

- {HOST.HOST}, {HOST.NAME}, {HOST.CONN}, {HOST.DNS}, {HOST.IP}, {HOST.PORT} un {HOST.ID} iebūvētie makrosi
- {INVENTORY.*} iebūvētie makrosi

- Lietotāja makrosi
- Zema līmeņa atklāšanas makrosi (tikai hosta un vienuma prototipu tagos)

Tālāk norādītos makrosus var izmantot uz trigeriem balstītos paziņojumos:

- {EVENT.TAGS} un {EVENT.RECOVERY.TAGS} iebūvētie makrosi (šie makrosi tiks atrisināti kā ar komatiem atdalīts notikuma tagu vai atkopšanas notikuma tagu saraksts)
- {EVENT.TAGSJSON} un {EVENT.RECOVERY.TAGSJSON} iebūvētie makrosi (šie makrosi tiks atrisināti kā JSON masīvs, kas satur notikuma taga objektus vai atkopšanas notikuma taga objektus)

Apakšvirknes izvilkšana trigeru tagos

Apakšvirknes izvilkšana tiek atbalstīta, lai aizpildītu taga nosaukumu vai taga vērtību, izmantojot makro funkciju. Funkcija piemēro regulāro izteiksmi vērtībai, kas iegūta no atbalstītā makro. Piemēram:

```
{{ITEM.VALUE}.regsub(pattern, output)}
{{ITEM.VALUE}.iregsub(pattern, output)}

{{#LLDMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.iregsub(pattern, output)}
```

Ja taga nosaukums vai vērtība pēc makro atrisināšanas pārsniedz 255 rakstzīmes, tā tiks saīsināta līdz 255 rakstzīmēm.

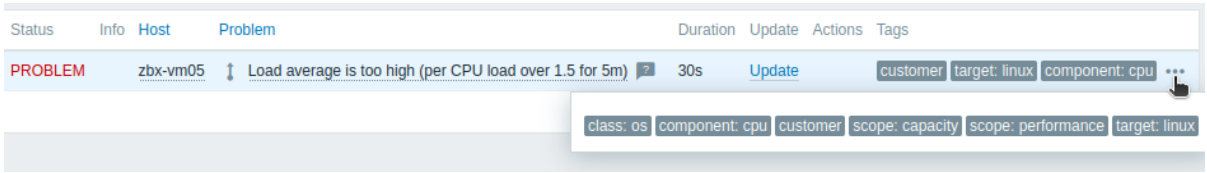
Skatiet arī: Makro funkciju izmantošana zema līmeņa atklāšanas makro notikumu tagošanai.

Notikumu birku skatīšana

Birku, ja tās ir definētas, var skatīt kopā ar jauniem notikumiem šeit:

- *Monitoring* > *Problems*
- *Monitoring* > *Problems* > *Event details*
- *Dashboards* > *Problems* widget

Parādīto birku secību un skaitu nosaka filtrēšanas opcijas *Tag display priority* un *Show tags* sadaļā *Monitoring* > *Problems* vai paneļa *Problems* logrīkā. Ņemiet vērā, ka var parādīt ne vairāk kā trīs birkas; ja birku ir vairāk, novietojot kursoru virs trim punktiem, visas birkas tiek parādītas uznirstošajā logā.



7 Vizualizācija

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam sadaļā Vizualizācija.

1 Grafiki

Pārskats

Kad Zabbix ieplūst liels datu apjoms, lietotājiem kļūst daudz vieglāk, ja viņi var aplūkot vizuālu notiekošā attēlojumu, nevis tikai skaitļus.

Te noder grafiki. Grafiki ļauj vienā acu uzmetienā uztvert datu plūsmu, saistīt problēmas, noteikt, kad kaut kas ir sācies, vai parādīt, kad kaut kas varētu pārvērsties par problēmu.

Zabbix lietotājiem nodrošina:

- iebūvētus vienkāršos grafikus viena vienuma datiem
- iespēju izveidot sarežģītākus pielāgotus grafikus
- ātru piekļuvi vairāku vienumu salīdzināšanai ad-hoc grafikos
- modernus pielāgojamus vektoru grafikus un sektoru diagrammas

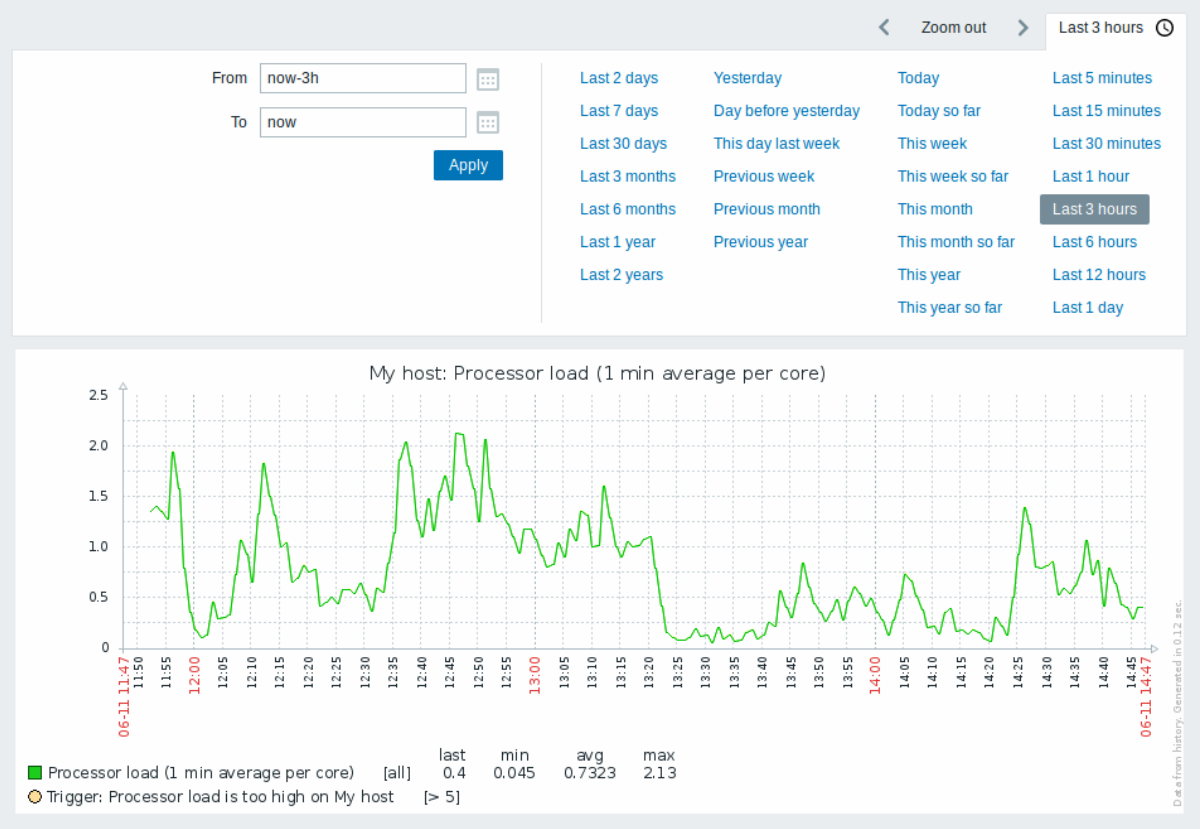
1 Vienkārši grafiki

Pārskats

Vienkāršie grafiki ir pieejami, lai vizualizētu datus, kas savākti ar vienumiem.

Lai skatītu vienkāršos grafikus, no lietotāja puses nav nepieciešama nekāda konfigurēšana. Tos brīvi nodrošina Zabbix.

Vienkārši dodieties uz *Uzraudzība* > *Jaunākie dati* un noklikšķiniet uz saites Graph attiecīgajam vienumam, un tiks parādīts grafiks.



Note:
Vienkāršie grafiki ir pieejami visiem skaitliskajiem vienumiem. Tekstuāliem vienumiem sadaļā *Uzraudzība* > *Jaunākie dati* ir pieejama saite uz vēsturi.

Laika perioda atlasītājs

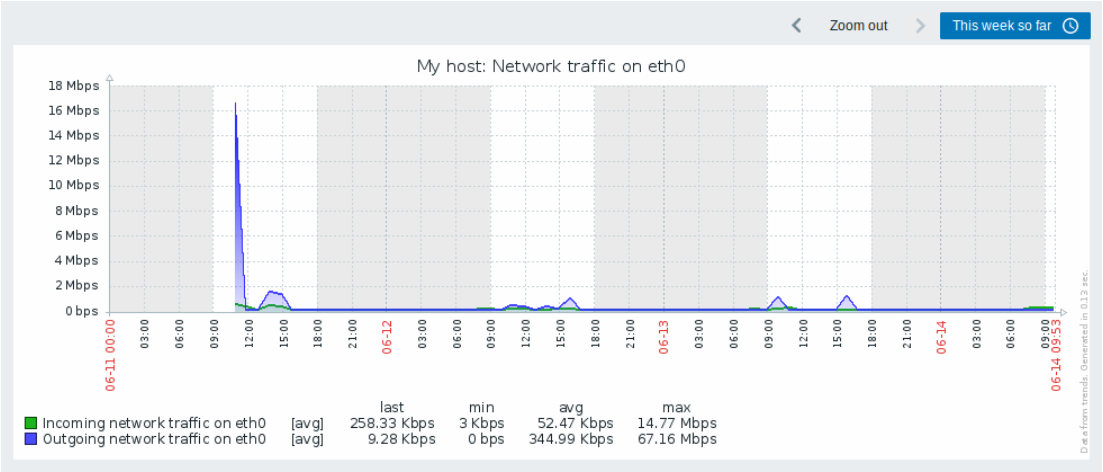
Laika perioda atlasītājs virs diagrammas ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamus periodus. Papildinformāciju skatiet sadaļā Laika perioda un hosta atlasītāji.

Jaunākie dati salīdzinājumā ar garākiem periodiem

Ļoti neseniem datiem tiek zīmēta **viena** līnija, kas savieno katru saņemto vērtību. Viena līnija tiek zīmēta tik ilgi, kamēr vienai vērtībai ir pieejams vismaz viens horizontāls pikselis.

Datiem, kas attēlo garāku periodu, tiek zīmētas **trīs līnijas** — tumši zaļa līnija rāda vidējo vērtību, savukārt gaiši rozā un gaiši zaļa līnija rāda maksimālās un minimālās vērtības attiecīgajā laika brīdī. Telpa starp augstākajām un zemākajām vērtībām tiek aizpildīta ar dzeltenu fonu.

Darba laiks (darba dienas) grafikos tiek attēlots ar baltu fonu, savukārt ārpusdarba laiks tiek attēlots pelēkā krāsā (ar noklusējuma lietotāja saskarne tēmu *Original blue*).



Darba laiks vienmēr tiek attēlots vienkāršajos grafikos, savukārt tā attēlošana pielāgotajos grafikos ir lietotāja izvēle.

Darba laiks netiek attēlots, ja grafiks rāda periodu, kas pārsniedz 3 mēnešus.

Trigeru līnijas

Vienkārši trigeri tiek attēloti kā līnijas ar melniem pārtraukumiem virs trigeru nopietnības krāsas — pievērsiet uzmanību zilajai līnijai grafikā un trigeru informācijai, kas parādīta leģendā. Grafikā var attēlot līdz 3 trigeru līnijām; ja trigeru ir vairāk, prioritāte tiek piešķirta trigeriem ar zemāku nopietnību. Trigeri vienmēr tiek attēloti vienkāršajos grafikos, savukārt to attēlošana pielāgotajos grafikos ir lietotāja izvēle.

Ģenerēšana no vēstures/tendencēm

Grafikus var zīmēt, pamatojoties vai nu uz vienuma vēsturi vai tendencēm.

Lietotājiem, kuriem ir aktivizēts lietotāja saskarnes atkļūdošanas režīms, grafika apakšējā labajā stūrī tiek parādīts pelēks, vertikāls uzraksts, kas norāda, no kurienes nāk dati.

To, vai tiek izmantota vēsture vai tendences, ietekmē vairāki faktori:

- vienuma vēstures glabāšanas ilgums. Piemēram, vienuma vēsturi var glabāt 14 dienas. Šādā gadījumā visi dati, kas ir vecāki par četrpadsmit dienām, tiks ņemti no tendencēm.
- datu blīvums grafikā. Ja sekundžu skaits, kas jāattēlo vienā horizontālā grafika pikselī, pārsniedz 3600/16, tiek attēloti tendenču dati (pat ja vienuma vēsture par to pašu periodu joprojām ir pieejama).
- ja tendences ir atspējotas, grafika veidošanai tiek izmantota vienuma vēsture — ja tā ir pieejama attiecīgajam periodam.

Datu neesamība

Vienumiem ar regulāru atjaunināšanas intervālu grafikā nekas netiek attēlots, ja vienuma dati netiek ievākti.

Tomēr trapper vienumiem un vienumiem ar plānotu atjaunināšanas intervālu (un regulāro atjaunināšanas intervālu, kas iestatīts uz 0) tiek uzzīmēta taisna līnija līdz pirmajai ievāktajai vērtībai un no pēdējās ievāktās vērtības līdz grafika beigām; līnija attiecīgi atrodas pirmās/pēdējās vērtības līmenī.

Pārslēgšanās uz neapstrādātām vērtībām

Nolaižamā izvēlne augšējā labajā stūrī ļauj pārslēgties no vienkāršā grafika uz *Vērtības/500 jaunākās vērtības* sarakstiem. Tas var būt noderīgi, lai skatītu skaitliskās vērtības, kas veido grafiku.

Šeit attēlotās vērtības ir neapstrādātas, t.i., netiek izmantotas ne mērvienības, ne vērtību pēcapstrāde. Tomēr vērtību kartēšana tiek piemērota.

Zināmās problēmas

Skatiet zināmās problēmas attiecībā uz grafikiem.

2 Pielāgoti grafiki

Pārskats

Pielāgotie grafiki, kā jau nosaukums liecina, piedāvā pielāgošanas iespējas.

Lai gan vienkāršie grafiki ir piemēroti viena vienuma datu skatīšanai, tie nepiedāvā konfigurēšanas iespējas.

Tādēļ, ja vēlaties mainīt grafika stilu vai līniju attēlošanas veidu, vai salīdzināt vairākus vienumus, piemēram, ienākošo un izejošo datplūsmu vienā grafikā, jums ir nepieciešams pielāgots grafiks.

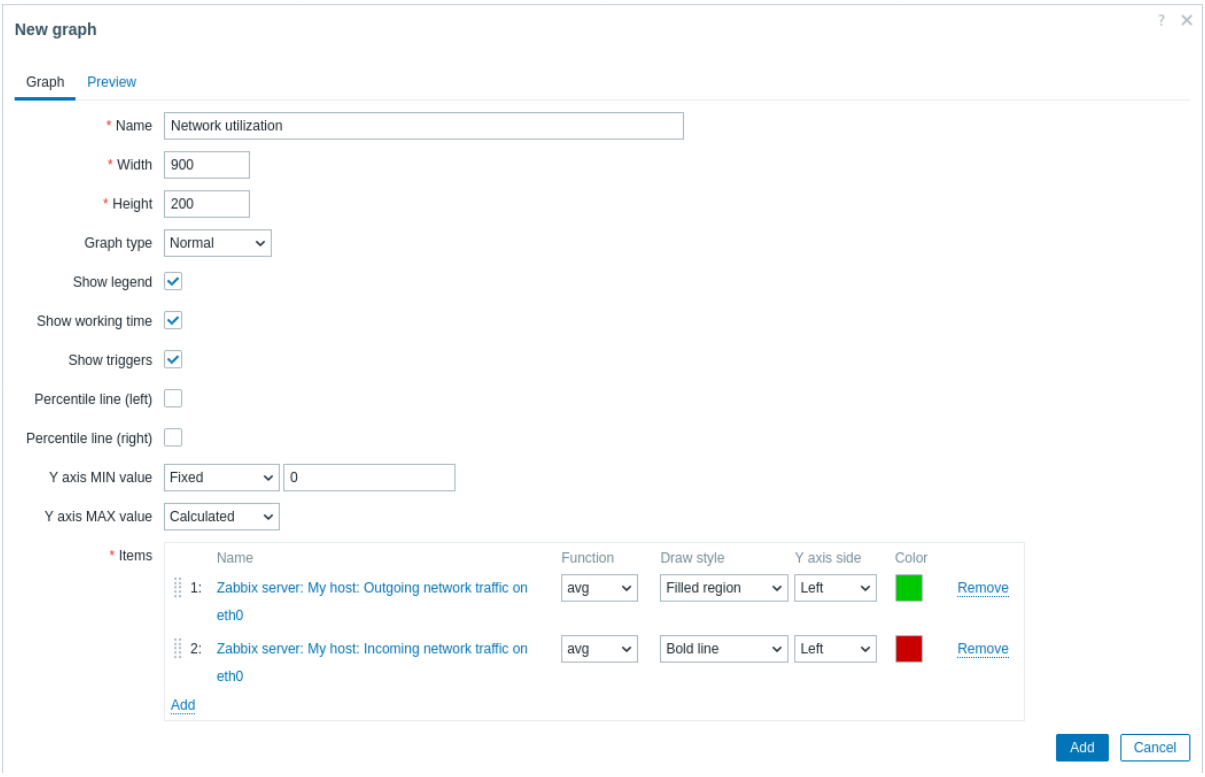
Pielāgotie grafiki tiek konfigurēti manuāli.

Tos var izveidot vienam hostam, vairākiem hostiem vai vienai veidnei.

Pielāgotu grafiku konfigurēšana

Lai izveidotu pielāgotu grafiku, rīkojieties šādi:

- Dodieties uz *Datu ievākšana > Hosti (vai Veidnes)*
- Noklikšķiniet uz *Grafiki* rindā blakus vajadzīgajam hostam vai veidnei
- Grafiku ekrānā noklikšķiniet uz *Izveidot grafiku*
- Rediģējiet grafika atribūtus



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

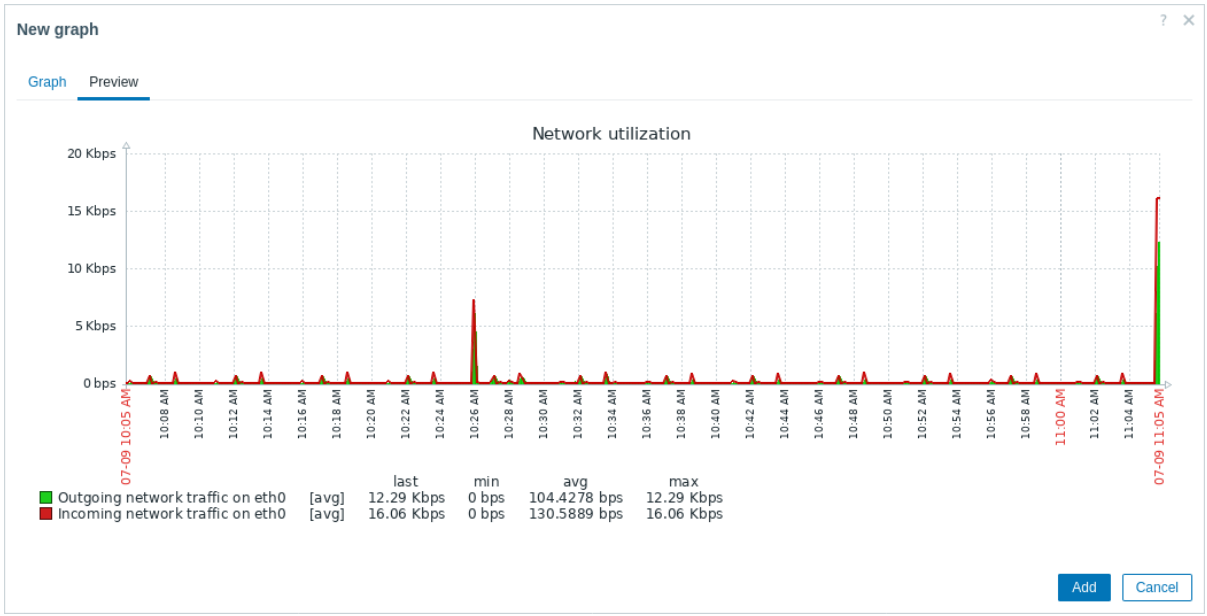
Grafika atribūti:

Parameter	Description
Name	Unikāls grafika nosaukums. Šajā laukā tiek atbalstītas izteiksmes makrosu formā, bet tikai ar `avg`, `last`, `min` un `max` funkcijām, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`). {HOST.HOST<1-9>} makrosi tiek atbalstīti izmantošanai šajā makrosā, atsaucoties uz pirmo, otro, trešo utt. hostu grafikā, piemēram, `{?avg(/{HOST.HOST2}/key,1h)}`. Ņemiet vērā, ka atsauce uz pirmo hostu ar šo makrosu ir lieka, jo uz pirmo hostu var atsaukties netieši, piemēram, `{?avg(//key,1h)}`.
Width	Grafika platums pikseļos (tikai priekšskatījumam un sektoru/izceltajiem sektoru grafikiem).
Height	Grafika augstums pikseļos.
Graph type	Grafika tips: **Normal** - parasts grafiks, vērtības tiek attēlotas kā līnijas **Stacked** - slāņots grafiks, tiek attēloti aizpildīti apgabali **Pie** - sektoru grafiks **Exploded** - ”izcelts” sektoru grafiks, daļas tiek attēlotas kā ”atdalītas” no sektoru diagrammas
Show legend	Atzīmējot šo izvēles rūtiņu, tiks parādīta grafika leģenda.
Show working time	Ja atlasīts, ārpus darba laika stundas tiks parādītas ar pelēku fonu. Šis parametrs nav pieejams sektoru un izceltajiem sektoru grafikiem.
Show triggers	Ja atlasīts, vienkāršie trigeri tiks attēloti kā līnijas ar melniem pārtraukumiem virs trigera svarīguma krāsas. Šis parametrs nav pieejams sektoru un izceltajiem sektoru grafikiem.
Percentile line (left)	Parādīt procentili kreisajai Y asij. Ja, piemēram, ir iestatīta 95% procentile, tad procentiles līnija atradīsies līmenī, zem kura ietilpst 95 procenti vērtību. Tiek attēlota kā spilgti zaļa līnija. Pieejams tikai parastajiem grafikiem.
Percentile line (right)	Parādīt procentili labajai Y asij. Ja, piemēram, ir iestatīta 95% procentile, tad procentiles līnija atradīsies līmenī, zem kura ietilpst 95 procenti vērtību. Tiek attēlota kā spilgti sarkana līnija. Pieejams tikai parastajiem grafikiem.
Y axis MIN value	Y ass minimālā vērtība: **Calculated** - Y ass minimālā vērtība tiks aprēķināta automātiski. **Fixed** - fiksēta Y ass minimālā vērtība. **Item** - atlasītā vienuma pēdējā vērtība būs minimālā vērtība. Šis parametrs nav pieejams sektoru un izceltajiem sektoru grafikiem.
Y axis MAX value	Y ass maksimālā vērtība: **Calculated** - Y ass maksimālā vērtība tiks aprēķināta automātiski. **Fixed** - fiksēta Y ass maksimālā vērtība. **Item** - atlasītā vienuma pēdējā vērtība būs maksimālā vērtība Šis parametrs nav pieejams sektoru un izceltajiem sektoru grafikiem.
3D view	Iespējot 3D stilu. Tikai sektoru un izceltajiem sektoru grafikiem.
Items	Vienumi, kuru dati tiks attēloti šajā grafikā. Noklikšķiniet uz *Pievienot*, lai atlasītu vienumus. Varat arī atlasīt dažādas attēlošanas opcijas (funkcija, zīmēšanas stils, kreisās/labās ass attēlojums, krāsa).
Sort order (0→100)	Zīmēšanas secība. Vispirms tiks apstrādāts 0. To var izmantot, lai zīmētu līnijas vai apgabalus aiz (vai priekšā) citam elementam. Varat vilkt un nomest vienumus, izmantojot ikonu rindas sākumā, lai iestatītu kārtošanas secību vai noteiktu, kurš vienums tiek attēlots cita priekšā.
Name	Atlasītā vienuma nosaukums tiek parādīts kā saite. Noklikšķinot uz saites, tiek atvērts citu pieejamo vienumu saraksts.
Type	Tips (pieejams tikai sektoru un izceltajiem sektoru grafikiem): **Simple** - vienuma vērtība tiek proporcionāli attēlota sektoru diagrammā **Graph sum** - vienuma vērtība attēlo visu sektoru diagrammu Ņemiet vērā, ka ”graph sum” vienuma krāsojums būs redzams tikai tādā apjomā, kādā to neaizņem ”proportional” vienumi.

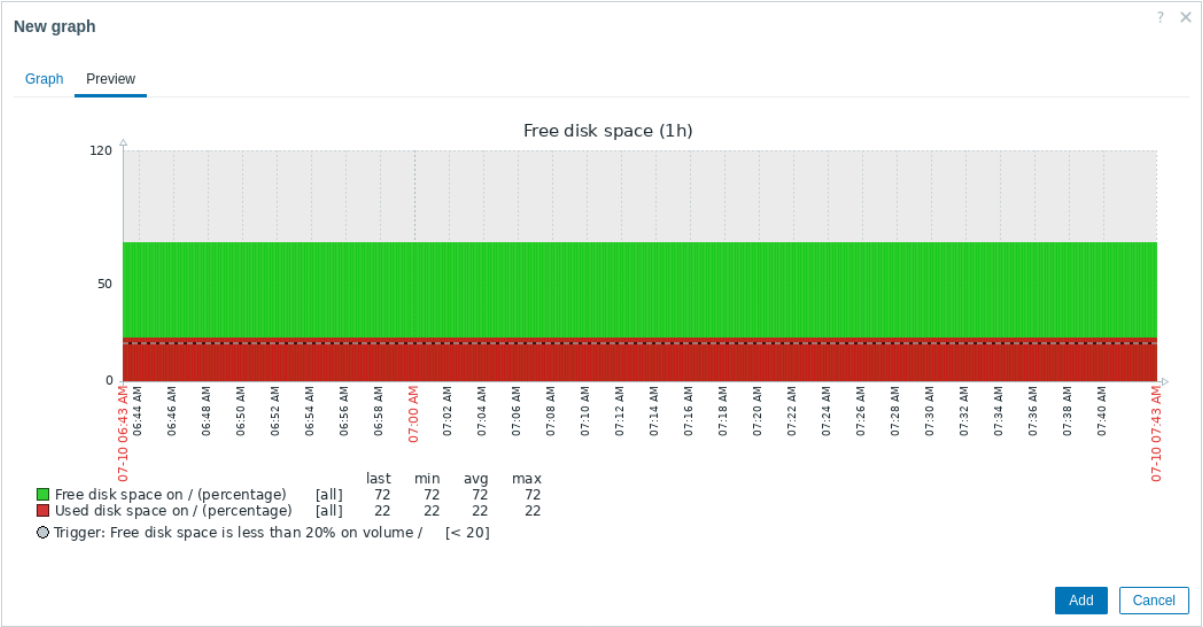
Parameter	Description
Function	Atlasiet, kuras vērtības tiks attēlotas, ja vienam vienumam uz vienu vertikālo grafika pikseli ir vairāk nekā viena vērtība: **all** - grafikā attēlot visas iespējamās vērtības (minimālo, maksimālo, vidējo). Ņemiet vērā, ka īsākiem periodiem šim iestatījumam nav ietekmes; tikai garākiem periodiem, kad datu blīvums vienā vertikālajā grafika pikselī palielinās, 'all' sāk attēlot minimālās, maksimālās un vidējās vērtības. Šī funkcija ir pieejama tikai *Normal* grafika tipam. Skatiet arī: Grafiku ģenerēšana no history/trends. **avg** - attēlot vidējās vērtības **last** - attēlot jaunākās vērtības. Šī funkcija ir pieejama tikai tad, ja kā grafika tips ir atlasīts *Pie/Exploded pie*. **max** - attēlot maksimālās vērtības **min** - attēlot minimālās vērtības
Draw style	Atlasiet zīmēšanas stilu (pieejams tikai parastajiem grafikiem; slāņotajiem grafikiem vienmēr tiek izmantots aizpildīts apgabals), ko piemērot vienuma datiem - *Līnija*, *Treknā līnija*, *Aizpildīts apgabals*, *Punkts*, *Pārtraukta līnija*, *Gradienta līnija*.
Y axis side	Atlasiet Y ass pusi, kurā parādīt vienuma datus - *Kreisā*, *Labā*.
Color	Atlasiet krāsu, ko piemērot vienuma datiem.

Grafika priekšskatījums

Cilnē *Priekšskatījums* tiek parādīts grafika priekšskatījums, lai jūs varētu uzreiz redzēt, ko veidojat.



Ņemiet vērā, ka priekšskatījumā netiks rādīti nekādi dati veidnes vienumiem.



Šajā piemērā pievērsiet uzmanību pārtrauktajai treknajai līnijai, kas attēlo trigera līmeni, un trigera informācijai, kas parādīta leģendā.

Note:
Var tikt parādītas ne vairāk kā 3 trigeru līnijas. Ja trigeru ir vairāk, parādīšanai prioritāri tiek izvēlēti trigeri ar zemāku nopietnības līmeni.

 Ja grafika augstums ir iestatīts mazāks par 120 pikseļiem, leģendā netiks parādīts neviens trigeris.

3 Ad-hoc grafiki

Pārskats

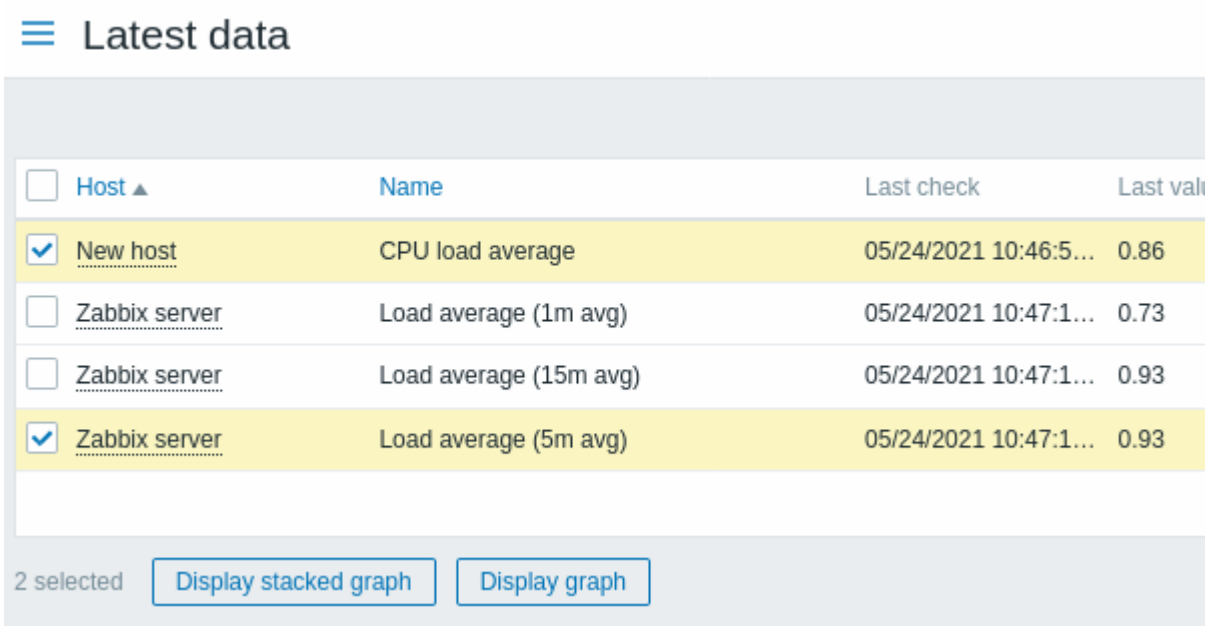
Lai gan vienkāršs grafiks ir lieliski piemērots viena vienuma datu apskatei un pielāgoti grafiki piedāvā pielāgošanas iespējas, neviens no tiem neļauj ātri izveidot salīdzinājuma grafiku vairākiem vienumiem ar nelielu piepūli un bez uzturēšanas.

Lai risinātu šo problēmu, ir iespējams ļoti ātri izveidot ad hoc grafikus vairākiem vienumiem.

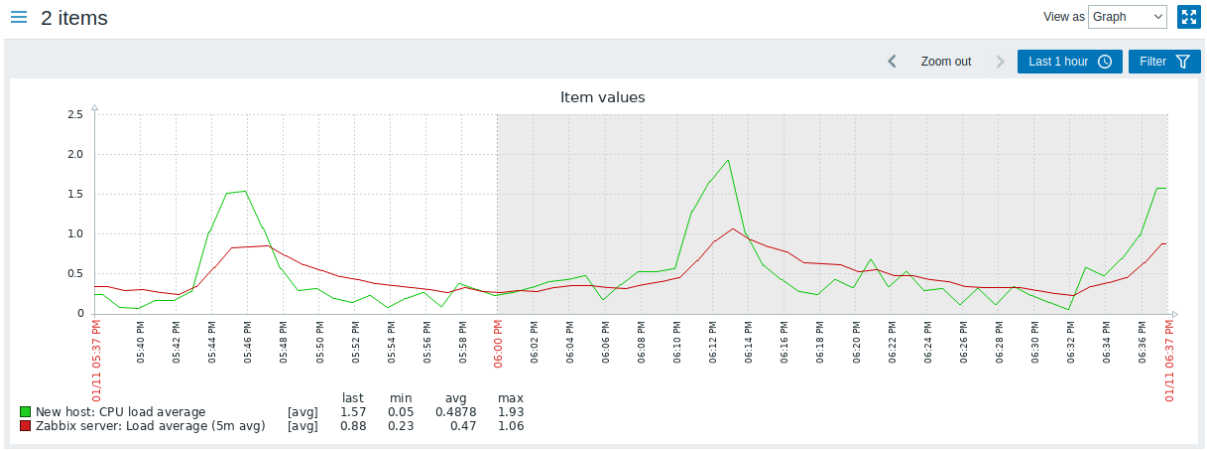
Konfigurācija

Lai izveidotu ad hoc grafiku, rīkojieties šādi:

- Dodieties uz *Uzraudzība → Jaunākie dati*
- Izmantojiet filtru, lai parādītu vajadzīgos vienumus
- Atzīmējiet to vienumu izvēles rūtiņas, kurus vēlaties attēlot grafikā
- Noklikšķiniet uz pogas *Display stacked graph* vai *Display graph*

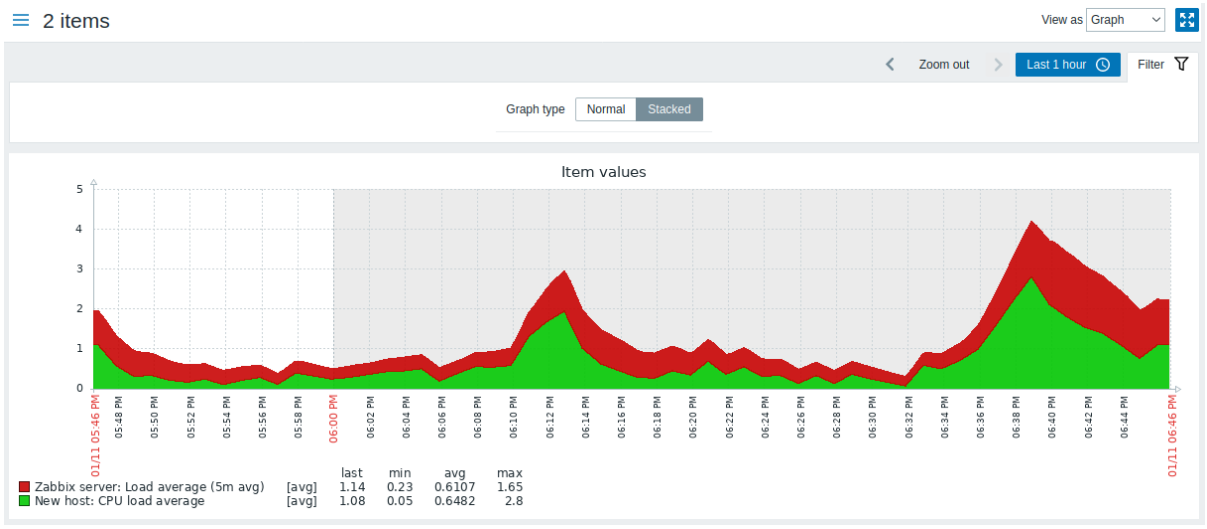


Jūsu grafiks tiek izveidots uzreiz:



Ņemiet vērā, ka, lai grafikā netiktu parādīts pārāk daudz līniju, katram vienumam tiek attēlota tikai vidējā vērtība (min/max vērtību līnijas netiek parādītas). Trigeri un trigeru informācija grafikā netiek parādīta.

Izveidotajā grafika logā ir pieejams laika perioda atlasītājs, kā arī iespēja pārslēgties no “parastā” līniju grafika uz saliktu grafiku (un atpakaļ).



4 Agregēšana grafikos

Pārskats

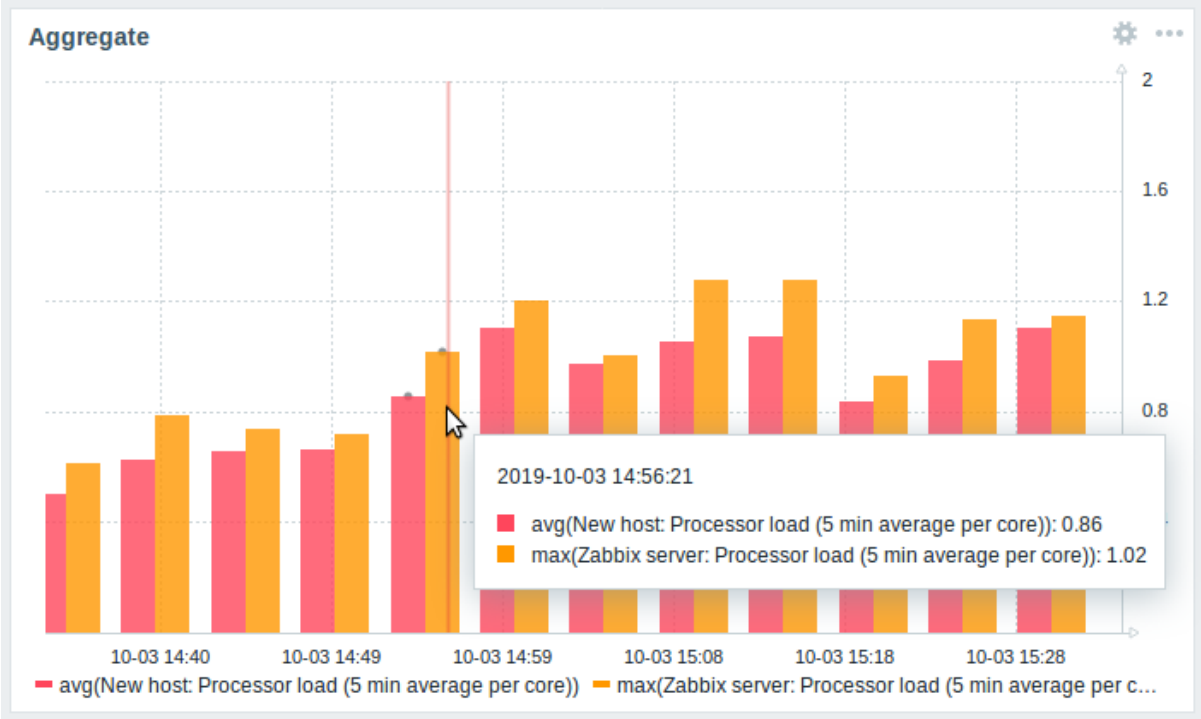
Apkopošanas funkcijas, kas pieejamas paneļa grafika un sektoru diagrammas logrīkos, ļauj attēlot apkopotu vērtību izvēlētajam intervālam (5 minūtes, stunda, diena), nevis visas vērtības.

Šajā sadaļā ir sniegta sīkāka informācija par apkopošanas iespējām grafika logrīkā.

Apkopošanas iespējas ir šādas:

- min
- max
- avg
- count
- sum
- first (pirmā attēlotā vērtība)
- last (pēdējā attēlotā vērtība)

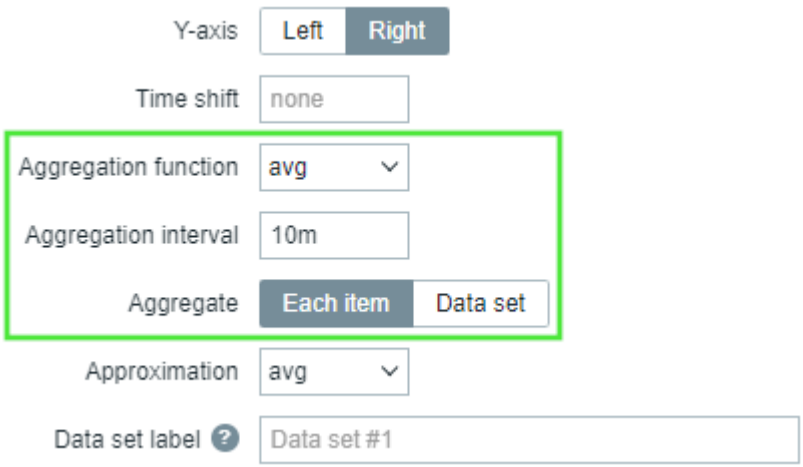
Visnoderīgākais datu apkopošanas pielietojums ir iespēja izveidot ērtus datu salīdzinājumus blakus izvietojumā noteiktam periodam:



Novietojot kursoru virs noteikta laika punkta grafikā, papildus vienumiem un to apkopotajām vērtībām tiek attēlots arī datums un laiks. Vienumi tiek attēloti iekavās ar izmantotās apkopošanas funkcijas prefiksu. Ja grafika logrīkam ir konfigurēta *Datu kopas etiķete*, etiķete tiek attēlota iekavās ar izmantotās apkopošanas funkcijas prefiksu. Ņemiet vērā, ka tas ir grafikā redzamā punkta datums un laiks, nevis faktisko vērtību datums un laiks.

Konfigurācija

Apkopošanas opcijas ir pieejamas datu kopas iestatījumos, konfigurējot grafika logrīku.



Jūs varat izvēlēties apkopošanas funkciju un laika intervālu. Tā kā datu kopa var ietvert vairākus vienumus, ir pieejama arī papildu opcija, kas ļauj rādīt apkopotos datus katram vienumam atsevišķi vai visiem datu kopas vienumiem kā vienu apkopotu vērtību.

Lietošanas gadījumi

Vidējais pieprasījumu skaits uz Nginx serveri

Skatiet vidējo pieprasījumu skaitu sekundē dienā uz Nginx serveri:

- pievienojiet datu kopai pieprasījumu skaita sekundē vienumu
- atlasiet apkopošanas funkciju `avg` un norādiet intervālu `1d`
- tiek attēlots joslu grafiks, kur katra josla attēlo vidējo pieprasījumu skaitu sekundē dienā

Minimālā nedēļas diska vieta starp klasteriem

Skatiet mazāko diska vietu starp klasteriem nedēļas griezumā.

- pievienojiet datu kopai: hosti `cluster*`, atslēga "Brīvā diska vieta uz /data"
- atlasiet apkopojošo funkciju `min` un norādiet intervālu `1w`
- tiek attēlots stabiņu grafiks, kur katrs stabiņš attēlo minimālo diska vietu nedēļā katram klastera /data sējumam

2 Tīkla kartes

Pārskats

Ja jums ir tīkls, par kuru jārūpējas, iespējams, vēlēsieties kaut kur iegūt savas infrastruktūras pārskatu. Šim nolūkam jūs varat izveidot kartes Zabbix vidē — tīklu kartes un jebkādas citas kartes pēc saviem ieskatiem.

Visi lietotāji var izveidot tīkla kartes. Kartes var būt publiskas (pieejamas visiem lietotājiem) vai privātas (pieejamas atlasītiem lietotājiem).

Turpiniet ar tīkla kartes konfigurēšanu.

1 Tīkla kartes konfigurēšana

Pārskats

Lai konfigurētu karti Zabbix, vispirms ir jāizveido karte, definējot tās vispārīgos parametrus, un pēc tam jāsāk aizpildīt pati karte ar elementiem un to saitēm.

Karti var aizpildīt ar elementiem, kas ir hosts, hostu grupa, trigeris, attēls vai cita karte.

Ikonas tiek izmantotas, lai attēlotu kartes elementus. Varat definēt informāciju, kas tiks parādīta kopā ar ikonām, un iestatīt, lai nesenās problēmas tiktu attēlotas īpašā veidā. Varat savienot ikonas un definēt informāciju, kas tiks parādīta uz saitēm.

Varat pievienot pielāgotus URL, kuriem var piekļūt, noklikšķinot uz ikonām. Tādējādi varat sasaistīt hosta ikonu ar hosta rekvizītiem vai kartes ikonu ar citu karti.

Problēmu skaits kartēs tiek parādīts tikai cēloņproblēmām.

Kartes tiek pārvaldītas sadaļā *Monitoring* > *Maps*, kur tās var konfigurēt, pārvaldīt un skatīt. Uzraudzības skatā varat noklikšķināt uz ikonām un izmantot saites uz dažiem skriptiem un URL.

Tīkla kartes ir balstītas uz vektorgrafiku (SVG).

Publiskās un privātās kartes

Visi Zabbix lietotāji (tostarp lietotāji bez administratora tiesībām) var izveidot tīkla kartes. Kartēm ir īpašnieks — lietotājs, kurš tās izveidojis. Kartes var būt publiskas vai privātas.

- *Publiskās* kartes ir redzamas visiem lietotājiem, taču, lai tās redzētu, lietotājam jābūt lasīšanas piekļuvei vismaz vienam kartes elementam. Publiskās kartes var rediģēt, ja lietotājam/lietotāju grupai ir lasīšanas-rakstīšanas atļaujas šai kartei un vismaz lasīšanas atļaujas visiem atbilstošās kartes elementiem, tostarp trigeriem saitēs.
- *Privātās* kartes ir redzamas tikai to īpašniekam un lietotājiem/lietotāju grupām, ar kurām karte ir kopīgota no īpašnieka puses. Parastie lietotāji (nevis Super admin) var kopīgot tikai ar grupām, kurām viņi pieder, un lietotājiem, kuri pieder šīm grupām. Administratora līmeņa lietotāji var redzēt privātās kartes neatkarīgi no tā, vai viņi ir īpašnieki vai ir iekļauti kopīgotāju lietotāju sarakstā. Privātās kartes var rediģēt kartes īpašnieks, kā arī gadījumā, ja lietotājam/lietotāju grupai ir lasīšanas-rakstīšanas atļaujas šai kartei un vismaz lasīšanas atļaujas visiem atbilstošās kartes elementiem, tostarp trigeriem saitēs.

Kartes elementi, kuriem lietotājam nav lasīšanas atļaujas, tiek attēloti ar pelēkotu ikonu, un visa teksta informācija par elementu ir paslēpta. Tomēr trigeru etiķete ir redzama pat tad, ja lietotājam nav atļaujas šim trigerim.

Lai pievienotu elementu kartei, lietotājam jābūt arī vismaz lasīšanas atļaujai šim elementam.

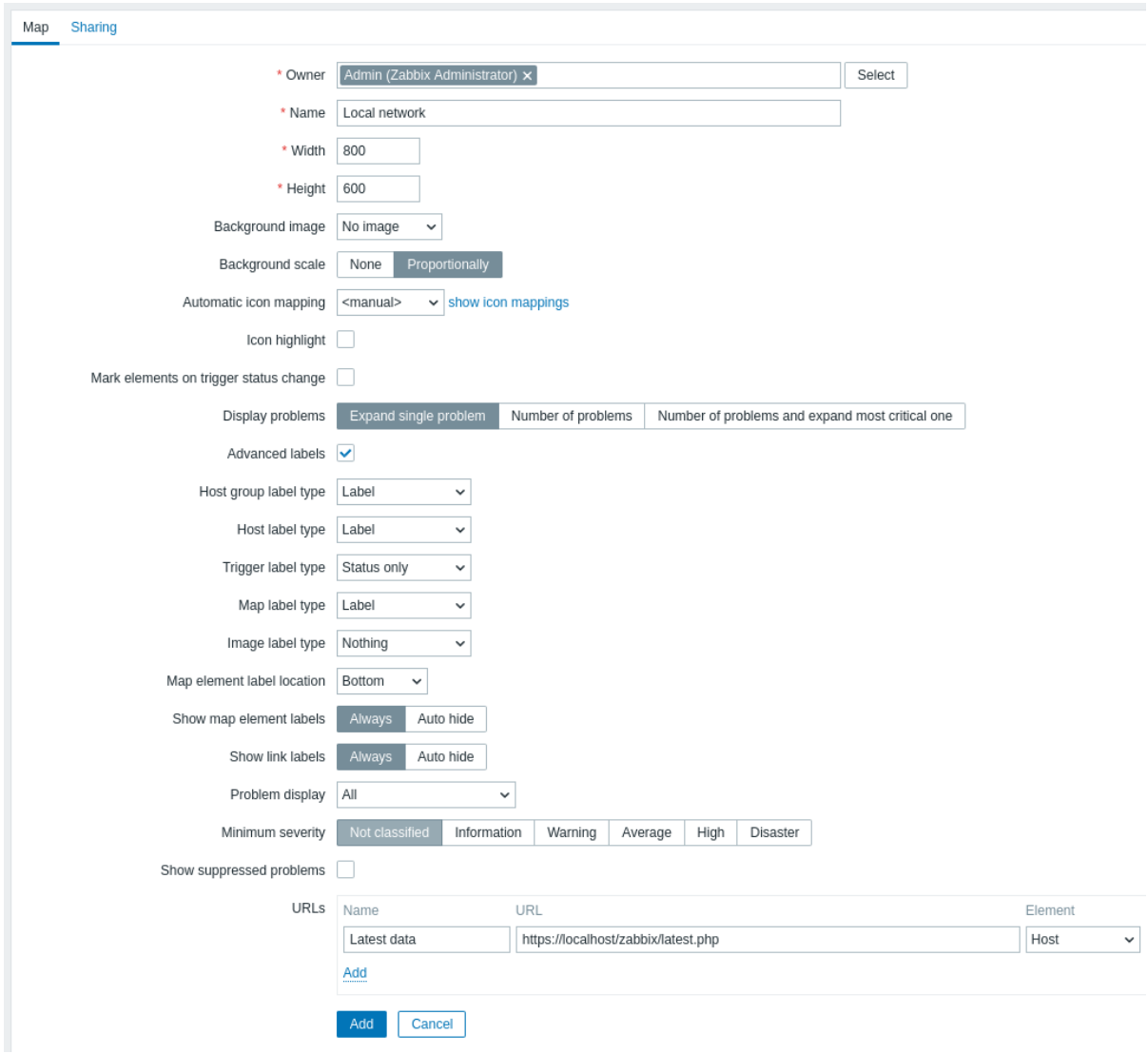
Kartes izveide

Lai izveidotu karti, rīkojieties šādi:

1. Dodieties uz *Monitoring* > *Maps*.
2. Atveriet skatu ar visām kartēm.
3. Noklikšķiniet uz *Create map*.

Varat arī izmantot pogu *Clone* esošas kartes konfigurācijas formā, lai izveidotu jaunu karti. Šai kartei būs visas esošās kartes īpašības, tostarp vispārīgie izkārtojuma atribūti, kā arī esošās kartes elementi.

Cilnē **Map** ir pieejami vispārīgie kartes atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Vispārīgie kartes atribūti:

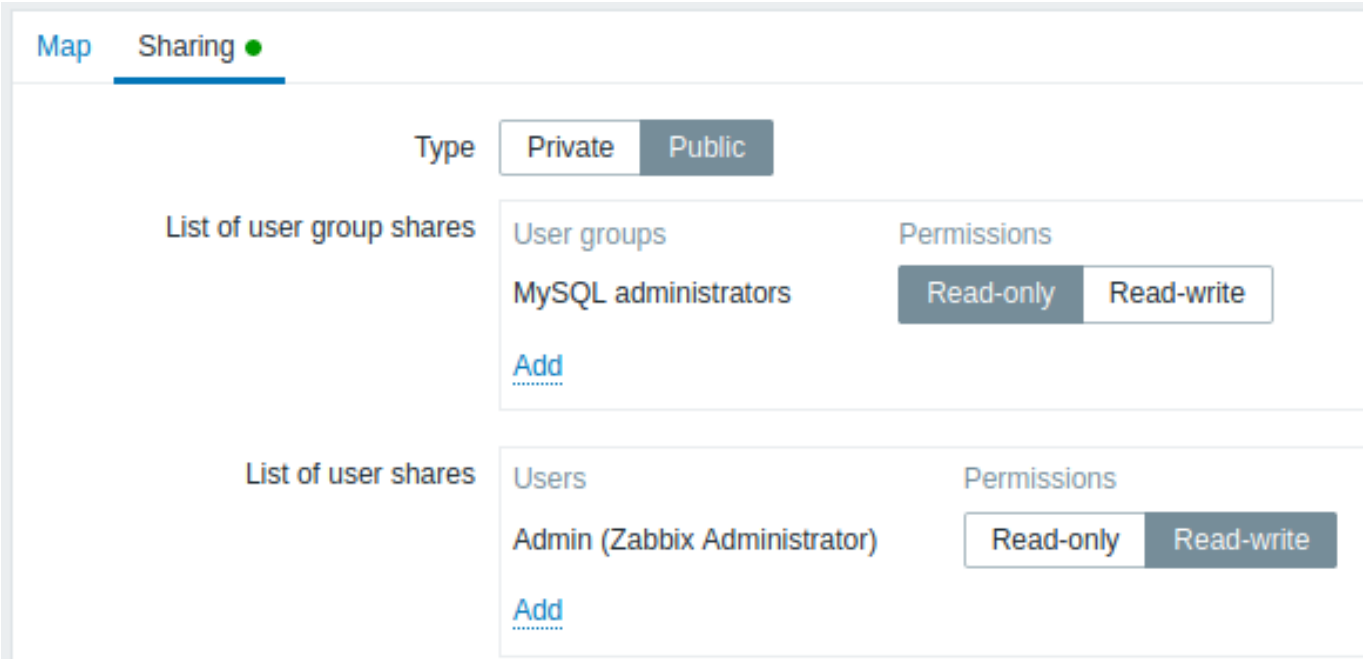
Parameter	Description
Owner	Kartes īpašnieka nosaukums.
Name	Unikāls kartes nosaukums.
Width	Kartes platums pikseļos.
Height	Kartes augstums pikseļos.
Background image	Fona attēla izmantošana: **No image** - bez fona attēla (balts fons) **Image** - atlasītais attēls tiks izmantots kā fona attēls. Varat izmantot ģeogrāfisko karti vai jebkuru citu attēlu, lai uzlabotu savu karti.

Parameter	Description
Background scale	Fona mērogošanas izmantošana: **None** - bez mērogošanas; **Proportionally** - aizpilda visu kartes fonu, bet saglabā attēla proporcijas (noklusējums).
Automatic icon mapping	Varat iestatīt automātisku ikonu kartēšanu, kas konfigurēta sadaļā *Administration* > *General* > *Icon mapping*. Ikonu kartēšana ļauj piesaistīt noteiktas ikonas noteiktiem hostu inventāra laukiem.
Icon highlight	Ja atzīmēsiet šo izvēles rūtiņu, kartes elementi tiks izcelti. Elementi ar aktīvu trigeri saņems apaļu fonu tādā pašā krāsā kā augstākās nopietnības trigeris. Turklāt ap apli tiks parādīta bieza zaļa līnija, ja visas problēmas ir apstiprinātas. Elementi ar statusu ”disabled” vai ”in maintenance” saņems attiecīgi pelēku un oranžu kvadrātveida fonu. Skatiet arī: Karšu skatīšana
Mark elements on trigger status change	Nesenās trigera stāvokļa izmaiņas (nesena problēma vai atrisinājums) tiks izceltas ar marķieriem (uz iekšu vērstiem sarkaniem trijstūriem) trijās elementa ikonas pusēs, kuras neaizņem etiķete. Marķieri tiek rādīti 30 minūtes.
Display problems	Izvēlieties, kā problēmas tiek parādītas kopā ar kartes elementu: **Expand single problem** - ja ir tikai viena problēma, tiek parādīts problēmas nosaukums. Pretējā gadījumā tiek parādīts kopējais problēmu skaits. **Number of problems** - tiek parādīts kopējais problēmu skaits **Number of problems and expand most critical one** - tiek parādīts kritiskākās problēmas nosaukums un kopējais problēmu skaits. ’Viskritiskākā’ tiek noteikta pēc problēmas nopietnības un, ja tā ir vienāda, pēc problēmas notikuma ID (vispirms tiek parādīts lielāks ID jeb vēlākā problēma). *Trigera kartes elementam* tas tiek noteikts pēc problēmas nopietnības un, ja tā ir vienāda, pēc trigera pozīcijas trigeru sarakstā. Ja vienam trigerim ir vairākas problēmas, tiks parādīta jaunākā.
Advanced labels	Ja atzīmēsiet šo izvēles rūtiņu, varēsiet definēt atsevišķus etiķešu tipus atsevišķiem elementu tipiem.
Host group label type	Kartes elementam izmantotais etiķetes tips: **Label** - kartes elementa etiķete **IP address** - IP adrese **Element name** - elementa nosaukums (piemēram, hosta nosaukums) **Status only** - tikai statuss (OK vai PROBLEM) **Nothing** - etiķetes netiek rādītas
Host label type *Trigger label type* *Map label type* *Image label type*	
Map element label location	Etiķetes novietojums attiecībā pret kartes elementu: **Bottom** - zem kartes elementa **Left** - pa kreisi **Right** - pa labi **Top** - virs kartes elementa
Show map element labels	Izvēlieties, kā rādīt kartes elementu etiķetes: **Always** - vienmēr rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja uz tās nav novietots kursors un tā nav atlasīta.
Show link labels	Izvēlieties, kā rādīt saišu etiķetes: **Always** - vienmēr rādīt (noklusējums); **Auto hide** - paslēpt etiķeti, ja uz tās nav novietots kursors un tā nav atlasīta.
Problem display	Parādīt problēmu skaitu kā: **All** - tiks parādīts pilns problēmu skaits **Separated** - neapstiprināto problēmu skaits tiks parādīts atsevišķi no kopējā problēmu skaita **Unacknowledged only** - tiks parādīts tikai neapstiprināto problēmu skaits
Minimum severity	Problēmas, kas ir zem izvēlētā minimālā nopietnības līmeņa, kartē netiks rādītas. Piemēram, ja ir izvēlēts *Warning*, izmaiņas ar *Information* un *Not classified* līmeņa trigeriem kartē netiks atspoguļotas.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai rādītu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas dēļ.

Parameter	Description
URLs	Šeit var definēt URL katram elementa tipam, kur: **Name** - URL etiķete; **URL** - URL (līdz 2048 rakstzīmēm); **Element** - elementa tips (piemēram, *Host*, *Host group* u.c.). Tie tiks parādīti kā saites, kad lietotājs kartes skatīšanas režīmā noklikšķinās uz elementa. Kartes URL nosaukumos un vērtībās var izmantot makrosus. Pilnu sarakstu skatiet sadaļā supported macros un meklējiet ’map URL names and values’.

Kopīgošana

Cilnē **Kopīgošana** ir ietverts kartes tips, kā arī kopīgošanas opcijas (lietotāju grupas, lietotāji) privātām kartēm:



Parametrs	Apraksts
Tips	Atlasiet kartes tipu: **Privāta** - karte ir redzama tikai atlasītajām lietotāju grupām un lietotājiem **Publiska** - karte ir redzama visiem
Lietotāju grupu kopīgojumu saraksts	Atlasiet lietotāju grupas, kurām karte ir pieejama. Varat atļaut tikai lasīšanas vai lasīšanas-rakstīšanas piekļuvi.
Lietotāju kopīgojumu saraksts	Atlasiet lietotājus, kuriem karte ir pieejama. Varat atļaut tikai lasīšanas vai lasīšanas-rakstīšanas piekļuvi.

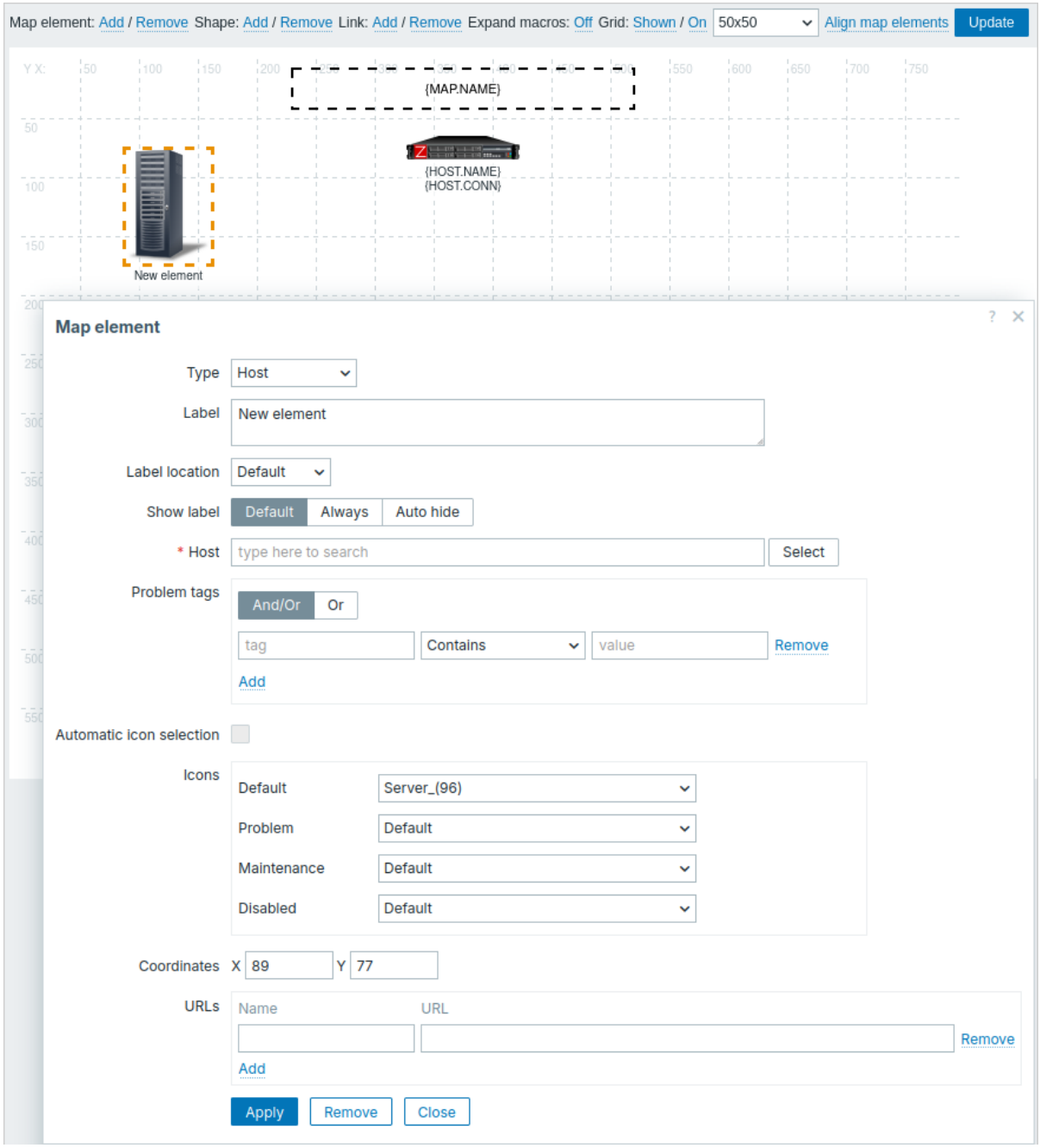
Kad noklikšķināt uz *Pievienot*, lai saglabātu šo karti, esat izveidojis tukšu karti ar nosaukumu, izmēriem un noteiktiem iestatījumiem. Tagad jums jāpievieno daži elementi. Lai to izdarītu, kartes sarakstā noklikšķiniet uz *Rediģēt*, lai atvērtu rediģējamo apgabalu.

Elementu pievienošana

Lai pievienotu elementu, noklikšķiniet uz *Add* blakus `Map element`. Jaunais elements parādīsies kartes augšējā kreisajā stūrī. Pārvelciet to uz jebkuru vēlamo vietu.

Ņemiet vērā, ka, ja režģa opcija ir iestatīta uz ”On”, elementi vienmēr tiks līdzināti režģim (nolaižamajā sarakstā varat izvēlēties dažādus režģa izmērus, kā arī paslēpt/parādīt režģi). Ja vēlaties novietot elementus jebkur bez līdzināšanas, pārslēdziet opciju uz ”Off”. (Nejauši izvietotus elementus vēlāk varat līdzināt režģim, noklikšķinot uz *Align map elements*.)

Tagad, kad daži elementi jau ir izvietoti, iespējams, vēlēsieties sākt tos atšķirt, piešķirot nosaukumus utt. Noklikšķinot uz elementa, tiek parādīta forma, kurā varat iestatīt elementa tipu, piešķirt nosaukumu, izvēlēties citu ikonu utt.



Kartes elementa atribūti:

Parameter	Description
Type	Elementa tips: **Hosts** - ikona, kas attēlo visu izvēlētā hosta trigeru statusu **Map** - ikona, kas attēlo visu kartes elementu statusu **Trigger** - ikona, kas attēlo viena vai vairāku trigeru statusu **Host group** - ikona, kas attēlo visu izvēlētajai grupai piederošo hostu visu trigeru statusu **Image** - ikona, kas nav saistīta ne ar vienu resursu
Label	Ikonas etiķete, jebkura virkne. Var izmantot makrosus un vairākrindu virknes. Šajā laukā tiek atbalstīti izteiksmju macros, bet tikai ar `avg`, `last`, `min` un `max` funkcijām, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`). Pilnu atbalstīto makrosu sarakstu skatiet sadaļā supported macros un meklējiet 'map element labels'.
Label location	Etiķetes novietojums attiecībā pret ikonu: **Default** - kartes noklusējuma etiķetes novietojums; **Bottom** - zem ikonas; **Left** - pa kreisi; **Right** - pa labi; **Top** - virs ikonas.

Parameter	Description
Show label	Izvēlieties, kā attēlot elementa etiķeti: **Default** - izmantot kartes konfigurācijas iestatījumu *Show map element labels* (noklusējums); **Always** - vienmēr attēlot (noklusējums); **Auto hide** - paslēpt etiķeti, ja uz tās nav novietots kursors un tā nav atlasīta.
Host	Izvēlieties hostu; alternatīvi ievadiet hosta nosaukumu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks parādīts atbilstošu hostu nolaižamais saraksts. Ritiniet uz leju, lai izvēlētos vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Host*.
Map	Izvēlieties karti; alternatīvi ievadiet kartes nosaukumu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt kartes nosaukumu, tiks parādīts atbilstošu karšu nolaižamais saraksts. Ritiniet uz leju, lai izvēlētos vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Map*.
Triggers	Izvēlieties vienu vai vairākus trigerus zemāk esošajā laukā *New triggers* un noklikšķiniet uz *Add*. Alternatīvi sāciet rakstīt trigera nosaukumu un izvēlieties to no atbilstošo trigeru nolaižamā saraksta. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Trigger*. Izvēlēto trigeru secību var mainīt, bet tikai vienas un tās pašas trigeru nopietnības ietvaros. Vairāku trigeru atlase ietekmē arī {HOST.*} makrosu atrisināšanu gan rediģēšanas, gan skatīšanas režīmā. Rediģēšanas režīmā pirmie attēlotie {HOST.*} makrosi tiks atrisināti atkarībā no pirmā trigera sarakstā (pamatojoties uz trigera nopietnību). Skatīšanas režīms ir atkarīgs no parametra Display problems kartes vispārīgajos atribūtos: - Ja ir izvēlēts režīms *Expand single problem*, pirmie attēlotie {HOST.*} makrosi tiks atrisināti atkarībā no pēdējā konstatētā problēmas trigera (neatkarīgi no nopietnības) vai pirmā trigera sarakstā (ja problēma nav konstatēta). - Ja ir izvēlēts režīms *Number of problems and expand most critical one*, pirmie attēlotie {HOST.*} makrosi tiks atrisināti atkarībā no trigera nopietnības.
Host group	Izvēlieties hostu grupu; alternatīvi ievadiet hostu grupu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hostu grupas nosaukumu, tiks parādīts atbilstošu hostu grupu nolaižamais saraksts. Ritiniet uz leju, lai izvēlētos vienu. Noklikšķiniet uz noņemšanas (x) ikonas, lai noņemtu atlasi. Šis lauks ir pieejams tikai elementiem ar tipu *Host group*.
Problem tags	Norādiet birkas, lai ierobežotu logrīkā attēloto problēmu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Does not exist** - izslēgt norādītos birku nosaukumus **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina tipi: **And/Or** - visiem nosacījumiem jābūt izpildītiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Or nosacījuma **Or** - pietiek, ja ir izpildīts viens nosacījums Šis lauks ir pieejams tikai elementiem ar tipu *Host* un *Host group*.
Automatic icon selection	Ja izvēles rūtiņa ir atzīmēta, ikonu kartēšana tiks izmantota, lai noteiktu, kuru ikonu attēlot.
Icons	Varat izvēlēties attēlot dažādas ikonas elementam šādos gadījumos: *Default*, *Problem*, *Maintenance* un *Disabled*. Šis lauks ir pieejams tikai elementiem ar tipu *Host*.
Coordinates	Kartes elementa **X** un **Y** koordinātas.

Parameter	Description
URLs	Šeit elementam var iestatīt elementam specifiskus URL, kur: **Name** - URL etiķete; **URL** - URL (līdz 2048 rakstzīmēm). Tie tiks attēloti kā saites, kad lietotājs kartes skatīšanas režīmā noklikšķinās uz elementa. Ja elementam ir savi URL un tā tipam ir definēti kartes līmeņa URL, tie tiks apvienoti vienā un tajā pašā izvēlnē. Kartes elementu nosaukumos un vērtībās var izmantot makrosus. Pilnu sarakstu skatiet sadaļā supported macros un meklējiet 'map URL names and values'.

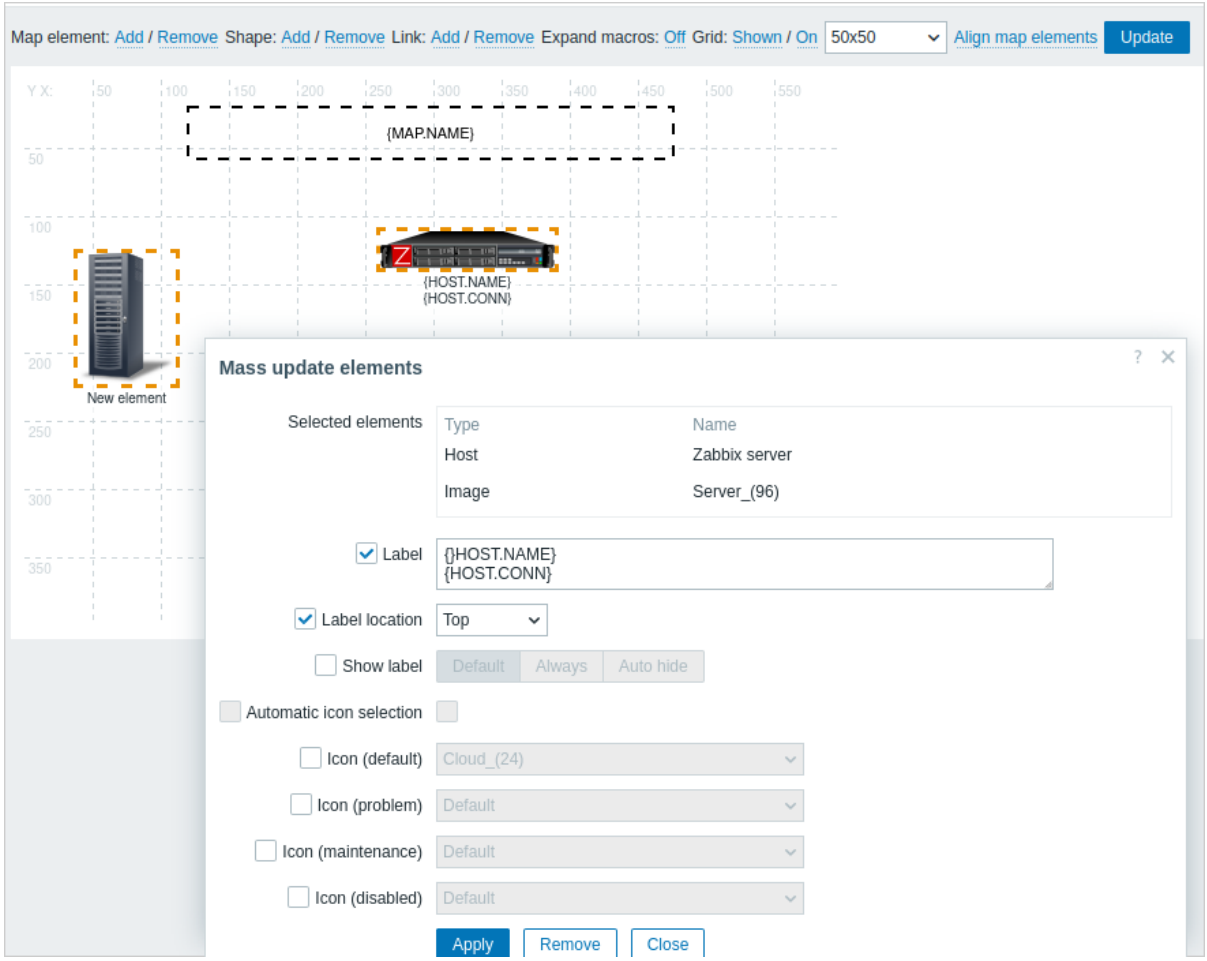
Attention:
Pievienotie elementi netiek saglabāti automātiski. Ja pametīsiet šo lapu, visas izmaiņas var tikt zaudētas.
Tāpēc ieteicams noklikšķināt uz pogas **Update** augšējā labajā stūrī. Pēc noklikšķināšanas izmaiņas tiek saglabātas neatkarīgi no tā, ko izvēlaties nākamajā uznirstošajā logā.
Izvēlētās režģa opcijas arī tiek saglabātas kopā ar katru karti.

Elementu atlasīšana

Lai atlasītu elementus, atlasiet vienu elementu un pēc tam turiet nospiestu *Ctrl*, lai atlasītu pārējos.

Varat arī atlasīt vairākus elementus, velkot taisnstūri rediģējamajā apgabalā un atlasot visus tajā esošos elementus.

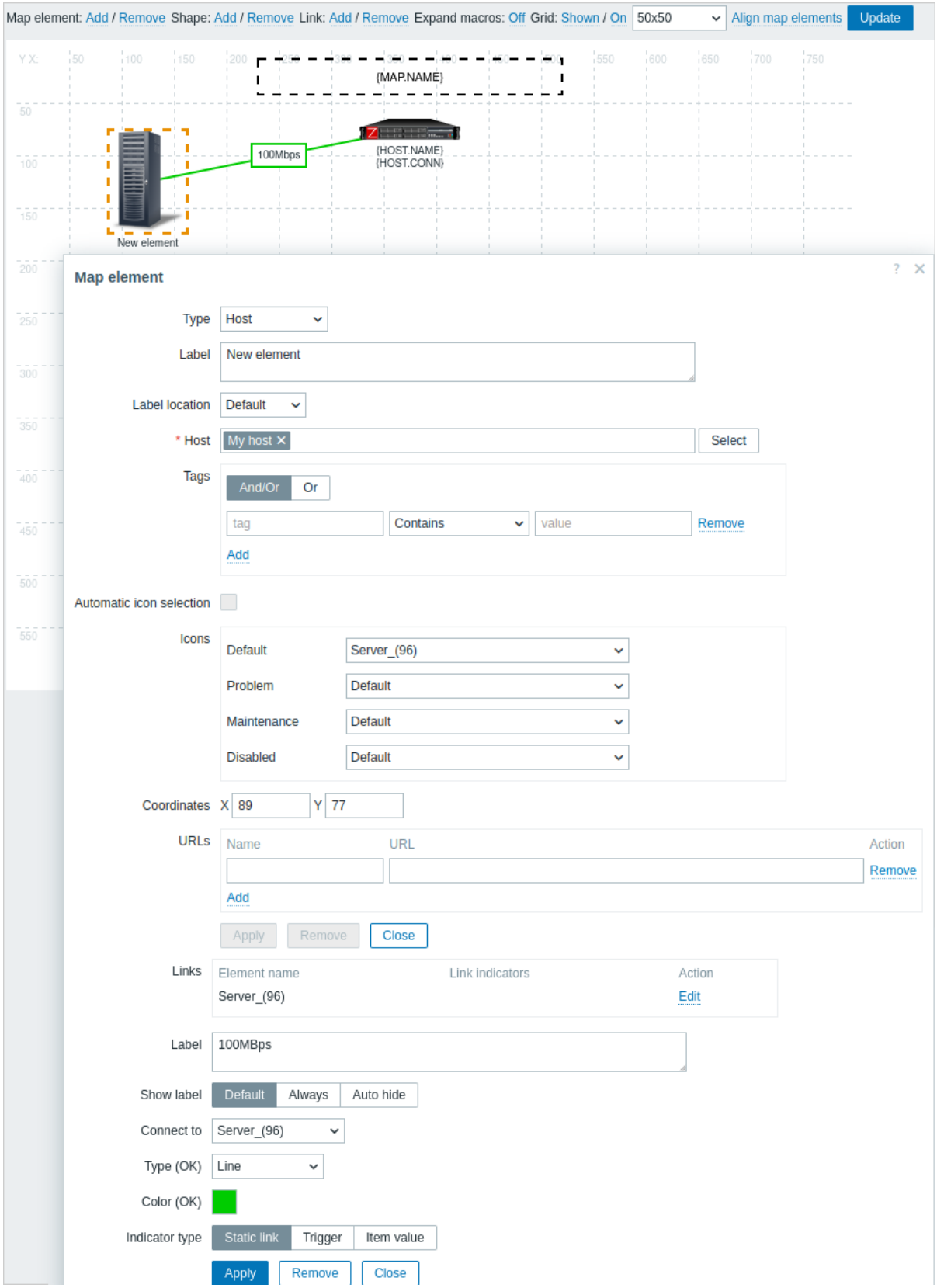
Kad atlasāt vairāk nekā vienu elementu, elementa īpašību forma pārslēdzas uz masveida atjaunināšanas režīmu, lai jūs varētu vienlaikus mainīt atlasīto elementu atribūtus. Lai to izdarītu, atzīmējiet atribūtu, izmantojot izvēles rūtiņu, un ievadiet tam jaunu vērtību. Šeit varat izmantot makrosus (piemēram, {HOST.NAME} elementa etiķetei).



Elementu sasaistīšana

Kad kartē esat izvietojis dažus elementus, ir laiks sākt tos sasaistīt. Lai sasaistītu divus elementus, vispirms tie ir jāatlasa. Kad elementi ir atlasīti, noklikšķiniet uz *Add* blakus Link.

Kad saite ir izveidota, viena elementa forma tagad satur papildu sadaļu *Links*. Noklikšķiniet uz *Edit*, lai rediģētu saites atribūtus.



Saites atribūti:

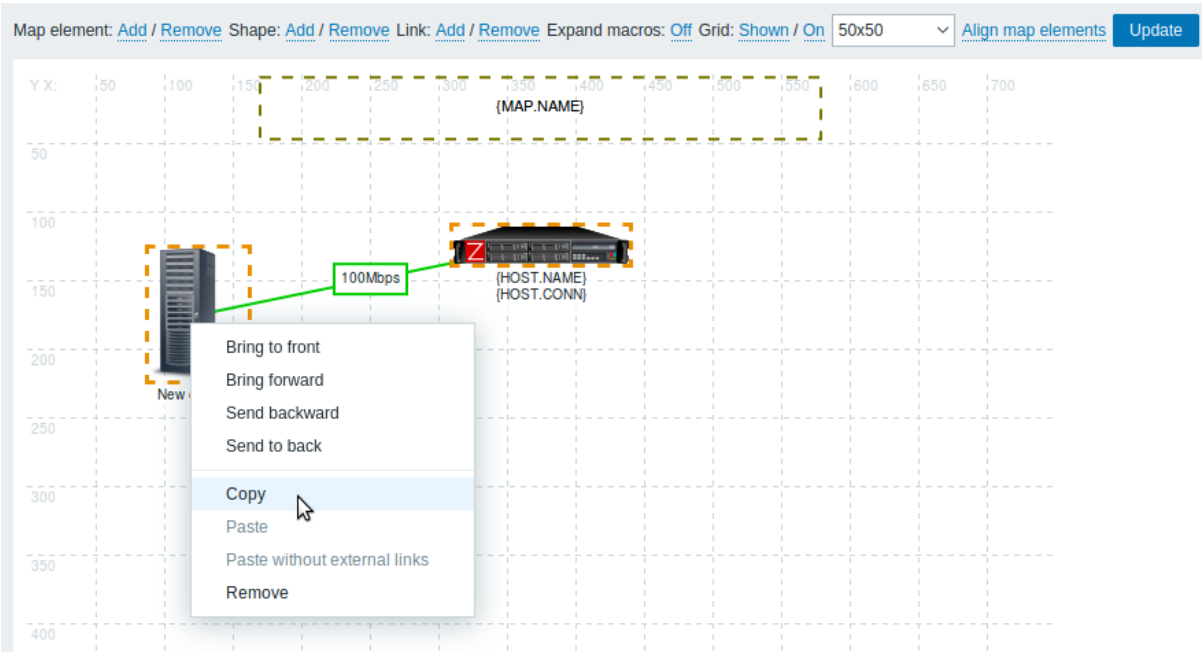
Parametrs	Apraksts
Label	Etiķete, kas tiks attēlota virs saites. Šajā laukā tiek atbalstītas izteiksmes makrosi, bet tikai ar `avg`, `last`, `min` un `max` funkcijām, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`).

Parametrs	Apraksts
Show label	Izvēlieties, kā attēlot saites etiķeti: **Default** - izmantot *Show map element labels* iestatījumus no kartes konfigurācijas (noklusējums); **Always** - vienmēr attēlot (noklusējums); **Auto hide** - paslēpt etiķeti, ja kursors nav virs tās vai tā nav atlasīta.
Connect to	Elements, ar kuru saite tiek savienota.
Type (OK)	Noklusējuma saites stils. Izvēlieties stilu: *Line*, *Bold line*, *Dot* vai *Dashed line*.
Color (OK)	Noklusējuma saites krāsa. Noklikšķiniet uz krāsas parauga, lai izvēlētos citu krāsu.
Indicator type	Izvēlieties saites indikatora tipu: **Static link** - bez indikatoriem; **Trigger** - atļaut uz trigeriem balstītus saites indikatorus; **Item value** - atļaut uz vienuma vērtību balstītus saites indikatorus.
Item	Izvēlieties vienums. Šī vienuma vērtība ietekmēs saites stilu. Šis lauks ir pieejams, ja laukam *Indicator type* ir atlasīts *Item value*.
Indicators	Saites indikatoru saraksts. Šis lauks ir pieejams, ja *Indicator type* ir iestatīts uz *Trigger* vai *Item value*. Ja ir iestatīts *Trigger*, trigerus var piešķirt kā saites indikatorus. Kad trigeris ir *Problem* stāvoklī, tā stils (izvēlētā krāsa un līnijas tips) tiks piemērots saitei. Ja ir iestatīts *Item value* un laukā *Item* ir atlasīts vienums, vienuma vērtības var piešķirt kā saites indikatorus. Kad vienuma vērtība sasniedz norādīto slieksni (skaitliskam vienumam) vai atbilst paraugam (teksta vienumam), tā stils (izvēlētā krāsa un līnijas tips) tiek piemērots saitei.

Elementu pārvietošana un kopēšana/ielīmēšana

Vairākus atlasītus elementus var **pārvietot** uz citu vietu kartē, noklikšķinot uz viena no atlasītajiem elementiem, turot nospiestu peles pogu un pārvietojot kursoru uz vēlamo vietu.

Vienu vai vairākus elementus var **kopēt**, atlasot elementus, pēc tam noklikšķinot uz atlasīta elementa ar peles labo pogu un izvēloties izvēlnē *Copy*.

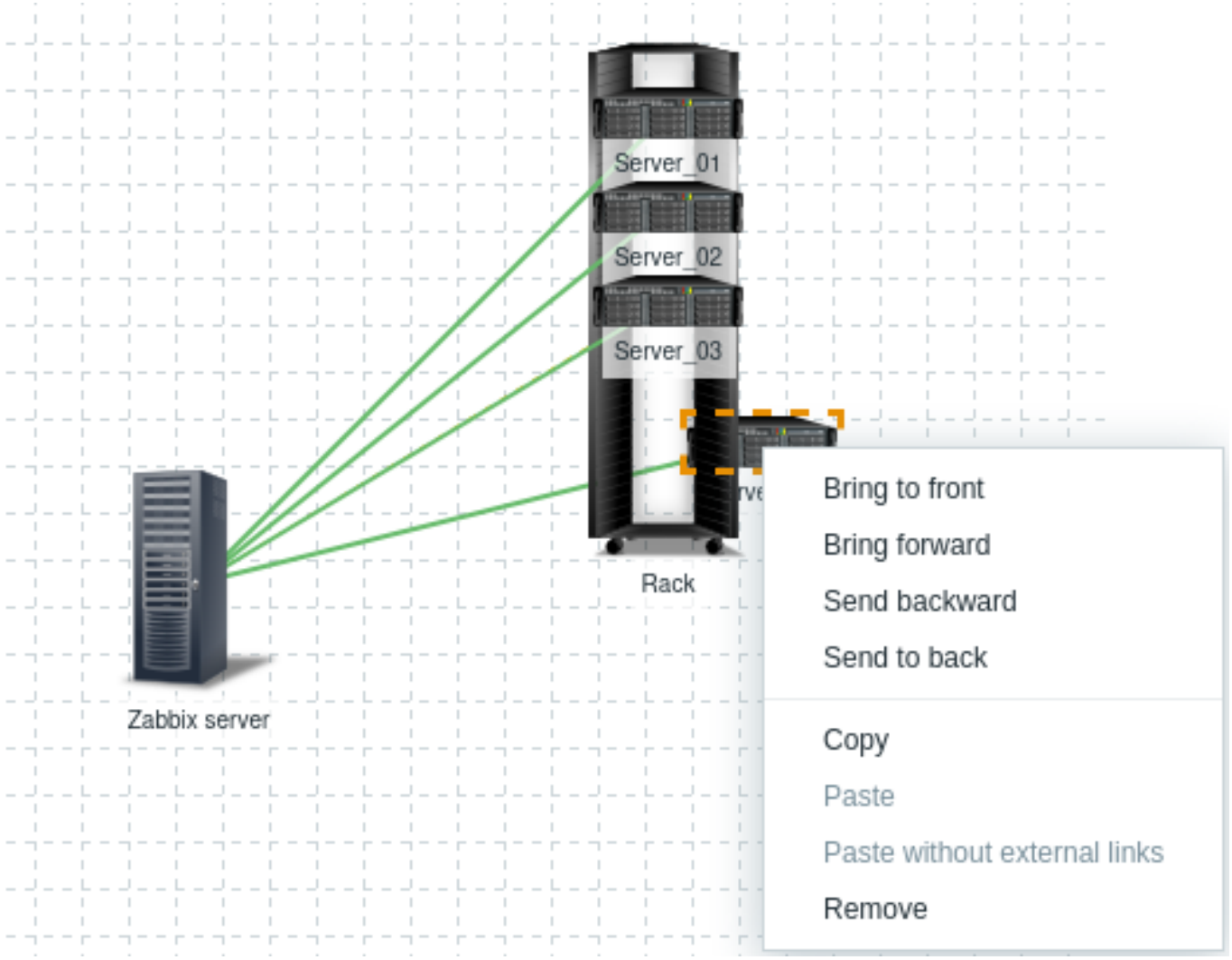


Lai ielīmētu elementus, noklikšķiniet kartes apgabalā ar peles labo pogu un izvēlieties izvēlnē *Paste*. Opcija *Paste without external links* ielīmēs elementus, saglabājot tikai tās saites, kas ir starp atlasītajiem elementiem.

Kopēšana un ielīmēšana darbojas vienā un tajā pašā pārlūkprogrammas logā. Tastatūras saīsnes netiek atbalstītas.

Elementu secības noteikšana

Lai pārvietotu vienu elementu priekšplānā attiecībā pret citu (vai otrādi), noklikšķiniet uz elementa ar peles labo pogu un atlasiet *Pārvietot uz priekšu/Pārvietot priekšplānā* vai *Pārvietot atpakaļ/Pārvietot fonā*.

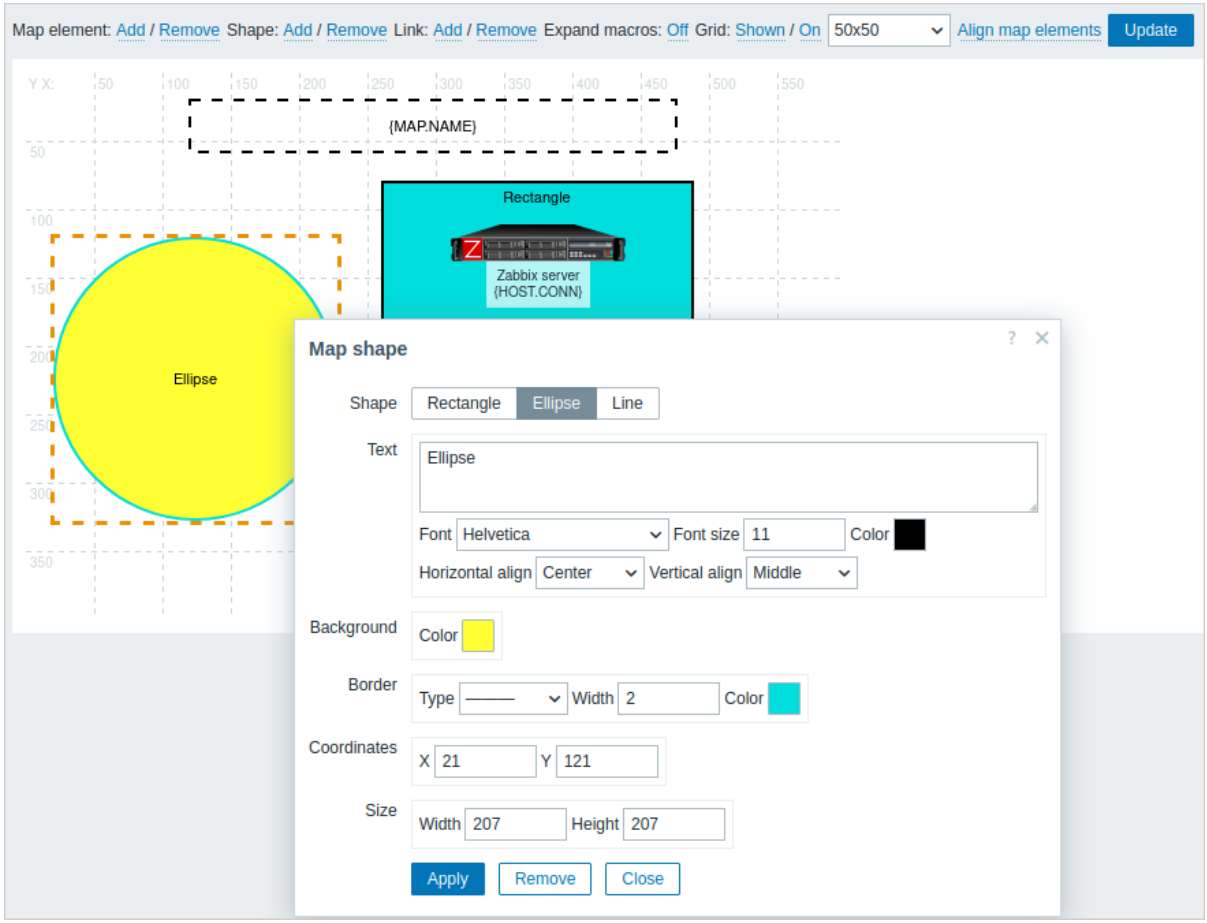


Formu pievienošana

Papildus kartes elementiem ir iespējams pievienot arī dažas formas. Formas nav kartes elementi; tās ir tikai vizuāls attēlojums. Piemēram, taisnstūra formu var izmantot kā fonu, lai grupētu vairākus hostus. Var pievienot taisnstūra un elipses formas.

Lai pievienotu formu, noklikšķiniet uz *Add* blakus laukam Shape. Jaunā forma parādīsies kartes augšējā kreisajā stūrī. Velciet un nometiet to jebkurā vēlamajā vietā.

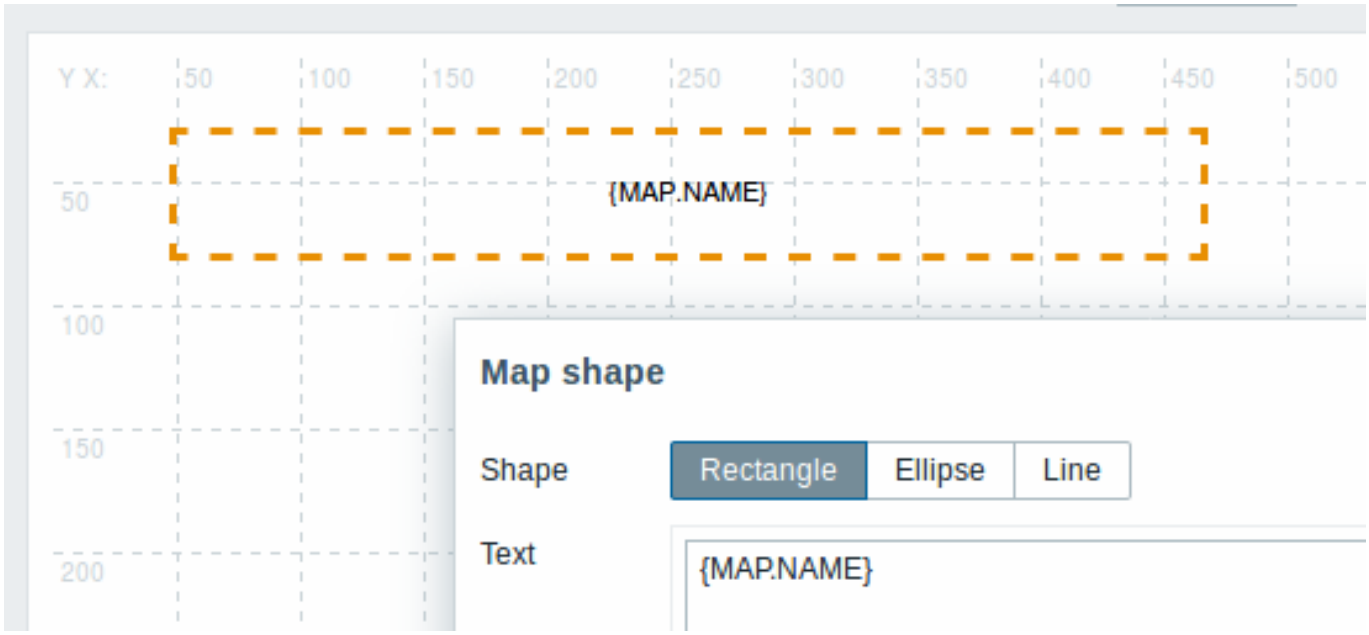
Jauna forma tiek pievienota ar noklusējuma krāsām. Noklikšķinot uz formas, tiek parādīta veidlapa, kurā varat pielāgot formas izskatu, pievienot tekstu utt.



Lai atlasītu formas, atlasiet vienu un pēc tam turiet nospiestu *Ctrl*, lai atlasītu pārējās. Ja ir atlasītas vairākas formas, kopīgās īpašības var masveidā atjaunināt līdzīgi kā elementiem.

Formās var pievienot tekstu. Tekstā tiek atbalstīti izteiksmju makrosi, taču tikai ar funkcijām `avg`, `last`, `min` un `max`, izmantojot laiku kā parametru (piemēram, `{?avg(/host/key,1h)}`).

Lai attēlotu tikai tekstu, formu var padarīt neredzamu, noņemot formas apmali (laukā *Border* atlasiet 'None'). Piemēram, pievērsiet uzmanību tam, ka iepriekš redzamajā ekrānuzņēmumā redzamais makross {MAP.NAME} patiesībā ir taisnstūra forma ar tekstu, ko var redzēt, noklikšķinot uz makrosa:



Skatot karti, {MAP.NAME} tiek aizstāts ar konfigurēto kartes nosaukumu.

Ja tekstā tiek izmantotas hipersaites, tās kļūst klikšķināmas, skatot karti.

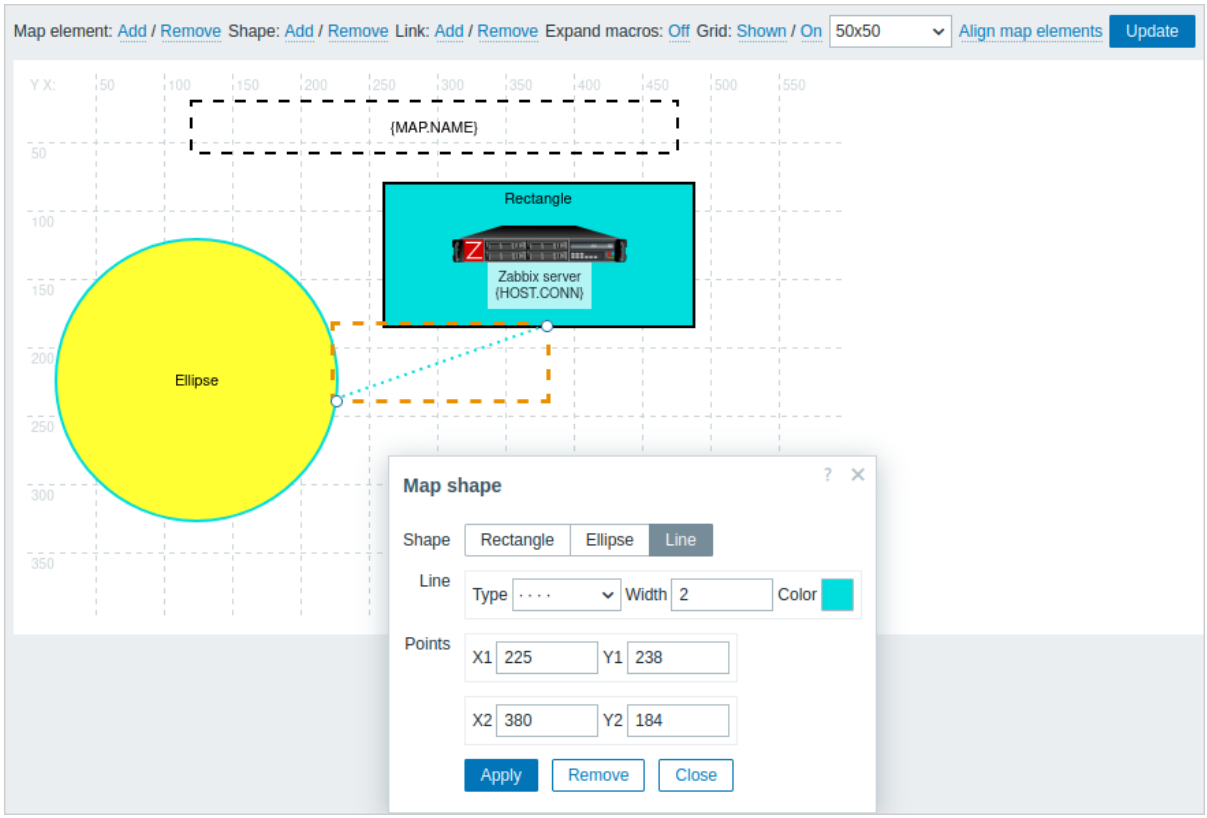
Teksta rindu aplaušana formās vienmēr ir ieslēgta. Tomēr elipsē rindas tiek aplauztas tā, it kā elipse būtu taisnstūris. Vārdu aplaušana nav ieviesta, tāpēc gari vārdi (vārdi, kas neietilpst formā) netiek aplauzti, bet tiek maskēti (kartes rediģēšanas lapā) vai

apgriezti (citās lapās ar kartēm).

Līniju pievienošana

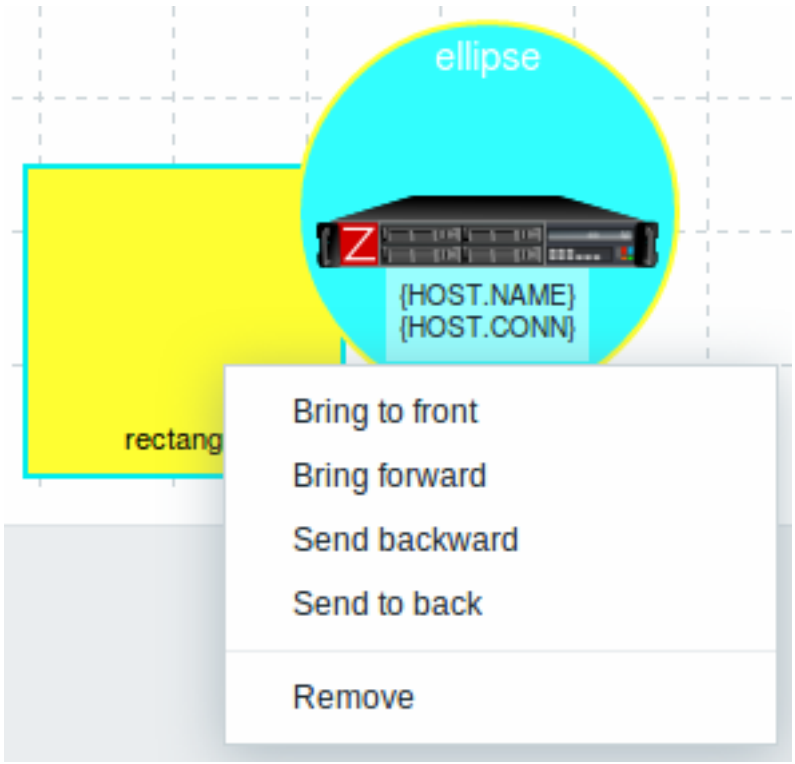
Papildus figūrām ir iespējams pievienot arī līnijas. Līnijas var izmantot, lai kartē savienotu elementus vai figūras.

Lai pievienotu līniju, noklikšķiniet uz *Pievienot* blakus Shape. Kartes augšējā kreisajā stūrī parādīsies jauna figūra. Atlasiet to un rediģēšanas formā noklikšķiniet uz *Line*, lai pārveidotu figūru par līniju. Pēc tam pielāgojiet līnijas īpašības, piemēram, līnijas tipu, platumu, krāsu utt.



Formu un līniju secības maiņa

Lai vienu formu novietotu otras priekšā (vai otrādi), noklikšķiniet uz formas ar peles labo pogu, lai atvērtu kartes formas izvēlni.

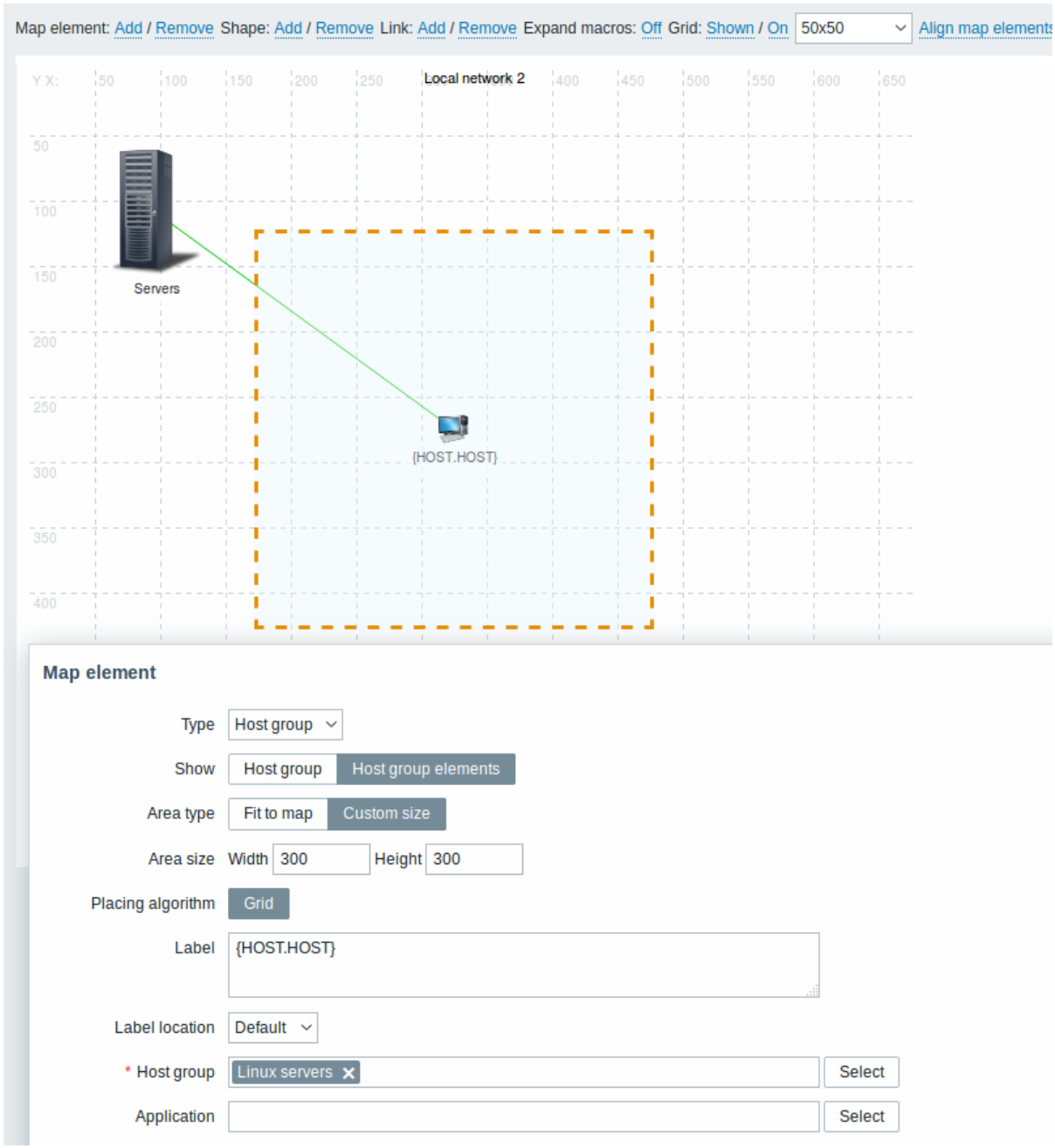


2 Hostu grupu elementi

Pārskats

Šajā sadaļā ir paskaidrots, kā pievienot *Host group* tipa elementu, konfigurējot tīkla karti.

Konfigurācija



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

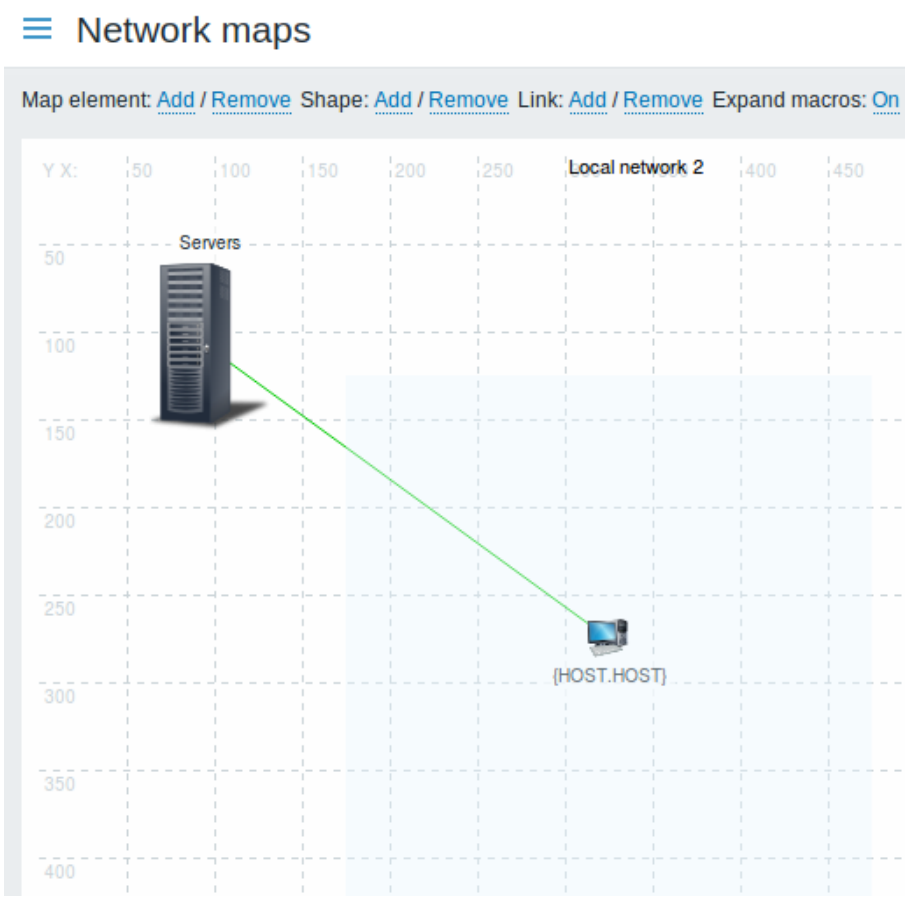
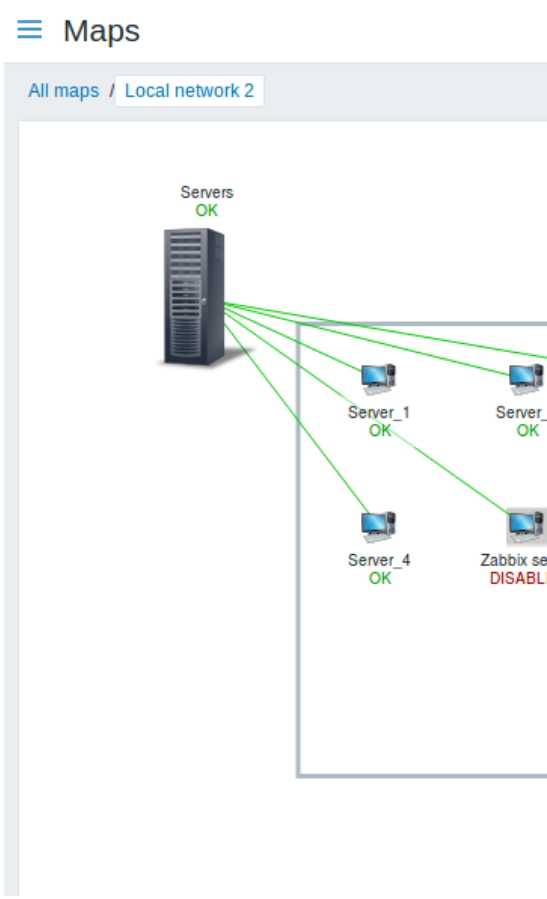
Šī tabula sastāv no parametriem, kas ir raksturīgi elementa tipam *Host group*:

Parametrs	Apraksts
Type	Atlasiet elementa tipu: **Host group** - ikona, kas attēlo visu atlasītajai grupai piederošo visu hostu visu trigeru statusu.
Show	Rādīšanas opcijas: **Host group** - izvēloties šo opciju, tiks parādīta viena ikona, kas attēlo atbilstošo informāciju par noteiktu hostu grupu. **Host group elements** - izvēloties šo opciju, tiks parādītas vairākas ikonas, kas attēlo atbilstošo informāciju par katru atsevišķo noteiktās hostu grupas elementu (hostu).

Parametrs	Apraksts
Area type	Šis iestatījums ir pieejams, ja *Show* ir iestatīts uz *Host group elements*: **Fit to map** - visi hostu grupas elementi tiek vienmērīgi izvietoti kartē. **Custom size** - manuāls kartes apgabala iestatījums visu hostu grupas elementu attēlošanai.
Area size	Šis iestatījums ir pieejams, ja *Area type* ir iestatīts uz *Custom size*: **Width** - vērtība pikseļos, lai norādītu kartes apgabala platumu. **Height** - vērtība pikseļos, lai norādītu kartes apgabala augstumu.
Placing algorithm	**Grid** - vienīgā pieejamā opcija visu hostu grupas elementu attēlošanai.
Label	Ikonas etiķete, jebkura virkne. Etiķetēs var izmantot makrosus un vairākrindu virknes. Ja kartes elementa tips ir *Host group*, noteiktu makrosu norādīšana ietekmē kartes skatu, attēlojot atbilstošo informāciju par katru atsevišķo hostu. Piemēram, ja tiek izmantots makross {HOST.IP}, kartes rediģēšanas skatā tiks attēlots tikai pats makross {HOST.IP}, savukārt kartes skatā tiks iekļauta un attēlota katra hosta unikālā IP adrese.

Hostu grupas elementu skatīšana

Šī opcija ir pieejama, ja ir izvēlēta *Host group elements* rādīšanas opcija. Izvēloties *Host group elements* kā *show* opciju, sākumā redzēsiet tikai vienu hostu grupas ikonu. Tomēr, kad saglabāsiet karti un pēc tam atvērsiet kartes skatu, redzēsiet, ka kartē ir iekļauti visi konkrētās hostu grupas elementi (hosti):

Kartes rediģēšanas skats	Kartes skats
	

Pievērsiet uzmanību tam, kā tiek izmantots {HOST.HOST} makross. Kartes rediģēšanas skatā makrosa nosaukums netiek atrisināts, savukārt kartes skatā tiek parādīti visu hostu unikālie nosaukumi.

3 Saites indikatori

Pārskats

Jūs varat piešķirt indikatorus saitei starp elementiem tīkla kartē.

Indikatori var būt balstīti uz trigeriem vai vienumu vērtībām. Ir iespējams attēlot atšķirīgu saites stilu un krāsu:

- kad trigeri pāriet problēmas stāvoklī;
- kad vienuma vērtība:
  - sasniedz slieksni (skaitliskiem vienumiem);
  - atbilst regulārajai izteiksmei (teksta vienumiem).

Kad konfigurējat saiti, jūs iestatāt noklusējuma saites tipu un krāsu. Piešķirot saitei indikatorus, kļūst iespējams padarīt saites stilu un krāsu atkarīgu no trigeru stāvokļa vai vienuma vērtības.

Piemēram, ja kāds no piešķirtajiem trigeriem pāriet problēmas stāvoklī, saites stils un krāsa mainīsies, lai to atspoguļotu. Tātad, iespējams, jūsu noklusējuma saite bija zaļa līnija. Tagad, kad trigeris ir problēmas stāvoklī, jūsu saite var kļūt par biezu sarkanu līniju (ja esat to tā definējis).

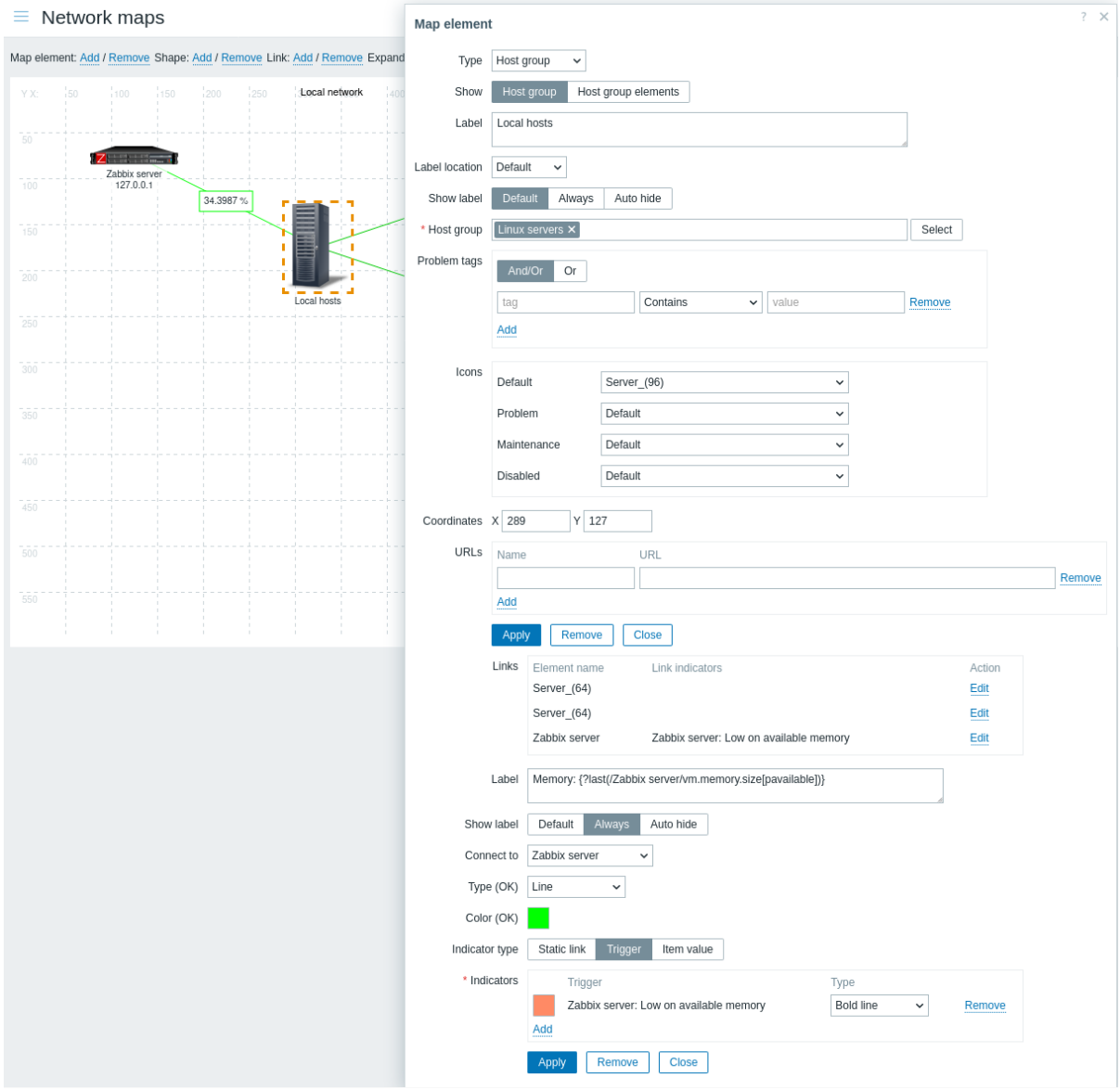
Līdzīgi, ja vienuma vērtība sasniedz norādīto slieksni vai atbilst norādītajai regulārajai izteiksmei, saites stils to var atspoguļot.

Konfigurācija

Trigeri

Lai piešķirtu trigerus kā saites indikatorus, rīkojieties šādi:

- atlasiet kartes elementu
- attiecīgajai saitei sadaļā *Links* noklikšķiniet uz *Edit*
- kā indikatora tipu atlasiet *Trigger*
- blokā *Indicators* noklikšķiniet uz *Add* un atlasiet vienu vai vairākus trigerus

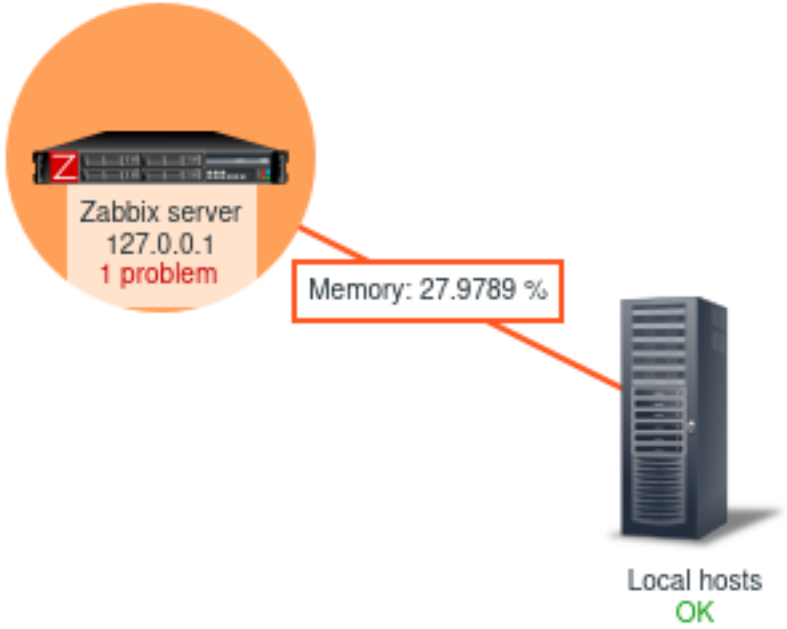


Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Pievienotie trigeri ir redzami sarakstā *Indicators*.

Katram indikatoram varat iestatīt saites tipu un krāsu tieši no saraksta. Kad tas ir izdarīts, noklikšķiniet uz *Apply*, aizveriet formu un noklikšķiniet uz *Update*, lai saglabātu kartes izmaiņas.

Sadaļā *Monitoring* → *Maps* attiecīgais saites tips un krāsa tiks parādīti, ja **trigeris** pāries problēmas stāvoklī.



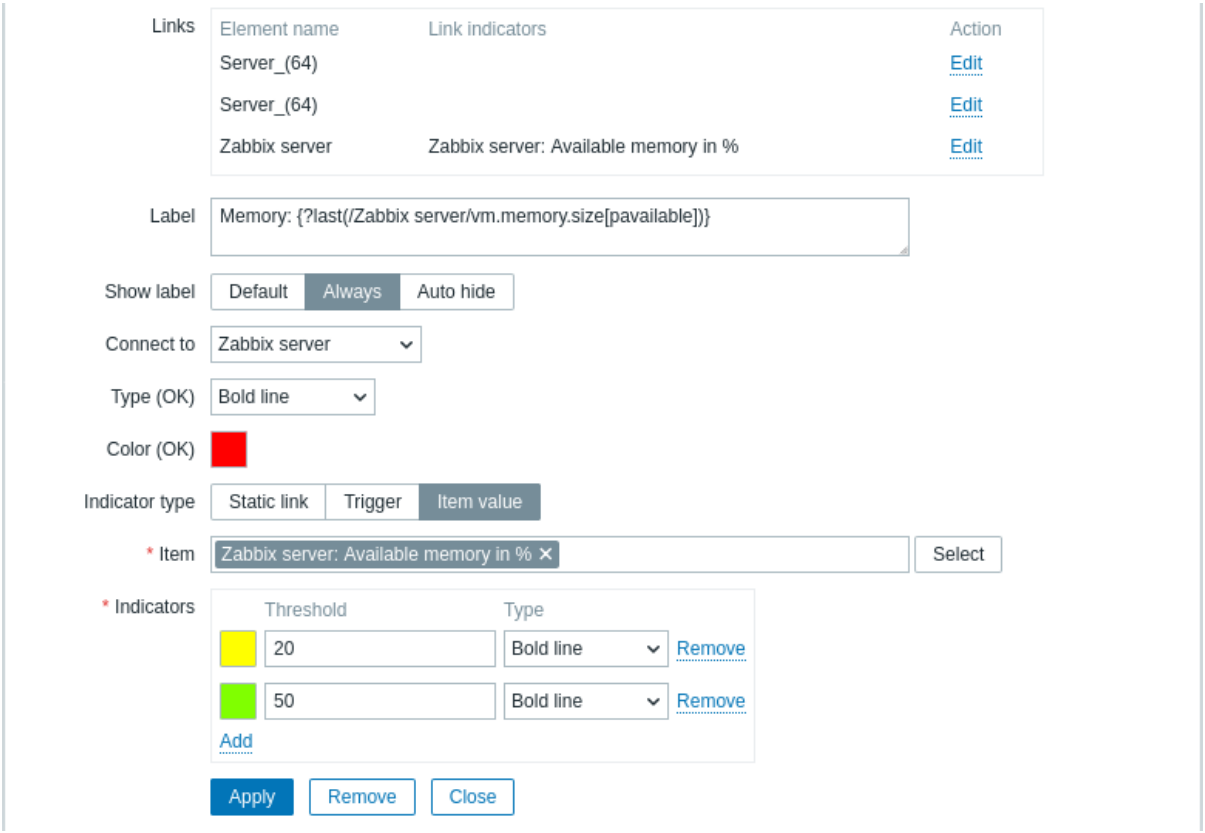
Note:
Ja vairāki trigeri pāriet problēmas stāvoklī, saites stilu un krāsu noteiks problēma ar augstāko nopietnības līmeni. Ja vienai un tai pašai kartes saitei ir piešķirti vairāki trigeri ar vienādu nopietnības līmeni, priekšroka tiek dota tam, kuram ir mazākais ID. Ņemiet vērā arī to, ka:

1. Kartes konfigurācijas iestatījumi *Minimum trigger severity* un *Show suppressed problem* ietekmē to, kuras problēmas tiek ņemtas vērā.
2. Trigeru gadījumā ar vairākām problēmām (vairākkārtēja problēmu ģenerēšana) katrai problēmai var būt nopietnības līmenis, kas atšķiras no trigera nopietnības līmeņa (mainīts manuāli), var būt atšķirīgas birkas (makrosu dēļ), un tā var būt apspiesta.

Vienumu vērtības

Lai piešķirtu vienumu vērtības kā saites indikatorus, rīkojieties šādi:

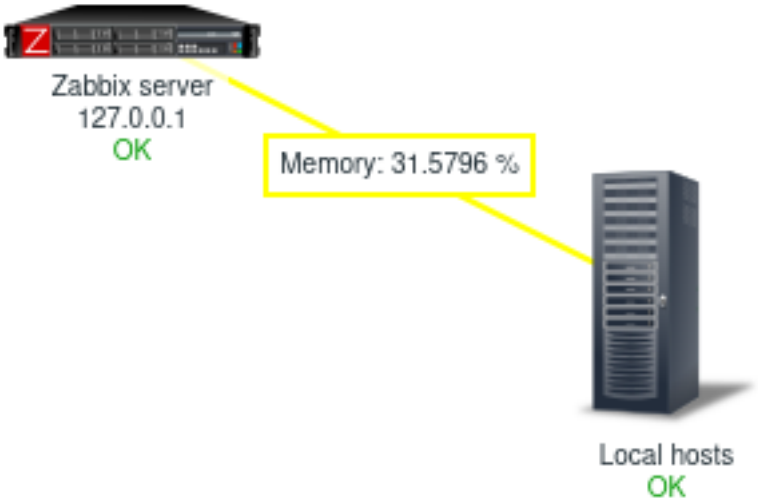
- atlasiet kartes elementu
- attiecīgajai saitei sadaļā *Links* noklikšķiniet uz *Edit*
- kā indikatora tipu atlasiet *Item value*
- atlasiet vienumu
- blokā *Indicators* pievienojiet vienu vai vairākas vienuma vērtību robežvērtības vai modeļus



Pievienotās vienumu robežvērtības/modeļi ir redzami sarakstā *Indicators*.

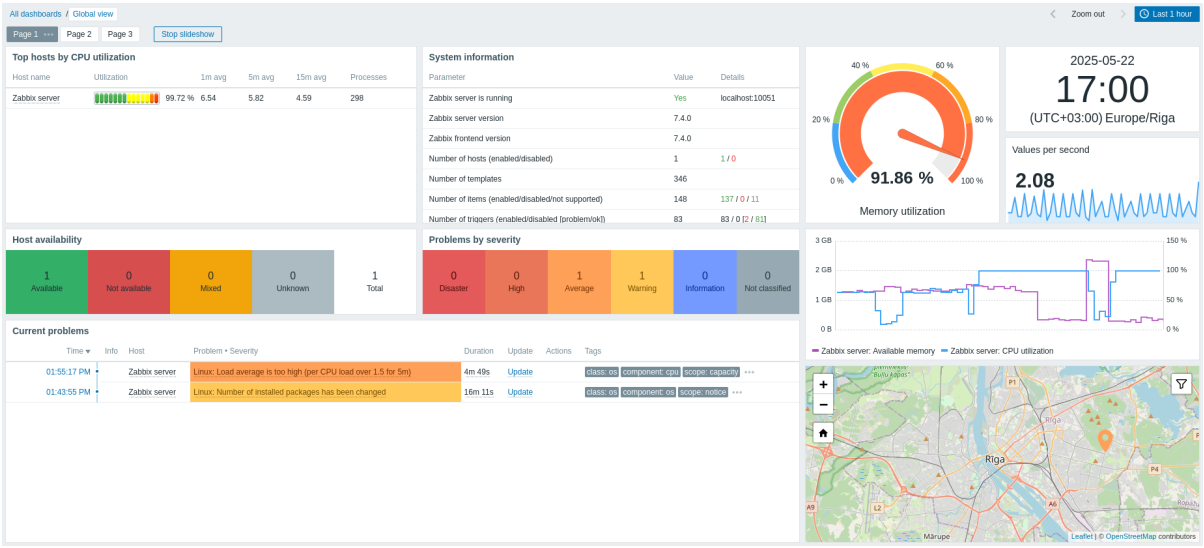
Katram indikatoram varat iestatīt saites tipu un krāsu tieši no saraksta. Kad tas ir izdarīts, noklikšķiniet uz *Apply*, aizveriet formu un noklikšķiniet uz *Update*, lai saglabātu kartes izmaiņas.

Sadaļā *Monitoring* → *Maps* attiecīgais saites tips un krāsa tiks parādīti, ja **vienuma vērtība** sasniegs iestatīto robežvērtību (skaitliskiem datu tipiem) vai atbildīs regulārās izteiksmes modelim (teksta datu tipiem).



3 Paneļi

Paneļi — gan globālie paneļi, gan hostu paneļi — nodrošina jaudīgu vizualizācijas platformu ar tādiem logrīkiem un rīkiem kā moderni grafiki, kartes, slaidrādes un citi.



8 Veidnes un veidņu grupas

Pārskats

Veidņu izmantošana ir lielisks veids, kā samazināt darba apjomu un racionalizēt Zabbix konfigurāciju. Veidne ir entītiju kopa, ko var ērti piemērot vairākiem hostiem.

Šīs entītijas var būt:

- vienumi
- trigeri
- grafiki
- paneļi
- zema līmeņa atklāšanas kārtulas
- tīmekļa scenāriji

Tā kā daudzi hosti reālajā vidē ir identiski vai diezgan līdzīgi, ir dabiski, ka entītiju kopa (vienumi, trigeri, grafiki, ...) , ko esat izveidojis vienam hostam, var būt noderīga daudziem citiem. Protams, jūs varētu tās kopēt uz katru jaunu hostu, taču tas prasītu daudz manuāla darba. Tā vietā, izmantojot veidnes, varat tās nokopēt uz vienu veidni un pēc tam piemērot šo veidni tik daudziem hostiem, cik nepieciešams.

Kad veidne ir piesaistīta hostam, visas veidnes entītijas (vienumi, trigeri, grafiki, ...) tiek pievienotas hostam. Veidnes tiek piešķirtas katram atsevišķam hostam tieši (nevis hostu grupai).

Veidnes bieži izmanto, lai grupētu entītijas konkrētiem pakalpojumiem vai lietotnēm (piemēram, Apache, MySQL, PostgreSQL, Postfix ...) un pēc tam piemērotu tās hostiem, kuros šie pakalpojumi darbojas.

Vēl viena veidņu izmantošanas priekšrocība ir situācijās, kad kaut kas ir jāmaina visiem hostiem. Vienreiz veicot izmaiņas veidnes līmenī, tās tiks izplatītas uz visiem piesaistītajiem hostiem.

Veidnes ir organizētas veidņu grupās.

Turpiniet ar veidnes izveidi un konfigurēšanu.

9 Gatavas veidnes

Pārskats

Zabbix nodrošina arvien plašāku iepriekš konfigurētu veidņu kopu, lai vienkāršotu un paātrinātu uzraudzības mērķu iestatīšanu.

Visas standarta veidnes ir pieejamas sadaļā *Datu ievākšana* > *Veidnes*.

Jauninot Zabbix, esošās veidnes netiek atjauninātas automātiski, lai izvairītos no pielāgoto izmaiņu pārrakstīšanas. Lai jaunināt esošās veidnes vai pievienotu jaunas, skatiet Veidņu jaunināšana.

Lūdzu, izmantojiet sānjoslu, lai piekļūtu informācijai par konkrētiem veidņu tipiem un darbības prasībām.

Skatiet arī:

- Veidnes piesaistīšana
- Zināmās problēmas ar veidnēm

Veidņu jaunināšana

Lai pēc Zabbix jaunināšanas jauninātu vienu veidni vai pievienotu jaunu:

1. Dodieties uz Zabbix Git repozitoriju.
2. Zaru atlasītājā (zem etiķetes *Source*) izvēlieties zaru, kas atbilst jūsu Zabbix versijai (piemēram, *master* jaunākajai izstrādes versijai).
3. Atveriet veidni, kuru vēlaties pievienot vai jaunināt (piemēram, MySQL by Zabbix aģents).
4. Nokopējiet veidnes faila saturu (piemēram, `template_db_mysql_agent.yaml`) un saglabājiet to lokālā failā.
5. Zabbix lietotāja saskarnē dodieties uz *Data collection* > *Templates* un importējiet lokālo failu.

Varat arī vienlaikus jaunināt visas veidnes un pievienot jaunas:

1. Lejupielādējiet veidņu pakotni atkarībā no jūsu tīmekļa servera: `templates_1M_8.0.7z` (Nginx) vai `templates_2M_8.0.7z` (Apache).
2. Izpakojiet pakotni. Tā satur YAML failus (katrs ar vairākām veidnēm) un indeksu (`index-1M_80.md`), kurā uzskaitītas katrā YAML failā iekļautās veidnes.
3. Zabbix lietotāja saskarnē dodieties uz *Data collection* > *Templates* un importējiet YAML failus pa vienam.

Alternatīvi, varat importēt visus YAML failus uzreiz, izmantojot skriptu `import_templates.sh`. Skripts prasa utilītas `curl` un `jq`, un izmanto Zabbix API metodi `configuration.import`. Palaižot skriptu, norādiet direktoriju, kurā atrodas YAML faili, kā arī savu Zabbix API URL un autentifikācijas marķieri.

> **Warning:**
> Skripts `import_templates.sh` pārraksta visas esošās veidnes bez iespējas pārskatīt vai atsaukt veidņu izmaiņas.

Piemērs:

```
./import_templates.sh /tmp/templates_2M_8.0 https://example.com/zabbix/api_jsonrpc.php d8d6b5c78ee2a8333db
```

Veidņu saderība ar Host Wizard

Host Wizard ir saderīgs ar visām standarta veidnēm.

Varat arī padarīt savas pielāgotās veidnes saderīgas ar Host Wizard:

1. Zabbix lietotāja saskarnē atveriet *Data collection* > *Templates* un eksportējiet pielāgoto veidni.
2. Rediģējiet eksportēto failu:
   - Pievienojiet `wizard_ready` veidnes elementu, lai norādītu saderību ar Host Wizard.
   - Pēc nepieciešamības pievienojiet `readme` vai `config` veidnes elementu.
3. Zabbix lietotāja saskarnē atveriet *Data collection* > *Templates* un importējiet atjaunināto veidnes failu.

1 Zabbix aģenta veidņu darbība

Darbības, lai nodrošinātu pareizu to veidņu darbību, kas apkopo metriku ar Zabbix aģentu:

1. Pārliecinieties, ka Zabbix aģents ir instalēts hostā. Aktīvo pārbaužu gadījumā pārliecinieties arī, ka Zabbix servera/starpniekservera adrese ir pievienota aģenta konfigurācijas faila parametram 'ServerActive'.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms tā ir jāimportē).
3. Ja nepieciešams, pielāgojiet veidnes makrosu vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- Apache ar Zabbix aģentu
- Apache ar aktīvo Zabbix aģentu
- HAProxy ar Zabbix aģentu
- IIS ar Zabbix aģentu
- IIS ar aktīvo Zabbix aģentu
- Microsoft Exchange Server 2016 ar Zabbix aģentu
- Microsoft Exchange Server 2016 ar aktīvo Zabbix aģentu
- MySQL ar Zabbix aģentu

- MySQL ar aktīvo Zabbix aģentu
- Nginx ar Zabbix aģentu
- Nginx ar aktīvo Zabbix aģentu
- PHP-FPM ar Zabbix aģentu
- PHP-FPM ar aktīvo Zabbix aģentu
- PostgreSQL ar Zabbix aģentu
- PostgreSQL ar aktīvo Zabbix aģentu
- RabbitMQ klasteris ar Zabbix aģentu

2 Zabbix aģenta 2 veidņu darbība

Darbības, lai nodrošinātu pareizu to veidņu darbību, kas apkopo metriku ar Zabbix aģentu 2:

1. Pārliecinieties, ka aģents 2 ir instalēts hostā un ka instalētajā versijā ir iekļauts nepieciešamais spraudnis. Dažos gadījumos vispirms var būt nepieciešams jaunināt aģentu 2.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms būs nepieciešams importēt veidni).
3. Ja nepieciešams, pielāgojiet veidnes makrosu vērtības. Ņemiet vērā, ka lietotāja makrosus var izmantot, lai pārrakstītu konfigurācijas parametrus.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Attention:
Zabbix aģenta 2 veidnes darbojas kopā ar spraudņiem. Pamata konfigurāciju var veikt, vienkārši pielāgojot lietotāja makrosus, savukārt padziļinātu pielāgošanu var panākt, konfigurējot pašu spraudni. Piemēram, ja spraudnis atbalsta nosauktās sesijas, ir iespējams uzraudzīt vairākas viena veida entītijas (piemēram, MySQL1 un MySQL2), konfigurācijas failā katrai entītijai norādot nosaukto sesiju ar savu URI, lietotājvārdu un paroli.

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- Ceph ar Zabbix aģentu 2
- Docker
- Memcached
- MongoDB klasteris ar Zabbix aģentu 2
- MongoDB mezgls ar Zabbix aģentu 2
- MySQL ar Zabbix aģentu 2
- MySQL ar Zabbix aģentu 2 aktīvais
- Oracle ar Zabbix aģentu 2
- PostgreSQL ar Zabbix aģentu 2
- PostgreSQL ar Zabbix aģentu 2 aktīvais
- Redis
- SMART ar Zabbix aģentu 2
- SMART ar Zabbix aģentu 2 aktīvais
- Systemd ar Zabbix aģentu 2
- Tīmekļa vietnes sertifikāts ar Zabbix aģentu 2

3 HTTP veidņu darbība

Darbības, lai nodrošinātu pareizu to veidņu darbību, kas apkopo metriku ar HTTP aģentu:

1. Izveidojiet Zabbix hostu, kas atbilst uzraudzības mērķim. Norādiet uzraudzības mērķa IP adresi/DNS nosaukumu kā galveno hosta saskarni. Tas ir nepieciešams, lai makross {HOST.CONN} veidnes vienumos tiktu korekti atrisināts.
2. Piesaistiet veidni 1. darbībā izveidotajam hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms būs nepieciešams veidni importēt).
3. Ja nepieciešams, pielāgojiet veidnes līmeņa makrosu vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- 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 IPMI veidnes darbība

IPMI veidnēm nav nepieciešama īpaša iestatīšana. Lai sāktu uzraudzību, saistiet veidni ar mērķa hostu (ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms būs nepieciešams importēt veidni).

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Pieejamā veidne:

- Šasija pēc IPMI

5 JMX veidņu darbība

Darbības, lai nodrošinātu pareizu veidņu darbību, kas vāc metriku, izmantojot JMX:

1. Pārliecinieties, ka Zabbix Java gateway ir instalēts un pareizi konfigurēts.
2. Piesaistiet veidni mērķa hostam. Hostam jābūt konfigurētai JMX saskarnei. Ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms būs nepieciešams importēt veidni.
3. Ja nepieciešams, pielāgojiet veidnes makrosu vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- 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 ODBC veidņu darbība

Darbības, lai nodrošinātu pareizu veidņu darbību, kas vāc metriku, izmantojot ODBC monitoringu:

1. Pārliecinieties, ka nepieciešamais ODBC draiveris ir instalēts Zabbix serverī vai starpniekserverī.
2. Piesaistiet veidni mērķa hostam (ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms būs nepieciešams importēt veidni).
3. Ja nepieciešams, pielāgojiet veidnes makrosu vērtības.
4. Konfigurējiet uzraugāmo instanci tā, lai tā atļautu datu koplietošanu ar Zabbix.

Detalizēts veidnes apraksts, tostarp pilns makrosu, vienumu un trigeru saraksts, ir pieejams veidnes README failā (tam var piekļūt, noklikšķinot uz veidnes nosaukuma).

Ir pieejamas šādas veidnes:

- MariaDB ar ODBC
- MSSQL ar ODBC
- MySQL ar ODBC
- Oracle ar ODBC
- Percona ar ODBC
- PostgreSQL ar ODBC

7 Standartizētas veidnes tīkla ierīcēm

Pārskats

Lai nodrošinātu tīkla ierīču, piemēram, komutatoru un maršrutētāju, uzraudzību, esam izveidojuši divus tā sauktos modeļus: tīkla ierīcei pašai (būtībā tās šasijai) un tīkla saskarnei.

Ir pieejamas veidnes daudzām tīkla ierīču saimēm. Visas veidnes aptver (ja no ierīces ir iespējams iegūt šos vienumus):

- Šasijas kļūmju uzraudzību (barošanas bloki, ventilatori un temperatūra, kopējais statuss)
- Šasijas veiktspējas uzraudzību (CPU un atmiņas vienumi)
- Šasijas inventāra apkopošanu (sērijas numuri, modeļa nosaukums, aparātprogrammatūras versija)
- Tīkla saskarņu uzraudzību ar IF-MIB un EtherLike-MIB (saskarnes statuss, saskarnes datplūsmas slodze, dupleksa statuss Ethernet tīkliem)

Ja veidne nav pieejama jūsu Zabbix instalācijā, iespējams, vispirms tā ir jāimportē.

Ja importējat jaunās iebūvētās veidnes, iespējams, vēlēsieties atjaunināt arī globālo regulāro izteiksmi `@Network interfaces for discovery` uz:

```
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.]*$
```

lai lielākajā daļā sistēmu atfiltrētu loopback un null saskarnes.

Ierīces

To ierīču saimju saraksts, kurām ir pieejamas veidnes:

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
Alcatel Timetra TiMOS SNMP	Alcatel	Alcatel Timetra	ALCATEL SR 7750	TiMOS	TIMETRA-SYSTEM-MIB,TIMETRA-CHASSIS-MIB	Sertificēts
Aruba CX 8300s by SNMP	HPE Aruba Network-ing	Aruba CX komuta-toru sērija	Aruba JL636A 8325, Aruba JL717A 8360	Aruba AOS-CX	ARUBAWIRED-FAN-MIB, ARUBAWIRED-POWERSUPPLY-MIB, ARUBAWIRED-SYSTEMINFO-MIB, ARUBAWIRED-TEMPSENSOR-MIB, OSPF-MIB	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
Brocade FC SNMP	Brocade	Brocade FC komutatori	Brocade 300 SAN Switch-	-	SW-MIB,ENTITY-MIB	Veiktspēja Kļūmes
Brocade_Foundry Stackable SNMP	Brocade	Brocade ICX	Brocade ICX6610, Brocade ICX7250-48, Brocade ICX7450-48F		FOUNDRY-SN-AGENT-MIB, FOUNDRY-SN-STACKING-MIB	Sertificēts
Brocade_Foundry Nonstackable SNMP	Brocade, Foundry	Brocade MLX, Foundry	Brocade MLXe, Foundry FLS648, Foundry FWSX424		FOUNDRY-SN-AGENT-MIB	Veiktspēja Kļūmes
Check Point Next Generation Firewall by SNMP	Check Point	Nākamās paaudzes ugunsmūris	-	Gaia	HOST-RESOURCES-MIB, CHECKPOINT-MIB, UCD-SNMP-MIB, SNMPv2-MIB, IF-MIB	Sertificēts
Ciena 3906 by SNMP	Ciena	Ciena CPE	Ciena 3906	SAOS	WWP-LEOS-BLADE-MIB, WWP-LEOS-CHASSIS-MIB, WWP-LEOS-SYSTEM-CONFIG-MIB	Sertificēts
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	Sertificēts
Cisco IOS SNMP	Cisco	Cisco IOS ver > 12.2 3.5	Cisco C2950	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Sertificēts
Cisco IOS versions 12.0_3_T-12.2_3.5 SNMP	Cisco	Cisco IOS > 12.0 3 T and 12.2 3.5	-	IOS	CISCO-PROCESS-MIB,CISCO-MEMORY-POOL-MIB,CISCO-ENVMON-MIB	Sertificēts
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	Sertificēts
D-Link DES_DGS Switch SNMP	D-Link	DES/DGX komutatori	D-Link DES-xxxx/DGS-xxxx,DLINK DGS-3420-26SC	-	DLINK-AGENT-MIB,EQUIPMENT-MIB,ENTITY-MIB	Sertificēts
D-Link DES 7200 SNMP	D-Link	DES-7xxx	D-Link DES 7206	-	ENTITY-MIB,MY-SYSTEM-MIB,MY-PROCESS-MIB,MY-MEMORY-MIB	Veiktspēja Kļūmes Saskarnes
Dell Force S-Series SNMP	Dell	Dell Force S-Series	S4810		F10-S-SERIES-CHASSIS-MIB	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
Extreme Exos SNMP	Extreme	Extreme EXOS	X670V-48x	EXOS	EXTREME-SYSTEM-MIB,EXTREME-SOFTWARE-MONITOR-MIB	Sertificēts
FortiGate by SNMP	Fortinet	FortiGate (NGFW)	-	FortiOS	HOST-RESOURCES-MIB FORTINET-FORTIGATE-MIB FORTINET-CORE-MIB SNMPv2-MIB IF-MIB ENTITY-MIB	Veiktspēja Inventārs
HP Comware HH3C SNMP	HP	HP (H3C) Comware	HP A5500-24G-4SFP HI Switch		HH3C-ENTITY-EXT-MIB,ENTITY-MIB	Sertificēts
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	Sertificēts
Huawei AR600 by SNMP	Huawei	Huawei NetEngine AR600 sērijas maršrutētāji	Huawei AR600/AR6100/AR6200/AR6300 sērijas maršrutētāji ar VRP programmatūru (V300R019/V300R021/V300R023 vai jaunāku)		HUAWEI-CBQOS-MIB, HUAWEI-NQA-MIB, ENTITY-MIB, IF-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Sertificēts
Huawei OceanStor 5300 V5 by SNMP	Huawei	Huawei OceanStor Dorado	Huawei OceanStor Dorado ar V5 programmatūru		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Sertificēts
Huawei OceanStor Dorado by SNMP	Huawei	Huawei OceanStor, Huawei OceanStor Dorado	Huawei OceanStor Dorado 3000/5000/6000/8000/18000/5300V/5500/5600/6800/18500/18800 ar V6/V700 programmatūru		HUAWEI-STORAGE-HARDWARE-MIB, HUAWEI-STORAGE-SPACE-MIB, ISM-PERFORMANCE-MIB, HOST-RESOURCES-MIB, SNMPv2-MIB	Sertificēts
Huawei VRP by SNMP	Huawei	Huawei VRP	S2352P-EI		ENTITY-MIB,HUAWEI-ENTITY-EXTENT-MIB	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
Intel_Qlogic Infiniband SNMP	Intel/QLogic	Intel/QLogic Infini-band ierīces	Infiniband 12300		ICS-CHASSIS-MIB	Kļūmes Inventārs
Juniper SNMP	Juniper	MX,SRX,EX modeļi	Juniper MX240, Juniper EX4200-24F	JunOS	JUNIPER-MIB	Sertificēts
Juniper MX NETCONF	Juniper	MX modeļi	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	*Nav piemērojams (izmanto NETCONF over SSH)*	Sertificēts
Juniper MX SNMP	Juniper	MX modeļi	Juniper MX204 Edge Router	JunOS 24.2R1-S1.10	OSPF-MIB, JUNIPER-DOM-MIB, JUNIPER-MIB, BGP4-V2-MIB-JUNIPER, OSPFV3-MIB-JUNIPER	Sertificēts
Mellanox SNMP	Mellanox	Mellanox Infini-band ierīces	SX1036	MLNX-OS	HOST-RESOURCES-MIB,ENTITY-MIB,ENTITY-SENSOR-MIB,MELLANOX-MIB	Sertificēts
MikroTik CCR<device model> SNMP	MikroTik	MikroTik Cloud Core maršrutētāji (CCR sērija)	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: 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	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
MikroTik CRS<device model> SNMP	MikroTik	MikroTik Cloud Router komutatori (CRS sērija)	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: 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	Sertificēts
MikroTik CSS<device model> SNMP	MikroTik	MikroTik Cloud Smart komutatori (CSS sērija)	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: MikroTik CSS326-24G-2S+RM, MikroTik CSS610-8G-2S+IN	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts
MikroTik FiberBox SNMP	MikroTik	MikroTik FiberBox	MikroTik FiberBox	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
MikroTik hEX <device model> SNMP	MikroTik	MikroTik hEX	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: MikroTik hEX, MikroTik hEX lite, MikroTik hEX PoE, MikroTik hEX PoE lite, MikroTik hEX S	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts
MikroTik netPower <device model> SNMP	MikroTik	MikroTik net-Power	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: MikroTik netPower 15FR, MikroTik netPower 16P SNMP, MikroTik netPower Lite 7R	RouterOS/SwitchOS, SwitchOS Lite	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts
MikroTik PowerBox <device model> SNMP	MikroTik	MikroTik Power-Box	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: MikroTik PowerBox, MikroTik PowerBox Pro	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts
MikroTik RB<device model> SNMP	MikroTik	MikroTik RB sērijas maršrutētāji	Ir pieejamas atsevišķas specializētas veidnes šādiem modeļiem: 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	Sertificēts

Veidnes nosaukums	Piegādātājs	Ierīču saime	Zināmie modeļi	OS	Izmantotie MIB	**Birkas**
MikroTik SNMP	MikroTik	MikroTik RouterOS ierīces	MikroTik CCR1016-12G, MikroTik RB2011UAS-2HnD, MikroTik 912UAG-5HPnD, MikroTik 941-2nD, MikroTik 951G-2HnD, MikroTik 1100AHx2	RouterOS	MIKROTIK-MIB,HOST-RESOURCES-MIB	Sertificēts
Netgear Fastpath SNMP	Netgear	Netgear Fastpath	M5300-28G		FASTPATH-SWITCHING-MIB,FASTPATH-BOXSERVICES-PRIVATE-MIB	Kļūmes Inventārs
QTech QSW SNMP	QTech	Qtech ierīces	Qtech QSW-2800-28T	-	QTECH-MIB,ENTITY-MIB	Veiktspēja Inventārs
Stormshield SNS by SNMP	Stormshield	Stormshield Network Security (SNS) ugun-smūri	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	Sertificēts
TP-LINK SNMP	TP-LINK	TP-LINK	T2600G-28TS v2.0		TPLINK-SYSMONITOR-MIB,TPLINK-SYSINFO-MIB	Veiktspēja Inventārs
Ubiquiti AirOS SNMP	Ubiquiti	Ubiquiti AirOS bezvadu ierīces	NanoBridge,NanoStation,Unifi	AirOS	FROGFOOT-RESOURCES-MIB,IEEE802dot11-MIB	Veiktspēja
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	Veiktspēja Inventārs

Veidnes dizains

Veidnes tika izstrādātas, ņemot vērā sekojošo:

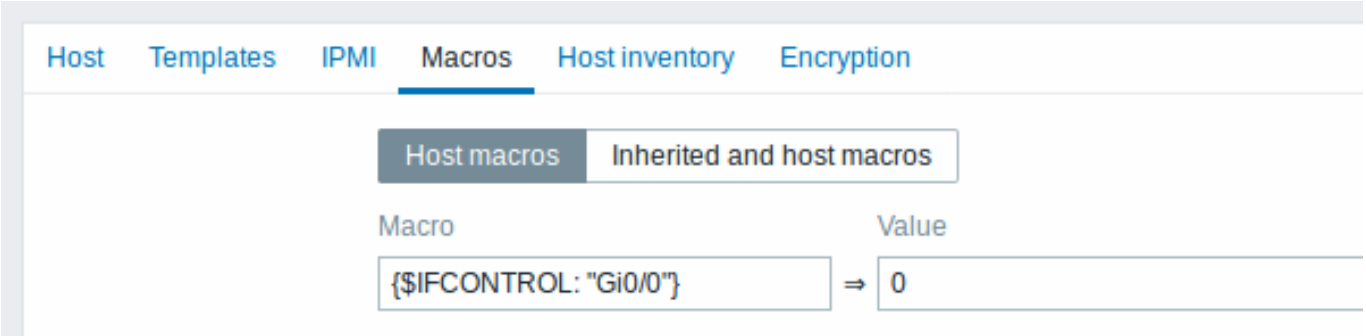
- Lietotāja makrosi tiek izmantoti pēc iespējas vairāk, lai lietotājs varētu pielāgot trigerus;
- Zema līmeņa atklāšana tiek izmantota pēc iespējas vairāk, lai samazinātu neatbalstīto vienumu skaitu;

- Visas veidnes ir atkarīgas no Template ICMP Ping, lai visas ierīces tiktu pārbaudītas arī ar ICMP;
- Vienumi neizmanto nekādus MIB — vienumos un zema līmeņa atklāšanā tiek izmantoti SNMP OID. Tāpēc, lai veidnes darbotos, Zabbix nav nepieciešams ielādēt nekādus MIB;
- Atklāšanas laikā tiek filtrētas loopback tīkla saskarnes, kā arī saskarnes ar ifAdminStatus = down(2)
- Ja iespējams, tiek izmantoti 64 bitu skaitītāji no IF-MIB::ifXTable. Ja tas netiek atbalstīts, tā vietā tiek izmantoti noklusējuma 32 bitu skaitītāji.

Visām atklātajām tīkla saskarnēm ir trigeris, kas uzrauga to darbības statusu (saite), piemēram:

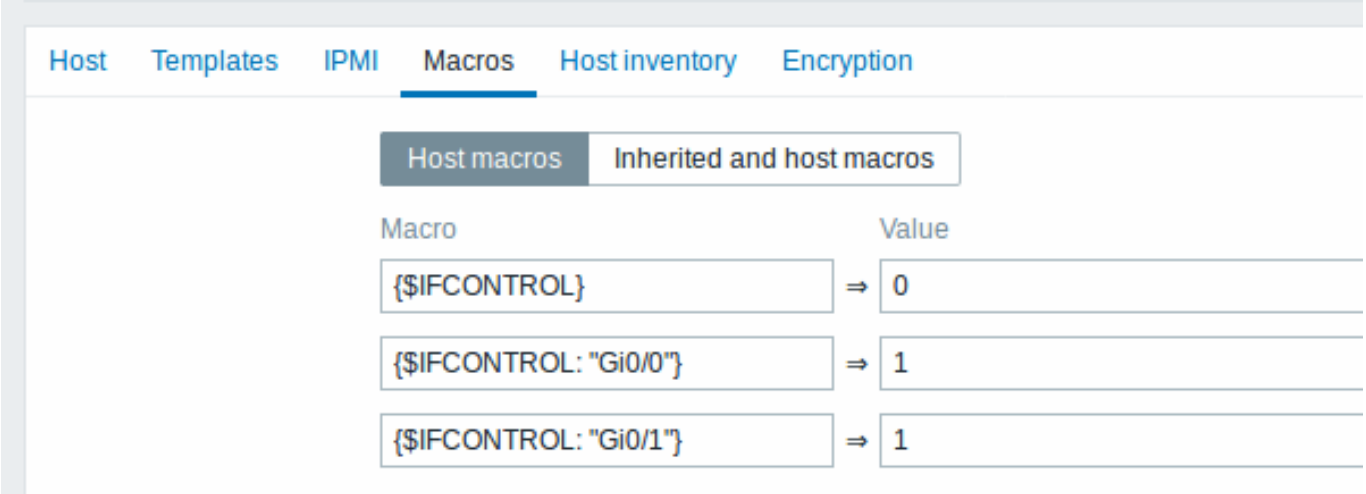
```
{$IFCONTROL:"{#IFNAME}"}=1 and last(/Alcatel Timetra TiMOS SNMP/net.if.status[ifOperStatus.{#SNMPINDEX}])
```

- Ja nevēlaties uzraudzīt šo nosacījumu konkrētai saskarnei, izveidojiet lietotāja makrosu ar kontekstu un vērtību 0. Piemēram:



kur Gi0/0 ir {#IFNAME}. Tādā veidā šis trigeris šai konkrētajai saskarnei vairs netiek izmantots.

- Varat arī mainīt noklusējuma darbību tā, lai visi trigeri nenostrādātu, un aktivizēt šo trigeri tikai ierobežotam saskarņu skaitam, piemēram, augšupsaitēm:



Birkas

- Veiktspēja – ierīču saimes MIB nodrošina veidu, kā uzraudzīt CPU un atmiņas vienumus;
- Kļūmes - ierīču saimes MIB nodrošina veidu, kā uzraudzīt vismaz vienu temperatūras sensoru;
- Inventārs – ierīču saimes MIB nodrošina veidu, kā apkopot vismaz ierīces sērijas numuru un modeļa nosaukumu;
- Sertificēts – ir aptvertas visas trīs iepriekš minētās galvenās kategorijas.

8 VMware veidnes darbība

Pārskats

Zabbix ietver virkni lietošanai gatavu veidņu virtuālo mašīnu uzraudzībai, kas paredzētas VMware vCenter un ESXi hipervizoriem. Pieejamās veidnes ir iekļautas divos atsevišķos veidņu komplektos:

- VMware - atbilstošajiem makrosiem izmanto UUID datus
- VMware FQDN - atbilstošajiem makrosiem izmanto FQDN datus

Veidnes, kas paredzētas VMware vCenter vai ESXi hipervizoru uzraudzībai, satur iepriekš konfigurētus zema līmeņa atklāšanas noteikumus, kā arī dažādas iebūvētas pārbaudes virtuālo instalāciju uzraudzībai.

Note:
Lai veidne *VMware FQDN* darbotos pareizi, katrai uzraudzītajai virtuālajai mašīnai jābūt unikālam OS nosaukumam, kas atbilst FQDN noteikumiem. Turklāt katrā mašīnā jābūt instalētiem VMware Tools/Open Virtual Machine Tools. Ja šie priekšnosacījumi ir izpildīti, ieteicams izmantot veidni *VMware FQDN*. Veidne *VMware FQDN* ir pieejama kopš Zabbix 5.2, kad tika ieviesta iespēja izveidot hostus ar pielāgotām saskarnēm.

 Ir pieejama arī klasiskā *VMware* veidne, ko var izmantot, ja FQDN prasības nav izpildītas. Tomēr veidnei *VMware* ir zināma problēma. Atklātajām virtuālajām mašīnām hosti tiek izveidoti ar nosaukumiem, kas ir saglabāti vCenter (piemēram, "VM1", "VM2" utt.). Ja šajos hostos ir instalēts Zabbix aģents un ir iespējota aktīvā Zabbix aģenta automātiskā reģistrācija, automātiskās reģistrācijas process nolasīs hostu nosaukumus tādus, kādi tie tika reģistrēti palaišanas laikā (piemēram, "vm1.example.com", "vm2.example.com" utt.). Tas var izraisīt jaunu hostu izveidi jau esošām virtuālajām mašīnām (jo netika atrastas atbilstošas nosaukumu sakritības), kā rezultātā rodas dublēti hosti ar atšķirīgiem nosaukumiem.

Lūdzu, ņemiet vērā:

- Veidni *VMware Hypervisor* var manuāli piesaistīt hostam, kā arī izmantot atklāšanā;
- Veidni *VMware Guest* nevajadzētu manuāli piesaistīt hostam, un to var izmantot tikai atklāšanā.

Hosta makro konfigurācija

Lai izmantotu VMware vienkāršās pārbaudes, hostam jābūt definētiem šādiem lietotāja makro:

- {$VMWARE.URL} - VMware pakalpojuma (vCenter vai ESXi hipervizora) SDK URL (https://servername/sdk)
- {$VMWARE.USERNAME} - VMware pakalpojuma lietotājvārds
- {$VMWARE.PASSWORD} - VMware pakalpojuma lietotāja {$VMWARE.USERNAME} parole

10 Paziņojumi par notikumiem

Pārskats

Pieņemot, ka esam konfigurējuši dažus vienumus un trigerus un tagad saņemam dažus notikumus, kas rodas trigeru stāvokļa maiņas rezultātā, ir laiks apsvērt dažas darbības.

Vispirms mēs nevēlētos visu laiku skatīties uz trigeru vai notikumu sarakstu. Būtu daudz labāk saņemt paziņojumu, ja ir noticis kaut kas būtisks (piemēram, problēma). Tāpat, kad rodas problēmas, mēs vēlētos redzēt, ka visas iesaistītās personas ir informētas.

Tāpēc paziņojumu sūtīšana ir viena no galvenajām darbībām, ko piedāvā Zabbix. Var noteikt, kam un kad būtu jāsaņem paziņojums par noteiktu notikumu.

Lai varētu sūtīt un saņemt paziņojumus no Zabbix, jums ir:

- jādefinē daži mediji
- jākonfigurē darbība, kas nosūta ziņojumu uz vienu no definētajiem medijiem

Darbības sastāv no *nosacījumiem* un *operācijām*. Pamatā, kad nosacījumi ir izpildīti, tiek veiktas operācijas. Divas galvenās operācijas ir ziņojuma (paziņojuma) nosūtīšana un attālinātas komandas izpilde.

Atklāšanas un automātiskās reģistrācijas izveidotajiem notikumiem ir pieejamas dažas papildu operācijas. Tās ietver hosta pievienošanu vai noņemšanu, veidnes piesaistīšanu utt.

1 Mediju tipi

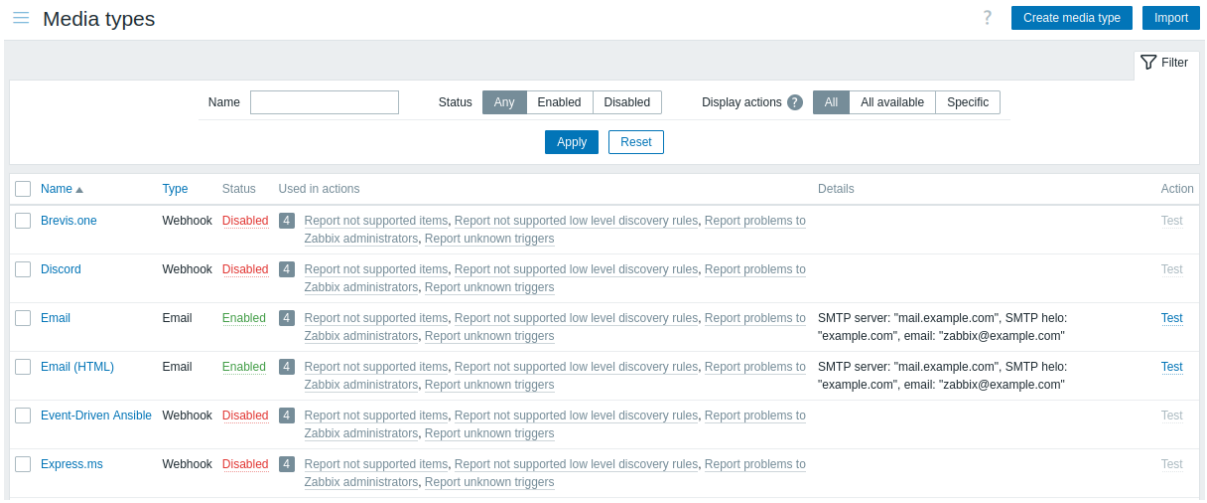
Pārskats

Mediju tipi ir piegādes kanāli, ko izmanto paziņojumu un brīdinājumu sūtīšanai no Zabbix.

Mediju tipi atbalsta šādas piegādes metodes:

- E-pasts
- SMS
- Pielāgots skripts
- Webhook

Mediju tipi tiek uzturēti sadaļā *Brīdinājumi* > *Mediju tipi*. Daži mediju tipi jau ir iepriekš definēti noklusējuma datu kopā. Jums tikai jāpielāgo to parametri, lai tie darbotos.



Lai redzētu, kā mediju tipi iekļaujas brīdināšanas procesā, aplūkosim trīs prasības, kas nepieciešamas, lai piegādātu paziņojumus no Zabbix gala lietotājiem:

1. Jābūt definētai darbības operācijai, kas sūta paziņojumus
2. Jābūt definētam strādājošam mediju tipam (piemēram, *E-pasts*, kas sūta brīdinājumus, izmantojot SMTP)
3. Lietotāja medijā jābūt definētai lietotāja līmeņa piegādes informācijai (piemēram, e-pasta adresēm, tālruņu numuriem utt.)

Medija tipa testēšana

Lai pārbaudītu, vai konfigurētais medija tips darbojas, medija tipu sarakstā noklikšķiniet uz *Test*.

Testēšanas pieprasījums tiks nosūtīts uz Zabbix serveris. Zabbix serveris mēģinās nosūtīt brīdinājumu, izmantojot norādīto medija tipu, un atgriezīs rezultātu lietotāja saskarnei. Lietotāja saskarne gaidīs, līdz serveris atgriezīs rezultātus. Medija tipa testēšanai pēc noklusējuma ir 65 sekunžu noildze (konfigurējama sadaļā *Administration* > *General* > *Timeouts*).

Skatiet arī testēšanas informāciju par:

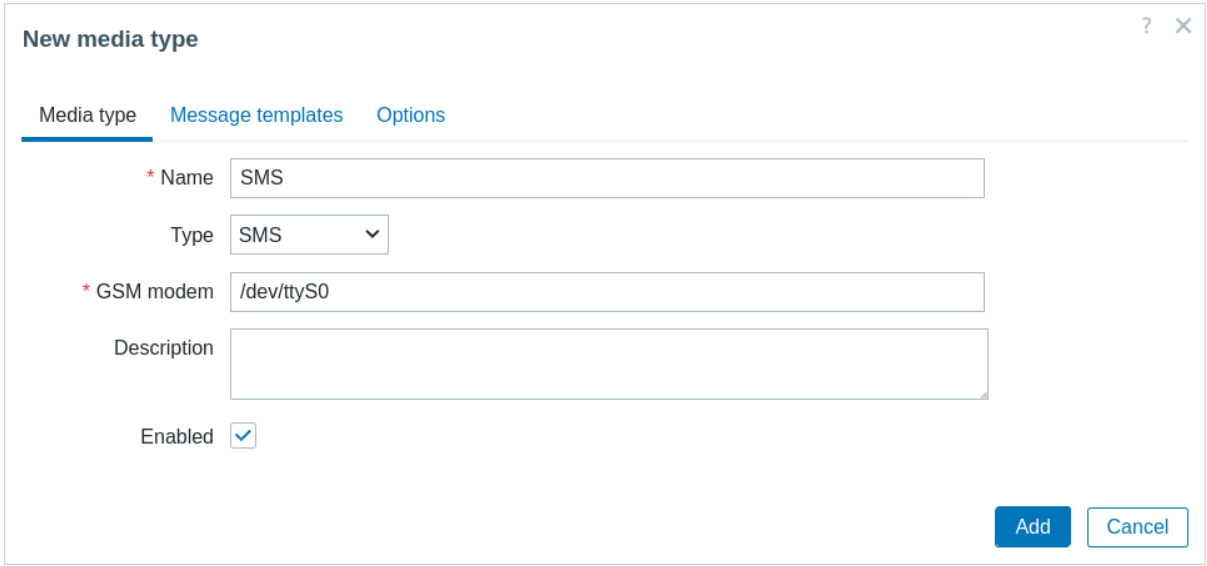
- E-pasts
- Webhook
- Skripts

Konfigurācija

Lai Zabbix lietotāja saskarnē izveidotu medija tipu:

- Dodieties uz: *Brīdinājumi* > *Mediju tipi*
- Noklikšķiniet uz *Izveidot medija tipu*
- Ievadiet medija tipa parametrus formā

Daži parametri ir **kopīgi** visām piegādes metodēm.



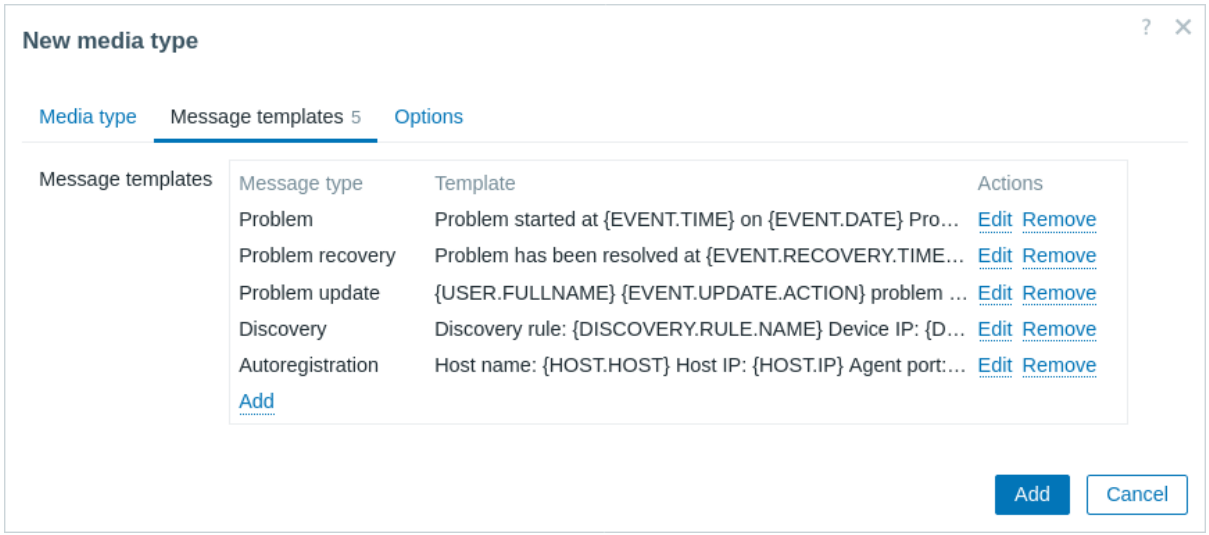
Parametrs	Apraksts
Name	Medija tipa nosaukums.

Parametrs	Apraksts
Type	Atlasiet medija tipa piegādes metodi.
Description	Ievadiet medija tipa aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu medija tipu.

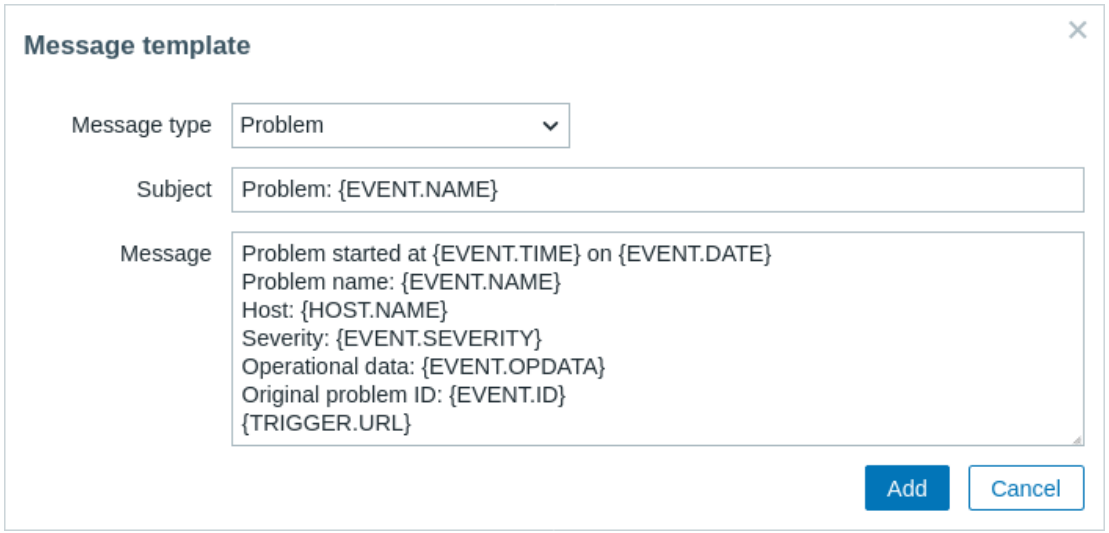
Ar metodei specifiskajiem parametriem iepazīstieties lapās e-pasts, SMS, pielāgots brīdinājuma skripts vai webhook.

Ziņojumu veidnes

Cilnē **Ziņojumu veidnes** ir noklusējuma ziņojumi katram notikuma tipam (problēma, problēmas atkopšana, atklāšana u.c.).



Noklikšķiniet uz **Pievienot**, lai definētu noklusējuma ziņojumu (vai uz **Rediģēt**, lai atjauninātu esošu ziņojumu):



Parametrs	Apraksts
Ziņojuma tips	Notikuma tips, kuram jāizmanto noklusējuma ziņojums. Katram notikuma tipam var definēt tikai vienu noklusējuma ziņojumu.
Temats	Noklusējuma ziņojuma temats. Temats var saturēt makrosus. Tas ir ierobežots līdz 255 rakstzīmēm. Temats nav pieejams SMS medija tipam.
Ziņojums	Noklusējuma ziņojums. Tas ir ierobežots līdz noteiktam rakstzīmju skaitam atkarībā no datubāzes tipa (plašāku informāciju skatiet Ziņojumu sūtīšana). Ziņojums var saturēt atbalstītos makrosus. Problēmu un problēmu atjaunināšanas ziņojumos tiek atbalstīti izteiksmju makrosi (piemēram, `{?avg(/host/key,1h)}`).

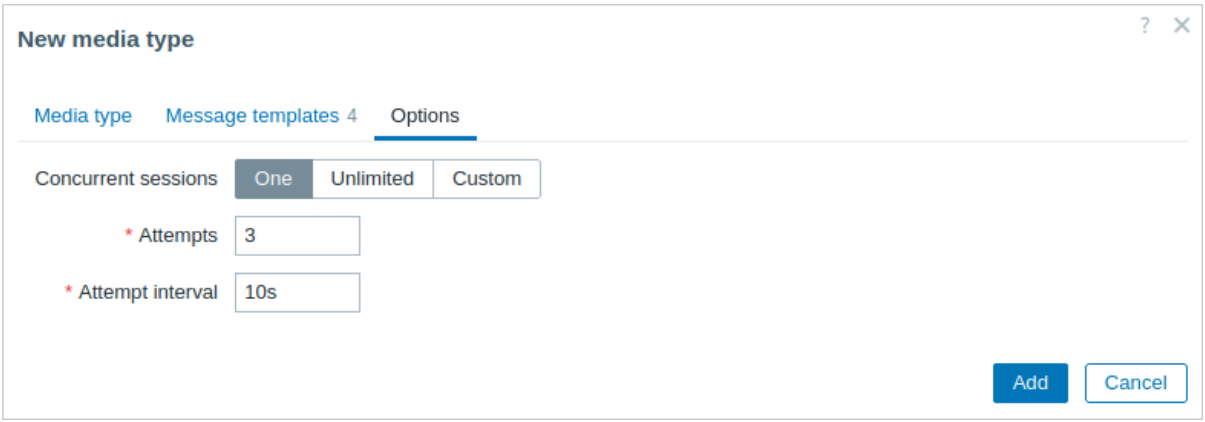
Ņemiet vērā, ka noklusējuma ziņojumus pārraksta pielāgotie ziņojumi, ja tie ir definēti darbību operācijās.

Warning:
Ziņojumu veidņu definēšana ir obligāta visām piegādes metodēm, tostarp webhook vai pielāgotiem brīdinājumu skriptiem, kas paziņojumiem neizmanto noklusējuma ziņojumus. Piemēram, darbība "Sūtīt ziņojumu uz Pushover webhook" neizdosies nosūtīt problēmas paziņojumus, ja Pushover webhook problēmas ziņojums nav definēts.

Opcijas

Cilnē **Opcijas** ir brīdinājumu apstrādes iestatījumi. Vienu un to pašu opciju kopu var konfigurēt katram mediju tipam.

Visi mediju tipi tiek apstrādāti paralēli. Lai gan maksimālo vienlaicīgo sesiju skaitu var konfigurēt katram mediju tipam atsevišķi, kopējo brīdinājumu apstrādātāju procesu skaitu serverī var ierobežot tikai ar StartAlerters parametru. Brīdinājumi, ko ģenerē viens trigeris, tiek apstrādāti secīgi. Tāpēc vairākus paziņojumus vienlaikus var apstrādāt tikai tad, ja tos ģenerē vairāki trigeri.



Parametrs	Apraksts
Vienlaicīgās sesijas	Izvēlieties paralēlo brīdinājumu apstrādātāja sesiju skaitu mediju tipam: **Viena** - viena sesija **Neierobežoti** - neierobežots sesiju skaits **Pielāgots** - izvēlieties pielāgotu sesiju skaitu Neierobežotas/lielas vērtības nozīmē vairāk paralēlu sesiju un lielāku paziņojumu sūtīšanas kapacitāti. Neierobežotas/lielas vērtības ieteicams izmantot lielās vidēs, kur vienlaikus var būt nepieciešams nosūtīt daudz paziņojumu. Ja jānosūta vairāk paziņojumu, nekā ir pieejamu vienlaicīgo sesiju, atlikušie paziņojumi tiks ievietoti rindā; tie netiks zaudēti.
Mēģinājumi	Mēģinājumu skaits paziņojuma nosūtīšanai. Var norādīt līdz 100 mēģinājumiem; noklusējuma vērtība ir '3'. Ja ir norādīts '1', Zabbix nosūtīs paziņojumu tikai vienu reizi un neveiks atkārtotu mēģinājumu, ja nosūtīšana neizdosies.
Mēģinājumu intervāls	Biežums, ar kādu tiek mēģināts atkārtoti nosūtīt paziņojumu, ja nosūtīšana neizdevās, sekundēs (0-3600). Ja ir norādīts '0', Zabbix mēģinās vēlreiz nekavējoties. Tiek atbalstīti laika sufiksi, piemēram, 5s, 3m, 1h.

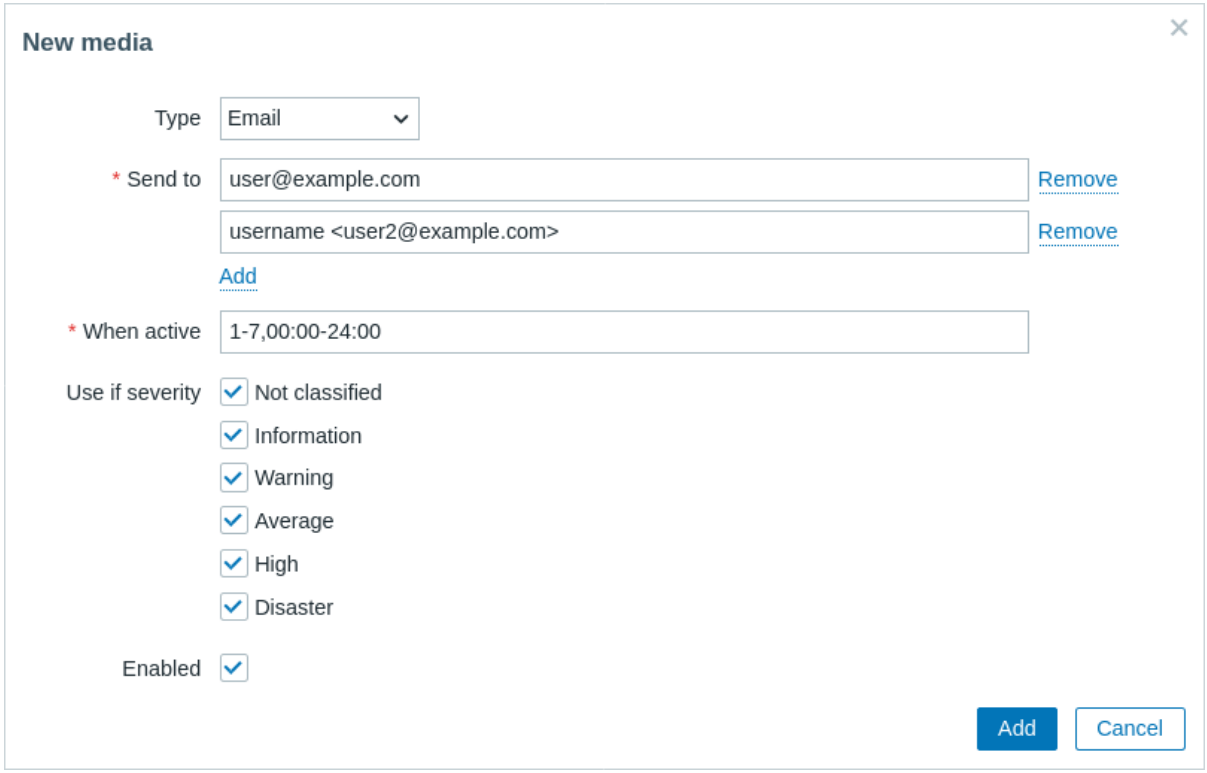
Lietotāja mediji

Kamēr mediju tipi nosaka, **kā** paziņojums tiks nosūtīts, lietotāja mediji nosaka, **kur** paziņojums ir jānosūta.

Lietotāja mediji (piemēram, e-pasta adrese, webhook lietotāja ID u.c.) ir jādefinē lietotāja profilā neatkarīgi no piegādes metodes. Darbība, kas sūta ziņojumus lietotājam *Admin*, izmantojot webhook *X*, nevarēs piegādāt ziņojumu, ja webhook *X* piegādes informācija nav definēta Admin lietotāja profilā.

Lai definētu lietotāja medijus:

- Dodieties uz *Users* > *Users* un atveriet lietotāja īpašību formu (vai dodieties uz *User settings* > *Notifications* savā lietotāja profilā)
- Cilnē Media noklikšķiniet uz *Add*



Parameter	Description
Type	Nolaižamajā sarakstā ir iekļauti iespējoto mediju tipu nosaukumi. Ņemiet vērā, ka, rediģējot mediju ar atspējotu mediju tipu, tips tiks attēlots sarkanā krāsā.
Send to	Ievadiet kontaktinformāciju, uz kuru jānosūta ziņojumi. E-pasta mediju tipam var pievienot vairākas adreses, noklikšķinot uz pogas Add zem adreses lauka. Šajā gadījumā paziņojumi tiks nosūtīti uz visām norādītajām adresēm. Adrešu piemērus skatiet mediju tipa email parametra *Email* aprakstā.
When active	Varat ierobežot laiku, kad ziņojumi tiek sūtīti, piemēram, norādīt tikai darba dienas (1-5,09:00-18:00). Ņemiet vērā, ka šis ierobežojums ir balstīts uz lietotāja laika joslu. Ja lietotāja laika josla tiek mainīta un atšķiras no sistēmas laika joslas, šo ierobežojumu var būt nepieciešams attiecīgi pielāgot, lai nepalaistu garām svarīgus ziņojumus. Formāta aprakstu skatiet lapā Laika perioda specifikācija. Tiek atbalstīti lietotāja makrosi.
Use if severity	Atzīmējiet to trigeru nopietnības līmeņu izvēles rūtiņas, par kuriem vēlaties saņemt paziņojumus. *Ņemiet vērā*, ka noklusējuma nopietnības līmenim ('Not classified') **jābūt** atzīmētam, ja vēlaties saņemt paziņojumus par ne-trigeru notikumiem. Pēc saglabāšanas atlasītie trigeru nopietnības līmeņi tiks attēloti atbilstošajās nopietnības krāsās, bet neatlasītie būs pelēkoti.
Status	Lietotāja medija statuss. **Enabled** - tiek izmantots. **Disabled** - netiek izmantots.

1 E-pasts

Pārskats

Lai konfigurētu e-pastu kā ziņojumu piegādes kanālu, jums jākonfigurē e-pasts kā medija tips un jāpiešķir lietotājiem konkrētas adreses.

> **Note:**
> Vairāki paziņojumi par vienu notikumu tiks grupēti vienā un tajā pašā e-pasta sarakstē.

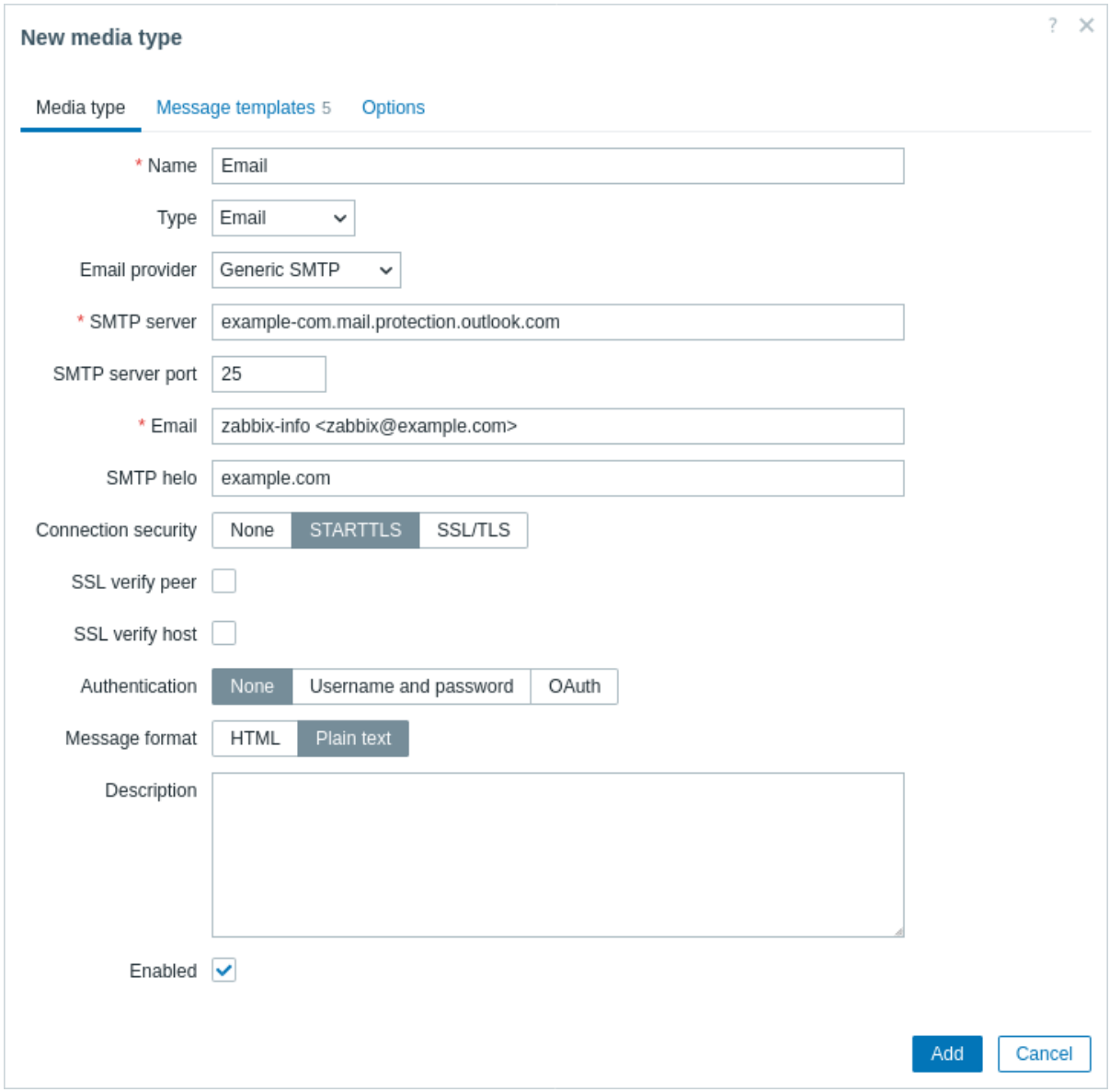
Konfigurācija

Lai konfigurētu e-pastu kā medija tipu:

1. Dodieties uz *Brīdinājumi* > *Mediju tipi*.

2. Noklikšķiniet uz *Izveidot medija tipu* (vai noklikšķiniet uz *E-pasts* iepriekš definēto mediju tipu sarakstā).

Cilnē **Medija tips** ir ietverti vispārīgie medija tipa atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Warning:
Parole tiek eksportēta atklātā tekstā, eksportējot e-pasta medija tipa konfigurāciju.

Tālāk norādītie parametri ir specifiski e-pasta medija tipam:

Parametrs	Apraksts
E-pasta pakalpojumu sniedzējs	Izvēlieties e-pasta pakalpojumu sniedzēju: *Generic SMTP*, *Gmail*, *Gmail relay*, *Office365* vai *Office365 relay*. Ja izvēlēsieties ar Gmail/Office365 saistītās opcijas, jums būs jānorāda tikai sūtītāja e-pasta adrese un parole; tādas opcijas kā *SMTP serveris*, *SMTP servera ports*, *SMTP helo* un *Savienojuma drošība* tiks automātiski aizpildītas no Zabbix puses. Skatiet arī: Automatizēti Gmail/Office365 mediju tipi.
SMTP serveris	Iestatiet SMTP serveri izejošo ziņojumu apstrādei. Šis lauks ir pieejams, ja kā e-pasta pakalpojumu sniedzējs ir izvēlēts *Generic SMTP*.
SMTP servera ports	Iestatiet SMTP servera portu izejošo ziņojumu apstrādei. Šis lauks ir pieejams, ja kā e-pasta pakalpojumu sniedzējs ir izvēlēts *Generic SMTP*.

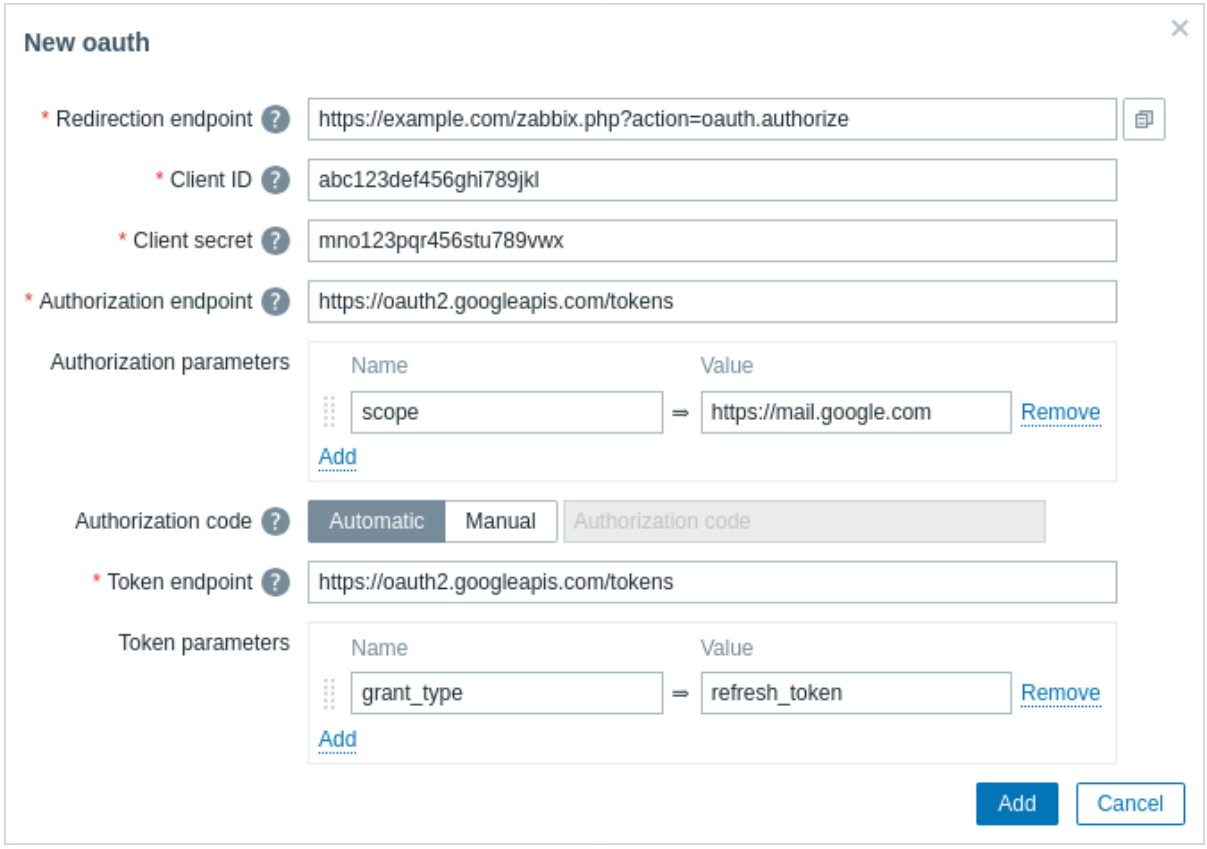
Parametrs	Apraksts
E-pasts	Šeit ievadītā adrese tiks izmantota kā nosūtīto ziņojumu **No** adrese. Tiek atbalstīta sūtītāja parādāmā vārda pievienošana (piemēram, ”Zabbix_info” no *Zabbix_info <zabbix@company.com>* iepriekš redzamajā ekrānuzņēmumā) kopā ar faktisko e-pasta adresi. Zabbix e-pastos parādāmajiem vārdiem ir daži ierobežojumi salīdzinājumā ar to, ko pieļauj RFC 5322, kā parādīts piemēros: Derīgi piemēri: *zabbix@company.com* (tikai e-pasta adrese, nav nepieciešams izmantot leņķiekavas) *Zabbix_info <zabbix@company.com>* (parādāmais vārds un e-pasta adrese leņķiekavās) *∑Ω-monitoring <zabbix@company.com>* (UTF-8 rakstzīmes parādāmajā vārdā) Nederīgi piemēri: *Zabbix HQ zabbix@company.com* (ir parādāmais vārds, bet e-pasta adrese nav leņķiekavās) *”Zabbix\@\<H(comment)Q\>” <zabbix@company.com>* (lai gan tas ir derīgs saskaņā ar RFC 5322, Zabbix e-pastos netiek atbalstīti quoted pairs un komentāri)
SMTP helo	Iestatiet korektu SMTP helo vērtību, parasti domēna nosaukumu. Ja lauks ir tukšs, tiks nosūtīts e-pasta domēna nosaukums (t.i., tas, kas seko aiz @ laukā *E-pasts*). Ja domēna nosaukumu nav iespējams iegūt, tiks reģistrēts atkļūdošanas līmeņa brīdinājums un servera resursdatora nosaukums tiks nosūtīts kā domēns HELO komandai. Šis lauks ir pieejams, ja kā e-pasta pakalpojumu sniedzējs ir izvēlēts *Generic SMTP*.
Savienojuma drošība	Izvēlieties savienojuma drošības līmeni: **Nav** - neizmantot opciju CURLOPT_USE_SSL **STARTTLS** - izmantot opciju CURLOPT_USE_SSL ar vērtību CURLUSESSL_ALL **SSL/TLS** - CURLOPT_USE_SSL izmantošana nav obligāta
SSL pārbaudīt vienādranga sertifikātu	Atzīmējiet izvēles rūtiņu, lai pārbaudītu SMTP servera SSL sertifikātu. Servera konfigurācijas direktīvas ”SSLCALocation” vērtība sertifikāta validācijai jāievieto CURLOPT_CAPATH. Tas iestata cURL opciju CURLOPT_SSL_VERIFYPEER.
SSL pārbaudīt hostu	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai SMTP servera sertifikāta lauks *Common Name* vai lauks *Subject Alternate Name* atbilst. Tas iestata cURL opciju CURLOPT_SSL_VERIFYHOST.
Autentifikācija	Izvēlieties autentifikācijas līmeni: **Nav** - netiek iestatītas nekādas cURL opcijas **Lietotājvārds un parole** - nozīmē ”AUTH=*”, atstājot autentifikācijas mehānisma izvēli cURL ziņā **OAuth** - OAuth autentifikācija OAuth autentifikācija netiek atbalstīta e-pasta pakalpojumu sniedzējam *Office365 relay*.
Lietotājvārds	Lietotājvārds, ko izmantot autentifikācijai. Tas iestata CURLOPT_USERNAME vērtību. Tiek atbalstīti lietotāja makrosi.
Parole	Parole, ko izmantot autentifikācijai. Tas iestata CURLOPT_PASSWORD vērtību. Tiek atbalstīti lietotāja makrosi. Šeit ievadītā vērtība tiek eksportēta bez izmaiņām, eksportējot medija tipa konfigurāciju. Lai izvairītos no sensitīvu akreditācijas datu atklāšanas eksportētajos failos, literālas paroles vietā izmantojiet lietotāja makrosu (piemēram, `{$EMAIL_SMTP_PASSWORD}`) — tiks eksportēta makrosa atsauce, savukārt slepenā vērtība ir jā(ie)konfigurē mērķa sistēmā.
OAuth tokeni	Noklikšķiniet uz *Konfigurēt*, lai jaunā logā iestatītu parametrus OAuth tokenu iegūšanai. Šis lauks ir pieejams tikai tad, ja laukā *Autentifikācija* ir izvēlēts ”OAuth”.
Ziņojuma formāts	Izvēlieties ziņojuma formātu: **HTML** - sūtīt kā HTML **Vienkāršs teksts** - sūtīt kā vienkāršu tekstu

Attention:
Lai iespējotu SMTP autentifikāciju, Zabbix serverim jābūt kompilētam ar `--with-libcurl` kompilācijas opciju, kurai nepieciešams libcurl (versijas informāciju skatiet serveris prasībās).

Skatiet arī kopīgos mediju tipu parametrus, lai iegūtu sīkāku informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu.

OAuth tokeni

Lai iegūtu OAuth tokenus, ir nepieciešami šādi parametri:



Parametrs	Apraksts
Pāradresācijas galapunkts	Ievadiet Zabbix lietotāja saskarnes URL, uz kuru OAuth pakalpojums pāradresēs atpakaļ OAuth autorizāciju (izmantojot formātu `https://<zabbix-frontend-url>/zabbix.php?action=oauth.authorize`). Jaunam e-pasta medija tipam tas tiek iestatīts automātiski, ja ir definēts lietotāja saskarnes URL.
Klienta ID	Ievadiet unikālu klienta lietotnes identifikatoru, kas reģistrēta OAuth autorizācijas serverī.
Klienta slepenā atslēga	Ievadiet klienta lietotnes privāto slepeno atslēgu, kas reģistrēta OAuth autorizācijas serverī.
Autorizācijas galapunkts	Ievadiet OAuth autorizācijas servera URL lietotāja autorizācijas pieprasīšanai.
Autorizācijas parametri	Ievadiet parametrus autorizācijas galapunktam.
Autorizācijas kods	Ievadiet autorizācijas kodu: **Automātiski** — kods tiks iegūts automātiski, izmantojot pāradresācijas lapu **Manuāli** — ievadiet kodu manuāli, ja automātiskā iegūšana neizdodas
Tokena galapunkts	Ievadiet OAuth autorizācijas servera URL, lai apmainītu autorizācijas kodu pret piekļuves un atsvaidzināšanas tokeniem.
Tokena parametri	Ievadiet parametrus piekļuves tokenam.

Note:
OAuth tokenu iegūšanai ir automatizētas iespējas e-pasta pakalpojumu sniedzējiem **Gmail**, **Gmail relay** un **Office365**. Nepieciešams norādīt tikai parametru *Pāradresācijas galapunkts*, *Klienta ID* un *Klienta slepenā atslēga* vērtības. Zabbix automātiski aizpildīs pārējās nepieciešamās vērtības (skatiet OAuth URL noklusējuma vērtības pēc pakalpojumu sniedzēja).

OAuth piekļuves un atsvaidzināšanas marķiera iegūšana

Pēc OAuth parametru iesniegšanas:

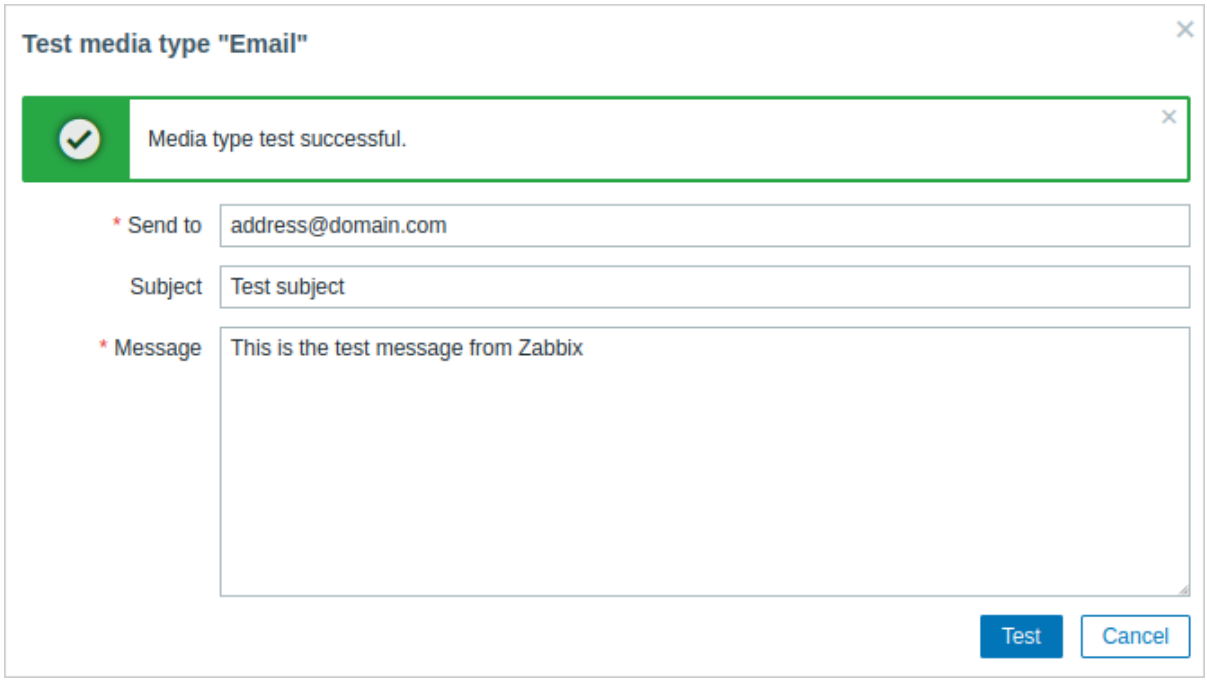
1. Tiek atvērts pārlūkprogrammas uznirstošais logs, novirzot lietotāju uz *autorizācijas galapunktu*.
2. Lietotājs autorizē Zabbix OAuth servisā.
3. OAuth serviss novirza lietotāju uz Zabbix lietotāja saskarnes darbību `oauth.authorize` ar autorizācijas kodu un tvēruma vērtību.
4. Atbildot uz to, Zabbix veiks pieprasījumu uz *marķiera galapunktu*, lai apmainītu autorizācijas kodu pret piekļuves un atsvaidzināšanas marķieriem.

Testēšana

Lai pārbaudītu, vai konfigurētais e-pasta medija tips darbojas pareizi:

1. Atrodiet attiecīgo e-pastu mediju tipu sarakstā.
2. Saraksta pēdējā kolonnā noklikšķiniet uz *Test* (tiks atvērts testēšanas logs).
3. Ievadiet adresāta adresi laukā *Send to*, ziņojuma tekstu un, pēc izvēles, tēmu.
4. Noklikšķiniet uz *Test*, lai nosūtītu testa ziņojumu.

Ziņojums par testa veiksmi vai neveiksmi tiks parādīts tajā pašā logā:



Lietotāja mediji

Kad e-pasta medija tips ir konfigurēts, dodieties uz sadaļu *Users* > *Users* un rediģējiet lietotāja profilu, lai piešķirtu lietotājam e-pasta mediju. Darbības lietotāja mediju iestatīšanai, kas ir kopīgas visiem mediju tipiem, ir aprakstītas lapā Media types.

1 Automatizēti Gmail/Office365 mediju tipi

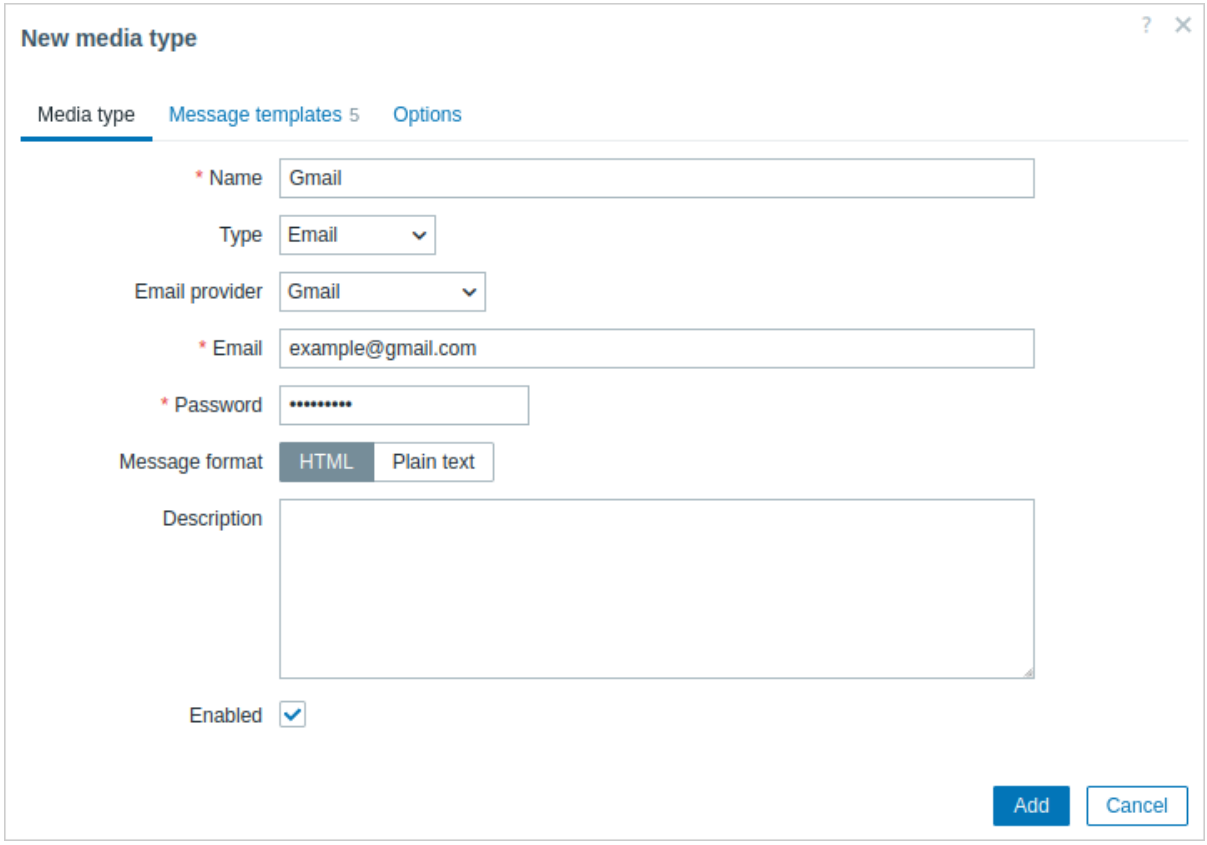
Pārskats

Gmail vai Office365 lietotāji var gūt labumu no automatizētām funkcijām mediju tipa konfigurācijā.

Konfigurācija

Lauks *E-pasta pakalpojumu sniedzējs* e-pasta medija tipa konfigurācijā ļauj atlasīt iepriekš konfigurētas opcijas pakalpojumiem Gmail un Office 365.

Atlasot ar Gmail/Office365 saistītās opcijas, lai izveidotu darbojošos medija tipu, ir jānorāda tikai sūtītāja e-pasta adrese/parole.



Tiklīdz ir norādīta e-pasta adrese/parole, Zabbix varēs automātiski aizpildīt visus nepieciešamos iestatījumus Gmail/Office365 medija tipiem ar faktiskajām/ieteicamajām vērtībām, t. i., *SMTP serveris*, *SMTP servera ports*, *SMTP helo* un *Savienojuma drošība*. Šīs automatizācijas dēļ šie lauki pat netiek rādīti, tomēr SMTP servera un e-pasta informāciju ir iespējams redzēt medija tipu sarakstā (skatiet kolonnu *Detaļas*).

Ņemiet vērā arī to, ka:

- Releja opcijām parole nav nepieciešama.
- Office365 relejam norādītās e-pasta adreses domēna nosaukums tiks izmantots, lai dinamiski aizpildītu SMTP serveri (t. i., aizstātu "example.com" vērtībā `example-com.mail.protection.outlook.com` ar reālo vērtību).

OAuth marķieri

OAuth marķieru iegūšanai ir automatizētas iespējas *Gmail*, *Gmail relay* un *Office365* e-pasta pakalpojumu sniedzējiem.

OAuth piekļuves un atsvaidzināšanas marķiera iegūšanai nepieciešams norādīt tikai ierobežotu parametru kopu — *Redirection endpoint*, *Client ID* un *Client secret*. Zabbix automātiski aizpildīs pārējās nepieciešamās vērtības (skatiet OAuth URL noklusējuma vērtības pēc pakalpojumu sniedzēja).

Ņemiet vērā, ka ir iespējams izmantot arī Generic SMTP formu, lai iestatītu OAuth autorizāciju šiem pakalpojumu sniedzējiem.

Note:
Office365 vidē SmtpClientAuthentication jābūt iespējotam vai nu katram lietotājam/koplietojamai pastkastei atsevišķi, vai visam tenant kopumā (pēc noklusējuma iestatījumiem tas nav iespējots).

OAuth URL noklusējuma vērtības pēc nodrošinātāja

Šajā tabulā ir uzskaitītas Zabbix izmantotās OAuth URL noklusējuma vērtības un parametri katram nodrošinātājam.

Parametrs	Gmail	Office365	Vispārīgs SMTP
Autorizācijas gala-punkts	`https://accounts.google.com/o/oauth2/auth`	`https://login.microsoftonline.com/common/oauth2/v2.0/authorize`	Nav noklusējuma vērtības
scope	`https://mail.google.com`	`https://outlook.office.com/SMTP.Send offline_access`	Nav noklusējuma vērtības
access_type	`offline`	Netiek izmantots	Nav noklusējuma vērtības
prompt	`consent`	Netiek izmantots	Nav noklusējuma vērtības
redirect_uri	Lauka *Redirection endpoint* vērtība tiek automātiski pievienota ar kodu. Tā nav norādīta *Authorization endpoint* parametros.		
state	Unikāla jaucējvērtība, lai identificētu datu nesēju, kas tiek atjaunināts ar OAuth pilnvaru, tiek automātiski pievienota ar kodu. Tā nav norādīta *Authorization endpoint* parametros.		
Pilnvaras gala-punkts	`https://oauth2.googleapis.com/token`	`https://login.microsoftonline.com/common/oauth2/v2.0/token`	Nav noklusējuma vērtības
grant_type	`authorization_code`	`authorization_code`	Nav noklusējuma vērtības
redirect_uri	Lauka *Redirection endpoint* vērtība tiek automātiski pievienota ar kodu. Tā nav norādīta *Token endpoint* parametros.		
state	Unikāla jaucējvērtība, lai identificētu datu nesēju, kas tiek atjaunināts ar OAuth pilnvaru, tiek automātiski pievienota ar kodu. Tā nav norādīta *Token endpoint* parametros.		

Office365 risinājums personīgajiem kontiem

Office365 personīgie konti neatbalsta vaicājuma virknes pāradresācijas URL.

Lai apietu šo problēmu, Apache tīmekļa servera administrators var pievienot šādu pārrakstīšanas kārtulu `.htaccess` failam:

```
RewriteEngine On
RewriteRule ^/oauth\.authorize$ /zabbix.php?action=oauth.authorize [QSA,L,PT]
```

Tas ļauj izmantot vienkāršotu pāradresācijas URL, piemēram:

```
http://server-name/zabbix/oauth.authorize
```

URL ar vaicājuma parametriem vietā, tādējādi nodrošinot saderību ar Office365 personīgajiem kontiem.

2 SMS

Pārskats

Zabbix atbalsta SMS ziņojumu sūtīšanu, izmantojot seriālo GSM modemu, kas pievienots Zabbix serveris seriālajam portam.

Pārliecinieties, ka:

- Seriālās ierīces ātrums (parasti /dev/ttyS0 Linux vidē) atbilst GSM modema ātrumam. Zabbix neiestata seriālā savienojuma ātrumu. Tas izmanto noklusējuma iestatījumus.
- Lietotājam 'zabbix' ir lasīšanas/rakstīšanas piekļuve seriālajai ierīcei. Izpildiet komandu ls –l /dev/ttyS0, lai skatītu pašreizējās seriālās ierīces atļaujas.

- GSM modemam ir ievadīts PIN kods un tas tiek saglabāts pēc barošanas atiestatīšanas. Alternatīvi varat atspējot PIN kodu SIM kartei. PIN kodu var ievadīt, termināļa programmatūrā izpildot komandu AT+CPIN="NNNN" (NNNN ir jūsu PIN kods, pēdiņām ir jābūt iekļautām), piemēram, Unix minicom vai Windows HyperTerminal.

Zabbix ir testēts ar šādiem GSM modēmiem:

- Siemens MC35
- Teltonika ModemCOM/G10

Lai konfigurētu SMS kā ziņojumu piegādes kanālu, jums ir arī jākonfigurē SMS kā medija tips un lietotājiem jānorāda attiecīgie tālruņu numuri.

Konfigurācija

Lai konfigurētu SMS kā medija tipu:

- Dodieties uz *Brīdinājumi* → *Mediju tipi*
- Noklikšķiniet uz *Izveidot medija tipu* (vai noklikšķiniet uz *SMS* iepriekš definēto mediju tipu sarakstā).

Tālāk norādītie parametri ir specifiski SMS medija tipam:

Parametrs	Apraksts
GSM modems	Iestatiet GSM modema seriālās ierīces nosaukumu. Šeit ievadītais ceļš tiks validēts attiecībā pret serveris parametru SMSDevices (ja tas ir norādīts).

Skatiet kopīgos medija tipa parametrus, lai iegūtu detalizētu informāciju par to, kā konfigurēt noklusējuma ziņojumus un brīdinājumu apstrādes opcijas. Ņemiet vērā, ka SMS paziņojumu sūtīšanas paralēla apstrāde nav iespējama.

Lietotāja mediji

Kad SMS medija tips ir konfigurēts, dodieties uz sadaļu *Lietotāji* → *Lietotāji* un rediģējiet lietotāja profilu, lai piešķirtu lietotājam SMS mediju. Darbības lietotāja mediju iestatīšanai, kas ir kopīgas visiem mediju tipiem, ir aprakstītas lapā Mediju tipi.

3 Pielāgoti brīdinājumu skripti

Pārskats

Ja jūs neapmierina esošie brīdinājumu sūtīšanas mediju tipi, ir alternatīvs veids, kā to izdarīt. Jūs varat izveidot skriptu, kas apstrādās paziņojumus jums vēlamajā veidā.

Pielāgotie brīdinājumu skripti tiek izpildīti Zabbix serveris. Šiem skriptiem jāatrodas direktorijā, kas norādīta servera konfigurācijas faila parametrā AlertScriptsPath.

Šeit ir pielāgota brīdinājumu skripta piemērs:

```
#####!/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 pārbauda izpildīto komandu un skriptu izejas kodu. Jebkurš izejas kods, kas atšķiras no **0**, tiek uzskatīts par komandas izpildes kļūdu. Šādos gadījumos Zabbix mēģinās atkārtot neveiksmīgo izpildi.

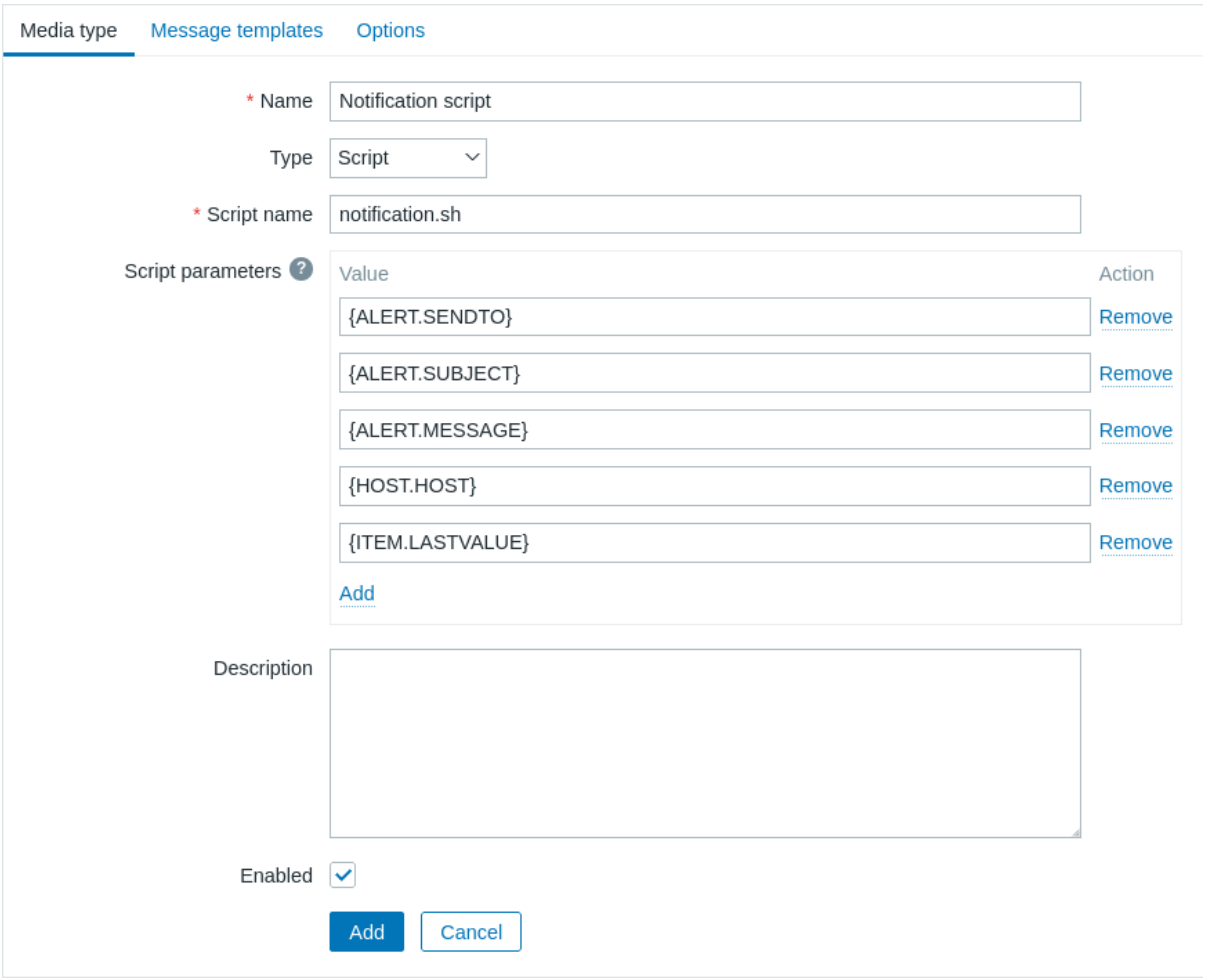
Skriptam vides mainīgie netiek saglabāti vai izveidoti, tāpēc tie ir jāapstrādā tieši.

Konfigurācija

Lai konfigurētu pielāgotus brīdinājumu skriptus kā datu nesēja tipu:

1. Dodieties uz *Brīdinājumi → Datu nesēju tipi*.
2. Noklikšķiniet uz *Izveidot datu nesēja tipu*.

Cilnē **Datu nesēja tips** ir ietverti vispārīgie datu nesēja tipa atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Tālāk norādītie parametri ir specifiski skripta datu nesēja tipam:

Parameter	Description
Skripta nosaukums	Ievadiet skripta faila nosaukumu (piemēram, notification.sh), kas atrodas direktorijā, kurš norādīts serveris konfigurācijas parametra AlertScriptsPath vērtībā.
Skripta parametri	Pievienojiet neobligātus skripta parametrus, kas tiks nodoti skriptam kā komandrindas argumenti to definēšanas secībā. Skripta parametri atbalsta makrosus {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} un visus makrosus, kas tiek atbalstīti paziņojumos, kā arī lietotāja makrosus.

Sīkāku informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu skatiet sadaļā kopīgie datu nesēju tipu parametri.

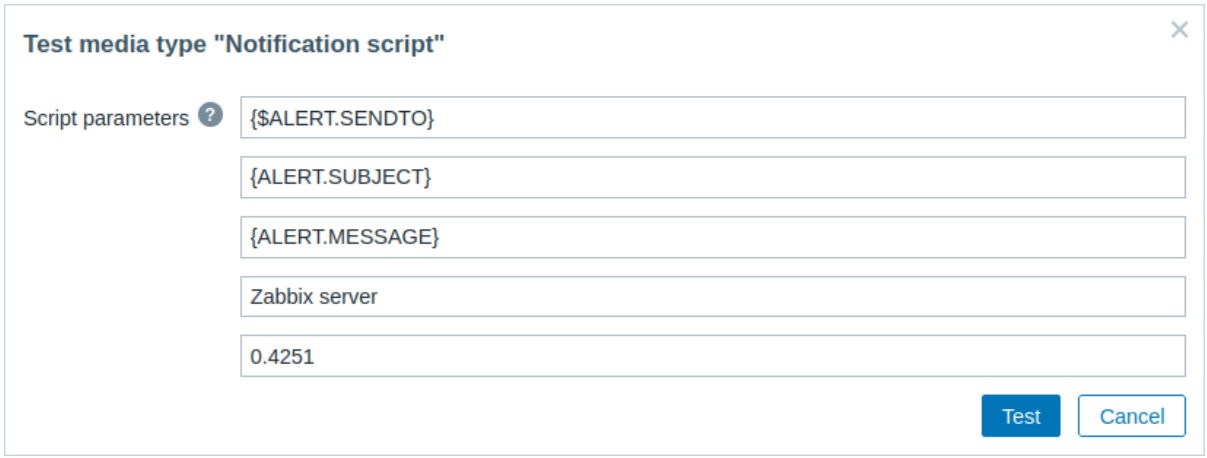
Warning:
Pat ja brīdinājumu skripts neizmanto noklusējuma ziņojumus, šī datu nesēja tipa izmantotajiem operāciju tipiem ziņojumu veidnes joprojām ir jādefinē. Pretējā gadījumā paziņojums netiks nosūtīts.

Attention:
Ja ir konfigurēts vairāk nekā viens skripta datu nesēja tips, šos skriptus alerter procesi var apstrādāt paralēli. Kopējo alerter procesu skaitu ierobežo serveris konfigurācijas faila parametrs `StartAlerters`.

Testēšana

Lai pārbaudītu konfigurētu skripta medija tipu:

1. Atrodiet attiecīgo skriptu mediju tipu sarakstā.
2. Noklikšķiniet uz *Test* saraksta pēdējā kolonnā; uznirstošajā logā tiks atvērta testēšanas forma. Testēšanas formā būs tāds pats parametru skaits, kāds ir konfigurēts skripta medija tipam.
3. Ja nepieciešams, rediģējiet skripta parametru vērtības. Rediģēšana ietekmē tikai testēšanas procesu; faktiskās vērtības netiks mainītas.
4. Noklikšķiniet uz *Test*.



Note:
Pārbaudot konfigurētu skripta medija tipu, {ALERT.SENDTO}, {ALERT.SUBJECT}, {ALERT.MESSAGE} un lietotāja makrosi tiks aizvietoti ar to vērtībām, taču makrosi, kas ir saistīti ar notikumiem (piemēram, {HOST.HOST}, {ITEM.LASTVALUE} u.c.), netiks aizvietoti, jo testēšanas laikā nav saistīta notikuma, no kura iegūt detalizētu informāciju. Ņemiet vērā, ka arī makrosi makrosos {ALERT.SUBJECT} un {ALERT.MESSAGE} netiks aizvietoti. Piemēram, ja {ALERT.SUBJECT} vērtība ir veidota kā "Problem: {EVENT.NAME}", tad makross {EVENT.NAME} netiks aizvietots.

Lietotāja medijs

Kad medija tips ir konfigurēts, dodieties uz sadaļu *Lietotāji* → *Lietotāji* un rediģējiet lietotāja profilu, piešķirot lietotājam šo medija tipu. Darbības lietotāja medija iestatīšanai, kas ir kopīgas visiem mediju tipiem, ir aprakstītas lapā Mediju tipi.

Ņemiet vērā, ka, definējot lietotāja mediju, lauks *Sūtīt uz* nedrīkst būt tukšs. Ja šis lauks netiek izmantots brīdinājuma skriptā, ievadiet jebkuru atbalstīto rakstzīmju kombināciju, lai apietu validācijas prasības.

4 Webhook

Pārskats

Webhook medija tips ir noderīgs HTTP izsaukumu veikšanai, izmantojot pielāgotu JavaScript kodu, lai nodrošinātu vienkāršu integrāciju ar ārēju programmatūru, piemēram, palīdzības dienesta sistēmām, tērzēšanas risinājumiem vai ziņapmaiņas lietotnēm. Jūs varat izvēlēties importēt Zabbix nodrošinātu integrāciju vai izveidot pielāgotu integrāciju no nulles.

Integrācijas

Ir pieejamas šādas integrācijas, kas ļauj izmantot iepriekš definētus webhook mediju tipus, lai nosūtītu Zabbix paziņojumus uz:

- 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:
Papildus šeit uzskaitītajiem pakalpojumiem Zabbix var integrēt ar **Spiceworks** (webhook nav nepieciešams). Lai pārveidotu Zabbix paziņojumus par Spiceworks biļetēm, izveidojiet e-pasta mediju tipu un norādītā Zabbix lietotāja profila iestatījumos ievadiet Spiceworks palīdzības dienesta e-pasta adresi (piemēram, help@zabbix.on.spiceworks.com).

Konfigurācija

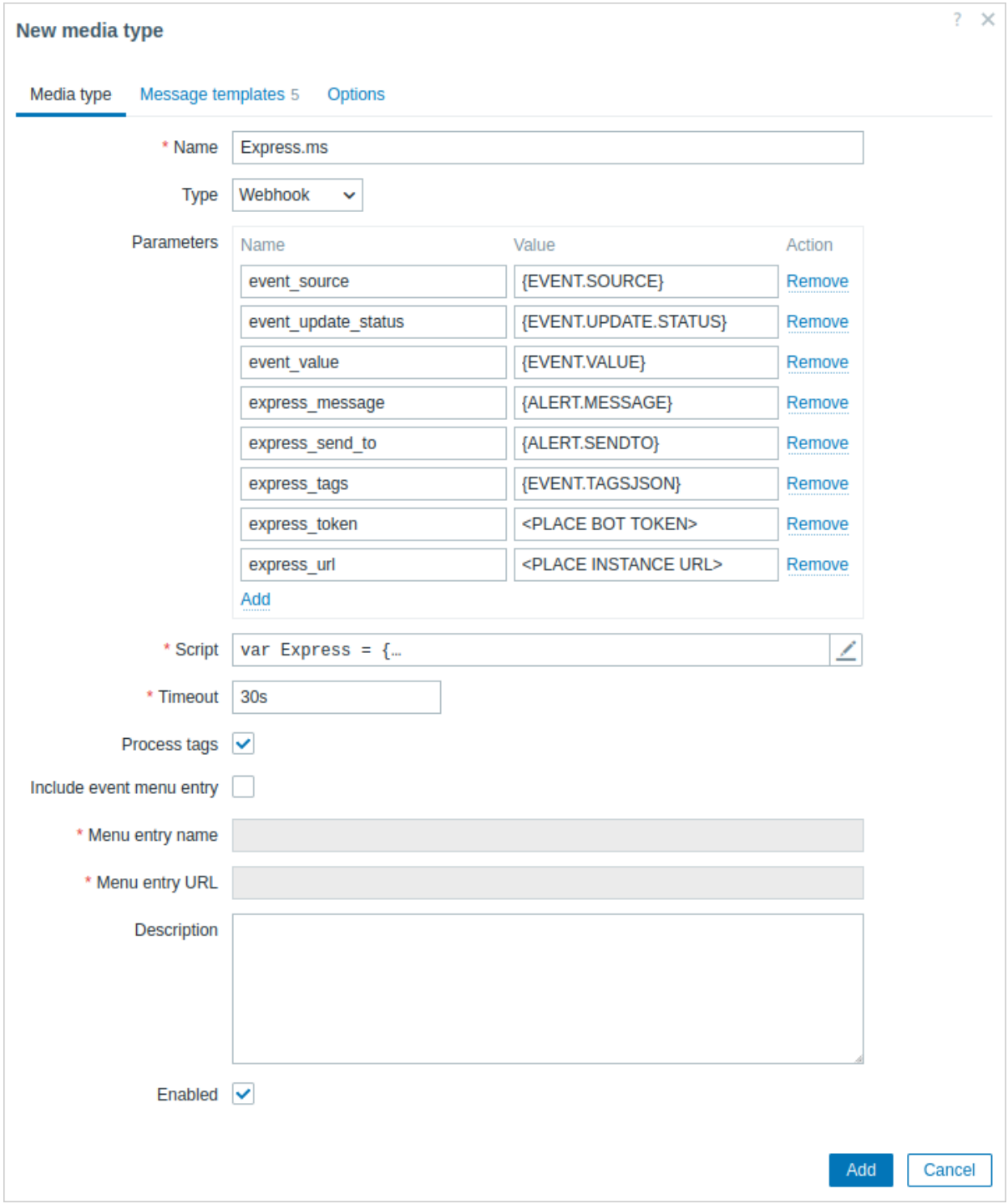
Lai sāktu izmantot webhook integrāciju:

1. Atrodiet nepieciešamo .yaml failu lejupielādētās Zabbix versijas `templates/media` direktorijā vai lejupielādējiet to no Zabbix git repozitorija.
2. Importējiet failu savā Zabbix instalācijā. Webhook parādīsies mediju tipu sarakstā.
3. Konfigurējiet webhook atbilstoši norādījumiem *Readme.md* failā (varat noklikšķināt uz webhook nosaukuma augstāk, lai ātri piekļūtu *Readme.md*).

Lai izveidotu pielāgotu webhook no nulles:

1. Dodieties uz *Brīdinājumi* > *Mediju tipi*.
2. Noklikšķiniet uz *Izveidot mediju tipu*.

Cilnē **Mediju tips** ir dažādi šim mediju tipam specifiski atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Webhook mediju tipam ir specifiski šādi parametri:

Parametrs	Apraksts
Parameters	Norādiet webhook mainīgos kā atribūta un vērtības pārus. Iepriekš konfigurētiem webhook parametru saraksts atšķiras atkarībā no pakalpojuma. Parametru aprakstu skatiet webhook *Readme.md* failā. Jauniem webhook pēc noklusējuma ir iekļauti vairāki bieži izmantoti mainīgie (URL:<empty>, HTTPProxy:<empty>, To:{ALERT.SENDTO}, Subject:{ALERT.SUBJECT}, Message:{ALERT.MESSAGE}); varat tos paturēt vai noņemt. Webhook parametri atbalsta lietotāja makrosus, visus makrosus, kas tiek atbalstīti problēmu paziņojumos, kā arī papildus makrosus {ALERT.SENDTO}, {ALERT.SUBJECT} un {ALERT.MESSAGE}. Ja norādāt HTTP starpniekserveri, lauks atbalsta to pašu funkcionalitāti kā vienuma konfigurācijas laukā HTTP proxy. Starpniekservera virknei var pievienot prefiksu `[scheme]://`, lai norādītu, kāda veida starpniekserveris tiek izmantots (piemēram, https, socks4, socks5; skatiet dokumentāciju).
Script	Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam. Šis kods izpildīs webhook darbību. Skripts ir funkcijas kods, kas pieņem parametru-vērtību pārus. Vērtības jāpārveido par JSON objektiem, izmantojot JSON.parse() metodi, piemēram: `var params = JSON.parse(value);`. Kodam ir piekļuve visiem parametriem; tas var izpildīt HTTP GET, POST, PUT un DELETE pieprasījumus, atbalsta arī papildu metodes, piemēram, CONNECT, PATCH, HEAD, OPTIONS un TRACE, un nodrošina kontroli pār HTTP galvenēm un pieprasījuma pamattekstu. Skriptā obligāti jābūt return operatoram, pretējā gadījumā tas nebūs derīgs. Tas var atgriezt OK statusu kopā ar neobligātu birku un birku vērtību sarakstu (skatiet opciju *Process tags*) vai kļūdas virkni. Atjaunošanas notikumus (neatkarīgi no tā, vai tie ģenerēti automātiski vai manuālas aizvēršanas rezultātā) izveido serveris, un tie ietver atrisinātās notikumu birkas (tostarp no veidnēm, hostiem un trigeriem mantotās birkas). Webhook skripti tiek izpildīti pēc brīdinājuma izveides; tādēļ webhook skripta atgrieztās birkas tiek pievienotas tikai pēc sākotnējās brīdinājuma izveides un nebūs pieejamas sākotnējā problēmas ziņojuma vai tūlītējā atjaunošanas ziņojuma makrosos {EVENT.TAGS} un {EVENT.RECOVERY.TAGS}. *Piezīme*: Ieteicams izmantot lokālos mainīgos (piemēram, `var local = 1`) globālo vietā (piemēram, `global = 1`), lai nodrošinātu, ka katrs skripts darbojas ar saviem datiem, un izvairītos no kolīzijām vienlaicīgu izsaukumu laikā (skatiet zināmās problēmas). Skatiet arī: Webhook izstrādes vadlīnijas, Webhook skriptu piemēri, Papildu JavaScript objekti.
Timeout	JavaScript izpildes noildze (1-60s, noklusējums 30s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m.
Process tags	Atzīmējiet izvēles rūtiņu, lai apstrādātu atgriezto JSON īpašību vērtības kā birkas. Šīs birkas tiek pievienotas visām esošajām problēmu birkām. Ņemiet vērā, ka, izmantojot webhook birkas, webhook jāatgriež JSON objekts, kurā ir vismaz tukšs tags objekts: `var result = {tags: {}};` Atgriežamo birku piemēri: *jira-id:prod-1234*, *responsible:John Smith*, *processed:<no value>*
Include event menu entry	Atzīmējiet izvēles rūtiņu, lai notikuma izvēlnē iekļautu ierakstu ar saiti uz izveidoto ārējo biļeti. Ieraksts tiks iekļauts katram webhook, kas ir iespējots un kam šī izvēles rūtiņa ir atzīmēta. Ņemiet vērā, ka, ja parametros *Menu entry name* un *Menu entry URL* ir ietverti makrosi {EVENT.TAGS.<tag name>}, ieraksts tiks iekļauts tikai tad, ja šos makrosus varēs atrisināt (tas ir, notikumam šīs birkas būs definētas). Ja šī opcija ir atzīmēta, webhook nevajadzētu izmantot paziņojumu sūtīšanai dažādiem lietotājiem (apsveriet iespēju izveidot atsevišķu lietotāju) un to nevajadzētu izmantot vairākās brīdinājumu darbībās vienam problēmas notikumam.
Menu entry name	Norādiet izvēlnes ieraksta nosaukumu. Tiek atbalstīts makross {EVENT.TAGS.<tag name>}. Šis lauks ir obligāts tikai tad, ja ir atzīmēta opcija *Include event menu entry*.
Menu entry URL	Norādiet izvēlnes ieraksta pamatā esošo URL. Tiek atbalstīts makross {EVENT.TAGS.<tag name>}. Šis lauks ir obligāts tikai tad, ja ir atzīmēta opcija *Include event menu entry*.

Sīkāku informāciju par noklusējuma ziņojumu un brīdinājumu apstrādes opciju konfigurēšanu skatiet sadaļā kopīgie mediju tipu parametri.

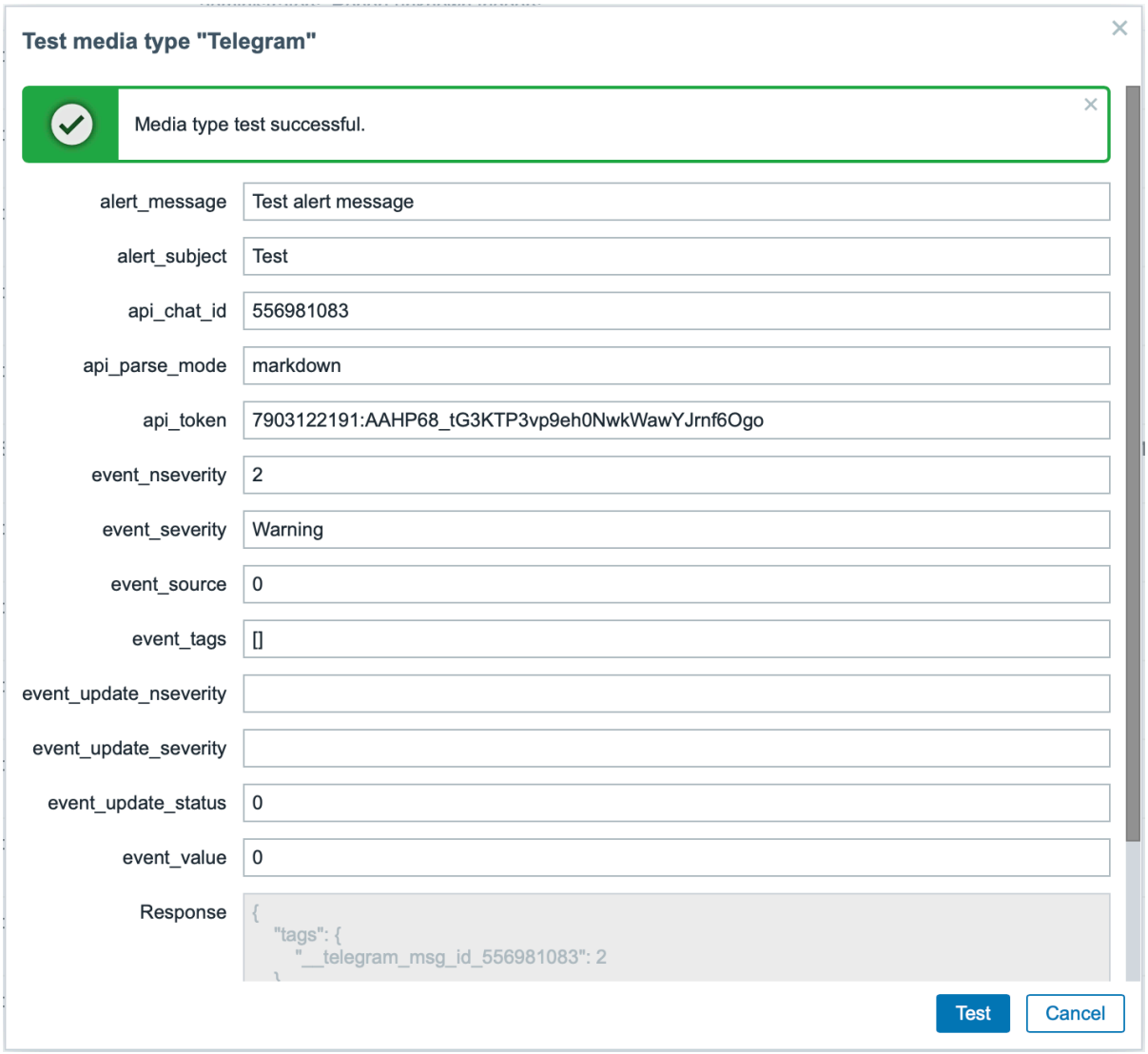
Warning:
Pat ja webhook neizmanto noklusējuma ziņojumus, šī webhook izmantotajiem darbību tipiem ziņojumu veidnēm joprojām jābūt definētām.

Testēšana

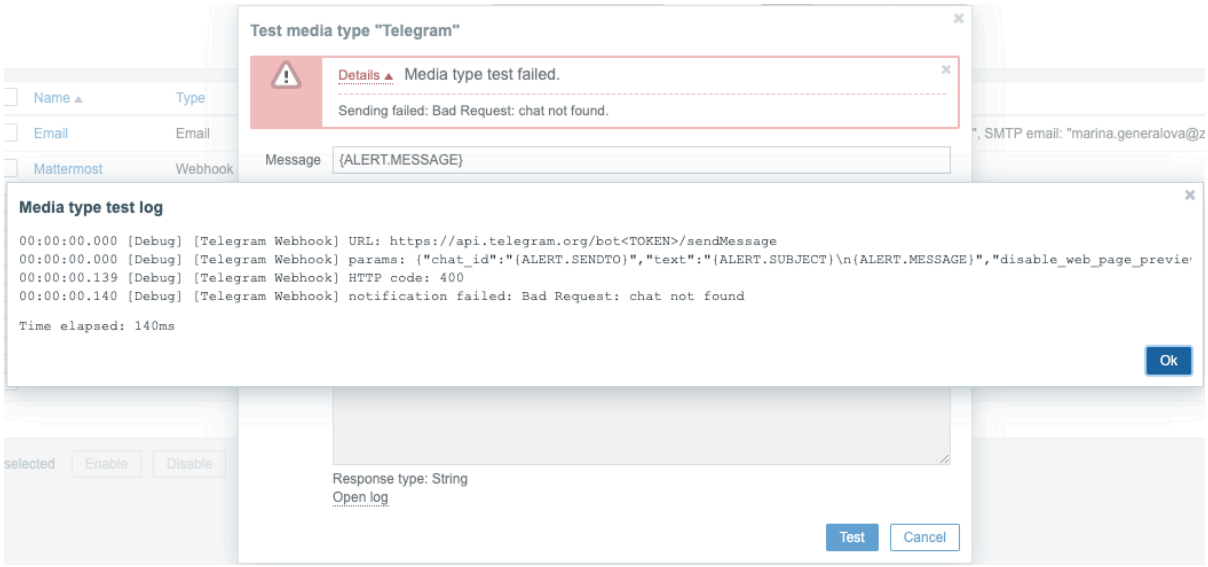
Lai pārbaudītu konfigurētu webhook medija tipu:

1. Atrodiet attiecīgo webhook medija tipu sarakstā.
2. Saraksta pēdējā kolonnā noklikšķiniet uz *Test* (tiks atvērts testēšanas logs).
3. Pēc nepieciešamības rediģējiet webhook parametru vērtības. Aizstājiet makrosus ar piemēra vērtībām; pretējā gadījumā makrosi netiks atrisināti, un tests neizdosies.
4. Noklikšķiniet uz *Test*.

Vērtību aizstāšana vai dzēšana testēšanas logā ietekmē tikai testēšanas procedūru; faktiskās webhook atribūtu vērtības paliks nemainītas.



Lai skatītu medija tipa testa žurnāla ierakstus, neatstājot testēšanas logu, noklikšķiniet uz *Open log* (tiks atvērts jauns uznirstošais logs).



Ja webhook tests ir veiksmīgs:

- tiek parādīts ziņojums *"Media type test successful."*.
- servera atbilde tiek parādīta pelēkajā laukā *Response*.
- zem lauka *Response* tiek norādīts atbildes tips (JSON vai String).

Ja webhook tests neizdodas:

- tiek parādīts ziņojums *"Media type test failed."*, kam seko papildu informācija par kļūmi.

Lietotāja medijs

Kad medija tips ir konfigurēts, dodieties uz sadaļu *Users* > *Users* un piešķiriet webhook mediju esošam lietotājam vai izveidojiet jaunu lietotāju, kas pārstāvēs webhook. Darbības lietotāja medija iestatīšanai esošam lietotājam, kas ir kopīgas visiem mediju tipiem, ir aprakstītas lapā Media types.

Ja webhook izmanto birkas, lai saglabātu biļetes\ziņojuma ID, nepiešķiriet vienu un to pašu webhook kā mediju dažādiem lietotājiem, jo tas var izraisīt webhook kļūdas (attiecas uz lielāko daļu webhook, kas izmanto opciju *Include event menu entry*). Šādā gadījumā ieteicamā prakse ir izveidot atsevišķu lietotāju, kas pārstāvēs webhook:

1. Pēc webhook medija tipa konfigurēšanas dodieties uz sadaļu *Users* > *Users* un izveidojiet atsevišķu Zabbix lietotāju, kas pārstāvēs webhook — piemēram, ar lietotājvārdu *Slack* Slack webhook vajadzībām. Visus iestatījumus, izņemot mediju, var atstāt pēc noklusējuma, jo šis lietotājs nepieslēgsies Zabbix.
2. Lietotāja profilā atveriet cilni *Media* un pievienojiet webhook, norādot nepieciešamo kontaktinformāciju. Ja webhook neizmanto lauku *Send to*, ievadiet jebkuru atbalstīto rakstzīmju kombināciju, lai apietu validācijas prasības.
3. Piešķiriet šim lietotājam vismaz lasīšanas atļaujas visiem hostiem, kuriem tam jānosūta brīdinājumi.

Konfigurējot brīdinājumu darbību, laukā *Send to users* sadaļā Operation details pievienojiet šo lietotāju — tas norādīs Zabbix izmantot webhook paziņojumiem no šīs darbības.

Brīdinājumu darbību konfigurēšana

Darbības nosaka, kuri paziņojumi jānosūta, izmantojot webhook. Darbību konfigurēšanas soļi, kuros tiek izmantoti webhook, ir tādi paši kā visiem citiem mediju tipiem, izņemot šādus gadījumus:

- Ja webhook izmanto webhook tagus, lai saglabātu biļetes\ziņojuma ID un apstrādātu atjaunināšanas\atrisināšanas darbības, izvairieties no viena un tā paša webhook izmantošanas vairākās brīdinājumu darbībās vienam problēmas notikumam. Ja {EVENT.TAGS.<tag name>} pastāv un tiek atjaunināts webhook, tā iegūtā vērtība būs nedefinēta. Lai no tā izvairītos, izmantojiet webhook jaunu taga nosaukumu atjaunināto vērtību glabāšanai. Tas attiecas uz Jira, Jira Service Desk, Mattermost, Opsgenie, OTRS, Redmine, ServiceNow, Slack, Zammad un Zendesk webhook, ko nodrošina Zabbix, kā arī uz lielāko daļu webhook, kas izmanto opciju *Include event menu entry*. Tomēr ņemiet vērā, ka vienu webhook var izmantot vairākās vienas un tās pašas darbības operācijās vai eskalācijas soļos, kā arī dažādās darbībās, kuras netiks aktivizētas ar vienu un to pašu problēmas notikumu atšķirīgu nosacījumu dēļ.
- Izmantojot webhook darbībās iekšējiem notikumiem, noteikti atzīmējiet izvēles rūtiņu *Custom message* un definējiet pielāgotu ziņojumu darbības operācijas konfigurācijā. Pretējā gadījumā paziņojums netiks nosūtīts.

1 Webhook skriptu piemēri

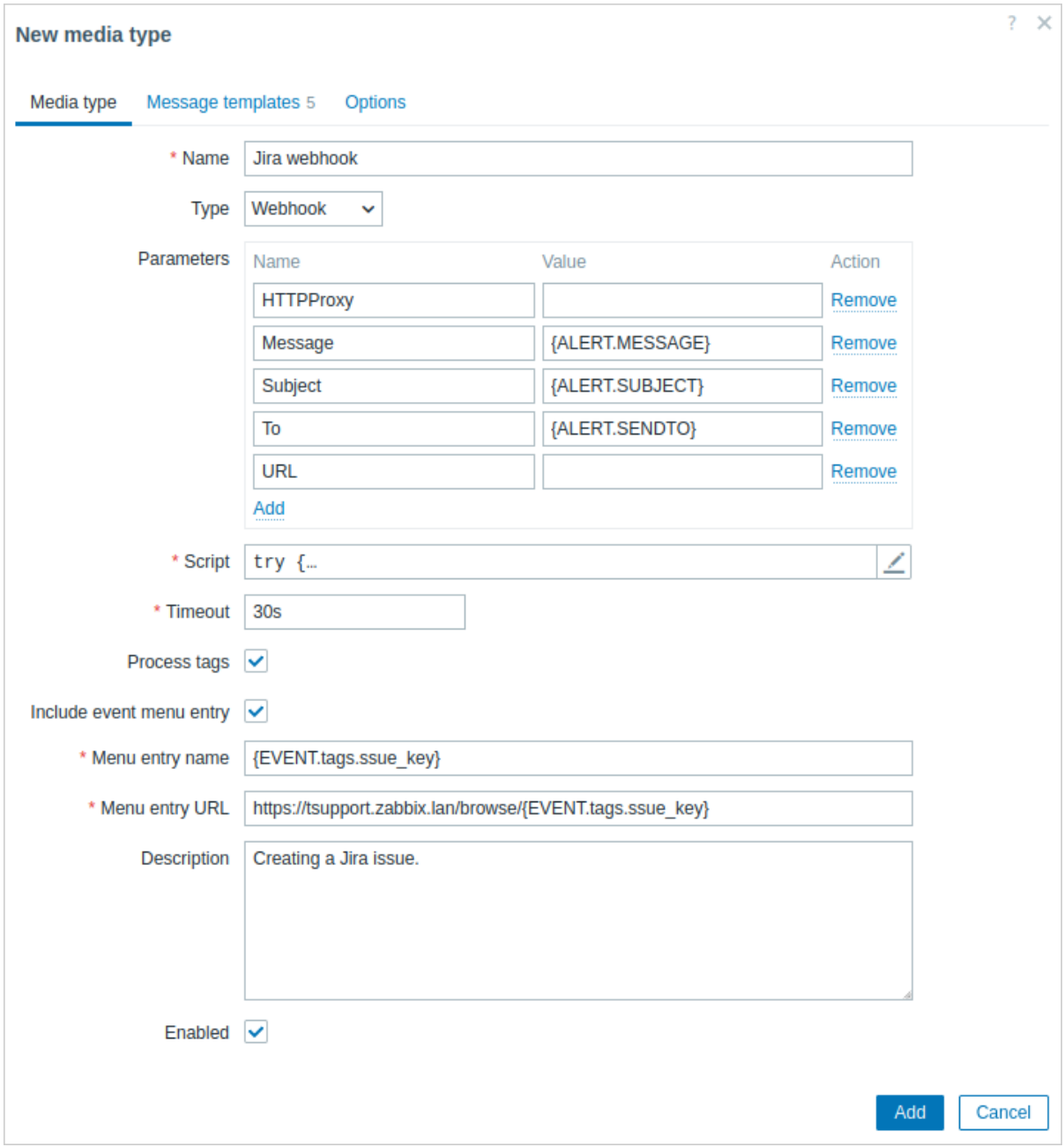
Pārskats

Lai gan Zabbix piedāvā lielu skaitu webhook integrāciju, kas ir pieejamas uzreiz, iespējams, vēlēsieties tā vietā izveidot savus webhook. Šajā sadaļā ir sniegti pielāgotu webhook skriptu piemēri (izmantojami parametrā *Script*). Citu webhook parametru aprakstu skatiet sadaļā webhook.

Attention:
Neizmantojiet neizdeklarētus piešķīrumus priekšapstrādes JavaScript kodā. Lietojiet `var`, lai deklarētu lokālos mainīgos.

Atkopšanas notikumi (neatkarīgi no tā, vai tie ģenerēti automātiski vai pēc manuālas aizvēršanas) ietver atrisinātā notikuma birkas (tostarp birkas, kas mantotas no veidnēm, hosts un trigeriem). Webhook skripti tiek izpildīti pēc brīdinājuma izveides; tādēļ birkas, ko atgriež webhook skripts, tiek piemērotas pēc sākotnējās brīdinājuma izveides un var nebūt ietvertas brīdinājumā, kas izraisīja webhook. Ja integrācijai ir nepieciešams, lai webhook izveidotās birkas būtu pieejamas sākotnējā paziņojumā, iegūstiet notikuma birkas no servera (piemēram, izmantojot Event API) vai saglabājiet webhook izveidotās birkas ārējā pastāvīgā krātuvē un veiciet korelāciju tur.

Jira webhook (pielāgots)



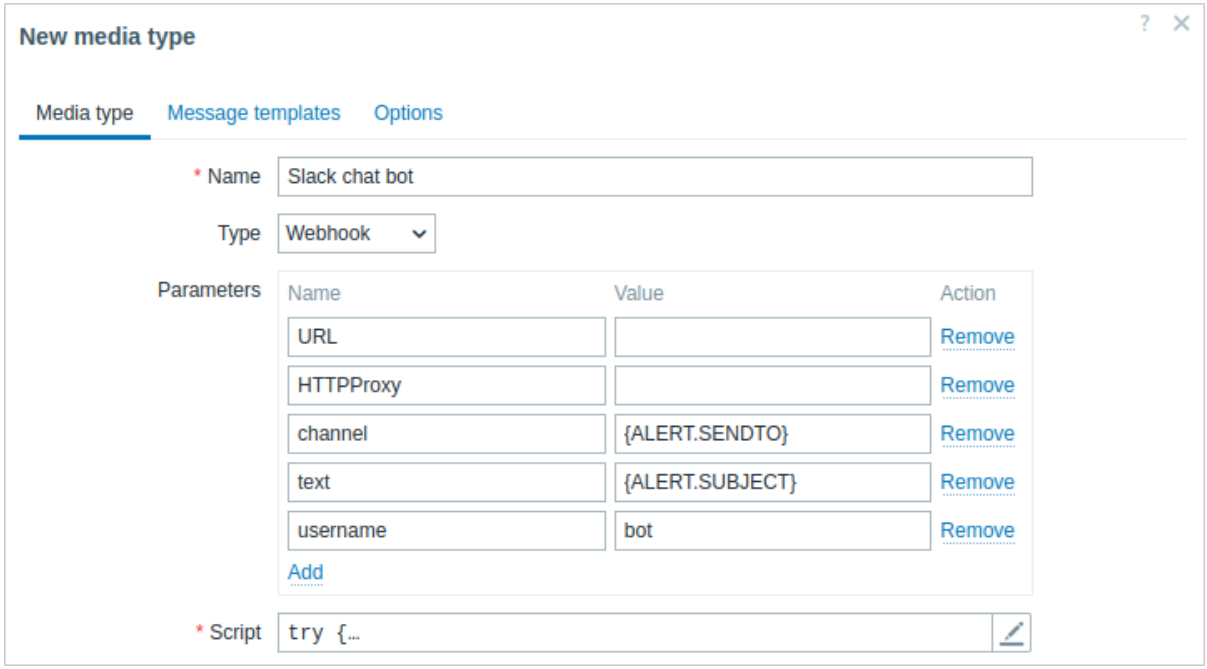
Šis skripts izveidos JIRA problēmu un atgriezīs informāciju par izveidoto problēmu.

```
try {
    Zabbix.log(4, '[ Jira webhook ] Started with params: ' + value);

    var result = {
            'tags': {
```

```
                'endpoint': 'jira'
            }
        },
        params = JSON.parse(value),
        req = new HttpRequest(),
        fields = {},
        resp;

    if (params.HTTPProxy) {
        req.setProxy(params.HTTPProxy);
    }

    req.addHeader('Content-Type: application/json');
    req.addHeader('Authorization: Basic ' + params.authentication);

    fields.summary = params.summary;
    fields.description = params.description;
    fields.project = {key: params.project_key};
    fields.issuetype = {id: params.issue_id};

    resp = req.post('https://jira.example.com/rest/api/2/issue/',
        JSON.stringify({"fields": fields})
    );

    if (req.getStatus() != 201) {
        throw 'Response code: ' + req.getStatus();
    }

    resp = JSON.parse(resp);
    result.tags.issue_id = resp.id;
    result.tags.issue_key = resp.key;

    return JSON.stringify(result);
}
catch (error) {
    Zabbix.log(4, '[ Jira webhook ] Issue creation failed json : ' + JSON.stringify({"fields": fields}));
    Zabbix.log(3, '[ Jira webhook ] issue creation failed : ' + error);

    throw 'Failed with error: ' + error;
}
```

Slack webhook (pielāgots)

Šis webhook pārsūtīs paziņojumus no Zabbix uz Slack kanālu.




```
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 Darbības

Pārskats

Ja vēlaties, lai notikumu rezultātā tiktu veiktas noteiktas darbības (piemēram, nosūtīti paziņojumi), jums jākonfigurē darbības.

Darbības var definēt, reaģējot uz visu atbalstīto tipu notikumiem:

- Trigeru darbības — notikumiem, kad trigeru stāvoklis mainās starp `OK` un `Problem`
- Pakalpojumu darbības — notikumiem, kad pakalpojuma stāvoklis mainās starp `OK` un `Problem`
- Atklāšanas darbības — notikumiem, kad notiek tīkla atklāšana
- Automātiskās reģistrācijas darbības — notikumiem, kad jauni aktīvie aģenti tiek automātiski reģistrēti (vai mainās reģistrēto hostu metadati)
- Iekšējās darbības — notikumiem, kad vienumi kļūst neatbalstīti vai trigeri nonāk nezināmā stāvoklī

Galvenās pakalpojumu darbību atšķirības ir šādas:

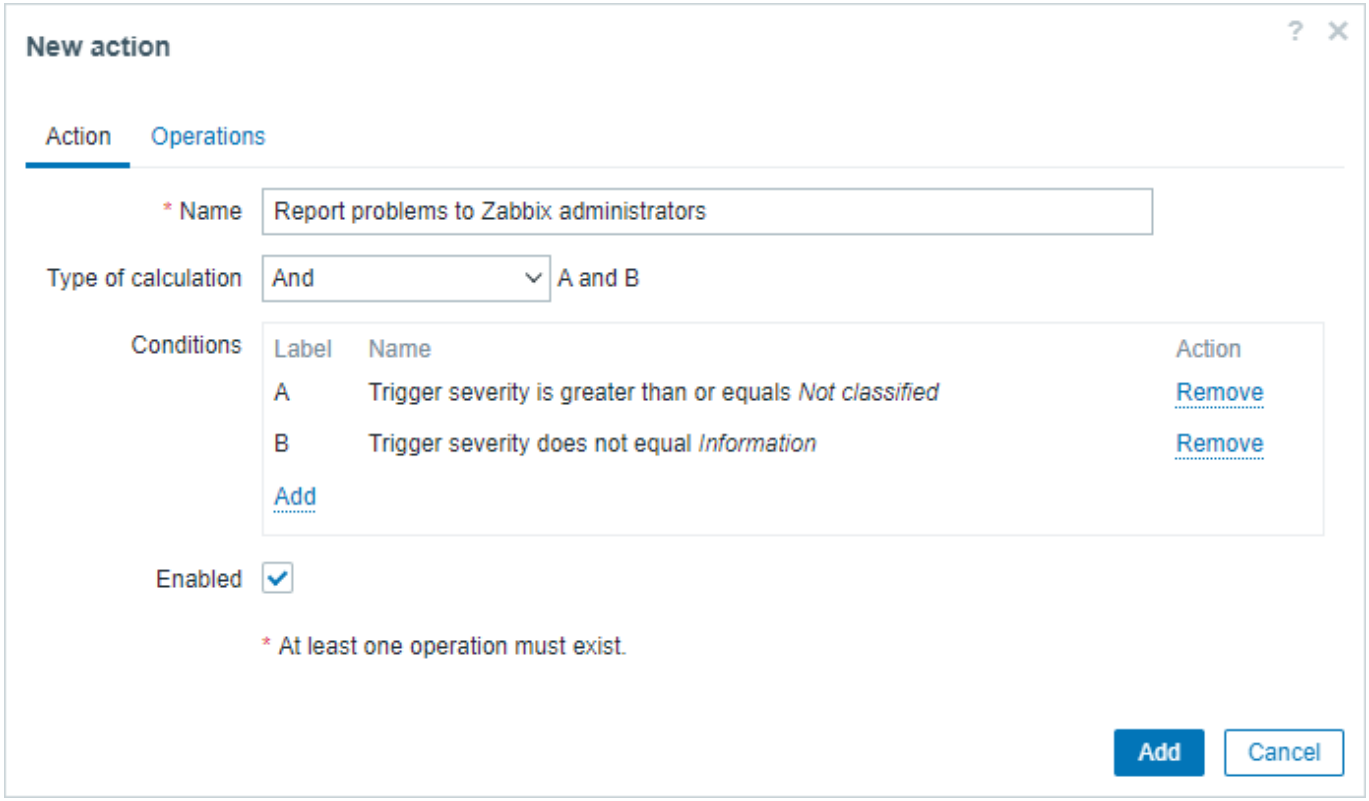
- Lietotāja piekļuve pakalpojumu darbībām ir atkarīga no piekļuves tiesībām pakalpojumiem, kas piešķirtas lietotāja lomai
- Pakalpojumu darbības atbalsta atšķirīgu nosacījumu kopu

Darbības konfigurēšana

Lai konfigurētu darbību, rīkojieties šādi:

- Dodieties uz *Brīdinājumi* > *Darbības* un apakšizvēlnē atlasiet nepieciešamo darbības tipu (vēlāk varat pārslēgties uz citu tipu, izmantojot virsraksta nolaižamo sarakstu).
- Noklikšķiniet uz *Izveidot darbību*.
- Nosauciet darbību.
- Izvēlieties nosacījumus, pēc kuriem tiek izpildītas operācijas.
- Izvēlieties izpildāmās operācijas.

Darbības vispārīgie atribūti:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Unikāls darbības nosaukums.

Parametrs	Apraksts
Type of calculation	Atlasiet darbības nosacījumu novērtēšanas opciju (ja ir vairāk nekā viens nosacījums): **And** - jābūt izpildītiem visiem nosacījumiem. **Or** - pietiek, ja ir izpildīts viens nosacījums. **And/Or** - abu kombinācija: AND dažādiem nosacījumu tipiem un OR vienam un tam pašam nosacījumu tipam. **Custom expression** - lietotāja definēta aprēķina formula darbības nosacījumu novērtēšanai.
Conditions	Darbības nosacījumu saraksts. Noklikšķiniet uz *Add*, lai pievienotu jaunu nosacījumu. Ja nav konfigurēts neviens nosacījums, darbība tiks izpildīta katram notikumam, kas atbilst konfigurējamajam darbības tipam.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu darbību. Pretējā gadījumā tā būs atspējota.

1 Nosacījumi

Pārskats

Ir iespējams noteikt, ka darbība tiek izpildīta tikai tad, ja notikums atbilst definētai nosacījumu kopai. Nosacījumi tiek iestatīti, konfigurējot darbību.

Nosacījumu atbilstība ir reģistrjutīga.

Trigeru darbības

Tālāk norādītos nosacījumus var izmantot uz trigeriem balstītās darbībās:

Nosacījuma tips	Atbalstītie operatori	Apraksts
Hostu grupa	vienāds ar nav vienāds ar	Norādiet hostu grupas vai hostu grupas, kas jāizslēdz. **vienāds ar** - notikums pieder šai hostu grupai. **nav vienāds ar** - notikums nepieder šai hostu grupai. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Lai norādītu tikai vecākgrupu, visām apakšgrupām papildus jāiestata operators **nav vienāds ar**.
Veidne	vienāds ar nav vienāds ar	Norādiet veidnes vai veidnes, kas jāizslēdz. **vienāds ar** - notikums pieder trigerim, kas mantots no šīs veidnes. **nav vienāds ar** - notikums nepieder trigerim, kas mantots no šīs veidnes.
Hosts	vienāds ar nav vienāds ar	Norādiet hostus vai hostus, kas jāizslēdz. **vienāds ar** - notikums pieder šim hostam. **nav vienāds ar** - notikums nepieder šim hostam.
Birka nosaukums	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma birku vai notikuma birku, kas jāizslēdz. **vienāds ar** - notikumam ir šī birka. **nav vienāds ar** - notikumam nav šīs birkas. **satur** - notikumam ir birka, kas satur šo virkni. **nesatur** - notikumam nav birkas, kas satur šo virkni.
Birkas vērtība	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma birkas un vērtības kombināciju vai birkas un vērtības kombināciju, kas jāizslēdz. **vienāds ar** - notikumam ir šī birka un vērtība. **nav vienāds ar** - notikumam nav šīs birkas un vērtības. **satur** - notikumam ir birka un vērtība, kas satur šīs virknes. **nesatur** - notikumam nav birkas un vērtības, kas satur šīs virknes.
Trigeris	vienāds ar nav vienāds ar	Norādiet trigerus vai trigerus, kas jāizslēdz. **vienāds ar** - notikumu ģenerē šis trigeris. **nav vienāds ar** - notikumu ģenerē jebkurš cits trigeris, izņemot šo.
Notikuma nosaukums	satur nesatur	Norādiet virkni trigera ģenerētā notikuma nosaukumā vai virkni, kas jāizslēdz. Pēc noklusējuma notikuma nosaukums atbilst trigera nosaukumam, ja vien trigera konfigurācijā nav norādīts pielāgots notikuma nosaukums. **satur** - notikuma nosaukums satur šo virkni. **nesatur** - šī virkne ir izslēgta no notikuma nosaukuma. Piezīme: Ievadītā vērtība tiks salīdzināta ar notikuma nosaukumu, kurā visas makrosu vērtības ir izvērstas.

Nosacījuma tips	Atbalstītie operatori	Apraksts
Trigera nopietnība	vienāds ar nav vienāds ar ir lielāks vai vienāds ar ir mazāks vai vienāds ar	Norādiet trigera nopietnību. **vienāds ar** - vienāds ar trigera nopietnību. **nav vienāds ar** - nav vienāds ar trigera nopietnību. **ir lielāks vai vienāds ar** - lielāks vai vienāds ar trigera nopietnību. **ir mazāks vai vienāds ar** - mazāks vai vienāds ar trigera nopietnību.
Laika periods	ietilpst neietilpst	Norādiet laika periodu vai laika periodu, kas jāizslēdz. **ietilpst** - notikuma laiks ir šajā laika periodā. **neietilpst** - notikuma laiks nav šajā laika periodā. Formāta aprakstu skatiet lapā laika perioda specifikācija. Tiek atbalstīti lietotāja makrosi.
Problēma ir apspiesta	nē jā	Norādiet, vai problēma ir apspiesta (netiek rādīta) hosta uzturēšanas dēļ. **nē** - problēma nav apspiesta. **jā** - problēma ir apspiesta.

Pakalpojumu darbības

Pakalpojumu darbībās var izmantot šādus nosacījumus:

Nosacījuma tips	Atbalstītie operatori	Apraksts
Pakalpojums	vienāds ar nav vienāds ar	Norādiet pakalpojumu vai pakalpojumu, kuru izslēgt. **vienāds ar** — notikums pieder šim pakalpojumam. **nav vienāds ar** — notikums nepieder šim pakalpojumam. Norādot vecākpakalpojumu, tiek netieši atlasīti arī visi bērnpakalpojumi. Lai norādītu tikai vecākpakalpojumu, visiem ligzdotajiem pakalpojumiem papildus jāiestata operators **nav vienāds ar**.
Pakalpojuma nosaukums	satur nesatur	Norādiet virkni pakalpojuma nosaukumā vai virkni, kuru izslēgt. **satur** — notikumu ģenerē pakalpojums, kura nosaukumā ir šī virkne. **nesatur** — šo virkni nevar atrast pakalpojuma nosaukumā.
Pakalpojuma birkas nosaukums	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma birku vai notikuma birku, kuru izslēgt. Pakalpojuma notikumu birkas var definēt pakalpojuma konfigurācijas sadaļā *Tags*. **vienāds ar** — notikumam ir šī birka. **nav vienāds ar** — notikumam nav šīs birkas. **satur** — notikumam ir birka, kas satur šo virkni. **nesatur** — notikumam nav birkas, kas satur šo virkni.
Pakalpojuma birkas vērtība	vienāds ar nav vienāds ar satur nesatur	Norādiet notikuma birkas un vērtības kombināciju vai birkas un vērtības kombināciju, kuru izslēgt. Pakalpojuma notikumu birkas var definēt pakalpojuma konfigurācijas sadaļā *Tags*. **vienāds ar** — notikumam ir šī birka un vērtība. **nav vienāds ar** — notikumam nav šīs birkas un vērtības. **satur** — notikumam ir birka un vērtība, kas satur šīs virknes. **nesatur** — notikumam nav birkas un vērtības, kas satur šīs virknes.

Attention:
Pārliecinieties, ka izvēlnē *Brīdinājumi* → *Mediju tipi* ir definētas ziņojumu veidnes pakalpojumu darbībām. Pretējā gadījumā paziņojumi netiks nosūtīti.

Atklāšanas darbības

Tālāk norādītos nosacījumus var izmantot uz atklāšanu balstītos notikumos:

Nosacījuma tips	Atbalstītie operatori	Apraksts
Host IP	vienāds ar nav vienāds ar	Norādiet atklāta hosta IP adrešu diapazonu vai izslēdzamo diapazonu. **vienāds ar** - hosta IP ir diapazonā. **nav vienāds ar** - hosta IP nav diapazonā. Tam var būt šādi formāti: Viena IP adrese: 192.168.1.33 IP adrešu diapazons: 192.168.1-10.1-254 IP maska: 192.168.4.0/24 Saraksts: 192.168.1.1-254, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Atstarpes saraksta formātā tiek atbalstītas.
Service type	vienāds ar nav vienāds ar	Norādiet atklāta pakalpojuma tipu vai izslēdzamo pakalpojuma tipu. **vienāds ar** - atbilst atklātajam pakalpojumam. **nav vienāds ar** - neatbilst atklātajam pakalpojumam. Pieejamie pakalpojumu tipi: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Zabbix aģents, SNMPv1 aģents, SNMPv2 aģents, SNMPv3 aģents, ICMP ping, telnet.
Service port	vienāds ar nav vienāds ar	Norādiet atklāta pakalpojuma TCP portu diapazonu vai izslēdzamo diapazonu. **vienāds ar** - pakalpojuma ports ir diapazonā. **nav vienāds ar** - pakalpojuma ports nav diapazonā.
Discovery rule	vienāds ar nav vienāds ar	Norādiet atklāšanas kārtulu vai izslēdzamu atklāšanas kārtulu. **vienāds ar** - izmanto šo atklāšanas kārtulu. **nav vienāds ar** - izmanto jebkuru citu atklāšanas kārtulu, izņemot šo.
Discovery check	vienāds ar nav vienāds ar	Norādiet atklāšanas pārbaudi vai izslēdzamu atklāšanas pārbaudi. **vienāds ar** - izmanto šo atklāšanas pārbaudi. **nav vienāds ar** - izmanto jebkuru citu atklāšanas pārbaudi, izņemot šo.
Discovery object	vienāds ar	Norādiet atklāto objektu. **vienāds ar** - vienāds ar atklāto objektu (ierīci vai pakalpojumu).
Discovery status	vienāds ar	**Up** - atbilst notikumiem 'Host Up' un 'Service Up'. **Down** - atbilst notikumiem 'Host Down' un 'Service Down'. **Discovered** - atbilst notikumiem 'Host Discovered' un 'Service Discovered'. **Lost** - atbilst notikumiem 'Host Lost' un 'Service Lost'.
Uptime/Downtime	ir lielāks vai vienāds ar ir mazāks vai vienāds ar	Darbspējas laiks notikumiem 'Host Up' un 'Service Up'. Dīkstāves laiks notikumiem 'Host Down' un 'Service Down'. **ir lielāks vai vienāds ar** - ir lielāks vai vienāds ar. Parametrs tiek norādīts sekundēs. **ir mazāks vai vienāds ar** - ir mazāks vai vienāds ar. Parametrs tiek norādīts sekundēs.
Received value	vienāds ar nav vienāds ar ir lielāks vai vienāds ar ir mazāks vai vienāds ar satur nesatur	Norādiet vērtību, kas saņemta no aģenta (Zabbix, SNMP) pārbaudes atklāšanas kārtulā. Virkņu salīdzināšana. Ja kārtulai ir konfigurētas vairākas Zabbix aģenta vai SNMP pārbaudes, tiek pārbaudītas no katras no tām saņemtās vērtības (katra pārbaude ģenerē jaunu notikumu, kas tiek salīdzināts ar visiem nosacījumiem). **vienāds ar** - vienāds ar vērtību. **nav vienāds ar** - nav vienāds ar vērtību. **ir lielāks vai vienāds ar** - lielāks vai vienāds ar vērtību. **ir mazāks vai vienāds ar** - mazāks vai vienāds ar vērtību. **satur** - satur apakšvirkni. Parametrs tiek norādīts kā virkne. **nesatur** - nesatur apakšvirkni. Parametrs tiek norādīts kā virkne.
Proxy	vienāds ar nav vienāds ar	Norādiet starpniekserveri vai izslēdzamu starpniekserveri. **vienāds ar** - izmanto šo starpniekserveri. **nav vienāds ar** - izmanto jebkuru citu starpniekserveri, izņemot šo.

Note:
Pakalpojumu pārbaudes atklāšanas kārtulā, kuru rezultātā rodas atklāšanas notikumi, nenotiek vienlaikus. Tāpēc, ja darbībā ir konfigurētas **vairākas** vērtības nosacījumiem `Service type`, `Service port` vai `Received value`, tās tiks salīdzinātas ar vienu atklāšanas notikumu vienlaikus, bet **nevis** ar vairākiem notikumiem vienlaikus. Tādēļ darbības ar vairākām vērtībām vieniem un tiem pašiem pārbaužu tipiem var netikt izpildītas pareizi.

Autoreģistrācijas darbības

Tālāk norādītos nosacījumus var izmantot darbībās, kas balstītas uz aktīvā aģenta autoreģistrāciju:

Nosacījuma tips	Atbalstītie operatori	Apraksts
Hosta metadati	satur nesatur atbilst neatbilst	Norādiet hosta metadatus vai izslēdzamos hosta metadatus. **satur** - hosta metadati satur virkni. **nesatur** - hosta metadati nesatur virkni. Hosta metadatus var norādīt aģenta konfigurācijas failā. **atbilst** - hosta metadati atbilst regulārajai izteiksmei. **neatbilst** - hosta metadati neatbilst regulārajai izteiksmei.
Hosta nosaukums	satur nesatur atbilst neatbilst	Norādiet hosta nosaukumu vai izslēdzamo hosta nosaukumu. **satur** - hosta nosaukums satur virkni. **nesatur** - hosta nosaukums nesatur virkni. **atbilst** - hosta nosaukums atbilst regulārajai izteiksmei. **neatbilst** - hosta nosaukums neatbilst regulārajai izteiksmei.
Starpniekserveris	ir vienāds ar nav vienāds ar	Norādiet starpniekserveri vai izslēdzamo starpniekserveri. **ir vienāds ar** - tiek izmantots šis starpniekserveris. **nav vienāds ar** - tiek izmantots jebkurš cits starpniekserveris, izņemot šo.

Internal event actions

The following conditions can be set for actions based on internal events:

Condition type	Supported operators	Description
Event type	equals	**Item in ”not supported” state** - matches events where an item goes from a ’normal’ to ’not supported’ state **Low-level discovery rule in ”not supported” state** - matches events where a low-level discovery rule goes from a ’normal’ to ’not supported’ state **Trigger in ”unknown” state** - matches events where a trigger goes from a ’normal’ to ’unknown’ state
Host group	equals does not equal	Specify host groups or host groups to exclude. **equals** - event belongs to this host group. **does not equal** - event does not belong to this host group.
Tag name	equals does not equal contains does not contain	Specify event tag or event tag to exclude. **equals** - event has this tag **does not equal** - event does not have this tag **contains** - event has a tag containing this string **does not contain** - event does not have a tag containing this string
Tag value	equals does not equal contains does not contain	Specify event tag and value combination or tag and value combination to exclude. **equals** - event has this tag and value **does not equal** - event does not have this tag and value **contains** - event has a tag and value containing these strings **does not contain** - event does not have a tag and value containing these strings
Template	equals does not equal	Specify templates or templates to exclude. **equals** - event belongs to an item/trigger/low-level discovery rule inherited from this template. **does not equal** - event does not belong to an item/trigger/low-level discovery rule inherited from this template.
Host	equals does not equal	Specify hosts or hosts to exclude. **equals** - event belongs to this host. **does not equal** - event does not belong to this host.

Aprēķina veids

Ir pieejamas šādas nosacījumu aprēķināšanas opcijas:

- **Un** — visiem nosacījumiem jābūt izpildītiem

Ņemiet vērā, ka aprēķina veida "Un" izmantošana nav atļauta starp vairākiem trigeriem, ja tie ir atlasīti kā `Trigger=` nosacījums. Darbības var tikt izpildītas tikai, pamatojoties uz viena trigeris notikumu.

- **Vai** — pietiek, ja ir izpildīts viens nosacījums
- **Un/Vai** — abu kombinācija: AND dažādiem nosacījumu tipiem un OR vienam un tam pašam nosacījuma tipam, piemēram:

Hostu grupa ir vienāda ar Oracle serveriem
 Hostu grupa ir vienāda ar MySQL serveriem
 Notikuma nosaukums satur 'Datubāze nedarbojas'
 Notikuma nosaukums satur 'Datubāze nav pieejama'

tiek novērtēts kā

(Hostu grupa ir vienāda ar Oracle serveriem **or** Hostu grupa ir vienāda ar MySQL serveriem**) and (**Notikuma nosaukums satur 'Datubāze nedarbojas' **or** Notikuma nosaukums satur 'Datubāze nav pieejama'**)**

- **Pielāgota izteiksme** — lietotāja definēta aprēķina formula darbības nosacījumu novērtēšanai. Tai jāietver visi nosacījumi (attēloti kā lielie burti A, B, C, ...) un tā var ietvert atstarpes, tabulācijas, iekavas ( ), **and** (reģistrjutīgs), **or** (reģistrjutīgs), **not** (reģistrjutīgs).

Lai gan iepriekšējais piemērs ar `And/Or` tiktu attēlots kā (A or B) and (C or D), pielāgotā izteiksmē var būt arī vairāki citi aprēķināšanas veidi:

(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
 utt.

Darbībās izmantoto objektu dzēšana

Mēģinot dzēst objektus, kas tiek izmantoti darbības nosacījumā/operācijā, lietotājs saņems atbilstošu kļūdas ziņojumu.

Darbības netiek atspējotas, ja darbību nosacījumiem vai operācijām:

- ir dzēsti objekti;
- ir objekti, kurus zema līmeņa atklāšana vairs neatklāj un kuri ir noņemti (pēc *Delete lost resources* perioda).

Darbību nosacījumi/operācijas netiek noņemti, ja tajos trūkst objektu. Tā vietā dzēstie objekti tiek attēloti kā *Deleted host group*, *Deleted host*, *Deleted trigger* utt. Darbību saraksta kolonnā "Info" tiek parādīta brīdinājuma ikona ar ziņojumu "This action has conditions or operations referencing deleted object(s).".

2 Darbības

Pārskats

Visiem notikumiem varat definēt šādas operācijas:

- Nosūtīt ziņojumu
- Izpildīt attālo komandu

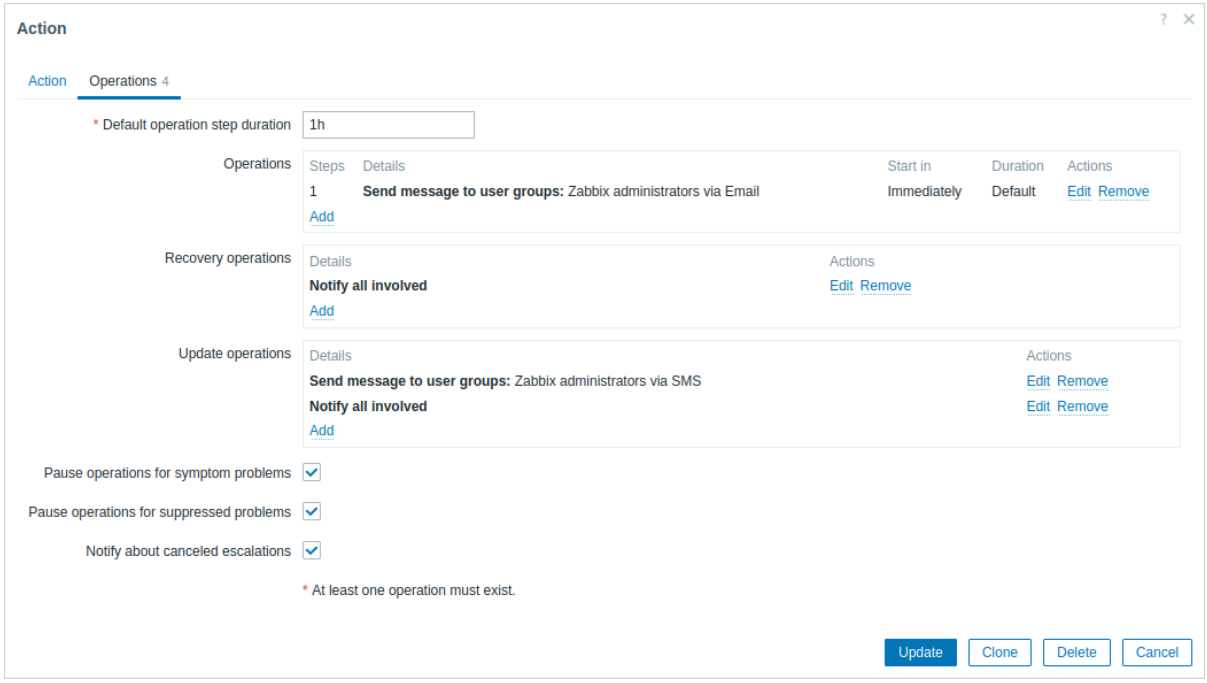
> **Attention:**
> Zabbix serveris neveido brīdinājumus, ja piekļuve hostam ir nepārprotami "liegta" lietotājam, kas definēts kā darbības operācijas saņēmējs, vai ja lietotājam vispār nav definētas tiesības uz šo hostu.

Atklāšanas un automātiskās reģistrācijas notikumiem ir pieejamas papildu operācijas:

- Pievienot hostu
- Noņemt hostu
- Iespējot hostu
- Atspējot hostu
- Pievienot hostu grupai
- Noņemt no hostu grupas
- Pievienot hosta tagus
- Noņemt hosta tagus
- Piesaistīt veidni
- Atvienot veidni
- Iestatīt hosta inventarizācijas režīmu

Operācijas konfigurēšana

Lai konfigurētu operāciju, dodieties uz cilni *Operations* darbības konfigurācijā.



Vispārīgie operācijas atribūti:

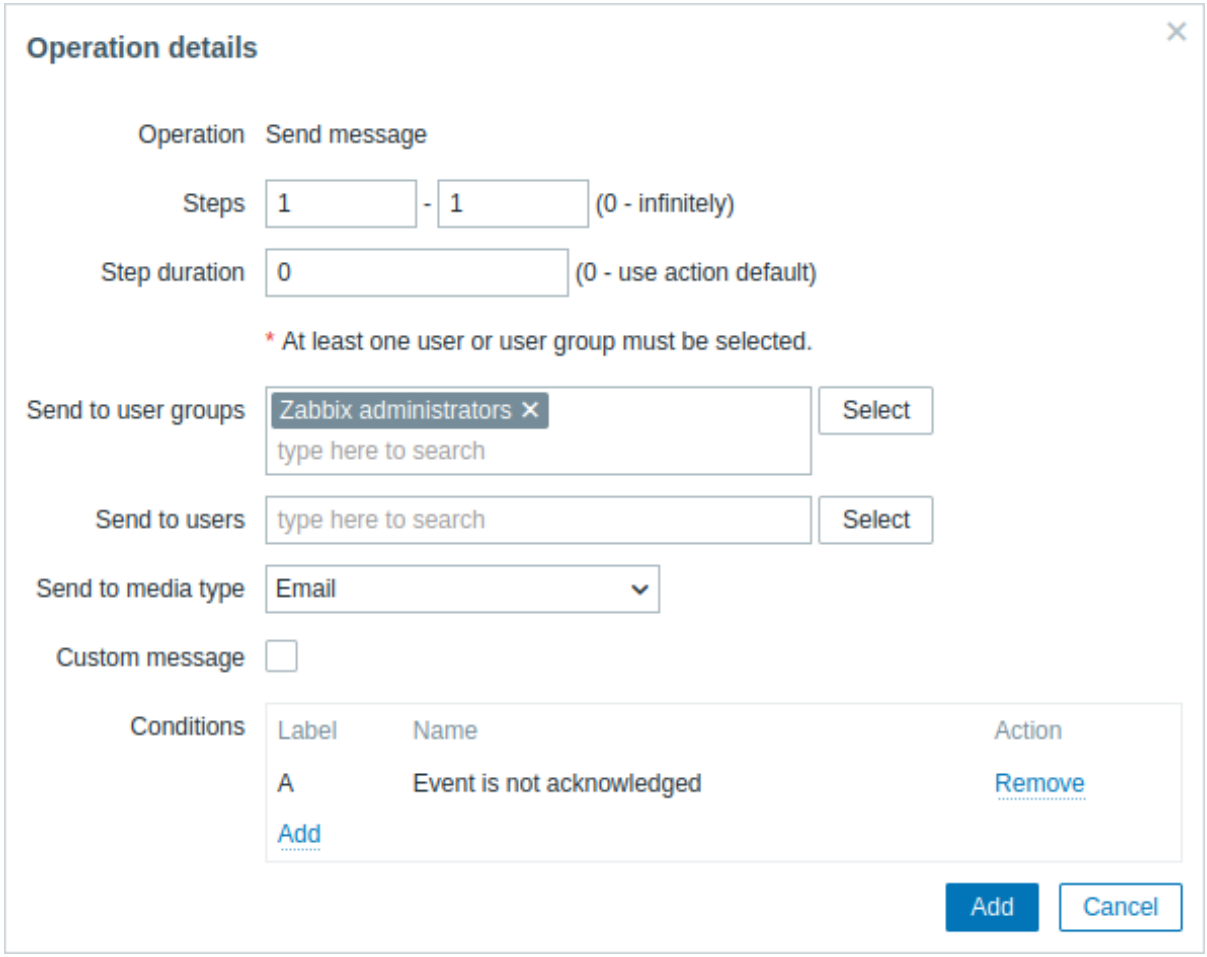
Parametrs	Apraksts
Default operation step duration	Vienas operācijas darbības noklusējuma ilgums (no 60 sekundēm līdz 1 nedēļai). Piemēram, vienu stundu ilgs darbības ilgums nozīmē, ka pēc operācijas izpildes paies viena stunda līdz nākamajai darbībai. Tiek atbalstīti laika sufiksi, piemēram, 60s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi.
Operations	Tiek parādītas darbības operācijas (ja tādas ir) ar šādu informāciju: **Steps** - eskalācijas darbība(-as), kurai(-ām) operācija ir piešķirta. **Details** - operācijas tips un tās saņēmējs/mērķis. Operāciju sarakstā tiek parādīts arī izmantotais medija tips (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Start in** - cik ilgi pēc notikuma operācija tiek izpildīta. **Duration (sec)** - tiek parādīts darbības ilgums. Tiek rādīts *Default*, ja darbībai tiek izmantots noklusējuma ilgums, un laiks, ja tiek izmantots pielāgots ilgums. **Actions** - tiek parādītas saites operācijas rediģēšanai un noņemšanai.
Recovery operations	Tiek parādītas darbības operācijas (ja tādas ir) ar šādu informāciju: **Details** - operācijas tips un tās saņēmējs/mērķis. Operāciju sarakstā tiek parādīts arī izmantotais medija tips (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Actions** - tiek parādītas saites operācijas rediģēšanai un noņemšanai.
Update operations	Tiek parādītas darbības operācijas (ja tādas ir) ar šādu informāciju: **Details** - operācijas tips un tās saņēmējs/mērķis. Operāciju sarakstā tiek parādīts arī izmantotais medija tips (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). **Actions** - tiek parādītas saites operācijas rediģēšanai un noņemšanai.
Pause operations for symptom problems	Atzīmējiet šo izvēles rūtiņu, lai apturētu operācijas (pēc pirmās operācijas) simptomu problēmām. Ņemiet vērā, ka šis iestatījums ietekmē tikai problēmu eskalācijas; atkopšanas un atjaunināšanas operācijas netiks ietekmētas. Šī opcija ir pieejama tikai *Trigger actions*.
Pause operations for suppressed problems	Atzīmējiet šo izvēles rūtiņu, lai aizkavētu operāciju sākumu uz uzturēšanas perioda laiku. Kad operācijas tiek sāktas pēc uzturēšanas, tiek izpildītas visas operācijas, tostarp tās, kas attiecas uz notikumiem uzturēšanas laikā. Ņemiet vērā, ka šis iestatījums ietekmē tikai problēmu eskalācijas; atkopšanas un atjaunināšanas operācijas netiks ietekmētas. Ja noņemsiet atzīmi no šīs izvēles rūtiņas, operācijas tiks izpildītas bez aizkaves pat uzturēšanas perioda laikā. Šī opcija nav pieejama *Service actions*.

Parametrs	Apraksts
Notify about canceled escalations	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu paziņojumus par atceltām eskalācijām (ja hosts, vienums, trigeris vai darbība ir atspējota).

Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Lai konfigurētu jaunas operācijas detalizēto informāciju, noklikšķiniet uz Add blokā *Operations*. Lai rediģētu esošu operāciju, noklikšķiniet uz Edit blakus operācijai. Tiks atvērts uznirstošais logs, kurā varēsiet rediģēt operācijas darbību detalizēto informāciju.

Darbības informācija



Parametrs	Apraksts
Darbība	Izvēlieties darbību: **Sūtīt ziņojumu** — nosūtīt ziņojumu lietotājam. **<attālās komandas nosaukums>** — izpildīt attālo komandu. Komandas ir pieejamas izpildei, ja tās iepriekš definētas globālajos skriptos, kā to tvērums izvēloties *Action operation*. Atklāšanas un automātiskās reģistrācijas notikumiem ir pieejamas arī citas darbības (skatiet iepriekš).
Soļi	Izvēlieties soli(-us), kuriem piešķirt darbību eskalācijas grafikā: **No** — izpildīt, sākot ar šo soli. **Līdz** — izpildīt līdz šim solim (0=bezgalība, izpilde netiks ierobežota).
Soļa ilgums	Pielāgots ilgums šiem soļiem (0=izmantot noklusēto soļa ilgumu). Tiek atbalstīti laika sufiksi, piemēram, 60s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. Vienam un tam pašam solim var piešķirt vairākas darbības. Ja šīm darbībām ir definēts atšķirīgs soļa ilgums, tiek ņemts vērā īsākais un piemērots šim solim.

Parametrs	Apraksts
Darbības tips: sūtīt ziņojumu	
Sūtīt lietotāju grupām	Izvēlieties lietotāju grupas, kurām nosūtīt ziņojumu. Lai saņemtu paziņojumu, lietotāju grupai jābūt vismaz ”read” atļaujām uz hostu.
Sūtīt lietotājiem	Izvēlieties lietotājus, kuriem nosūtīt ziņojumu. Lai saņemtu paziņojumu, lietotājam jābūt vismaz ”read” atļaujām uz hostu.
Sūtīt uz medija tipu	Nosūtīt ziņojumu uz visiem pieejamajiem (konfigurētajiem un iespējotajiem) mediju tipiem vai tikai uz konkrētu tipu.
Pielāgots ziņojums	Ja atlasīts, var konfigurēt pielāgotu ziņojumu. Paziņojumiem par iekšējiem notikumiem, izmantojot webhooks, pielāgots ziņojums ir obligāts.
Temats	Pielāgotā ziņojuma temats. Temats var saturēt makrosus. Tas ir ierobežots līdz 255 rakstzīmēm.
Ziņojums	Pielāgotais ziņojums. Ziņojums var saturēt makrosus. Tas ir ierobežots līdz noteiktam rakstzīmju skaitam atkarībā no datubāzes tipa (plašākai informācijai skatiet Ziņojuma sūtīšana).
Darbības tips: attālā komanda	
Mērķu saraksts	Izvēlieties mērķus, uz kuriem izpildīt komandu: **Pašreizējais hosts** — komanda tiek izpildīta uz trigera hosta, kas izraisīja problēmas notikumu. Šī opcija nedarbosies, ja trigerī ir vairāki hosti. **Hosts** — izvēlieties hostu(-s), uz kuriem izpildīt komandu. **Hostu grupa** — izvēlieties hostu grupu(-as), uz kurām izpildīt komandu. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas visas tajā ietvertās apakšgrupas. Tādējādi attālā komanda tiks izpildīta arī uz hostiem no apakšgrupām. Komanda uz hosta tiek izpildīta tikai vienu reizi, pat ja hosts atbilst vairāk nekā vienu reizi (piemēram, no vairākām hostu grupām; individuāli un no hostu grupas). Mērķu sarakstam nav nozīmes, ja pielāgots skripts tiek izpildīts uz Zabbix servera. Šajā gadījumā, atlasot vairāk mērķu, skripts tikai tiks izpildīts uz servera vairākas reizes. Ņemiet vērā, ka globālajiem skriptiem mērķa atlase ir atkarīga arī no iestatījuma *Host group* globālā skripta konfigurācijā. Opcija *Mērķu saraksts* nav pieejama *Pakalpojumu darbībām*, jo šajā gadījumā attālās komandas vienmēr tiek izpildītas uz Zabbix servera.
Nosacījumi	Nosacījums darbības izpildei: **Notikums nav apstiprināts** — tikai tad, ja notikums nav apstiprināts. **Notikums ir apstiprināts** — tikai tad, ja notikums ir apstiprināts. Opcija *Nosacījumi* ir pieejama tikai *Trigeru darbībām*.

Kad pabeigts, noklikšķiniet uz *Pievienot*, lai pievienotu darbību *Darbību* sarakstam.

1 Ziņojuma sūtīšana

Pārskats

Ziņojuma nosūtīšana ir viens no labākajiem veidiem, kā informēt cilvēkus par problēmu. Tāpēc tā ir viena no galvenajām darbībām, ko piedāvā Zabbix.

Konfigurācija

Lai varētu sūtīt un saņemt paziņojumus no Zabbix, jums ir:

- jādefinē datu nesējs, uz kuru sūtīt ziņojumu

Ja darbība notiek ārpus laika perioda When active, kas lietotāja konfigurācijā definēts izvēlētajam datu nesējam, ziņojums netiks nosūtīts.

Noklusējuma trigera nopietnības līmenim ('Not classified') lietotāja datu nesēja konfigurācijā **obligāti jābūt** atzīmētam, ja vēlaties saņemt paziņojumus par notikumiem, kas nav trigeru notikumi, piemēram, atklāšanu, aktīva aģenta automātisko reģistrāciju vai iekšējiem notikumiem.

- jākonfigurē darbības operācija, kas sūta ziņojumu uz vienu no definētajiem datu nesējiem

Attention:
Zabbix sūta paziņojumus tikai tiem lietotājiem, kuriem ir vismaz 'read' atļaujas uz hostu, kas ģenerēja notikumu. Jābūt pieejamam vismaz vienam trigera izteiksmes hostam.

Varat konfigurēt pielāgotus ziņojumu sūtīšanas scenārijus, izmantojot eskalācijas.

Lai veiksmīgi saņemtu un lasītu e-pastus no Zabbix, e-pasta serveriem/klientiem jāatbalsta standarta 'SMTP/MIME email' formāts, jo Zabbix sūta UTF-8 datus (ja temats satur tikai ASCII rakstzīmes, tas netiek kodēts UTF-8 formātā). Ziņojuma temats un pamatteksts tiek kodēti base64 formātā, lai atbilstu 'SMTP/MIME email' formāta standartam.

Ziņojuma ierobežojums pēc visu makrosu izvēršanas ir tāds pats kā ziņojuma ierobežojums attālinātajām komandām.

Ziņojumu izsekošana

Jūs varat skatīt nosūtīto ziņojumu statusu sadaļā *Uzraudzība* → *Problēmas*.

Kolonnā *Darbības* jūs varat redzēt kopsavilkuma informāciju par veiktajām darbībām. Tajā zaļie skaitļi apzīmē nosūtītos ziņojumus, bet sarkanie — neveiksmīgos ziņojumus. *Procesā* norāda, ka darbība ir uzsākta. *Neizdevās* informē, ka neviena darbība nav izpildīta veiksmīgi.

Ja noklikšķināsiet uz notikuma laika, lai skatītu notikuma detalizētu informāciju, jūs varēsiet redzēt informāciju par ziņojumiem, kas nosūtīti (vai nav nosūtīti) šī notikuma dēļ, blokā *Darbības*.

Sadaļā *Pārskati* → *Darbību žurnāls* jūs redzēsiet detalizētu informāciju par visām veiktajām darbībām tiem notikumiem, kuriem ir konfigurēta darbība.

2 Attālinātās komandas

Pārskats

Ar attālinātajām komandām jūs varat definēt, ka noteikta iepriekš definēta komanda tiek automātiski izpildīta uzraudzītajā hostā, iestājoties kādam nosacījumam.

Tādējādi attālinātās komandas ir jaudīgs mehānisms viedai, proaktīvai uzraudzībai.

Visredzamākajos šīs funkcijas izmantošanas gadījumos varat mēģināt:

- Automātiski restartēt kādu lietotni (tīmekļa serveri, starpprogrammatūru, CRM), ja tā neatbild
- Izmantot IPMI komandu 'reboot', lai pārstartētu kādu attālinātu serveri, ja tas neatbild uz pieprasījumiem
- Automātiski atbrīvot diska vietu (dzēšot vecākus failus, tīrot /tmp), ja sāk pietrūkt diska vietas
- Migrēt VM no vienas fiziskas iekārtas uz citu atkarībā no CPU noslodzes
- Pievienot jaunus mezglus mākoņvidē, ja nepietiek CPU (diska, atmiņas vai citu) resursu

Darbības konfigurēšana attālinātajām komandām ir līdzīga ziņojuma sūtīšanas konfigurēšanai; vienīgā atšķirība ir tā, ka Zabbix izpildīs komandu, nevis nosūtīs ziņojumu.

Attālinātās komandas var izpildīt Zabbix serveris, starpniekserveris vai aģents. Attālinātās komandas Zabbix aģentā var tikt izpildītas tieši no Zabbix servera vai caur Zabbix starpniekserveri. Gan Zabbix aģentā, gan Zabbix starpniekserverī attālinātās komandas pēc noklusējuma ir atspējotas. Tās var iespējot šādi:

- pievienojot `AllowKey=system.run[*]` parametru aģenta konfigurācijā;
- iestatot parametru EnableRemoteCommands uz '1' starpniekservera konfigurācijā.

Attālinātās komandas, ko izpilda Zabbix serveris, tiek palaistas, kā aprakstīts Komandu izpilde, ieskaitot izejas koda pārbaudi.

Attālinātās komandas tiek izpildītas pat tad, ja mērķa hosts atrodas apkopes režīmā.

Attālinātās komandas ierobežojums

Attālinātās komandas ierobežojums pēc visu makrosu atrisināšanas ir atkarīgs no datu bāzes tipa un rakstzīmju kopas (ne-ASCII rakstzīmju glabāšanai ir nepieciešams vairāk nekā viens baits):

Datu bāze	Ierobežojums rakstzīmēs	Ierobežojums baitos
MySQL	65535	65535
PostgreSQL	65535	nav ierobežots
SQLite (tikai Zabbix starpniekserveris)	65535	nav ierobežots

Attālinātās komandas izpildes izvade (atgrieztā vērtība) ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas). IPMI attālinātās komandas ierobežojums ir atkarīgs no instalētās IPMI bibliotēkas. Ņemiet vērā, ka datu bāzes ierobežojumi attiecas uz visām attālinātajām komandām.

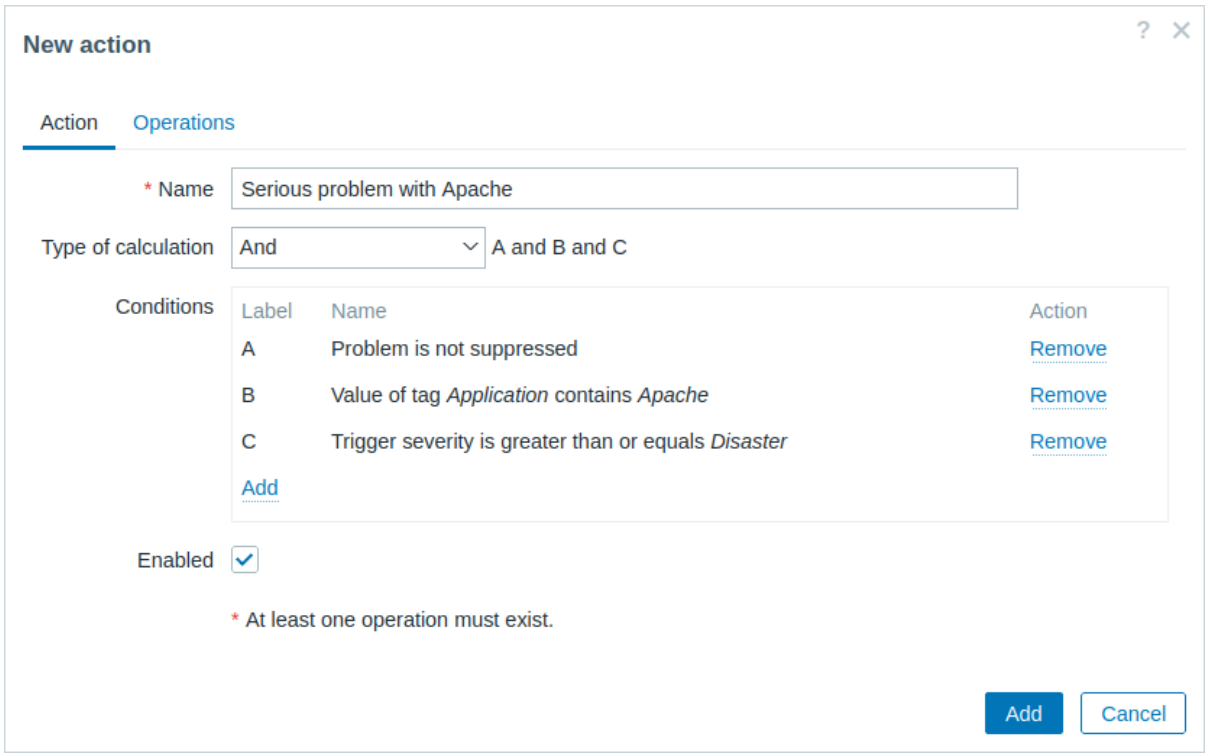
Konfigurācija

Šīs attālinātās komandas, kas tiek izpildītas Zabbix aģentā (pielāgoti skripti), vispirms ir jāiespējo aģenta konfigurācijā.

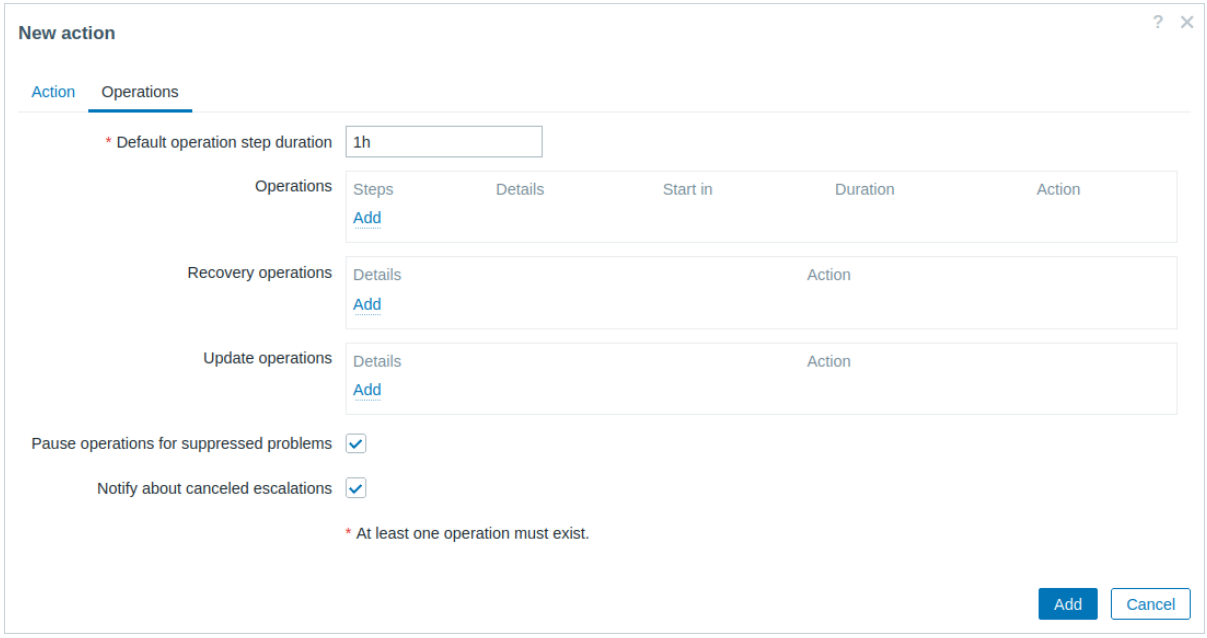
Pārliecinieties, ka parametrs AllowKey=system.run[<command>,*] ir pievienots katrai atļautajai komandai aģenta konfigurācijā, lai atļautu konkrētu komandu nowait režīmā. Ja mainât šo parametru, restartējiet aģenta dēmonu.

Pēc tam, konfigurējot jaunu darbību sadaļā *Brīdinājumi* → *Darbības* → *Trigeru darbības*:

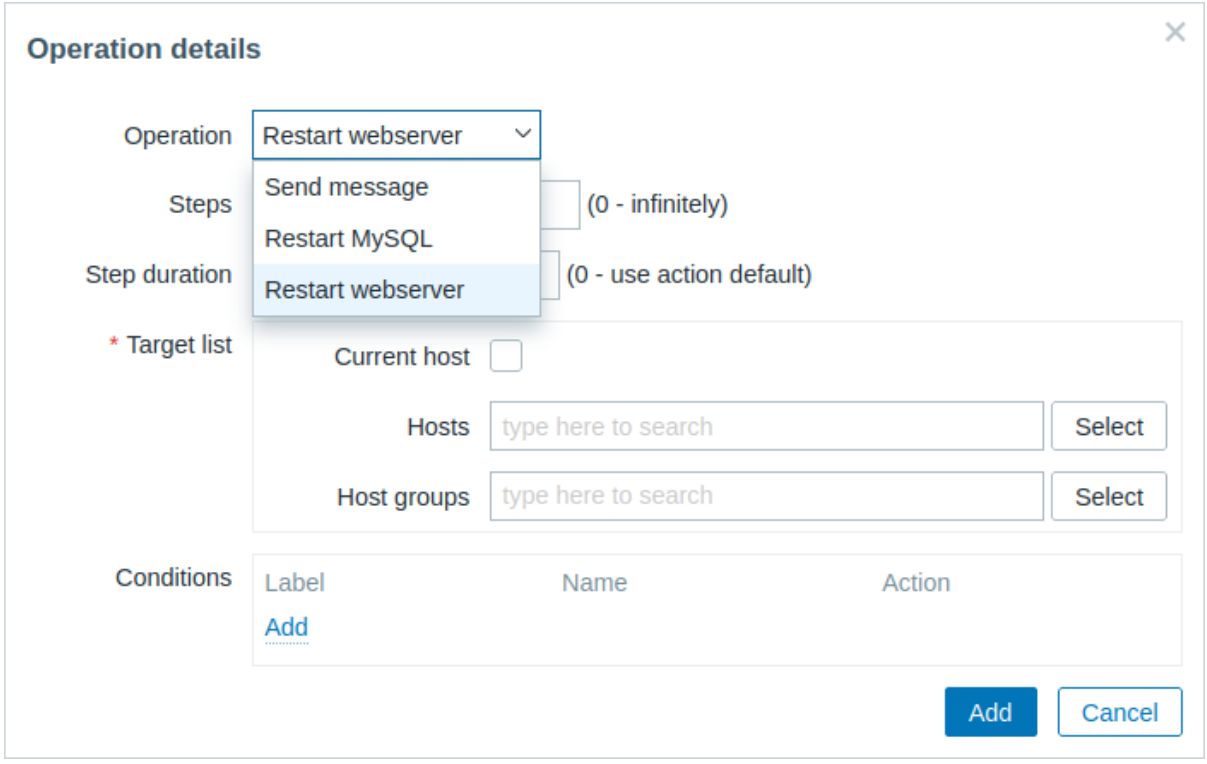
1. Definējiet atbilstošos nosacījumus, piemēram, iestatiet, ka darbība tiek aktivizēta jebkuru katastrofas līmeņa problēmu gadījumā kādā no Apache lietotnēm.



2. cilnē *Darbības* noklikšķiniet uz *Pievienot* blokā *Darbības*, *Atkopšanas darbības* vai *Atjaunināšanas darbības*.



3. Nolaižamajā sarakstā *Darbība* atlasiet vienu no iepriekš definētajiem skriptiem un iestatiet skripta *Mērķu sarakstu*.



Iepriekš definēti skripti

Skripti, kas ir pieejami darbību operācijām (webhook, skripts, SSH, Telnet, IPMI), ir definēti globālajos skriptos.

Piemēram:

```
sudo /etc/init.d/apache restart
```

Šajā gadījumā Zabbix mēģinās pārstartēt Apache procesu. Izmantojot šo komandu, pārliecinieties, ka komanda tiek izpildīta uz Zabbix aģenta (noklikšķiniet uz pogas *Zabbix agent* pretī laukam *Execute on*).

Attention:
Ņemiet vērā **sudo** lietojumu — Zabbix lietotājam pēc noklusējuma nav atļauju pārstartēt sistēmas servisus. Skatiet tālāk padomus par to, kā konfigurēt **sudo**.

Note:
Sākot ar Zabbix aģentu 7.0, attālinātās komandas var izpildīt arī uz aģenta, kas darbojas aktīvajā režīmā. Zabbix aģentam — neatkarīgi no tā, vai tas ir aktīvs vai pasīvs — jādarbojas uz attālā hosta, un tas izpilda komandas fonā.

Attālinātās komandas uz Zabbix aģenta tiek izpildītas bez noildzes, izmantojot atslēgu system.run[,nowait], un to izpildes rezultāti netiek pārbaudīti. Uz Zabbix servera un Zabbix starpniekservera attālinātās komandas tiek izpildītas ar noildzi, kas iestatīta TrapperTimeout parametrā failā zabbix_server.conf vai zabbix_proxy.conf, un to izpildes rezultāti tiek pārbaudīti. Papildinformāciju skatiet sadaļā *Skripta noildze*.

Piekļuves atļaujas

Pārliecinieties, ka lietotājam 'zabbix' ir izpildes atļaujas konfigurētajām komandām. Var būt lietderīgi izmantot **sudo**, lai piešķirtu piekļuvi priviliģētām komandām. Lai konfigurētu piekļuvi, izpildiet kā root:

```
visudo
```

Piemēra rindas, ko varētu izmantot *sudoers* failā:

```
# ļauj lietotājam 'zabbix' palaist visas komandas bez paroles.
zabbix ALL=NOPASSWD: ALL

# ļauj lietotājam 'zabbix' pārstartēt apache bez paroles.
zabbix ALL=NOPASSWD: /etc/init.d/apache restart
```

Note:
Dažās sistēmās *sudoers* fails neļaus nelokāliem lietotājiem izpildīt komandas. Lai to mainītu, izkomentējiet opciju **requiretty** failā */etc/sudoers*.

Attālinātās komandas ar vairākām saskarnēm

Ja mērķa sistēmai ir vairākas atlasītā tipa saskarnes (Zabbix aģents vai IPMI), attālinātās komandas tiks izpildītas noklusējuma saskarnē.

Ir iespējams izpildīt attālinātās komandas, izmantojot SSH un Telnet, caur citu saskarni, nevis Zabbix aģenta saskarni. Pieejamā izmantojamā saskarne tiek atlasīta šādā secībā:

- Zabbix aģenta noklusējuma saskarne
- SNMP noklusējuma saskarne
- JMX noklusējuma saskarne
- IPMI noklusējuma saskarne

IPMI attālinātās komandas

IPMI attālinātajām komandām jāizmanto šāda sintakse:

```
<command> [<value>]
```

kur

- <command> - viena no IPMI komandām bez atstarpēm
- <value> - 'on', 'off' vai jebkurš nenegatīvs vesels skaitlis. <value> ir neobligāts parametrs.

Piemēri

Piemēri globālajiem skriptiem, kurus var izmantot kā attālinātās komandas darbību operācijās.

Piemērs 1

Windows restartēšana noteiktā nosacījumā.

Lai automātiski restartētu Windows, kad Zabbix ir konstatējis problēmu, definējiet šādu skriptu:

Skripta parametrs	Vērtība
Darbības joma	'Darbības operācija'
Tips	'Skripts'
Komanda	c:\windows\system32\shutdown.exe -r -f

Piemērs 2

Pārstartējiet hostu, izmantojot IPMI vadību.

Skripta parametrs	Vērtība
Tvērums	'Darbības operācija'
Tips	'IPMI'
Komanda	reset

Piemērs 3

Izslēdziet hostu, izmantojot IPMI vadību.

Skripta parametrs	Vērtība
Tvērums	'Darbības operācija'
Tips	'IPMI'
Komanda	power off

3 Papildu darbības

Pārskats

Šajā sadaļā varat atrast papildu informāciju par papildu operācijām atklāšanas/automātiskās reģistrācijas notikumiem.

Hosta pievienošana

Hosti tiek pievienoti atklāšanas procesa laikā — tiklīdz hosts ir atklāts, nevis atklāšanas procesa beigās.

Note:
Tā kā tīkla atklāšana var aizņemt kādu laiku daudzu nepieejamu hostu/pakalpojumu dēļ, ieteicams būt pacietīgiem un izmantot saprātīgus IP diapazonus.

Pievienojot hostu, tā nosaukumu nosaka standarta **gethostbyname** funkcija. Ja hostu var atrisināt, tiek izmantots atrisinātais nosaukums. Ja nē, tiek izmantota IP adrese. Turklāt, ja hosta nosaukumam jāizmanto IPv6 adrese, tad visas ":" (kolu zīmes) tiek aizstātas ar "_" (pasvītrojuma zīmēm), jo kolu zīmes hostu nosaukumos nav atļautas.

Attention:
Ja atklāšana tiek veikta ar starpniekserveri, pašlaik hosta nosaukuma meklēšana joprojām notiek Zabbix serverī.

Attention:
Ja Zabbix konfigurācijā jau eksistē hosts ar tādu pašu nosaukumu kā jaunatklātajam hostam, Zabbix hosta nosaukumam pievienos **_N**, kur **N** ir pieaugošs skaitlis, sākot no 2.

4 Makrosu izmantošana ziņojumos

Pārskats

Ziņojumu tēmās un ziņojumu tekstā varat izmantot makrosus efektīvākai problēmu ziņošanai.

Papildus vairākiem iebūvētiem makrosiem tiek atbalstīti arī lietotāja makrosi un izteiksmju makrosi. Ir pieejams pilns Zabbix atbalstīto makrosu saraksts.

Piemēri

Šeit sniegtie piemēri ilustrē, kā varat izmantot makrosus ziņojumos.

Piemērs 1

Ziņojuma temats:

```
Problem: {TRIGGER.NAME}
```

Kad saņemsiet ziņojumu, ziņojuma temats tiks aizstāts ar, piemēram:

```
Problem: Processor load is too high on Zabbix server
```

Piemērs 2

Ziņojums:

```
Procesora slodze ir: {?last(/zabbix.zabbix.com/system.cpu.load[,avg1])}
```

Kad saņemsiet ziņojumu, tas tiks aizstāts ar kaut ko līdzīgu šim:

```
Procesora slodze ir: 1.45
```

Piemērs 3

Ziņojums:

```
Jaunākā vērtība: {?last(/{HOST.HOST}/{ITEM.KEY})}
MAX 15 minūtēm: {?max(/{HOST.HOST}/{ITEM.KEY},15m)}
MIN 15 minūtēm: {?min(/{HOST.HOST}/{ITEM.KEY},15m)}
```

Kad saņemsiet ziņojumu, tas tiks aizstāts ar kaut ko līdzīgu:

```
Jaunākā vērtība: 1.45
MAX 15 minūtēm: 2.33
MIN 15 minūtēm: 1.01
```

Piemērs 4

Ziņojums:

```
http://<server_ip_or_name>/zabbix/tr_events.php?triggerid={TRIGGER.ID}&eventid={EVENT.ID}
```

Kad saņemsiet ziņojumu, tas saturēs saiti uz *Notikuma informācijas* lapu, kurā ir sniegta informācija par notikumu, tā trigeri un to jaunāko notikumu sarakstu, kurus ģenerējis tas pats trigeris.

Piemērs 5

Informēšana par vērtībām no vairākiem hostiem trigera izteiksmē.

Ziņojums:

```
Problēmas nosaukums: {TRIGGER.NAME}
Trigera izteiksme: {TRIGGER.EXPRESSION}

1. Vienuma vērtība uz {HOST.NAME1}: {ITEM.VALUE1} ({ITEM.NAME1})
2. Vienuma vērtība uz {HOST.NAME2}: {ITEM.VALUE2} ({ITEM.NAME2})
```

Kad saņemsiet ziņojumu, tas tiks aizstāts ar apmēram šādu tekstu:

```
Problēmas nosaukums: Procesora slodze lokālajā hostā ir pārāk augsta
Trigera izteiksme: last(/Myhost/system.cpu.load[percpu,avg1])>5 or last(/Myotherhost/system.cpu.load[percp

1. Vienuma vērtība uz Myhost: 0.83 (Procesora slodze (1 min vidējā uz kodolu))
2. Vienuma vērtība uz Myotherhost: 5.125 (Procesora slodze (1 min vidējā uz kodolu))
```

Piemērs 6

Saņemt detalizētu informāciju gan par problēmas notikumu, gan atkopšanas notikumu atkopšanas ziņojumā:

Ziņojums:

```
Problēma:

Notikuma ID: {EVENT.ID}
Notikuma vērtība: {EVENT.VALUE}
Notikuma statuss: {EVENT.STATUS}
Notikuma laiks: {EVENT.TIME}
Notikuma datums: {EVENT.DATE}
Notikuma vecums: {EVENT.AGE}
Notikuma apstiprinājums: {EVENT.ACK.STATUS}
Notikuma atjauninājumu vēsture: {EVENT.UPDATE.HISTORY}

Atkopšana:

Notikuma ID: {EVENT.RECOVERY.ID}
Notikuma vērtība: {EVENT.RECOVERY.VALUE}
```

```
Notikuma statuss: {EVENT.RECOVERY.STATUS}
Notikuma laiks: {EVENT.RECOVERY.TIME}
Notikuma datums: {EVENT.RECOVERY.DATE}
Operatīvie dati: {EVENT.OPDATA}
```

Kad saņemsiet ziņojumu, makrosi tiks aizstāti ar kaut ko līdzīgu šim:

```
Problēma:

Notikuma ID: 21874
Notikuma vērtība: 1
Notikuma statuss: PROBLEM
Notikuma laiks: 13:04:30
Notikuma datums: 2018.01.02
Notikuma vecums: 5m 0s
Notikuma apstiprinājums: Jā
Notikuma atjauninājumu vēsture: 2018.01.02 13:05:51 "John Smith (Admin)"
Darbības: apstiprināts.

Atkopšana:

Notikuma ID: 21896
Notikuma vērtība: 0
Notikuma statuss: OK
Notikuma laiks: 13:10:07
Notikuma datums: 2018.01.02
Operatīvie dati: Pašreizējā vērtība ir 0.83
```

3 Atkopšanas darbības

Pārskats

Atkopšanas darbības ļauj jums saņemt paziņojumu, kad problēmas ir atrisinātas.

Atkopšanas darbībās tiek atbalstīti gan ziņojumi, gan attālinātās komandas. Lai gan var pievienot vairākas darbības, eskalācija netiek atbalstīta — visas darbības tiek piešķirtas vienam solim un tādēļ tiks izpildītas vienlaikus.

Lietošanas gadījumi

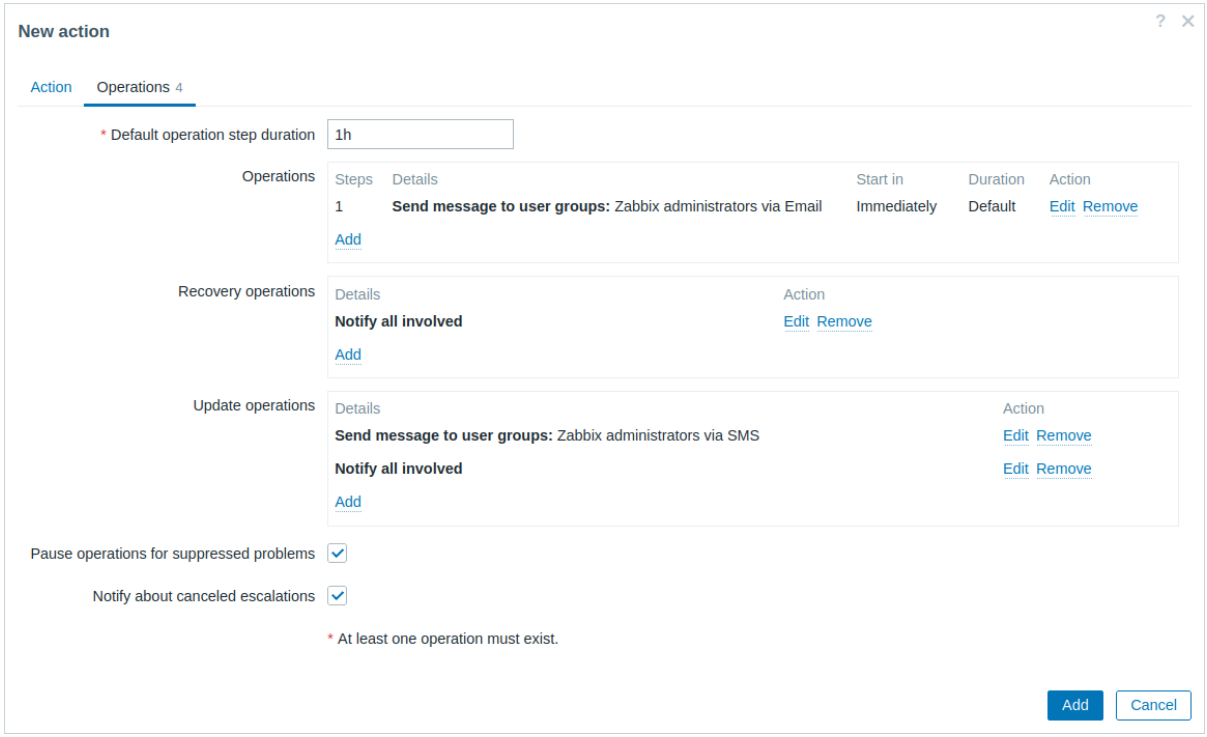
Daži atkopšanas operāciju lietošanas gadījumi ir šādi:

1. Paziņot par atkopšanu visiem lietotājiem, kuriem tika nosūtīts paziņojums par problēmu:
   - Kā operācijas tipu atlasiet *Paziņot visiem iesaistītajiem*.
2. Veikt vairākas operācijas atkopšanas laikā: nosūtīt paziņojumu un izpildīt attālo komandu:
   - Pievienojiet operāciju tipus ziņojuma nosūtīšanai un komandas izpildei.
3. Atvērt pieteikumu ārējā palīdzības dienesta/pieteikumu sistēmā un aizvērt to, kad problēma ir atrisināta:
   - Izveidojiet ārēju skriptu, kas sazinās ar palīdzības dienesta sistēmu.
   - Izveidojiet darbību ar operāciju, kas izpilda šo skriptu un tādējādi atver pieteikumu.
   - Izveidojiet atkopšanas operāciju, kas izpilda šo skriptu ar citiem parametriem un aizver pieteikumu.
   - Izmantojiet makrosu {EVENT.ID}, lai atsauktos uz sākotnējo problēmu.

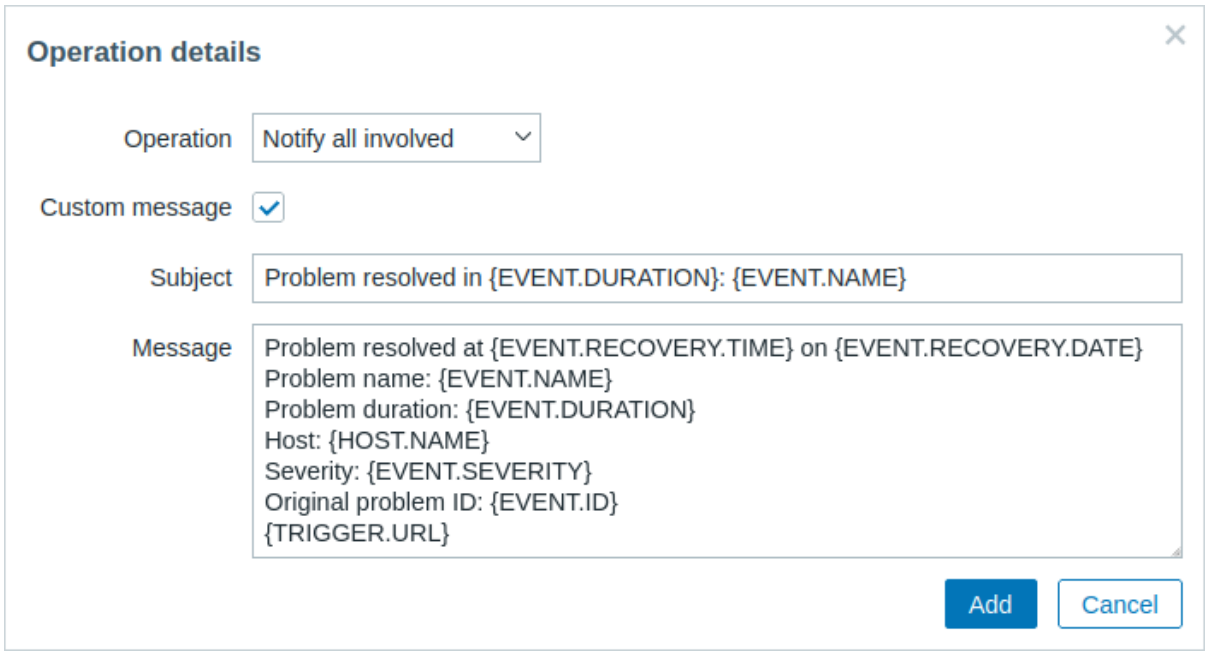
Atkopšanas operācijas konfigurēšana

Lai konfigurētu atkopšanas operāciju, veiciet šādas darbības:

1. Dodieties uz cilni *Operations* darbības konfigurācijas formā.
2. Lai konfigurētu jaunu atkopšanas operāciju, sadaļā *Recovery operations* noklikšķiniet uz *Add*. Lai rediģētu esošu operāciju, blakus operācijai noklikšķiniet uz *Edit*.
3. Konfigurējiet operācijas informāciju.



Atkopšanas operācijas informācija



Atkopšanas notikumiem ir pieejami trīs operāciju tipi:

- **Sūtīt ziņojumu** — nosūtīt atkopšanas ziņojumu norādītajam lietotājam;
- **Paziņot visiem iesaistītajiem** — nosūtīt atkopšanas ziņojumu visiem lietotājiem, kuri saņēma paziņojumu par problēmas notikumu;
- **<attālās komandas nosaukums>** — izpildīt attālo komandu. Komandas ir pieejamas izpildei, ja tās iepriekš definētas globālajos skriptos, izvēloties *Darbības operācija* kā to tvērumu.

Tālāk ir aprakstīti katra operācijas tipa parametri. Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti. Kad pabeigts, noklikšķiniet uz *Pievienot*, lai pievienotu operāciju *Atkopšanas operāciju* sarakstam.

Note:
Ņemiet vērā, ka, ja viens un tas pats saņēmējs ir definēts vairākos operāciju tipos bez norādīta *Pielāgota ziņojuma*, dublēti paziņojumi netiek sūtīti.

Operācijas tips: sūtīt ziņojumu

Parametrs		Apraksts
Sūtīt lietotāju grupām		Atlasiet lietotāju grupas, kurām nosūtīt atkopšanas ziņojumu. Lai saņemtu paziņojumu, lietotāju grupai jābūt vismaz ”read” atļaujām uz hostu.
Sūtīt lietotājiem		Atlasiet lietotājus, kuriem nosūtīt atkopšanas ziņojumu. Lai saņemtu paziņojumu, lietotājam jābūt vismaz ”read” atļaujām uz hostu.
Sūtīt uz medija tipu		Nosūtīt noklusējuma atkopšanas ziņojumu uz visiem pieejamajiem (konfigurētajiem un iespējotajiem) mediju tipiem vai tikai uz konkrētu tipu.
Pielāgots ziņojums		Ja atlasīts, var definēt pielāgotu ziņojumu.
	Temats	Pielāgotā ziņojuma temats. Temats var saturēt makrosus.
	Ziņojums	Pielāgotais ziņojums. Ziņojums var saturēt makrosus.

Operācijas tips: attālinātā komanda

Parametrs	Apraksts
Mērķu saraksts	Izvēlieties mērķus, uz kuriem izpildīt komandu: **Pašreizējais hosts** - komanda tiek izpildīta uz trigera hosta, kas izraisīja problēmas notikumu. Šī opcija nedarbosies, ja trigerī ir vairāki hosti. **Hosts** - izvēlieties hostu(s), uz kuriem izpildīt komandu. **Hostu grupa** - izvēlieties hostu grupu(as), uz kurām izpildīt komandu. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas visas tajā ietvertās apakšgrupas. Tādējādi attālinātā komanda tiks izpildīta arī uz hostiem no apakšgrupām. Komanda uz hosta tiek izpildīta tikai vienu reizi, pat ja hosts atbilst vairāk nekā vienu reizi (piemēram, no vairākām hostu grupām; individuāli un no hostu grupas). Mērķu sarakstam nav nozīmes, ja komanda tiek izpildīta uz Zabbix servera. Izvēloties vairāk mērķu šajā gadījumā, komanda tikai tiks izpildīta uz servera vairākas reizes. Ņemiet vērā, ka globālajiem skriptiem mērķa atlase ir atkarīga arī no iestatījuma *Hostu grupa* globālā skripta konfigurācijā.

Operācijas tips: paziņot visiem iesaistītajiem

Parametrs		Apraksts
Pielāgots ziņojums		Ja atlasīts, var definēt pielāgotu ziņojumu.
	Temats	Pielāgotā ziņojuma temats. Temats var saturēt makrosus.
	Ziņojums	Pielāgotais ziņojums. Ziņojums var saturēt makrosus.

4 Atjaunināšanas darbības

Pārskats

Atjaunināšanas darbības ir pieejamas darbībās ar šādiem notikumu avotiem:

- *Trigeri* — kad problēmas tiek atjaunināta citu lietotāju darbību rezultātā,
  t. i., tām tiek pievienoti komentāri, tās tiek apstiprinātas, tiek mainīta to nopietnība, tās tiek aizvērtas (manuāli);
- *Pakalpojumi* — kad pakalpojuma nopietnība ir mainījusies, bet pakalpojums vēl nav atjaunots.

Lūdzu, ņemiet vērā, ka lietotāji nesaņem paziņojumus par saviem pašu atjaunājumiem.

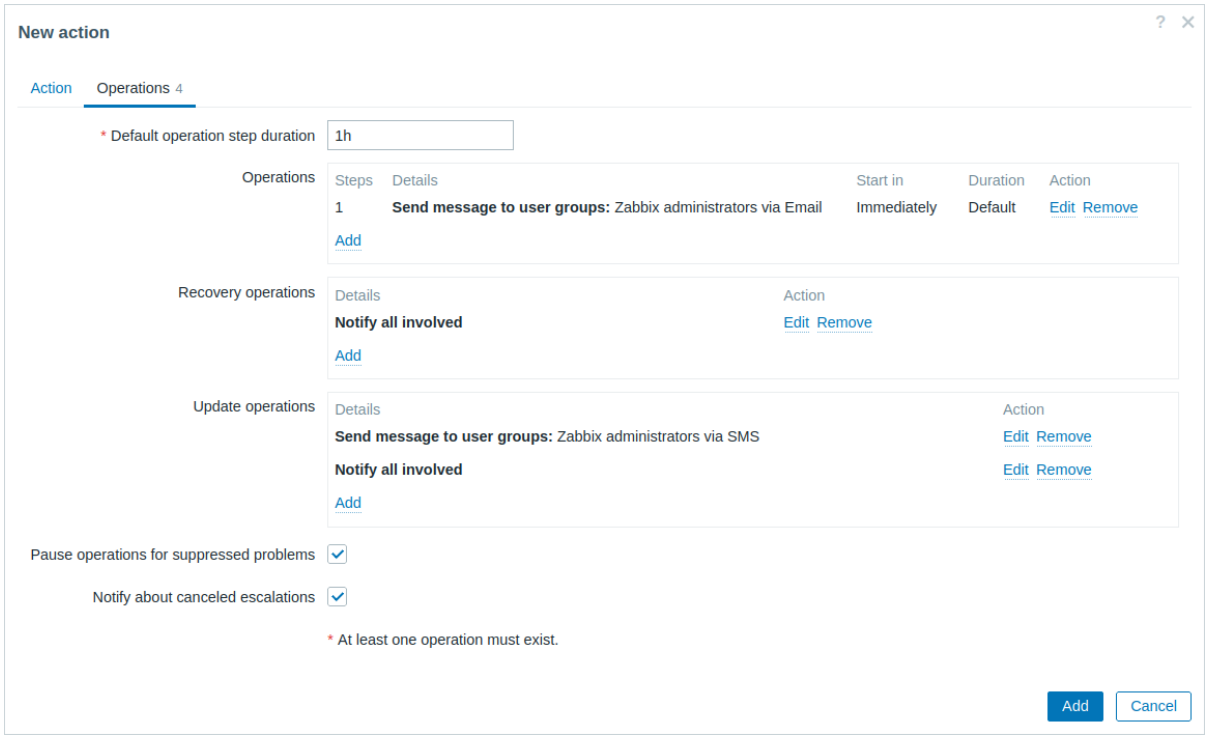
Atjaunināšanas darbībās tiek atbalstīti gan ziņojumi, gan attālinātās komandas. Lai gan var pievienot vairākas darbības, eskalācija netiek atbalstīta — visas darbības tiek piešķirtas vienam solim un tāpēc tiks izpildītas vienlaikus.

Atjaunināšanas operācijas konfigurēšana

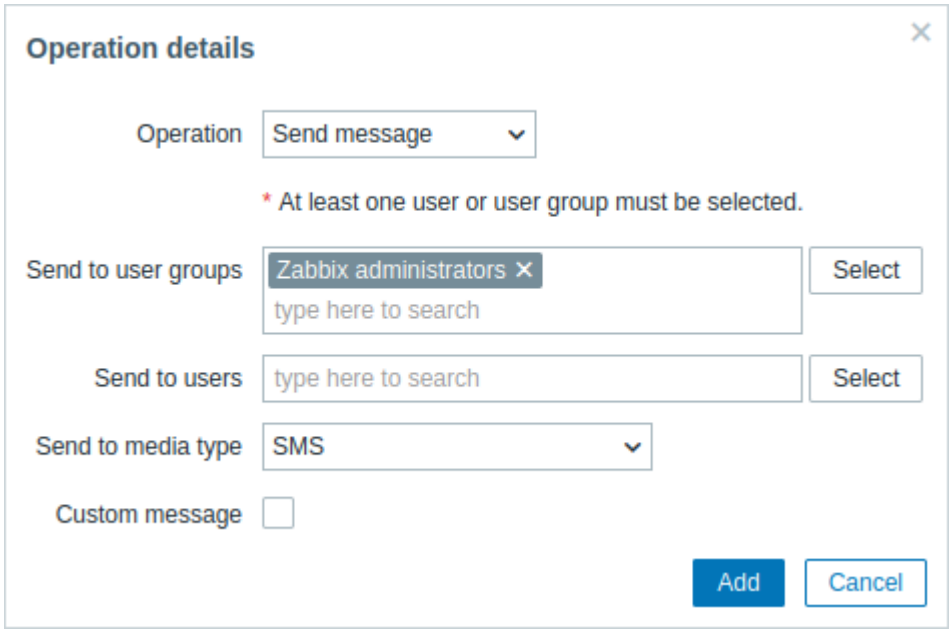
Lai konfigurētu atjaunināšanas operāciju, veiciet šādas darbības:

1. Atveriet cilni *Operations* darbības konfigurācijas formā.

2. Lai konfigurētu jaunu atjaunināšanas operāciju, sadaļā *Update operations* noklikšķiniet uz *Add*. Lai rediģētu esošu operāciju, blakus operācijai noklikšķiniet uz *Edit*.
3. Konfigurējiet operācijas informāciju.



Atjaunināšanas operācijas informācija



Atjaunināšanas operācijas piedāvā to pašu parametru kopu kā Atkopšanas operācijas.

5 Eskalācijas

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.

Dažādi eskalācijas darbības aspekti

Apskatīsim, kas notiek dažādos apstākļos, ja darbība satur vairākus eskalācijas soļus.

Situācija	Darbība
Attiecīgais hosts tiek ievietots uzturēšanā pēc sākotnējā problēmas paziņojuma nosūtīšanas	Atkarībā no iestatījuma *Pause operations for suppressed problems* darbības konfigurācijā, visi atlikušie eskalācijas soļi tiek izpildīti vai nu ar uzturēšanas perioda izraisītu aizkavi, vai bez aizkaves. Uzturēšanas periods neatceļ operācijas.
*Darbības nosacījumā **Time period** definētais laika periods beidzas pēc sākotnējā paziņojuma nosūtīšanas*	Visi atlikušie eskalācijas soļi tiek izpildīti. Nosacījums *Time period* nevar apturēt operācijas; tas ietekmē tikai to, kad darbības tiek sāktas vai netiek sāktas, nevis operācijas.
Problēma sākas uzturēšanas laikā un turpinās (netiek atrisināta) pēc uzturēšanas beigām	Atkarībā no iestatījuma *Pause operations for suppressed problems* darbības konfigurācijā, visi eskalācijas soļi tiek izpildīti vai nu no brīža, kad uzturēšana beidzas, vai nekavējoties.
Problēma sākas no-data uzturēšanas laikā un turpinās (netiek atrisināta) pēc uzturēšanas beigām	Jāgaida, līdz nostrādā trigeris, pirms tiek izpildīti visi eskalācijas soļi.
Dažādas eskalācijas seko cita citai īsā laikā un pārklājas	Katras jaunas eskalācijas izpilde aizstāj iepriekšējo eskalāciju, taču iepriekšējai eskalācijai vienmēr tiek izpildīts vismaz viens eskalācijas solis. Šī darbība ir būtiska darbībām uz notikumiem, kas tiek izveidoti ar KATRU trigeris problēmas novērtējumu.
Notiekošas eskalācijas laikā (piemēram, tiek sūtīts ziņojums), pamatojoties uz jebkura veida notikumu: - darbība tiek atspējota Pamatojoties uz trigeris notikumu: - trigeris tiek atspējots - hosts vai vienums tiek atspējots Pamatojoties uz iekšēju notikumu par trigeriem: - trigeris tiek atspējots Pamatojoties uz iekšēju notikumu par vienumiem/zema līmeņa atklāšanas kārtulām: - vienums tiek atspējots - hosts tiek atspējots	Notiekošais ziņojums tiek nosūtīts, un pēc tam eskalācijas ietvaros tiek nosūtīts vēl viens ziņojums. Sekojošā ziņojuma pamatteksta sākumā būs atcelšanas teksts (*NOTE: Escalation canceled*), norādot iemeslu (piemēram, *NOTE: Escalation canceled: action '<Action name>' disabled*). Tādējādi saņēmējs tiek informēts, ka eskalācija ir atcelta un turpmāki soļi netiks izpildīti. Šis ziņojums tiek nosūtīts visiem, kuri iepriekš saņēma paziņojumus. Atcelšanas iemesls tiek arī reģistrēts serveris žurnālfailā (sākot no Debug Level 3=Warning). Ņemiet vērā, ka ziņojums *Escalation canceled* tiek nosūtīts arī tad, ja operācijas ir pabeigtas, bet atkopšanas operācijas ir konfigurētas un vēl nav izpildītas.
Notiekošas eskalācijas laikā (piemēram, tiek sūtīts ziņojums) darbība tiek dzēsta	Vairāk ziņojumu netiek sūtīts. Informācija tiek reģistrēta serveris žurnālfailā (sākot no Debug Level 3=Warning), piemēram: `escalation canceled: action id:334 deleted`

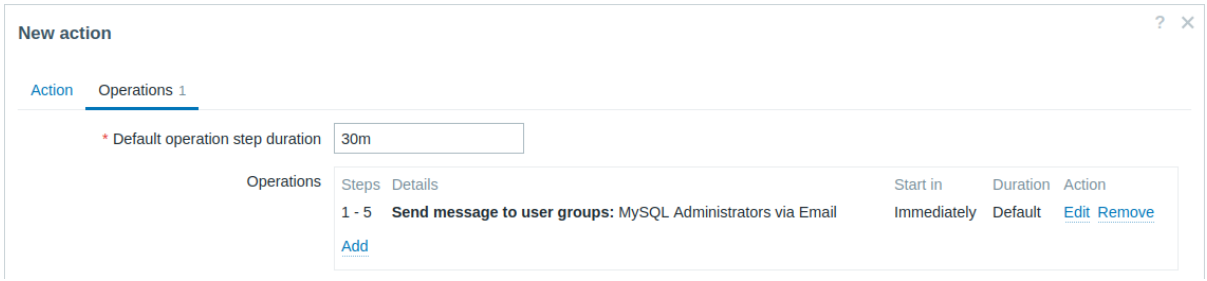
Eskalācijas piemēri

Piemērs 1

Atkārtota paziņojuma sūtīšana reizi 30 minūtēs (kopā 5 reizes) grupai "MySQL Administrators". Lai to konfigurētu:

- Cilnē *Operations* iestatiet *Default operation step duration* uz "30m" (30 minūtes).
- Iestatiet eskalācijas *Steps* no "1" līdz "5".

- Atlasiet grupu ”MySQL Administrators” kā ziņojuma saņēmējus.



Paziņojumi tiks nosūtīti pēc 0:00, 0:30, 1:00, 1:30 un 2:00 stundām kopš problēmas sākuma (ja vien, protams, problēma netiek atrisināta agrāk).

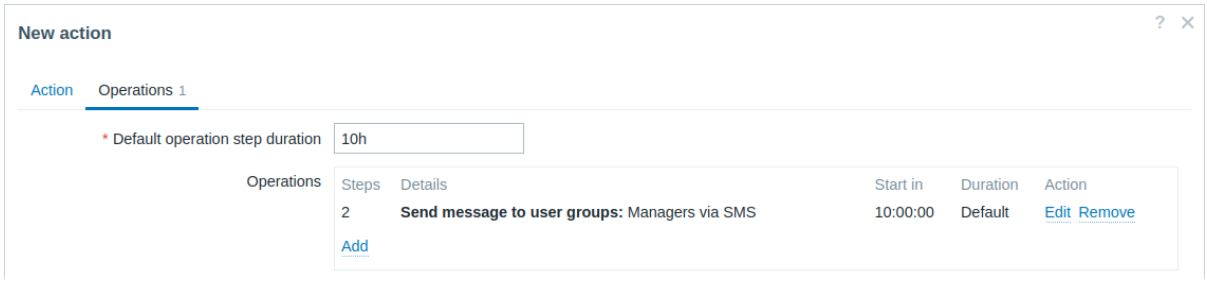
Ja problēma tiek atrisināta un ir konfigurēts atkopšanas ziņojums, tas tiks nosūtīts tiem, kuri šajā eskalācijas scenārijā saņēma vismaz vienu problēmas ziņojumu.

Note:
Ja trigeris, kas izveidoja aktīvu eskalāciju, tiek atspējots, Zabbix nosūta informatīvu ziņojumu visiem tiem, kuri jau ir saņēmuši paziņojumus.

Piemērs 2

Aizkavēta paziņojuma nosūtīšana par ilgstošu problēmu. Lai to konfigurētu:

- Cilnē *Operations* iestatiet *Default operation step duration* uz ”10h” (10 stundas).
- Iestatiet eskalācijas *Steps* no ”2” līdz ”2”.



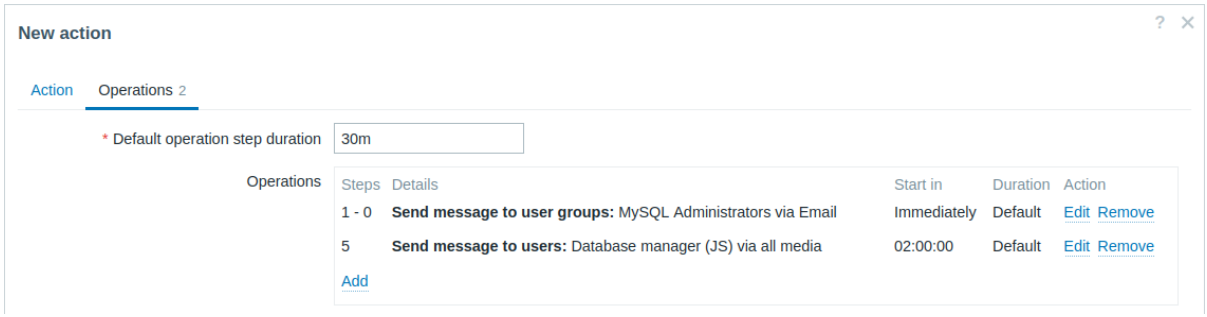
Paziņojums tiks nosūtīts tikai eskalācijas scenārija 2. solī, jeb 10 stundas pēc problēmas sākuma.

Varat pielāgot ziņojuma tekstu, piemēram: ”Problēma ilgst jau vairāk nekā 10 stundas”.

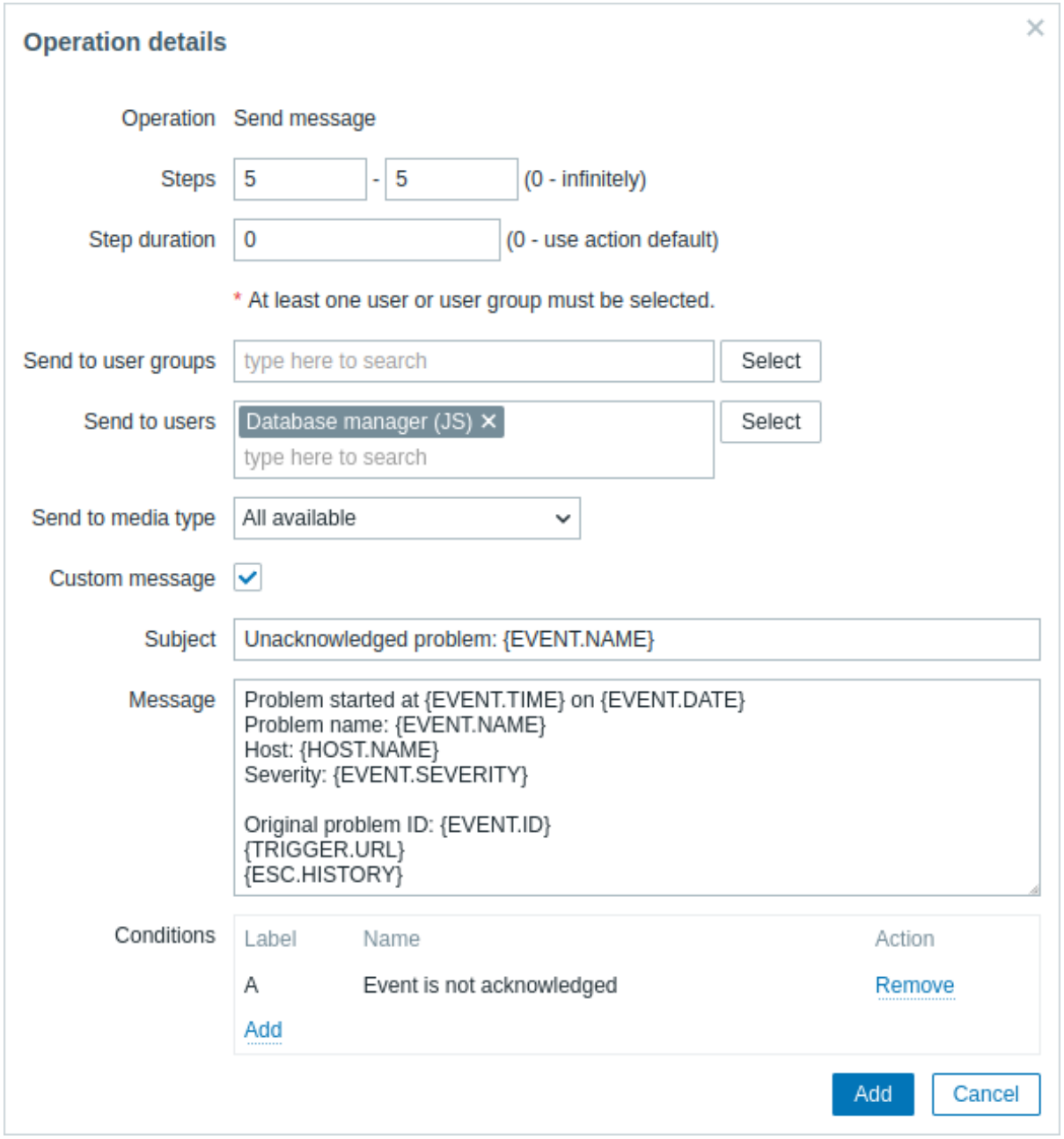
Piemērs 3

Problēmas eskalēšana vadītājam.

Pirmajā iepriekš minētajā piemērā mēs konfigurējām periodisku ziņojumu sūtīšanu MySQL administratoriem. Šajā gadījumā administratori saņems četrus ziņojumus, pirms problēma tiks eskalēta datubāzes pārvaldniekam. Ņemiet vērā, ka pārvaldnieks saņems ziņojumu tikai tad, ja problēma vēl nebūs apstiprināta, t. i., visticamāk, neviens ar to vēl nestrādā.



2. darbības informācija:



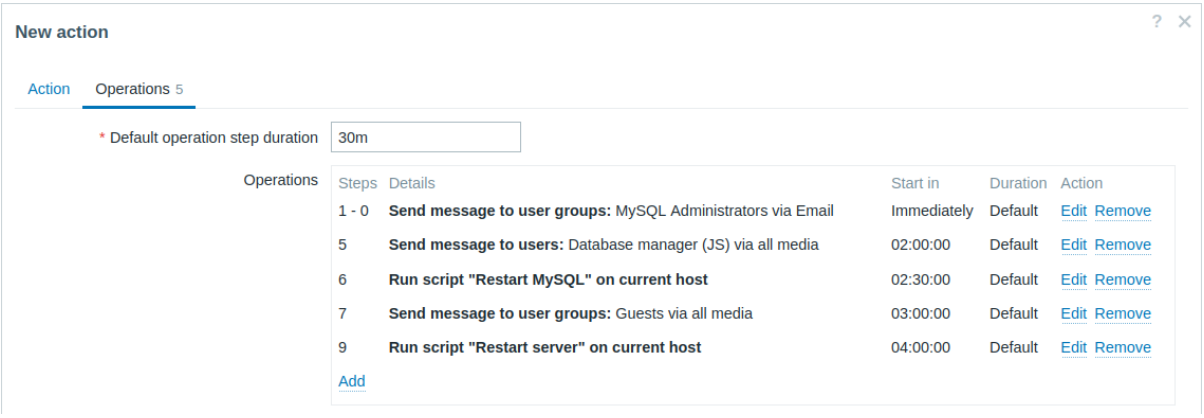
Ņemiet vērā pielāgotajā ziņojumā izmantoto makrosu {ESC.HISTORY}. Šis makross saturēs informāciju par visām iepriekš izpildītajām šīs eskalācijas darbībām, piemēram, nosūtītajiem paziņojumiem un izpildītajām komandām.

Piemērs 4

Sarežģītāks scenārijs. Pēc vairākiem ziņojumiem MySQL administratoriem un eskalācijas vadītājam, Zabbix mēģinās restartēt MySQL datu bāzi. Tas notiks, ja problēma pastāvēs 2:30 stundas un tā nebūs apstiprināta.

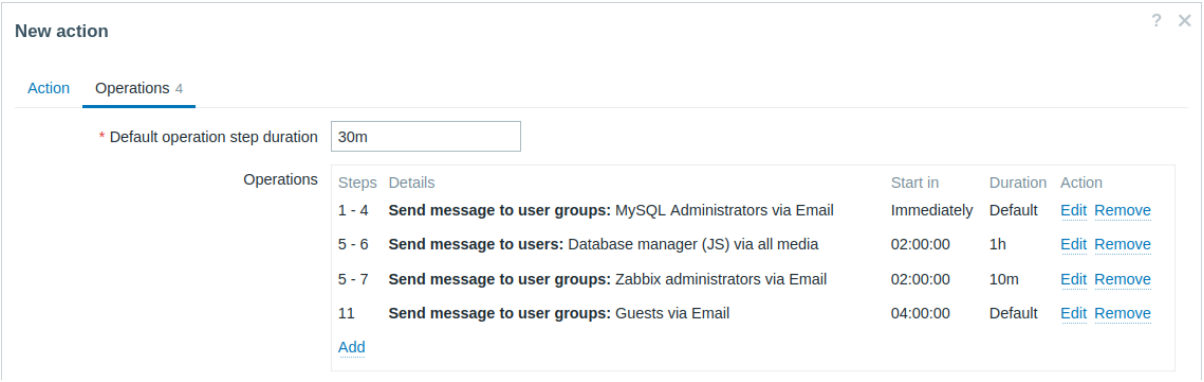
Ja problēma joprojām pastāvēs, pēc vēl 30 minūtēm Zabbix nosūtīs ziņojumu visiem vieslietotājiem.

Ja tas nepalīdzēs, pēc vēl vienas stundas Zabbix pārstartēs serveri ar MySQL datu bāzi (otrā attālā komanda), izmantojot IPMI komandas.



Piemērs 5

Eskalācija ar vairākām operācijām, kurām pārklājas soļu diapazoni un ir pielāgoti intervāli. Noklusējuma operācijas soļa ilgums ir 30 minūtes.



Paziņojumi tiks nosūtīti šādi:

- MySQL administratoriem 0:00, 0:30, 1:00 un 1:30 pēc problēmas sākuma.
- Database manager 2:00 un 2:10 (īsākais turpmākajā operācijā definētais pielāgotais soļa ilgums — 10 minūtes — aizstāj šeit konfigurēto garāko soļa ilgumu — 1 stundu, kā aprakstīts sadaļā Operation details parametram *Step duration*, ja soļi pārklājas).
- Zabbix administratoriem 2:00, 2:10 un 2:20 pēc problēmas sākuma (tiek piemērots pielāgotais soļa ilgums — 10 minūtes).
- vieslietotājiem 4:00 pēc problēmas sākuma (starp 8. un 11. soli stājas spēkā noklusējuma soļa ilgums — 30 minūtes).

3 Paziņojuma saņemšana par neatbalstītiem vienumiem

Pārskats

Zabbix ir iespējams saņemt paziņojumus par neatbalstītiem vienumiem.

Tā ir daļa no Zabbix iekšējo notikumu koncepcijas, kas ļauj lietotājiem saņemt paziņojumus šādos gadījumos. Iekšējie notikumi atspoguļo stāvokļa izmaiņas:

- kad vienumi pāriet no “normāls” uz “neatbalstīts” (un atpakaļ);
- kad trigeri pāriet no “normāls” uz “nezināms” (un atpakaļ);
- kad zema līmeņa atklāšanas kārtulas pāriet no “normāls” uz “neatbalstīts” (un atpakaļ).

Šajā sadaļā ir aprakstīts, kā **saņemt paziņojumu**, kad vienums kļūst neatbalstīts.

Konfigurācija

Kopumā paziņojuma iestatīšanas procesam vajadzētu būt pazīstamam tiem, kuri iepriekš jau ir konfigurējuši brīdinājumus Zabbix.

1. solis

Konfigurējiet kādu paziņojumu nesēju, piemēram, e-pastu, SMS vai skriptu, ko izmantot paziņojumiem. Lai veiktu šo uzdevumu, skatiet attiecīgās rokasgrāmatas sadaļas.

Attention:
Paziņojumiem par iekšējiem notikumiem tiek izmantota noklusējuma nopietnības pakāpe (’Nav klasificēts’), tāpēc atstājiet to atzīmētu, konfigurējot lietotāja paziņojumu nesējus, ja vēlaties saņemt paziņojumus par iekšējiem notikumiem.

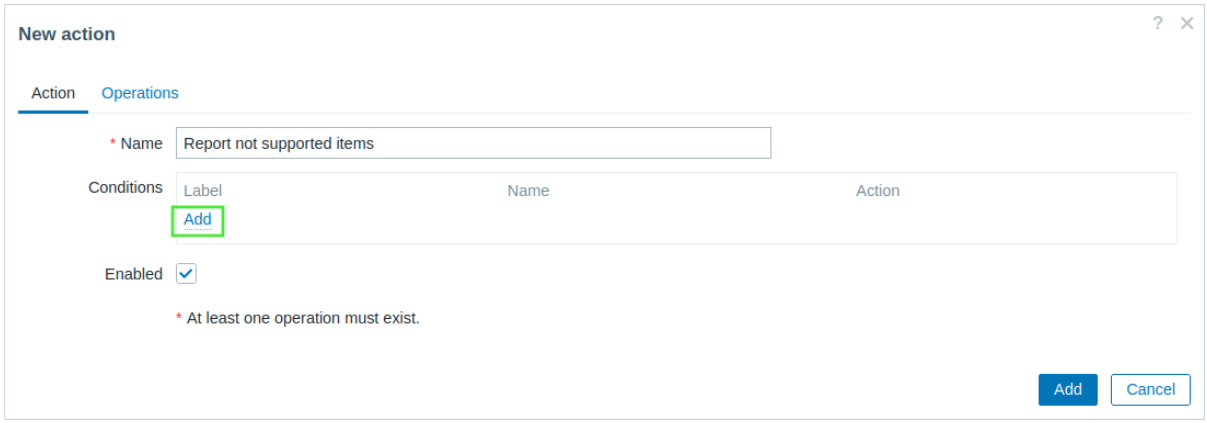
2. solis

Dodieties uz *Brīdinājumi → Darbības → Iekšējās darbības*.

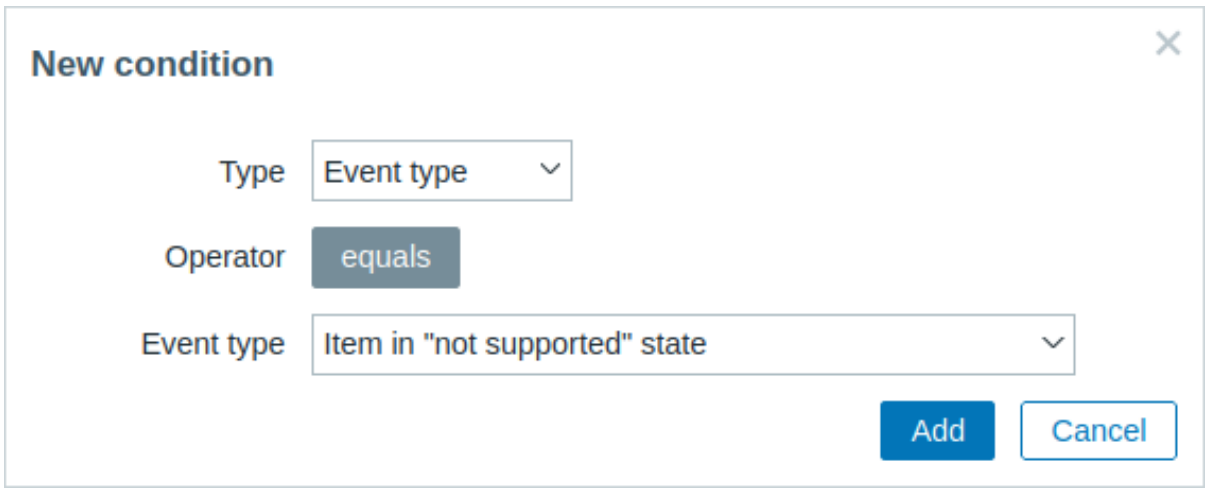
Lapas augšējā labajā stūrī noklikšķiniet uz *Izveidot darbību*, lai atvērtu darbības konfigurācijas formu.

3. solis

Cilnē *Action* ievadiet darbības nosaukumu. Pēc tam *Conditions* blokā noklikšķiniet uz *Add*, lai pievienotu jaunu nosacījumu.



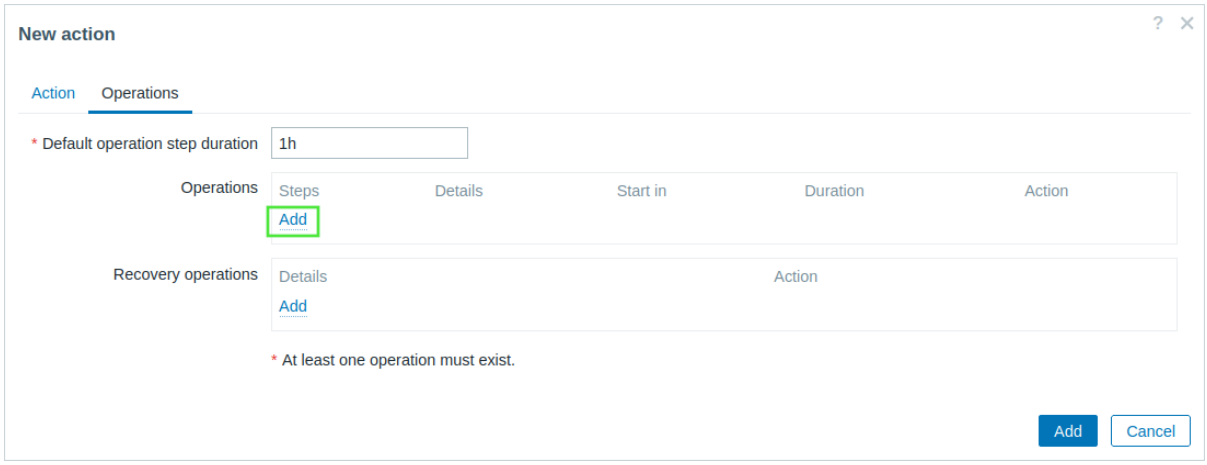
Uznirstošajā logā *New condition* atlasiet ”Event type” kā nosacījuma tipu un pēc tam atlasiet ”Item in ’not supported’ state” kā notikuma tipu.



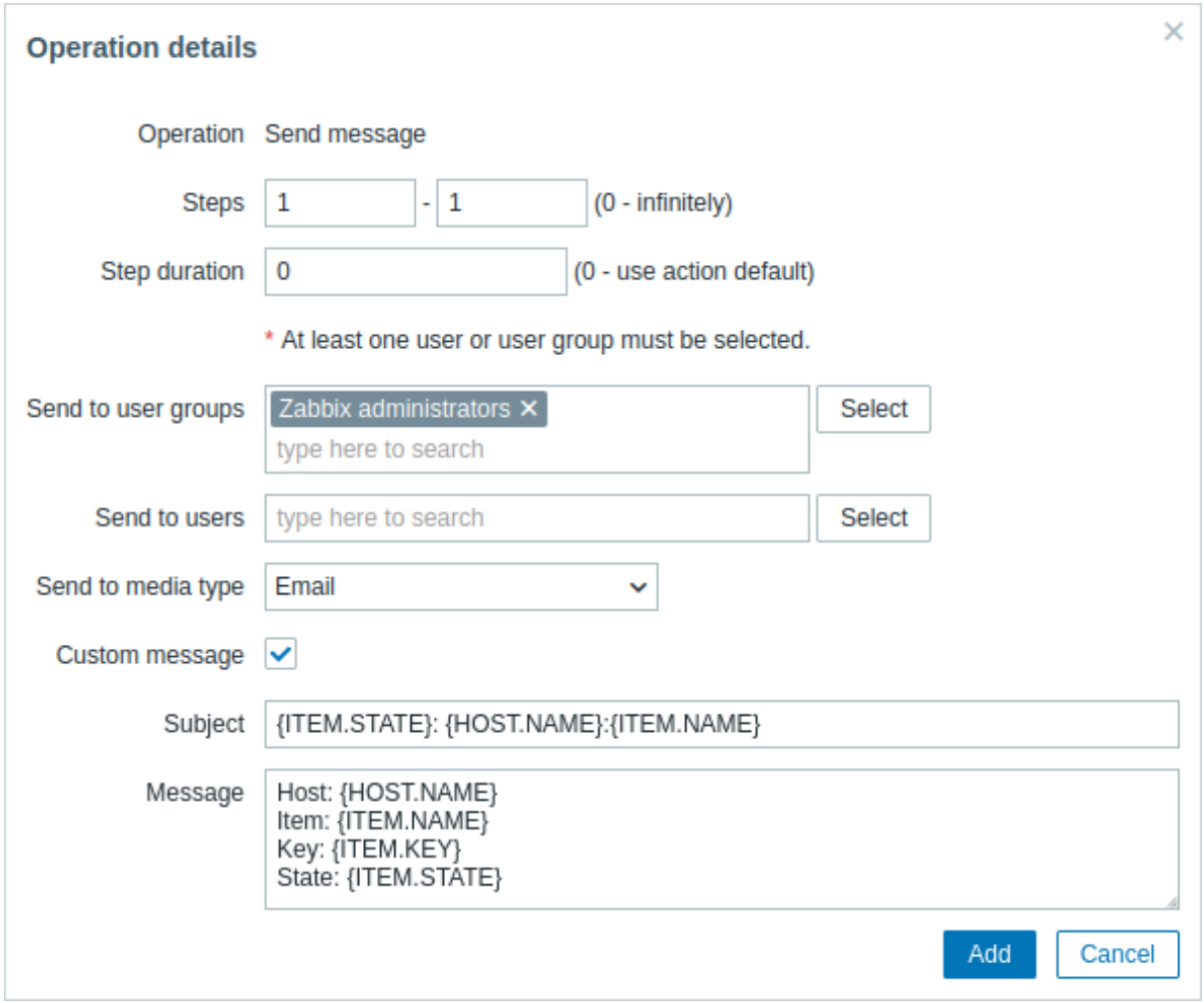
Neaizmirstiet noklikšķināt uz *Add*, lai nosacījums patiešām tiktu pievienots sarakstam *Conditions* blokā.

4. solis

Cilnē *Operations* noklikšķiniet uz *Add* blokā *Operations*, lai pievienotu jaunu operāciju.



Atlasiet dažus ziņojuma saņēmējus (lietotāju grupas/lietotājus) un piegādei izmantojamo medija tipu (vai ”All”). Atzīmējiet izvēles rūtiņu *Custom message*, ja vēlaties ievadīt pielāgotu problēmas ziņojuma tēmu/saturu.

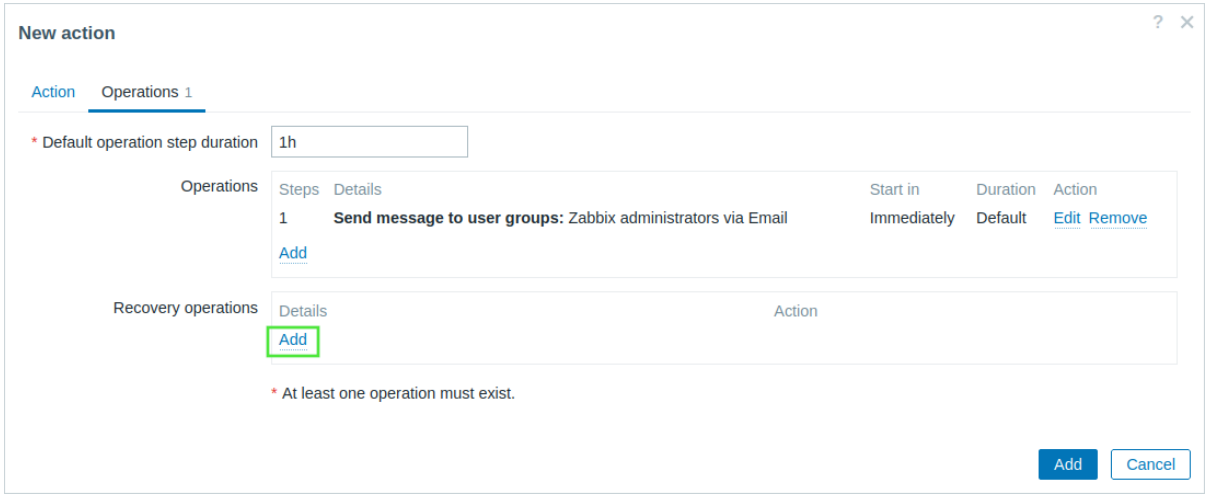


Noklikšķiniet uz *Add*, lai operācija patiešām tiktu parādīta blokā *Operations*.

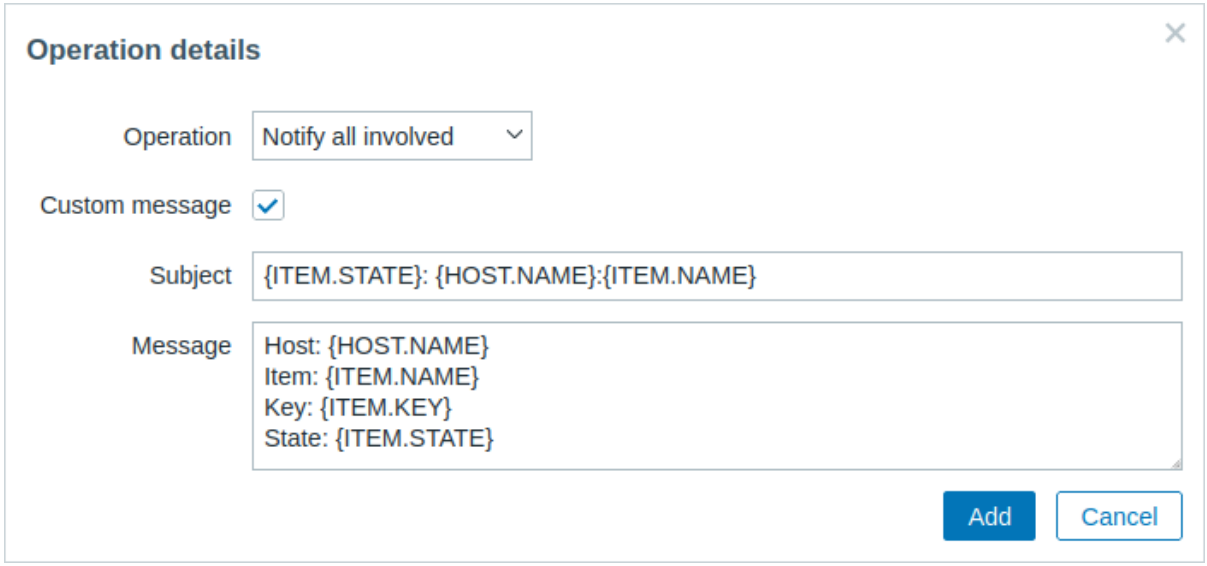
Ja vēlaties saņemt vairāk nekā vienu paziņojumu, iestatiet operācijas soļa ilgumu (intervālu starp nosūtītajiem ziņojumiem) un pievienojiet vēl vienu soli.

5. solis

Bloks *Atkopšanas darbības* ļauj konfigurēt atkopšanas paziņojumu, kad vienums atgriežas normālā stāvoklī. Noklikšķiniet uz *Pievienot* blokā *Atkopšanas darbības*, lai pievienotu jaunu atkopšanas darbību.



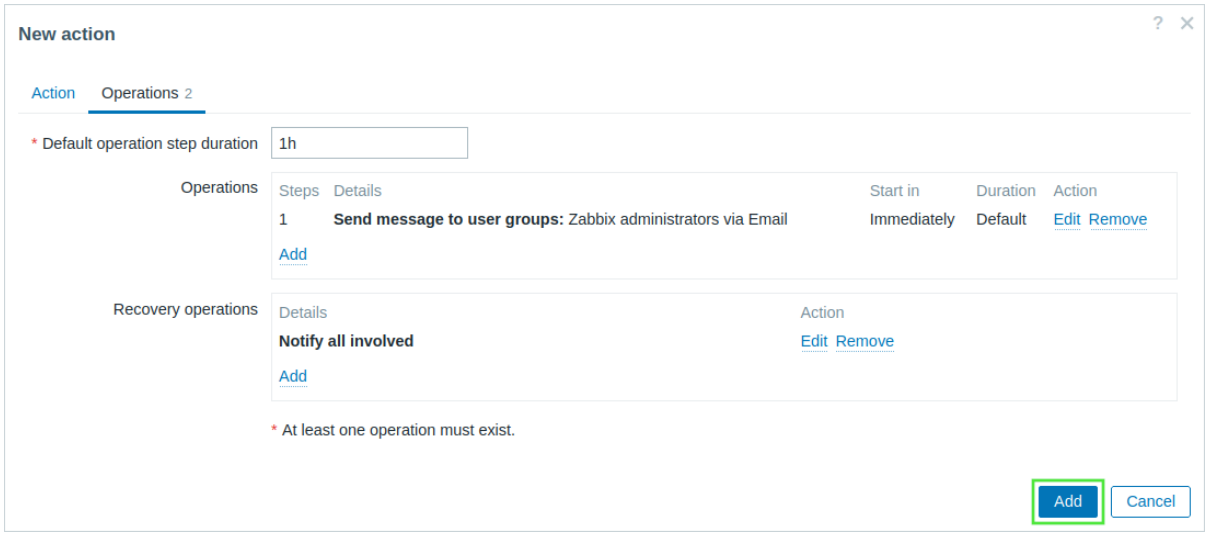
Atlasiet darbības tipu ”Paziņot visiem iesaistītajiem”. Atlasiet izvēles rūtiņu *Pielāgots ziņojums*, ja vēlaties ievadīt pielāgotu atkopšanas ziņojuma tēmu/saturu.



Noklikšķiniet uz *Pievienot* uznirstošajā logā *Darbības informācija*, lai darbība tiktu pievienota blokā *Atkopšanas darbības*.

6. solis

Kad esat pabeidzis, noklikšķiniet uz pogas *Pievienot* formas apakšdaļā.



Un tas arī viss — esat pabeidzis! Tagad varat gaidīt savu pirmo paziņojumu no Zabbix, ja kāds vienums kļūs neatbalstīts.

11 Makrosi

Pārskats

Zabbix atbalsta vairākus iebūvētus makrosus, kurus var izmantot dažādās situācijās. Šie makrosi ir mainīgie, kurus identificē pēc noteiktas sintakses:

`{MACRO}`

Makrosi tiek aizvietoti ar noteiktu vērtību atkarībā no konteksta.

Efektīva makrosu izmantošana ļauj ietaupīt laiku un padarīt Zabbix konfigurāciju pārskatāmāku.

Vienā no tipiskajiem lietošanas gadījumiem makrosu var izmantot veidnē. Tādējādi trigeri veidnē var nosaukt, piemēram, "Procesora slodze ir pārāk augsta uz {HOST.NAME}". Kad veidne tiek piemērota hostam, piemēram, Zabbix serverim, nosaukums tiks aizvietots ar "Procesora slodze ir pārāk augsta uz Zabbix servera", kad trigeris tiek parādīts sadaļā Uzraudzība.

Makrosus var izmantot vienuma atslēgas parametros. Makrosu var izmantot tikai parametra daļai, piemēram, `item.key[server_{HOST.HOS`
Parametru nav nepieciešams likt dubultpēdiņās, jo Zabbix parūpēsies par jebkādiem neviennozīmīgiem speciālajiem simboliem, ja tādi būs aizvietotajā makrosā.

Zabbix ir arī citi makrosu veidi.

Zabbix atbalsta šādus makrosus:

- {MACRO} - iebūvēts makross (skatiet pilnu sarakstu)
- {<macro>.<func>(<params>)} - makrosu funkcijas
- {$MACRO} - lietotāja definēts makross, pēc izvēles ar kontekstu
- {#MACRO} - makross zema līmeņa atklāšanai
- {?EXPRESSION} - izteiksmes makross

1 Makro funkcijas

Pārskats

Makro funkcijas piedāvā iespēju pielāgot makro vērtības (piemēram, saīsināt vai izvilkt noteiktas apakšvirknes), padarot tās ērtāk lietojamas.

Makro funkcijas sintakse ir:

```
{macro.func(params)}
```

kur

- **macro** - pielāgojamais makro;
- **func** - piemērojamā funkcija (skatiet atbalstītās funkcijas);
- **params** - ar komatiem atdalīts funkcijas parametru saraksts, kam jābūt **dubultpēdiņās**, ja tie:
  - sākas ar atstarpi vai dubultpēdiņām;
  - satur aizverošās iekavas vai komatu.

Piemēram:

```
{{TIME}.fmttime(format,time_shift)}
{{ITEM.VALUE}.regsub(pattern, output)}
{{$USERMACRO}.regsub(pattern, output)}
{{#LLDMACRO}.regsub(pattern, output)}
```

Makro funkcijas tiek atbalstītas:

- Iebūvētajiem makro
- Lietotāja makro
- Zema līmeņa atklāšanas makro
- Izteiksmju makro

Makro funkcijas var izmantot visās vietās, kur tiek atbalstīti uzskaitītie makro. Tas attiecas uz visiem gadījumiem, izņemot, ja ir skaidri norādīts, ka tiek sagaidīts tikai makro (piemēram, konfigurējot hostu makro vai zema līmeņa atklāšanas noteikumu filtrus).

Katram makro tiek atbalstīta viena funkcija; vairākas makro funkcijas ķēdē netiek atbalstītas.

> **Note:**
> Lūdzu, skatiet ekranēšanas piemērus gadījumiem, kad makro funkcijas tiek izmantotas citos kontekstos (funkcijā, vienuma atslēgā, citā makro utt.).

Atbalstītās funkcijas

Funkcijas ir uzskaitītas bez papildu informācijas. Noklikšķiniet uz funkcijas, lai skatītu pilnu informāciju.

Funkcija	Apraksts
btoa	Makro vērtības kodēšana Base64 kodējumā.
fmtnum	Skaitļu formatēšana, lai kontrolētu aiz komata drukāto ciparu skaitu.
fmttime	Laika formatēšana.
htmldecode	Makro vērtības dekodēšana no HTML kodējuma.
htmlencode	Makro vērtības kodēšana HTML kodējumā.
iregsub	Apakšvirknes iegūšana pēc atbilstības regulārajai izteiksmei (reģistrnejutīga).
lowercase	Makro vērtības rakstzīmju pārveidošana uz mazajiem burtiem.
regrepl	Rakstzīmes/apakšvirknes aizstāšana makro vērtībā.
regsub	Apakšvirknes iegūšana pēc atbilstības regulārajai izteiksmei (reģistrjutīga).
tr	Makro vērtības rakstzīmju transliterācija.
uppercase	Makro vērtības rakstzīmju pārveidošana uz lielajiem burtiem.
urldecode	Makro vērtības dekodēšana no URL kodējuma.

Funkcija	Apraksts
urlencode	Makro vērtības kodēšana URL kodējumā.

Funkcijas informācija

Neobligātie funkcijas parametri ir norādīti ar < >.

btoa

Makro vērtības kodēšana Base64 kodējumā. Base64 kodējums ir metode bināro datu attēlošanai teksta formā, kas ir noderīga bināra satura glabāšanai un drošai pārsūtīšanai, izmantojot uz tekstu balstītus protokolus.

Piemērs:

```
{{ITEM.VALUE}.btoa()} - Base64 kodēs vērtību, piemēram, "zabbix", par "emFiYml4"
```

fmtnum(digits)

Skaitļa formatēšana, lai kontrolētu pēc decimālpunkta izdrukāto ciparu skaitu.

Parametri:

- **digits** - ciparu skaits aiz decimālpunkta. Derīgais diapazons: 0-20. Beigu nulles netiks pievienotas.

Piemēri:

```
{{ITEM.VALUE}.fmtnum(2)} - atgriezīs "24.35" no saņemtās vērtības "24.3483523"
{{ITEM.VALUE}.fmtnum(0)} - atgriezīs "24" no saņemtās vērtības "24.3483523"
```

fmttime(format,<time_shift>)

Laika formatēšana.
 Ņemiet vērā, ka šo funkciju var izmantot ar makrosiem, kas tiek atrisināti uz vērtību vienā no šādiem laika formātiem:

- `hh:mm:ss`
- `yyyy-mm-ddThh:mm:ss[tz]` (ISO8601 standarts)
- UNIX laikspiedols

Parametri:

- **format** - obligāta formāta virkne, kas ir saderīga ar `strftime` funkcijas formatēšanu;
- **time_shift** (neobligāts) - laika nobīde, kas tiek piemērota laikam pirms formatēšanas; tai jāsākas ar `-<N><time_unit>` vai `+<N><time_unit>`, kur:
  - `N` - pievienojamo vai atņemamo laika vienību skaits;
  - `time_unit` - h (stunda), d (diena), w (nedēļa), M (mēnesis) vai y (gads).

Komentāri:

- Parametrs `time_shift` atbalsta daudzpakāpju laika operācijas un var ietvert `/<time_unit>`, lai pārbīdītu uz laika vienības sākumu (`/d` - pusnakts, `/w` - nedēļas 1. diena (pirmdiena), `/M` - mēneša 1. diena utt.). Piemēri: `-1w` - tieši 7 dienas atpakaļ; `-1w/w` - iepriekšējās nedēļas pirmdiena; `-1w/w+1d` - iepriekšējās nedēļas otrdiena.
- Laika operācijas tiek aprēķinātas no kreisās uz labo pusi bez prioritātēm. Piemēram, `-1M/d+1h/w` tiks interpretēts kā `((-1M/d)+1h)/w`.

Piemēri:

```
{{TIME}.fmttime(%B)} - atgriezīs "October" no saņemtās vērtības "1633098961"
{{TIME}.fmttime(%d %B,-1M/M)} - atgriezīs "1 September" no saņemtās vērtības "1633098961"
```

htmldecode

Makro vērtības dekodēšana no HTML kodējuma.

Tiek atbalstītas šādas rakstzīmes:

Vērtība	Dekodētā vērtība
&	&
<	<
>	>
"	"
'	'

Vērtība	Dekodētā vērtība
'	'

Piemērs:

```
{{ITEM.VALUE}.htmldecode()} - HTML dekodēs vērtību, piemēram, "&lt;" uz "<"
```

htmlencode

Makro vērtības kodēšana HTML kodējumā.

Tiek atbalstītas šādas rakstzīmes:

Vērtība	Kodētā vērtība
&	&
<	<
>	>
"	"
'	'

Piemērs:

```
{{ITEM.VALUE}.htmlencode()} - HTML kodēs rakstzīmi, piemēram, "<", par "&lt;"
```

iregsub(pattern,output)

Apakšvirknes izvilkšana pēc atbilstības regulārajai izteiksmei (bez reģistra jutības).

Parametri:

- **pattern** - regulārā izteiksme atbilstības noteikšanai;
- **output** - izvades opcijas. **\1 - \9** vietturi tiek atbalstīti tveršanas grupām. **\0** atgriež atbilstošo tekstu.

Komentāri:

- Ja regulārajai izteiksmei nav atbilstības, funkcija atgriež tukšu virkni.
- Ja funkcijas pattern ir nekorekta regulārā izteiksme, tad makro tiek novērtēts kā 'UNKNOWN' (izņemot zema līmeņa atklāšanas makro, kurā gadījumā funkcija tiks ignorēta un makro paliks neatrisināts).
- Atsauces uz neeksistējošām tveršanas grupām aizvietošanas virknē tiek aizstātas ar tukšu virkni.

Piemērs:

```
{{ITEM.VALUE}.iregsub("fail|error|fault|problem","ERROR")} - tiks atrisināts kā "ERROR", ja tiks saņemtas
```

mazie burti

Visu makro vērtības rakstzīmju pārveidošana par mazajiem burtiem.
Darbojas ar viena baita rakstzīmju kopām (piemēram, ASCII) un neatbalsta UTF-8.

Piemērs:

```
{{ITEM.VALUE}.lowercase()} - pārveidos tādu vērtību kā "Zabbix SERVER" par "zabbix server" (mazie burti)
```

regrepl(pattern,replacement,<pattern2>,<replacement2>,...)

Rakstzīmes/apakšvirknes aizstāšana makro vērtībā.

Parametri:

- **pattern** - regulārā izteiksme atbilstības meklēšanai;
- **replacement** - aizstāšanas virkne. Aizstāšanas virknēs tiek atbalstīti vietturi **\1 - \9** grupu uztveršanai.

Komentāri:

- Šabloni un aizstāšanas virknes tiek apstrādātas secīgi, katru nākamo pāri piemērojot atbilstoši iepriekšējās aizstāšanas rezultātam;
- Atsauces uz neeksistējošām uztveršanas grupām aizstāšanas virknē tiek aizstātas ar tukšu virkni.

Piemēri:

```
{{ITEM.VALUE}.regrepl("oldParam", "newParam")} - aizstās "oldParam" ar "newParam"
{{ITEM.VALUE}.regrepl("([^a-z])","\\\1")} - visas rakstzīmes, kas nav burti, tiks ekranētas ar slīpsvītru
{$THRESHOLD:"{{#FSNAME}.regrepl(\"\\$\",\"\")}"} - noņems beigu slīpsvītru (piemēram, lai aizstātu "C:\" a
{{ITEM.VALUE}.regrepl("_v1\.0", "_v2.0", "\(final\)", "")} - aizstās vairākas daļas vienuma vērtībā
```

regsub(pattern,output)

Apakšvirknes izvilkšana pēc atbilstības regulārajai izteiksmei (reģistrjutīga).

Parametri:

- **pattern** - regulārā izteiksme atbilstības noteikšanai;
- **output** - izvades opcijas. **\1 - \9** vietturi tiek atbalstīti grupu uztveršanai. **\0** atgriež atbilstošo tekstu.

Komentāri:

- Ja regulārajai izteiksmei nav atbilstības, funkcija atgriež tukšu virkni.
- Ja funkcijas pattern ir nekorekta regulārā izteiksme, tad makro tiek novērtēts kā 'UNKNOWN' (izņemot zema līmeņa atklāšanas makro, kur šī funkcija tiks ignorēta un makro paliks neatrisināts).
- Atsauces uz neeksistējošām uztveršanas grupām aizvietošanas virknē tiek aizstātas ar tukšu virkni.

Piemēri:

```
{{ITEM.VALUE}.regsub("^([0-9]+)", Problem ID: \1)} - tiks atrisināts kā "Problem ID: 123", ja tiek saņemta
{{ITEM.VALUE}.regsub("fail|error|fault|problem","ERROR")} - tiks atrisināts kā "ERROR", ja tiek saņemtas a
```

Skatiet vairāk piemēru.

tr(characters,replacement)

Makro vērtības rakstzīmju transliterācija.

- **characters** - aizstājamais rakstzīmju kopums;
- **replacement** - pozicionāli atbilstošo aizstājējrakstzīmju kopums.

Piemēri:

```
{{ITEM.VALUE}.tr(abc, xyz)} - aizstās visas "a" ar "x", "b" ar "y", "c" ar "z"
{{ITEM.VALUE}.tr(abc, xyzq)} - aizstās visas "a" ar "x", "b" ar "y", "c" ar "z" ("q" tiek ignorēts)
{{ITEM.VALUE}.tr(abcde, xyz)} - aizstās visas "a" ar "x", "b" ar "y", "c" ar "z", "d" ar "z", "e" ar "z" (
{{ITEM.VALUE}.tr("\\\'", "\/\"")} - aizstās visas atpakaļvērstās slīpsvītras ar slīpsvītru uz priekšu, vie
{{ITEM.VALUE}.tr(A-Z,a-z)} - pārveidos visus burtus par mazajiem burtiem
{{ITEM.VALUE}.tr(0-9a-z,*)} - aizstās visus ciparus un mazos burtus ar "*"
{{ITEM.VALUE}.tr(0-9,ab)} - aizstās visas 0 ar "a", un aizstās visas 1, 2, 3, 4, 5, 6, 7, 8 un 9 ar "b"
{{ITEM.VALUE}.tr(0-9abcA-L,*)} - aizstās visus ciparus, "abc" rakstzīmes un diapazonu A-L ar "*"
{{ITEM.VALUE}.tr("\n","*")} - aizstās visas rindas beigu virknes ar *
{{ITEM.VALUE}.tr("e", "\n")} - aizstās visus "e" ar rindas beigu virkni
```

Lai iekļautu literālās rakstzīmes:

```
atpakaļvērstā slīpsvītra - jāatveido kā \\
vienpēdiņa - jāatveido kā \'
dubultpēdiņa - jāatveido kā \"
```

Atbalstītās atsoļa secības ar atpakaļvērsto slīpsvītru:

```
\\\\ => \\ - dubultā atpakaļvērstā slīpsvītra uz vienu atpakaļvērsto slīpsvītru
\\a  => \a - brīdinājuma signāls
\\b  => \b - atpakaļatkāpe
\\f  => \f - lapas pārtraukums
\\n  => \n - jauna rinda
\\r  => \r - kursora atgriešana
\\t  => \t - horizontālā tabulācija
\\v  => \v - vertikālā tabulācija
```

lielie burti

Visu makro vērtības rakstzīmju pārveidošana par lielajiem burtiem.
Darbojas ar viena baita rakstzīmju kopām (piemēram, ASCII) un neatbalsta UTF-8.

Piemērs:

```
{{ITEM.VALUE}.uppercase()} - pārveidos vērtību, piemēram, "Zabbix Server", par "ZABBIX SERVER" (lielie bur
```

urldecode

Makro vērtības dekodēšana no URL kodējuma.

Piemērs:

```
{{ITEM.VALUE}.urldecode()} - URL dekodēs tādu vērtību kā "%2F" par "/"
```

urlencode

Makro vērtības kodēšana URL kodējumā.

Piemērs:

```
{{ITEM.VALUE}.urlencode()} - URL kodēs rakstzīmi, piemēram, "/" uz "%2F"
```

Papildu piemēri

Tālāk redzamajā tabulā ir parādīti vēl citi makro funkciju izmantošanas piemēri.

{#IFALIAS} ir LLD makross un ir definēts tikai zema līmeņa atklāšanas kontekstos (atklāšanas noteikumos, prototipos un no tiem izveidotajos vienumos/trigeros). Izmantojot to ārpus LLD, tokens paliks neaizvietots.

Makro funkcija	Saņemtā vērtība	Izvade
{{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* (nederīga regulārā izteiksme)
{{#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)} (nederīga regulārā izteiksme)
{$MACRO:"{{#IFALIAS}.regsub(\"(.*)_([0-9]+\", \1)}"}	customername_1	{$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)}"} (nederīga regulārā izteiksme)
"{$MACRO:"\{{#IFALIAS}.regsub("(.*)_([0-9]+", \1)}\"}"	customername_1	"{$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)}\"}" (nederīga regulārā izteiksme)

Pilnu vienuma vērtību skatīšana

Atrisināto {ITEM.VALUE} un {ITEM.LASTVALUE} makrosu garās vērtības teksta/žurnāla vienumiem dažās lietotāja saskarnes vietās tiek saīsinātas līdz 20 rakstzīmēm. Lai redzētu šo makrosu pilnās vērtības, varat izmantot makro funkcijas, piemēram:

```
{{ITEM.VALUE}.regsub("(.*)", \1)}
{{ITEM.LASTVALUE}.regsub("(.*)", \1)}
```

Skatiet arī: {ITEM.VALUE} un {ITEM.LASTVALUE} makrosu detalizēta informācija.

2 Lietotāja makrosi

Pārskats

Lietotāja makrosi tiek atbalstīti Zabbix, lai nodrošinātu lielāku elastību, papildus makrosiem, kas tiek atbalstīti pēc noklusējuma.

Lietotāja makrosus var definēt globālā, veidnes un hosta līmenī. Šiem makrosiem ir īpaša sintakse:

`{$MACRO}`

Zabbix apstrādā makrosus saskaņā ar šādu prioritātes secību:

1. hosta līmeņa makrosi (tiek pārbaudīti vispirms)
2. makrosi, kas definēti hosta pirmā līmeņa veidnēm (t.i., veidnēm, kas ir tieši piesaistītas hostam), sakārtoti pēc veidnes ID
3. makrosi, kas definēti hosta otrā līmeņa veidnēm, sakārtoti pēc veidnes ID
4. makrosi, kas definēti hosta trešā līmeņa veidnēm, sakārtoti pēc veidnes ID utt.
5. globālie makrosi (tiek pārbaudīti pēdējie)

Citiem vārdiem sakot, ja makross hostam nepastāv, Zabbix mēģinās to atrast hosta veidnēs ar pieaugošu dziļumu. Ja tas joprojām netiek atrasts, tiks izmantots globālais makross, ja tāds pastāv.

Warning:
Ja makross ar **vienādu nosaukumu** pastāv vairākās piesaistītās viena un tā paša līmeņa veidnēs, tiks izmantots makross no veidnes ar zemāko ID. Tādējādi makrosu ar vienādu nosaukumu izmantošana vairākās veidnēs rada konfigurācijas risku.

Ja Zabbix nevar atrast makrosu, tas netiks apstrādāts.

Attention:
Makrosi (tostarp lietotāja makrosi) pēc paredzētā principa paliek neatrisināti sadaļā Configuration (piemēram, trigeru sarakstā), lai padarītu sarežģītu konfigurāciju pārskatāmāku.

Lietotāja makrosus var izmantot:

- vienuma nosaukumā
- vienuma atslēgas parametrā
- vienuma atjaunināšanas intervālos un elastīgajos intervālos
- trigera nosaukumā un aprakstā
- trigera izteiksmes parametros un konstantēs (skatiet piemērus)
- daudzās citās vietās — skatiet pilno sarakstu

Globālo un hostu makrosu biežākie lietošanas gadījumi

- izmantot globālu makrosu vairākās vietās; pēc tam mainīt makrosa vērtību un ar vienu klikšķi piemērot konfigurācijas izmaiņas visām vietām
- izmantot veidņu priekšrocības ar hostam specifiskiem atribūtiem: paroles, portu numuri, failu nosaukumi, regulārās izteiksmes utt.

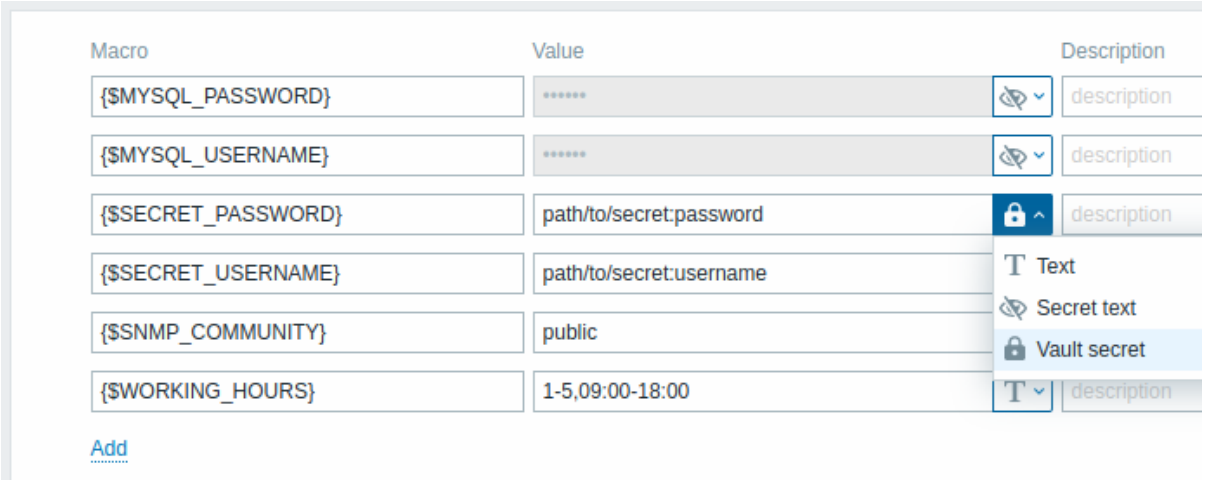
Note:
Ieteicams izmantot hostu makrosus globālo makrosu vietā, jo globālo makrosu pievienošana, atjaunināšana vai dzēšana piespiež veikt inkrementālu konfigurācijas atjaunināšanu visiem hostiem. Plašāku informāciju skatiet sadaļā Pasīvās un aktīvās aģenta pārbaudes.

Konfigurācija

Lai definētu lietotāja makrosus, dodieties uz atbilstošo vietu lietotāja saskarnē:

- globālajiem makrosiem atveriet *Administrēšana → Makrosi*
- hosta un veidnes līmeņa makrosiem atveriet hosta vai veidnes rekvizītus un meklējiet cilni *Makrosi*

Lietotāja makrosam ir šādi atribūti:



Parametrs	Apraksts
Macro	Makrosa nosaukums. Nosaukumam jābūt ietvertam figūriekavās un jāsākas ar dolāra zīmi. Piemērs: {$FRONTEND_URL}. Makrosu nosaukumos ir atļautas šādas rakstzīmes: **A-Z** (tikai lielie burti), **0-9**, _, **.**
Value	Makrosa vērtība. Tiek atbalstīti trīs vērtību tipi: **Text** (noklusējums) - vienkārša teksta vērtība **Secret text** - vērtība tiek maskēta ar zvaigznītēm **Vault secret** - vērtība satur ceļu/vaicājumu uz vault secret. Lai mainītu vērtības tipu, noklikšķiniet uz pogas vērtības ievades lauka beigās. Lietotāja makrosa vērtības maksimālais garums ir 2048 rakstzīmes.
Description	Teksta lauks, ko izmanto, lai sniegtu vairāk informācijas par šo makrosu.

Konfigurējot lietotāja makrosus, **ņemiet vērā** šādu kontekstam specifisku darbību:

- ja lietotāja makrosi tiek izmantoti veidnes vienumos vai trigeros, apsveriet iespēju pievienot šos makrosus arī veidnei (pat ja tie ir definēti globāli); tādā veidā *Text* tipa makrosi darbosies, kā paredzēts, pēc veidnes eksportēšanas uz XML un importēšanas citā sistēmā (slepeno makrosu vērtības netiek eksportētas)
- ja lietotāja makrosi tiek izmantoti trigeru izteiksmēs, šie makrosi tiks atrisināti tikai tad, ja tie atsaucas uz parametru vai konstanti; tie NETIKS atrisināti, ja tie atsaucas uz hostu, vienuma atslēgu, funkciju, operatoru vai citu trigera izteiksmi (slepenos makrosus nevar izmantot trigeru izteiksmēs)
- ja lietotāja makrosi tiek izmantoti hostā, kuram ir zema līmeņa atklāšanas kārtula ar hostu prototipiem, atklātie hosti mantos visus šajā hostā definētos lietotāja makrosus

Piemēri

Piemērs 1

Hosta līmeņa makrosa izmantošana vienuma atslēgā "SSH dēmona statuss":

```
net.tcp.service[ssh,,{$SSH_PORT}]
```

Šo vienumu var piešķirt vairākiem hostiem, ja šajos hostos ir definēta **{$SSH_PORT}** vērtība.

Piemērs 2

Hosta līmeņa makrosa izmantošana trigerī "CPU load is too high":

```
last(/ca_001/system.cpu.load[,avg1])>{$MAX_CPULOAD}
```

Šāds trigeris tiktu izveidots veidnē, nevis rediģēts atsevišķos hostos.

Note:
Ja vēlaties izmantot vērtību skaitu kā funkcijas parametru (piemēram, **max(/host/key,#3)**), iekļaujiet restītes zīmi makrosa definīcijā šādi: SOME_PERIOD => #3

Piemērs 3

Divu makrosu izmantošana trigerī "CPU load is too high":

```
min(/ca_001/system.cpu.load[,avg1],{$CPULOAD_PERIOD})>{$MAX_CPULOAD}
```

Ņemiet vērā, ka makrosu var izmantot kā trigera funkcijas parametru, šajā piemērā funkcijā **min()**.

Piemērs 4

Sinhronizējiet aģents nepieejamības nosacījumu ar vienums atjaunināšanas intervālu:

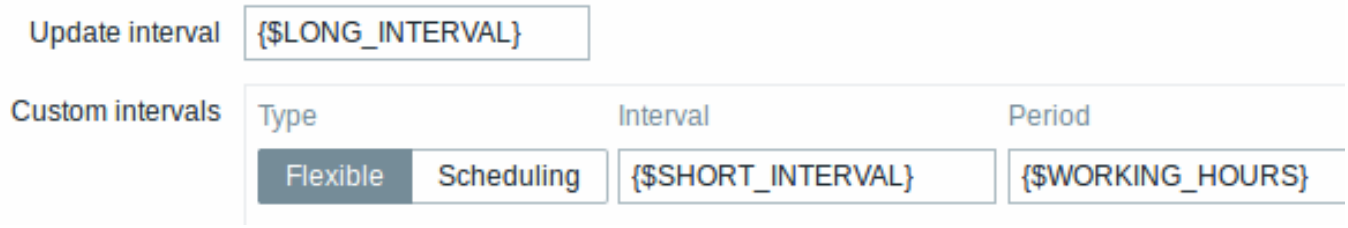
- definējiet {$INTERVAL} makrosu un izmantojiet to vienums atjaunināšanas intervālā;
- izmantojiet {$INTERVAL} kā aģents nepieejamības trigeris parametru:

```
nodata(/ca_001/agent.ping,{$INTERVAL})=1
```

Piemērs 5

Centralizējiet darba laika konfigurāciju:

- izveidojiet globālu {$WORKING_HOURS} makro ar vērtību `1-5,09:00-18:00`;
- izmantojiet to laukā *Working time* sadaļā *Administration → General → GUI*;
- izmantojiet to laukā *When active* sadaļā *Users → Users*, lietotāja cilnē *Media*;
- izmantojiet to, lai iestatītu biežāku vienumu aptaujāšanu darba laikā:

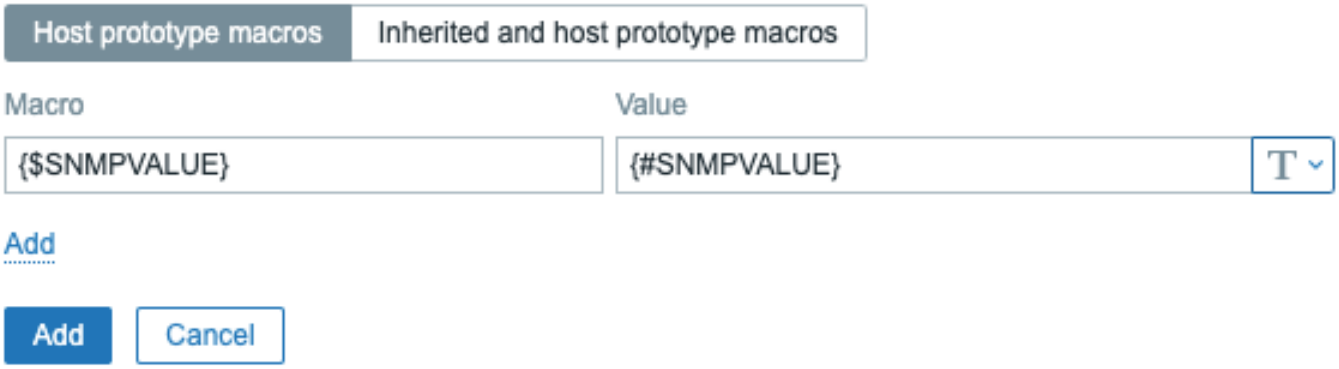


- izmantojiet to darbības nosacījumā *Time period*;
- ja nepieciešams, pielāgojiet darba laiku sadaļā *Administration → Macros*.

Piemērs 6

Izmantojiet hosta prototipa makrosu, lai konfigurētu vienumus atklātajiem hostiem:

- hosta prototipā definējiet lietotāja makrosu {$SNMPVALUE} ar {#SNMPVALUE} zemā līmeņa atklāšanas makrosu kā vērtību:



- piešķiriet hosta prototipam veidni *Generic SNMPv2*;
- izmantojiet {$SNMPVALUE} veidnes *Generic SNMPv2* vienumu laukā *SNMP OID*.

Lietotāja makro konteksts

Skatiet lietotāja makro ar kontekstu.

3 Lietotāja makrosi ar kontekstu

Pārskats

Var izmantot neobligātu kontekstu lietotāja makrosos, kas ļauj aizstāt noklusējuma vērtību ar kontekstam specifisku vērtību.

Konteksts tiek pievienots makrosa nosaukumam; sintakse ir atkarīga no tā, vai konteksts ir statiska teksta vērtība:

```
{$MACRO:"static text"}
```

vai regulārā izteiksme:

```
{$MACRO:regex:"regular expression"}
```

Ņemiet vērā, ka makrosu ar regulārās izteiksmes kontekstu var definēt tikai lietotāja makrosu konfigurācijā. Ja prefikss `regex:` tiek izmantots citur kā lietotāja makrosa konteksts, piemēram, trigera izteiksmē, tas tiks apstrādāts kā statisks konteksts.

Konteksta ievietošana pēdiņās nav obligāta (skatiet arī svarīgas piezīmes).

Makrosa konteksta piemēri:

Piemērs	Apraksts
`{$LOW_SPACE_LIMIT}`	Lietotāja makross bez konteksta.
`{$LOW_SPACE_LIMIT:/tmp}`	Lietotāja makross ar kontekstu (statiska virkne).
`{$LOW_SPACE_LIMIT:regex:"^/tmp$"}`	Lietotāja makross ar kontekstu (regulārā izteiksme). Tas pats, kas `{$LOW_SPACE_LIMIT:/tmp}`.
`{$LOW_SPACE_LIMIT:regex:"^/var/log/.*$"}`	Lietotāja makross ar kontekstu (regulārā izteiksme). Atbilst visām virknēm ar prefiksu /var/log/.

Lietošanas gadījumi

Lietotāja makrosus ar kontekstu var definēt, lai nodrošinātu elastīgākus sliekšņus trigeru izteiksmēs (pamatojoties uz vērtībām, kas iegūtas ar zema līmeņa atklāšanu). Piemēram, varat definēt šādus makrosus:

- {$LOW_SPACE_LIMIT} = 10
- {$LOW_SPACE_LIMIT:/home} = 20
- {$LOW_SPACE_LIMIT:regex:”^/[a-z]+$”} = 30

Pēc tam zema līmeņa atklāšanas makrosu var izmantot kā makrosa kontekstu trigera prototipā piemontēto failu sistēmu atklāšanai:

`last(/host/vfs.fs.size[{#FSNAME},pfree])<{$LOW_SPACE_LIMIT:"{#FSNAME}"}`

Pēc atklāšanas trigeros tiks piemēroti atšķirīgi zemas brīvās vietas sliekšņi atkarībā no atklātajiem piemontēšanas punktiem vai failu sistēmu tipiem. Problēmu notikumi tiks ģenerēti, ja:

- mapei /home ir mazāk nekā 20% brīvas diska vietas
- mapēm, kas atbilst regulārās izteiksmes paraugam (piemēram, /etc, /tmp vai /var), ir mazāk nekā 30% brīvas diska vietas
- mapēm, kas neatbilst regulārās izteiksmes paraugam un nav /home, ir mazāk nekā 10% brīvas diska vietas

Svarīgas piezīmes

- Ja pastāv vairāk nekā viens lietotāja makross ar kontekstu, Zabbix vispirms mēģinās saskaņot vienkāršos konteksta makrosus un pēc tam konteksta makrosus ar regulārajām izteiksmēm nenoteiktā secībā.

Warning:
Neveidojiet dažādus konteksta makrosus, kas atbilst tai pašai virknei, lai izvairītos no nenoteiktas darbības.

- Ja makross ar tā kontekstu netiek atrasts hostā, piesaistītajās veidnēs vai globāli, tad tiek meklēts makross bez konteksta.
- Kontekstā tiek atbalstīti tikai zema līmeņa atklāšanas makrosi. Jebkuri citi makrosi tiek ignorēti un apstrādāti kā vienkāršs teksts.

Tehniski makrosa konteksts tiek norādīts, izmantojot noteikumus, kas ir līdzīgi vienuma atslēgas parametriem, izņemot to, ka makrosa konteksts netiek parsēts kā vairāki parametri, ja tajā ir rakstzīme `,`:

- Makrosa konteksts jāliek pēdiņās " , ja konteksts satur rakstzīmi } vai sākas ar rakstzīmi ". Pēdiņas pēdiņās ietvertā kontekstā ir jāekranē ar rakstzīmi \.
- Pati rakstzīme \ netiek ekranēta, kas nozīmē, ka nav iespējams izveidot pēdiņās ietvertu kontekstu, kas beidzas ar rakstzīmi \ — makross {$MACRO:”a:\b\c\”} nav derīgs.
- Sākuma atstarpes kontekstā tiek ignorētas, beigu atstarpes — nē:
  - Piemēram, {$MACRO:A} ir tas pats, kas {$MACRO: A}, bet ne {$MACRO:A }.
- Visas atstarpes pirms sākuma pēdiņām un pēc beigu pēdiņām tiek ignorētas, bet visas atstarpes pēdiņu iekšpusē — nē:
  - Makrosi {$MACRO:”A”}, {$MACRO: ”A”}, {$MACRO:”A” } un {$MACRO: ”A” } ir vienādi, bet makrosi {$MACRO:”A”} un {$MACRO:” A ”} nav vienādi.

Tālāk norādītie makrosi ir ekvivalenti, jo tiem ir viens un tas pats konteksts: {$MACRO:A}, {$MACRO: A} un {$MACRO:”A”}. Tas atšķiras no vienumu atslēgām, kur ’key[a]’, ’key[a]’ un ’key[”a”]’ semantiski ir vienādi, bet unikalitātes nolūkos atšķiras.

4 Slepenie lietotāja makrosi

Pārskats

Zabbix nodrošina divas iespējas sensitīvas informācijas aizsardzībai lietotāja makro vērtībās:

- Slepens teksts
- Vault noslēpums

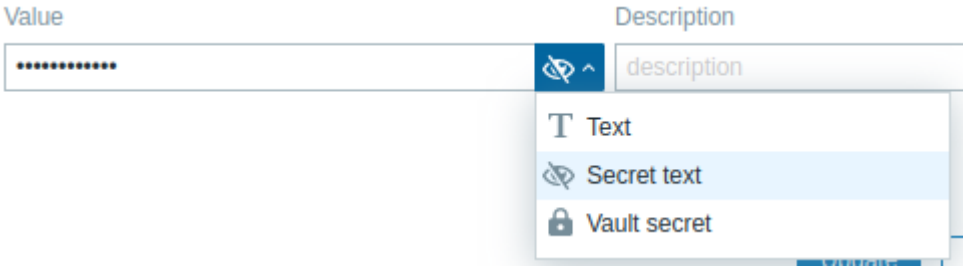
Note:
Lai gan slepena makro vērtība ir paslēpta, to var atklāt, izmantojot to vienumos. Piemēram, ārējā skriptā echo priekšrakstu, kas atsaucas uz slepenu makro, var izmantot, lai atklātu makro vērtību lietotāja saskarnē, jo Zabbix serveris ir piekļuve reālajai makro vērtībai. Skatiet vietas, kur slepeno makro vērtības tiek parādītas neatmaskotas.

Slepenos makro nevar izmantot trigeru izteiksmēs.

Slepens teksts

Izmantojot *Slepens teksts* makrosus, makrosa vērtība tiek maskēta ar zvaigznītēm.

Lai padarītu makrosa vērtību slepenu, noklikšķiniet uz pogas lauka *Vērtība* beigās un atlasiet opciju *Slepens teksts*:



Kad konfigurācija ir saglabāta, vērtību vairs nebūs iespējams apskatīt.

Lai mainītu makrosa vērtību, novietojiet kursoru virs lauka *Vērtība* un noklikšķiniet uz pogas *Iestatīt jaunu vērtību* (parādās, novietojot kursoru virs lauka):

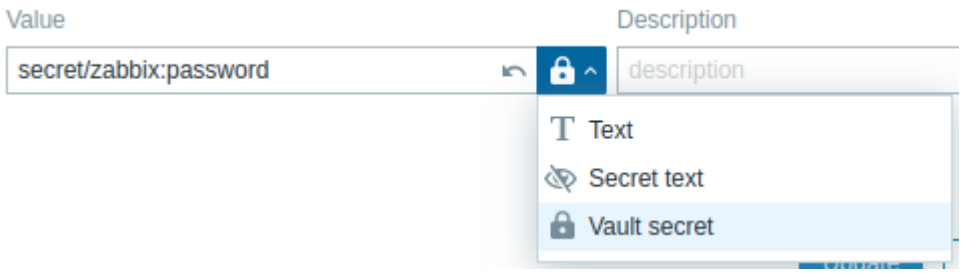
Noklikšķinot uz pogas *Iestatīt jaunu vērtību* (vai mainot makrosa vērtības tipu), pašreizējā vērtība tiks dzēsta. Sākotnējo vērtību var atjaunot, noklikšķinot uz bultiņas lauka *Vērtība* beigās (pieejams tikai pirms jaunās konfigurācijas saglabāšanas). Ņemiet vērā, ka sākotnējās vērtības atjaunošana to neatklās.

Note:
URL, kas satur slepenu makrosu, nedarbosies, jo makross tajos tiks atrisināts kā "******".

Vault slepenā vērtība

Izmantojot Vault slepeno vērtību makrosus, makrosa vērtība tiek glabāta ārējā slepeno vērtību pārvaldības programmatūrā (vault).

Lai konfigurētu Vault slepenās vērtības makrosu, noklikšķiniet uz pogas lauka *Value* beigās un atlasiet opciju *Vault secret*:



Makrosa vērtībai jānorāda uz vault slepeno vērtību. Ievades formāts ir atkarīgs no vault nodrošinātāja. Nodrošinātājam specifiskus konfigurācijas piemērus skatiet šeit:

- HashiCorp
- CyberArk

Vault slepeno makro vērtības no glabātuves iegūst Zabbix serveris (un Zabbix starpniekserveris, ja *Resolve secret vault macros by* ir iestatīts uz *Zabbix server and proxy*) katrā konfigurācijas datu atsvaidzināšanas reizē, un pēc tam tās tiek saglabātas konfigurācijas kešatmiņā. Zabbix serveris un Zabbix starpniekserveris var izmantot dažādas glabātuves.

Ja *Resolve secret vault macros by* ir iestatīts uz *Zabbix server*, tad glabātuves slepenos datus iegūst tikai serveris, un Zabbix starpniekserveris saņem Vault slepeno makro vērtības no Zabbix servera katrā konfigurācijas sinhronizācijas reizē un saglabā tās

savā konfigurācijas kešatmiņā. Tas nozīmē, ka Zabbix starpniekserveris pēc restartēšanas nevar sākt datu vākšanu, kamēr tas nesaņem konfigurācijas atjauninājumu no Zabbix servera.

Lai manuāli atsvaidzinātu slepenās vērtības no glabātuves, izmantojiet `secrets_reload` izpildlaika vadības opciju (tikai serverim).

Starp Zabbix serveri un starpniekserveri jābūt iespējotai šifrēšanai; pretējā gadījumā servera žurnālā tiek ierakstīts brīdinājuma ziņojums.

Warning:
Ja makro vērtību neizdodas veiksmīgi iegūt, atbilstošais vienums, kas izmanto šo vērtību, kļūs neatbalstīts.

Nemaskētās atrašanās vietas

Šajā sarakstā ir norādītas parametru atrašanās vietas, kur slepeno makrosu vērtības tiek rādītas nemaskētas.

Note:
Slepeno makrosu vērtības tālāk norādītajās vietās paliks maskētas, ja uz tām būs netieša atsauce. Piemēram, {ITEM.KEY}, {ITEM.KEY<1-9>}, {LLDRULE.KEY} iebūvētie makrosi, kas izmantoti mediju tipos (Script vai Webhook parametros), tiks atrisināti kā vienumu atslēgas, kurās slepenie makrosi ir maskēti, piemēram, `net.tcp.port[******,******]` nevis `net.tcp.port[192.0.2.0,80]`.

Konteksts		Parametrs
Vienumi, vienumu prototipi, LLD kārtulas		
	Vienums	*Vienuma atslēgas parametri*
	Vienuma prototips	*Vienuma prototipa atslēgas parametri*
	Zema līmeņa atklāšanas kārtula	*Atklāšanas vienuma atslēgas parametri*
	SNMP aģents	*SNMP kopiena*
		Konteksta nosaukums (SNMPv3)
		Drošības nosaukums (SNMPv3)
		Autentifikācijas parole (SNMPv3)
		Privātuma parole (SNMPv3)
	HTTP aģents	*URL*
		Vaicājuma lauki
		Pieprasījuma pamatteksts
		Galvenes
		Lietotājvārds
		Parole
		SSL atslēgas parole
	Skripts	*Parametri*
		Skripts
	Pārlūks	*Parametri*
		Skripts
	Datubāzes monitors	*SQL vaicājums*
	TELNET aģents	*Skripts*
		Lietotājvārds
		Parole
	SSH aģents	*Skripts*
		Lietotājvārds
		Parole
	Vienkāršā pārbaude	*Lietotājvārds*
		Parole
	JMX aģents	*Lietotājvārds*
		Parole
Vienuma vērtības priekšapstrāde		
	JavaScript priekšapstrādes solis	*Skripts*
Tīmekļa scenāriji		
	Tīmekļa scenārijs	*Mainīgā vērtība*
		Galvenes vērtība
		URL
		Vaicājuma lauka vērtība
		POST lauka vērtība
		Neapstrādāts POST
	Tīmekļa scenārija autentifikācija	*Lietotājs*

Konteksts		Parametrs
		Parole
		SSL atslēgas parole
Savienotāji		
	Savienotājs	*URL*
		Lietotājvārds
		Parole
		Tokens
		HTTP starpniekserveris
		SSL sertifikāta fails
		SSL atslēgas fails
		SSL atslēgas parole
Tīkla atklāšana		
	SNMP	*SNMP kopiena*
		Konteksta nosaukums (SNMPv3)
		Drošības nosaukums (SNMPv3)
		Autentifikācijas parole (SNMPv3)
		Privātuma parole (SNMPv3)
Globālie skripti		
	Webhook	*JavaScript skripts*
		JavaScript skripta parametra vērtība
	Telnet	*Lietotājvārds*
		Parole
	SSH	*Lietotājvārds*
		Parole
	Skripts	*Skripts*
Mediju tipi		
	Skripts	*Skripta parametri*
	Webhook	*Parametri*
IPMI pārvaldība		
	Hosts	*Lietotājvārds*
		Parole

5 Zema līmeņa atklāšanas makrosi

Pārskats

Pastāv makro veids, kas tiek izmantots zemā līmeņa atklāšanas (LLD) funkcijā:

```
{#MACRO}
```

Tas ir makro, kas tiek izmantots LLD kārtulā un atgriež reālās failu sistēmas nosaukuma, tīkla saskarnes, SNMP OID u.c. vērtības.

Šos makro var izmantot, lai izveidotu entītiju prototipus (vienuma, trigera, grafika, LLD kārtulas, hosta un hostu grupas prototipus). Atklājot reālas failu sistēmas, tīkla saskarnes, virtuālās mašīnas u.c., šie makro tiek aizstāti ar reālām vērtībām un kalpo par pamatu reālu entītiju izveidei.

Daži zemā līmeņa atklāšanas makro tiek piegādāti kopā ar LLD funkciju Zabbix sistēmā — {#FSNAME}, {#FSTYPE}, {#IFNAME}, {#SNMPINDEX}, {#SNMPVALUE}. Tomēr, veidojot pielāgotu zemā līmeņa atklāšanas kārtulu, šo nosaukumu ievērošana nav obligāta. Tādā gadījumā varat izmantot jebkuru citu LLD makro nosaukumu un atsaukties uz šo nosaukumu.

Atbalstītie datu tipi

Definējot pielāgotus atklāšanas noteikumus, JSON objektos atgrieztajām īpašību vērtībām LLD makro jābūt vienam no šiem primitīvajiem tipiem:

- string
- number
- boolean

Masīvi, objekti un null vērtības netiek atbalstītas. Jebkurš LLD makro, kas atsaucas uz šādu vērtību, paliks neaizvietots un tiks attēlots burtiski (piemēram, `'{#MY_MACRO}'`) vienuma priekšapstrādes un izveides laikā.

Atbalstītās vietas

LLD makrosus var izmantot:

- zema līmeņa atklāšanas kārtulas filtrā
- ligzdotās zema līmeņa atklāšanas kārtulās, šeit:
  - JSONPath priekšapstrādes parametros
  - JSONPath laukā pielāgotiem LLD makrosiem
- vienumu prototipiem un atklāšanas prototipiem, šeit:
  - nosaukumā
  - atslēgas parametros
  - mērvienībā
  - atjaunināšanas intervālā[1]
  - noildzē[1]
  - vēstures glabāšanas periodā[1]
  - tendenču glabāšanas periodā[1]
  - vienuma vērtības priekšapstrādes soļos
  - SNMP OID
  - IPMI sensora laukā
  - aprēķinātā/agregētā vienuma izteiksmē, šeit:
    * izteiksmes konstantēs un funkciju parametros
    * vienuma atslēgas parametros
  - agregētā vienuma filtra nosacījumos (hostu grupas nosaukums un birkas nosaukums)
  - SSH skriptā un Telnet skriptā
  - datubāzes uzraudzības SQL vaicājumā
  - JMX vienuma galapunkta laukā
  - aprakstā
  - HTTP aģenta URL laukā
  - HTTP aģenta HTTP vaicājuma lauku laukā
  - HTTP aģenta pieprasījuma pamatteksta laukā
  - HTTP aģenta nepieciešamo statusa kodu laukā
  - HTTP aģenta galveņu lauka atslēgā un vērtībā
  - HTTP aģenta HTTP autentifikācijas lietotājvārda laukā
  - HTTP aģenta HTTP autentifikācijas paroles laukā
  - HTTP aģenta HTTP starpniekservera laukā
  - HTTP aģenta HTTP SSL sertifikāta faila laukā
  - HTTP aģenta HTTP SSL atslēgas faila laukā
  - HTTP aģenta HTTP SSL atslēgas paroles laukā
  - birkās
- trigeru prototipiem, šeit:
  - nosaukumā
  - darbības datos
  - izteiksmē (tikai konstantēs un funkciju parametros)
  - URL
  - aprakstā
  - birkās
- grafiku prototipiem, šeit:
  - nosaukumā
- hostu prototipiem, šeit:
  - nosaukumā
  - redzamajā nosaukumā
  - pielāgotajos saskarnes laukos: IP, DNS, ports, SNMP v1/v2 kopiena, SNMP v3 konteksta nosaukums, SNMP v3 drošības nosaukums, SNMP v3 autentifikācijas parole, SNMP v3 privātuma parole
  - hostu grupas prototipa nosaukumā
  - hosta birkas vērtībā
  - hosta makrosa vērtībā
  - (skatiet pilnu sarakstu)

Visās šajās vietās, izņemot zema līmeņa atklāšanas kārtulas filtru, LLD makrosus var izmantot statiskā lietotāja makrosa kontekstā.

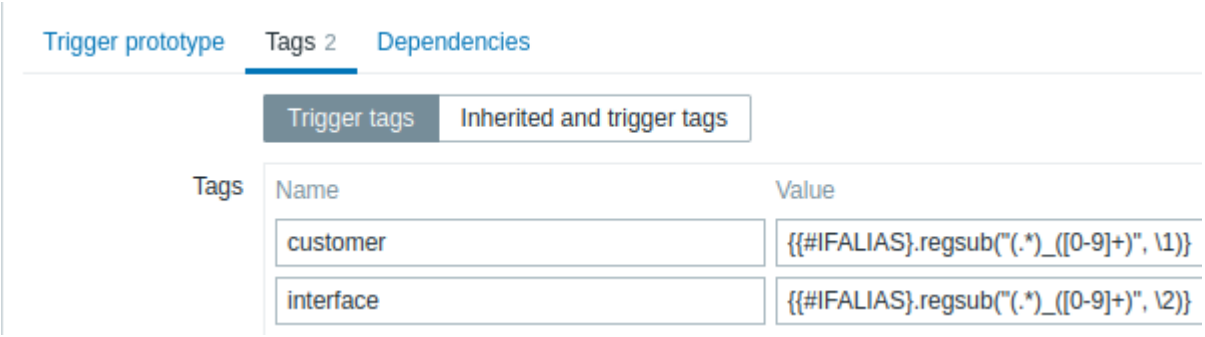
Makro funkciju izmantošana

Makro funkcijas tiek atbalstītas zema līmeņa atklāšanas makros (izņemot zema līmeņa atklāšanas kārtulas filtrā), ļaujot izvilkt noteiktu makro vērtības daļu, izmantojot regulāro izteiksmi.

Piemēram, iespējams, vēlēsieties izvilkt klienta nosaukumu un saskarnes numuru no šāda ZLA makro notikumu marķēšanas vajadzībām:

```
{#IFALIAS}=customername_1
```

Lai to izdarītu, var izmantot `regsub` makro funkciju ar makro trigera prototipa notikuma taga vērtības laukā:



Ņemiet vērā, ka neiekļautiem vienuma atslēgas parametriem komati nav atļauti, tāpēc parametrs, kas satur makro funkciju, ir jāliek pēdiņās. Atpakaļvērstās slīpsvītras rakstzīme (\) jāizmanto, lai parametrā atsoļotu dubultpēdiņas. Piemērs:

```
net.if.in["{{#IFALIAS}.regsub(\"(.*)_([0-9]+)\", \1)}",bytes]
```

Plašāku informāciju par makro funkciju sintaksi skatiet: Makro funkcijas

Makro funkcijas zema līmeņa atklāšanas makros tiek atbalstītas kopš Zabbix 4.0.

Vienumu prototipi bez atslēgas parametriem

Ja vienuma atslēgas parametri netiek izmantoti, ievietojiet LLD makrosu vienuma atslēgas parametra iekavās `[...]`, lai tas tiktu apstrādāts kā parametrs un aizstāts atklāšanas laikā:

```
v_[{#MACRO}]
```

Piezīmes

1 Laukos, kas atzīmēti ar [1], vienai makrokomandai ir jāaizpilda viss lauks. Vairākas makrokomandas vienā laukā vai makrokomandas kopā ar tekstu netiek atbalstītas.

6 Izteiksmju makrosi

Pārskats

Izteiksmju makrosi ļauj veikt aprēķinus laukos.

To vērtība tiek aprēķināta, vispirms atrisinot visus iekšējos makrosus un pēc tam novērtējot iegūto izteiksmi.

Sintakse:

```
{?EXPRESSION}
```

`EXPRESSION` izmanto to pašu sintaksi un atbalsta tās pašas funkcijas kā trigeru izteiksmes.

Piemērs:

```
{?trendavg(/host/item1,1M:now/M)/trendavg(/host/item1,1M:now/M-1y)*100}
```

Lai atrastu vietas, kas atbalsta izteiksmju makrosus, meklējiet "{?EXPRESSION}" atbalstīto makrosu tabulā.

Piezīmes par lietošanu:

- Izmantojiet {FUNCTION.*} makrosus, lai atsauktos uz trigeru izteiksmju/atkopšanas izteiksmju funkciju vērtībām.
- Izmantojiet {HOST.HOST<1-9>} un {ITEM.KEY<1-9>} makrosus, lai atsauktos uz hostiem un vienumiem.
- Veidnēs izmantojiet {HOST.HOST<1-9>} makrosus vai pirmajam hostam vispār nenorādiet hostu — piemēram, {?avg(//item1,1h)} — veidņu nosaukumu vietā, jo veidņu nosaukumi veidņu piesaistes laikā netiek aizstāti ar hostu nosaukumiem.

```
{?{FUNCTION.VALUE2} - {FUNCTION.VALUE3}}
{?max(/{HOST.HOST}/{ITEM.KEY},3h)}
```

Skatiet arī Trigeru izteiksmju piemērus, lai redzētu piemēru, kā izteiksmes makross tiek izmantots notikuma nosaukumā.

12 Lietotāji un lietotāju grupas

Pārskats

Visi Zabbix lietotāji piekļūst Zabbix lietojumprogrammai, izmantojot tīmekļa lietotāja saskarni. Katram lietotājam tiek piešķirts unikāls pieteikšanās vārds un parole.

Visas lietotāju paroles ir šifrētas un glabātas Zabbix datubāzē. Lietotāji nevar izmantot savu lietotāja ID un paroli, lai tieši pieteiktos UNIX serverī, ja vien tie nav attiecīgi konfigurēti arī UNIX vidē. Saziņu starp tīmekļa serveri un lietotāja pārlūku var aizsargāt, izmantojot SSL.

Izmantojot elastīgu lietotāju atļauju shēmu, jūs varat ierobežot un diferencēt tiesības:

- piekļūt Zabbix lietotāja saskarnes administratīvajām funkcijām
- veikt noteiktas darbības lietotāja saskarnē
- piekļūt uzraudzītajiem hostiem hostu grupās
- izmantot noteiktas API metodes

1 Lietotāja konfigurēšana

Pārskats

Sākotnējā Zabbix instalācijā ir divi iepriekš definēti lietotāji:

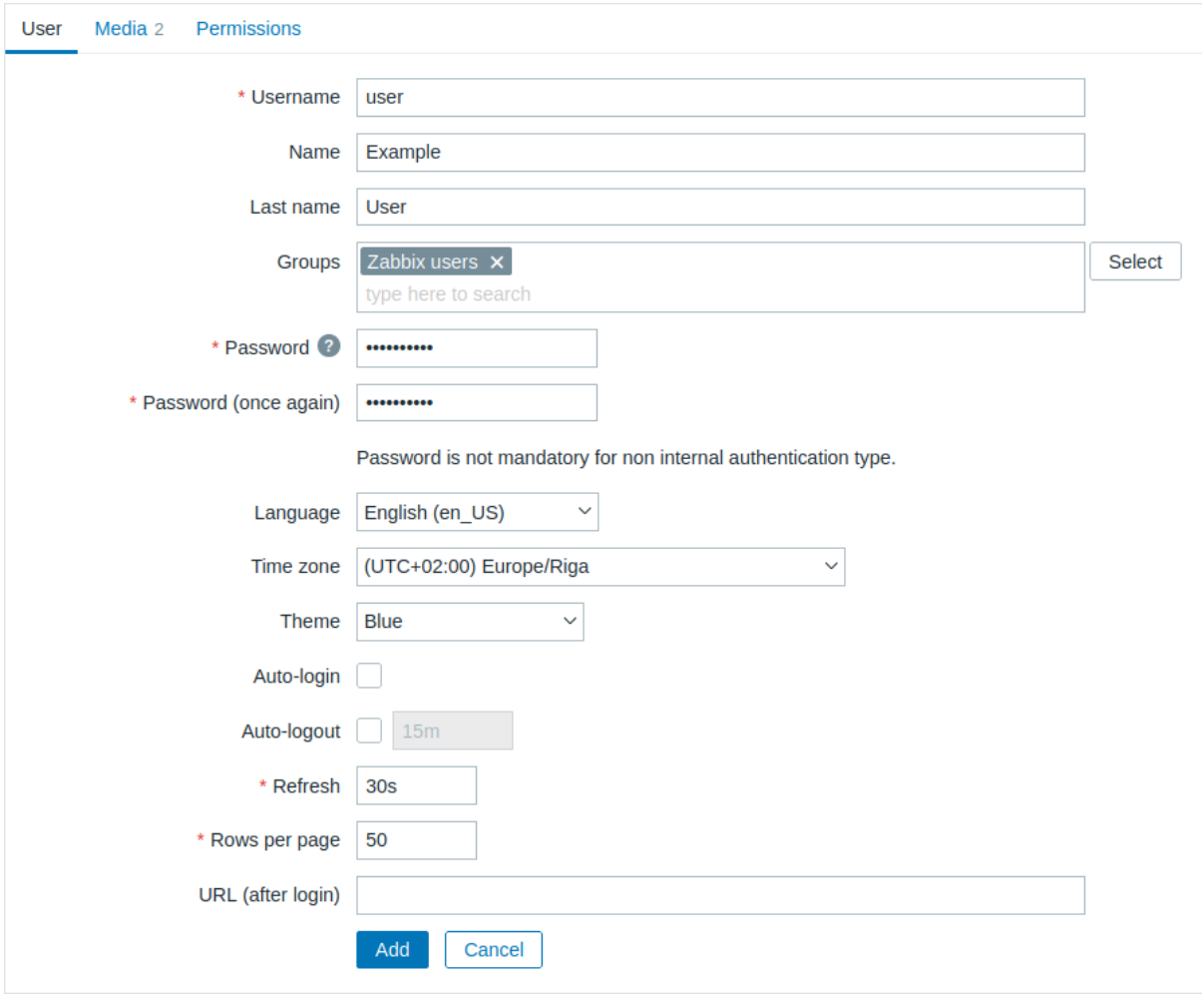
- *Admin* — Zabbix superlietotājs ar pilnām atļaujām.
- *guest* — īpašs Zabbix lietotājs. Lietotājs 'guest' pēc noklusējuma ir atspējots. Ja pievienosiet to lietotāju grupai Guests, varēsiet pieteikties ar šo lietotāju un piekļūt uzraudzības lapām Zabbix. Ņemiet vērā, ka pēc noklusējuma lietotājam 'guest' nav atļauju darbam ar Zabbix objektiem.

Lai konfigurētu lietotāju:

- Dodieties uz *Lietotāji* → *Lietotāji*.
- Noklikšķiniet uz *Izveidot lietotāju* (vai uz lietotāja vārda, lai rediģētu esošu lietotāju).
- Rediģējiet lietotāja atribūtus formā.

Vispārīgie atribūti

Cilnē *Lietotājs* ir pieejami vispārīgie lietotāja atribūti:



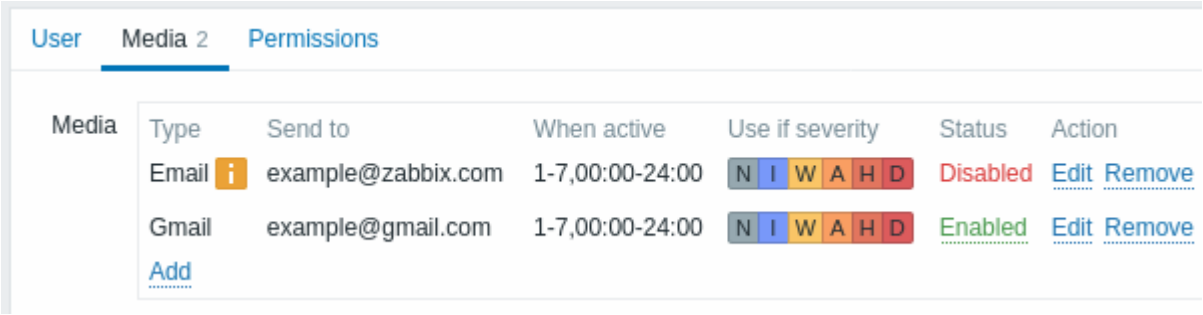
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Username	Unikāls lietotājvārds, kas tiek izmantots kā pieteikšanās vārds.
Name	Lietotāja vārds (nav obligāts). Ja nav tukšs, tas ir redzams apstiprinājuma informācijā un paziņojumu saņēmēja informācijā.
Last name	Lietotāja uzvārds (nav obligāts). Ja nav tukšs, tas ir redzams apstiprinājuma informācijā un paziņojumu saņēmēja informācijā.
Groups	Atlasiet lietotāju grupas, kurām lietotājs pieder. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt lietotāju grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Ritiniet uz leju, lai atlasītu. Alternatīvi, noklikšķiniet uz *Select*, lai pievienotu grupas. Noklikšķiniet uz 'x', lai noņemtu atlasīto. Piederība lietotāju grupām nosaka, kurām hostu grupām un hostiem lietotājam būs piekļuve.
Password	Divi lauki lietotāja paroles ievadīšanai vai poga *Change password*, ja lietotājs jau pastāv. Noklikšķinot uz pogas *Change password*, tiek atvērti divi lauki jaunas paroles ievadīšanai. Lietotājam ar *Super admin role*, kurš maina savu paroli, noklikšķinot uz pogas *Change password*, tiek atvērts papildu lauks pašreizējās (vecās) paroles ievadīšanai. Pēc veiksmīgas paroles maiņas lietotājs, kuram parole tika mainīta, tiks izrakstīts no visām aktīvajām sesijām. Ņemiet vērā, ka paroli var mainīt tikai lietotājiem, kuri izmanto Zabbix iekšējo autentifikāciju.
Language	Zabbix lietotāja saskarnes valoda. Lai tulkojumi darbotos, ir nepieciešams php gettext paplašinājums.
Time zone	Atlasiet laika joslu, lai lietotāja līmenī ignorētu globālo laika joslu, vai atlasiet **System default**, lai izmantotu globālos laika joslas iestatījumus.

Parametrs	Apraksts
Theme	Nosaka, kā izskatās lietotāja saskarne: **System default** - izmantot noklusējuma sistēmas iestatījumus **Blue** - standarta zilā tēma **Blue (classic)** - standarta zilā tēma, izmantojot mantotos fontus **Dark** - alternatīva tumšā tēma **Dark (classic)** - alternatīva tumšā tēma, izmantojot mantotos fontus **High-contrast light** - gaišā tēma ar augstu kontrastu **High-contrast dark** - tumšā tēma ar augstu kontrastu
Auto-login	Atzīmējiet šo izvēles rūtiņu, lai Zabbix atcerētos lietotāju un automātiski pieteiktu to sistēmā uz 30 dienām. Piesakoties ar *Remember for 30 days*: - Automātiskā izrakstīšanās tiek atiestatīta (sesija saglabājas 30 dienas). - Automātiskā pieteikšanās tiek iespējota nepārtrauktai atkārtotai autentifikācijai. Piesakoties bez *Remember for 30 days*: - Automātiskā pieteikšanās tiek iespējota nepārtrauktai atkārtotai autentifikācijai. - Automātisko izrakstīšanos joprojām nosaka standarta noildzes iestatījums. Tam tiek izmantotas pārlūkprogrammas sīkdatnes.
Auto-logout	Ja šī izvēles rūtiņa ir atzīmēta, lietotājs tiks automātiski izrakstīts pēc iestatītā sekunžu skaita (minimums 90 sekundes, maksimums 1 diena). Ņemiet vērā, ka šis iestatījums tiek ignorēts, ja ir iespējots *Remember for 30 days*, jo sesija tiek pagarināta uz visu periodu. Tiek atbalstīti laika sufiksi, piemēram, 90s, 5m, 2h, 1d. Ņemiet vērā, ka šī opcija nedarbosies: * Ja ir iespējota globālā konfigurācijas opcija ”Show warning if Zabbix server is down” un Zabbix lietotāja saskarne tiek turēta atvērta. * Kad izvēlnes Monitoring lapas fonā veic informācijas atsvaidzināšanu. * Ja pieteikšanās laikā ir atzīmēta opcija *Remember me for 30 days*.
Refresh	Iestatiet atsvaidzināšanas biežumu, kas tiek izmantots grafikiem, vienkārša teksta datiem utt. Var iestatīt uz 0, lai atspējotu. Tiek atbalstīti laika sufiksi, piemēram, 90s, 5m, 1h.
Rows per page	Varat noteikt, cik rindu katrā lapā tiks parādīts sarakstos.
URL (after login)	Varat iestatīt, lai Zabbix pēc veiksmīgas pieteikšanās novirzītu lietotāju uz noteiktu URL, piemēram, uz lapu *Problems*.

Lietotāja mediji

Cilnē *Mediji* ir redzams visu lietotājam definēto mediju saraksts. Mediji tiek izmantoti paziņojumu sūtīšanai.



Noklikšķiniet uz *Pievienot*, lai piešķirtu lietotājam mediju.

Ja medija tips ir atspējots:

- Pēc nosaukuma tiek parādīta dzeltena informācijas ikona.
- Kolonnā Statuss tiek parādīts *Atspējots*.

Note:
Lietotāja tiesības pašam mainīt savu mediju informāciju var piešķirt/atsaukt atbilstoši viņa lietotāja lomai (skatiet opciju *Create and edit own media*). Super admin lietotāja tiesības mainīt citu lietotāju mediju informāciju arī var piešķirt/atsaukt atbilstoši viņa lietotāja lomai (skatiet opciju *Create and edit user media*)

Sīkāku informāciju par lietotāja mediju konfigurēšanu skatiet sadaļā Mediju tipi.

Atļaujas

Cilnē *Atļaujas* ir informācija par šādiem elementiem:

- Lietotāja loma (obligāta jebkuram jaunizveidotam lietotājam), kuru var mainīt tikai *Super admin* tipa lietotājs.

Attention:
Lietotājus nevar izveidot bez lietotāja lomas (izņemot ar Zabbix User API). Iepriekš izveidotos lietotājus, kuriem nav lomas, joprojām var rediģēt, nepiešķirot tiem lomu. Tomēr, tiklīdz loma ir piešķirta, to var tikai mainīt, nevis noņemt.

 Ņemiet vērā, ka lietotāji bez lomas var pieteikties Zabbix tikai, izmantojot LDAP vai SAML autentifikāciju, ja viņu LDAP/SAML informācija atbilst Zabbix konfigurētajām lietotāju grupu piesaistēm.

- Lietotāja tips (*User*, *Admin*, *Super admin*), kas ir definēts lietotāja lomas konfigurācijā.
- Hostu un veidņu grupas, kurām lietotājam ir piekļuve.
  - *User* un *Admin* tipa lietotājiem pēc noklusējuma nav piekļuves nevienai grupai, veidnei un hostam. Lai piešķirtu šādu piekļuvi, lietotāji ir jāiekļauj lietotāju grupās, kurām ir konfigurētas atļaujas attiecīgajām entītijām.
- Piekļuves tiesības Zabbix lietotāja saskarnes sadaļām un elementiem, moduļiem un API metodēm.
  - Elementi ar atļautu piekļuvi tiek attēloti zaļā krāsā, savukārt tie, kuriem piekļuve ir liegta, — gaiši pelēkā krāsā.
- Tiesības veikt noteiktas darbības.
  - Darbības, kuras lietotājam ir atļauts veikt, tiek attēlotas zaļā krāsā, savukārt tās, kas ir liegtas, — gaiši pelēkā krāsā.

Sīkāku informāciju skatiet lapā Atļaujas.

2 Atļaujas

Pārskats

Atļaujas Zabbix ir atkarīgas no lietotāja tipa, pielāgotām lietotāju lomām un piekļuves hostiem, kas tiek noteikta, pamatojoties uz lietotāju grupu.

Lietotāju tipi

Atļaujas Zabbix vidē galvenokārt ir atkarīgas no lietotāja tipa:

- *Lietotājs* - ir ierobežotas piekļuves tiesības izvēlnes sadaļām (skatīt zemāk), un pēc noklusējuma tam nav piekļuves nekādiem resursiem. Jebkādas atļaujas hosts vai veidņu grupām ir skaidri jāpiešķir;
- *Administrators* - ir nepilnīgas piekļuves tiesības izvēlnes sadaļām (skatīt zemāk). Lietotājam pēc noklusējuma nav piekļuves nevienai hostu grupai. Jebkādas atļaujas hostu vai veidņu grupām ir skaidri jāpiešķir;
- *Super administrators* - ir piekļuve visām izvēlnes sadaļām. Lietotājam ir lasīšanas-rakstīšanas piekļuve visām hostu un veidņu grupām. Atļaujas nevar atsaukt, liedzot piekļuvi konkrētām grupām.

Piekļuve izvēlnei

Note:
Ierobežota piekļuve dažiem lietotāja saskarnes elementiem tikai novērš šīs lapas atvēršanu — tā neatceļ iespēju piekļūt pamatā esošajiem datiem citās saskarnes daļās.

Šajā tabulā ir parādīta piekļuve Zabbix izvēlnes sadaļām atbilstoši lietotāja tipam:

Izvēlnes sadaļa		Lietotājs	Administrators	Super administrators
Paneļi		+	+	+
Uzraudzība		+	+	+
	Problēmas	+	+	+
	Hosti	+	+	+
	Jaunākie dati	+	+	+
	Kartes	+	+	+
	Atklāšana		+	+
Pakalpojumi		+	+	+
	Pakalpojumi	+	+	+
	SLA		+	+
	SLA atskaite	+	+	+
Inventārs		+	+	+
	Pārskats	+	+	+
	Hosti	+	+	+
Atskaites		+	+	+

Izvēlnes sadaļa		Lietotājs	Administrators	Super administrators
	Sistēmas informācija			+
	Plānotās atskaites		+	+
	Pieejamības atskaite	+	+	+
	Top 100 trigeri	+	+	+
	Audita žurnāls			+
	Darbību žurnāls			+
	Paziņojumi		+	+
Datu ievākšana			+	+
	Veidņu grupas		+	+
	Hostu grupas		+	+
	Veidnes		+	+
	Hosti		+	+
	Uzturēšana		+	+
	Notikumu korelācija			+
	Atklāšana		+	+
Brīdinājumi			+	+
	Trigeru darbības		+	+
	Pakalpojumu darbības		+	+
	Atklāšanas darbības		+	+
	Automātiskās reģistrācijas darbības		+	+
	Iekšējās darbības		+	+
	Mediju tipi			+
	Skripti			+
Lietotāji				+
	Lietotāju grupas			+
	Lietotāju lomas			+
	Lietotāji			+
	API pilnvaras			+
	Autentifikācija			+
Administrēšana				+
	Vispārīgi			+
	Audita žurnāls			+
	Housekeeping			+
	Starpniekserveru grupas			+
	Starpniekserveri			+
	Makrosi			+
	Rinda			+

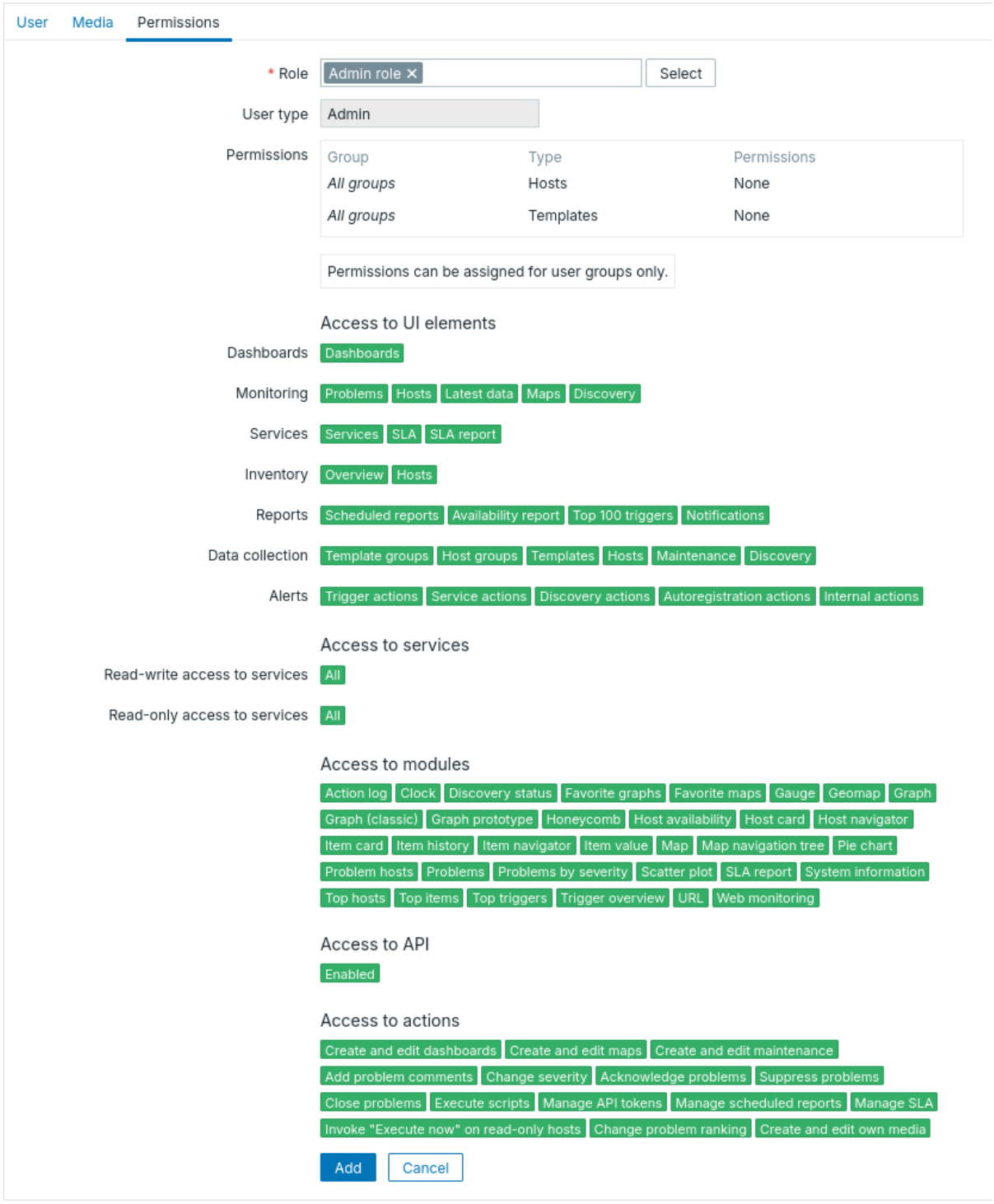
Lietotāju lomas

Lietotāju lomas ļauj veikt pielāgotas izmaiņas lietotāja tipa noteiktajās atļaujās. Lai gan jaunas atļaujas nevar pievienot (tas pārsniegtu lietotāja tipa atļaujas), dažas atļaujas var atsaukt.

Turklāt lietotāja loma nosaka piekļuvi ne tikai izvēlnes sadaļām, bet arī pakalpojumiem, moduļiem, API metodēm un dažādām darbībām lietotāja saskarnē.

Lietotāju lomas tiek konfigurētas sadaļā *Lietotāji* → *Lietotāju lomas*, un to veic Super admin lietotāji.

Lietotāju lomas tiek piešķirtas lietotājiem lietotāja konfigurācijas formā, cilnē *Atļaujas*, un to veic Super admin lietotāji.



Piekļuve hostiem

Piekļuve jebkura hosta un veidnes datiem Zabbix vidē tiek piešķirta lietotāju grupām tikai hostu/veidņu grupu līmenī.

Tas nozīmē, ka atsevišķam lietotājam nevar tieši piešķirt piekļuvi hostam (vai hostu grupai). Piekļuvi hostam var piešķirt tikai tad, ja lietotājs ir daļa no lietotāju grupas, kurai ir piešķirta piekļuve hostu grupai, kas satur šo hostu.

Līdzīgi lietotājam var piešķirt piekļuvi veidnei tikai tad, ja viņš ir daļa no lietotāju grupas, kurai ir piešķirta piekļuve veidņu grupai, kas satur šo veidni.

3 Lietotāju grupas

Pārskats

Lietotāju grupas ļauj grupēt lietotājus gan organizatoriskiem mērķiem, gan piekļuves tiesību piešķiršanai datiem. Piekļuves tiesības hostu grupu un veidņu grupu datu skatīšanai un konfigurēšanai tiek piešķirtas lietotāju grupām, nevis atsevišķiem lietotājiem.

Bieži ir lietderīgi nodalīt, kāda informācija ir pieejama vienai lietotāju grupai un kāda — citai. To var panākt, grupējot lietotājus un pēc tam piešķirot atšķirīgas piekļuves tiesības hostu un veidņu grupām.

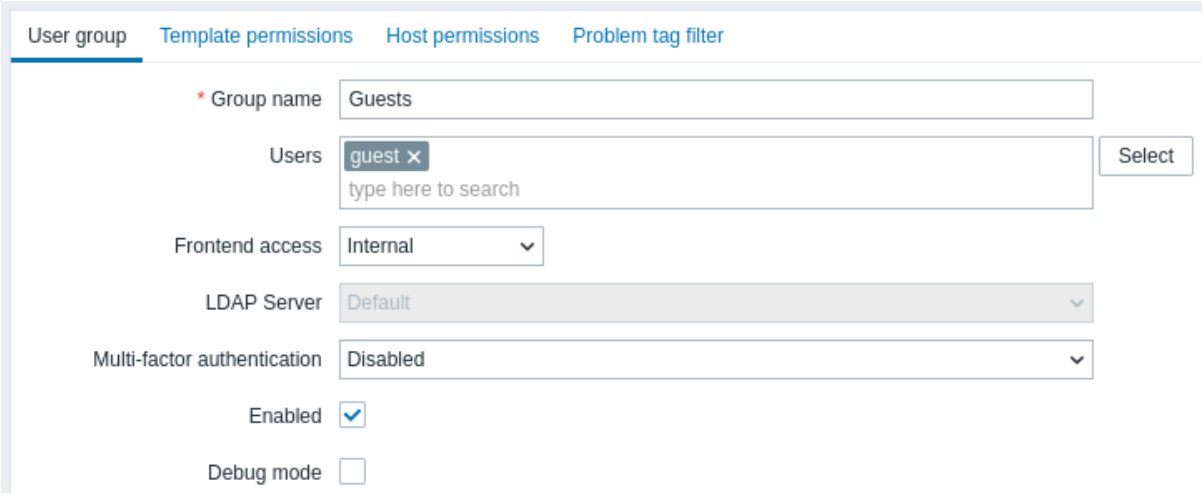
Lietotājs var piederēt jebkuram skaitam grupu.

Konfigurācija

Lai konfigurētu lietotāju grupu:

- Dodieties uz *Lietotāji* > *Lietotāju grupas*
- Noklikšķiniet uz *Izveidot lietotāju grupu* (vai uz grupas nosaukuma, lai rediģētu esošu grupu)
- Rediģējiet grupas atribūtus formā

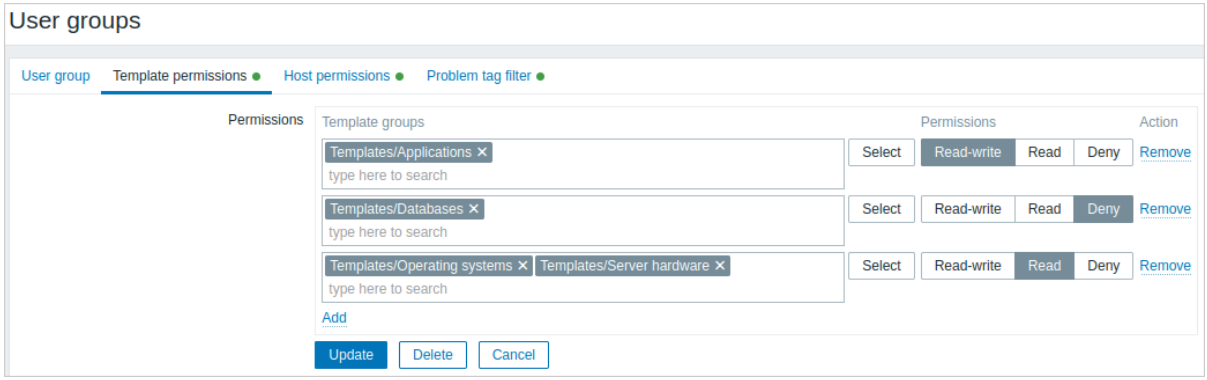
Cilnē **Lietotāju grupa** ir ietverti vispārīgie grupas atribūti:



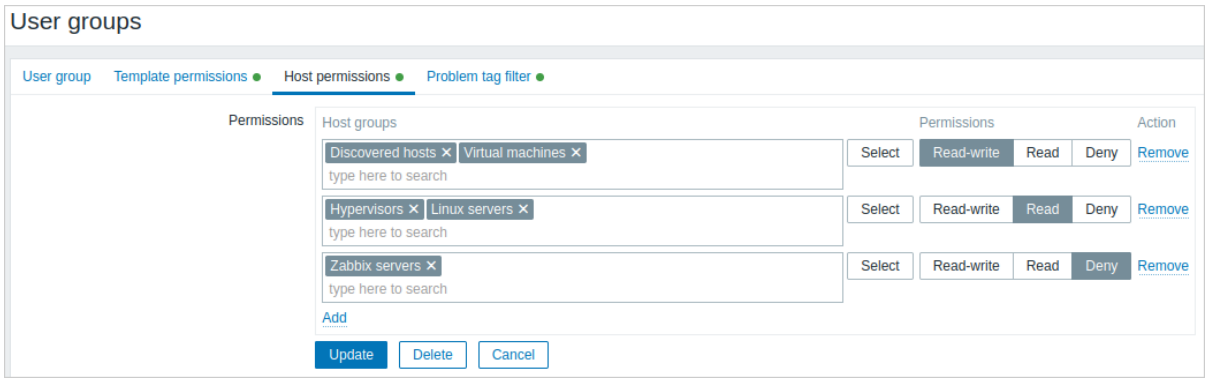
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Grupas nosaukums	Unikāls grupas nosaukums.
Lietotāji	Lai pievienotu lietotājus grupai, sāciet rakstīt esoša lietotāja vārdu. Kad parādās nolaižamais saraksts ar atbilstošiem lietotāju vārdiem, ritiniet uz leju, lai atlasītu. Alternatīvi varat noklikšķināt uz pogas *Atlasīt*, lai atlasītu lietotājus uznirstošajā logā.
Piekļuve lietotāja saskarnei	Kā tiek autentificēti grupas lietotāji. **Sistēmas noklusējums** - izmantot noklusēto autentifikācijas metodi (iestatīta globāli) **Iekšējā** - izmantot Zabbix iekšējo autentifikāciju (pat ja globāli tiek izmantota LDAP autentifikācija). Tiek ignorēts, ja HTTP autentifikācija ir globālais noklusējums. **LDAP** - izmantot LDAP autentifikāciju (pat ja globāli tiek izmantota iekšējā autentifikācija). Tiek ignorēts, ja HTTP autentifikācija ir globālais noklusējums. **Atspējota** - piekļuve Zabbix lietotāja saskarnei šai grupai ir aizliegta
LDAP serveris	Atlasiet, kuru LDAP serveri izmantot lietotāja autentificēšanai. Šis lauks ir pieejams tikai tad, ja *Piekļuve lietotāja saskarnei* ir iestatīta uz LDAP vai Sistēmas noklusējums.
Daudzfaktoru autentifikācija	Atlasiet, kuru daudzfaktoru autentifikācijas metodi izmantot lietotāja autentificēšanai: **Noklusētā** - izmantot metodi, kas MFA konfigurācijā iestatīta kā noklusētā; šī opcija pēc noklusējuma ir atlasīta jaunām lietotāju grupām, ja MFA ir iespējota; **<Metodes nosaukums>** - izmantot atlasīto metodi (piemēram, ”Zabbix TOTP”); **Atspējota** - MFA šai grupai ir atspējota; šī opcija pēc noklusējuma ir atlasīta jaunām lietotāju grupām, ja MFA ir atspējota. Ņemiet vērā, ka, ja lietotājs pieder vairākām lietotāju grupām ar iespējotu MFA (vai vismaz vienai grupai ir iespējota MFA), tiek piemēroti šādi autentifikācijas noteikumi: ja kāda grupa izmanto MFA metodi ”Noklusētā”, tā autentificēs lietotāju; pretējā gadījumā autentifikācijai tiks izmantota pirmā metode (alfabētiskā secībā).
Iespējota	Lietotāju grupas un grupas dalībnieku statuss. *Atzīmēts* - lietotāju grupa un lietotāji ir iespējoti *Nav atzīmēts* - lietotāju grupa un lietotāji ir atspējoti
Atkļūdošanas režīms	Atzīmējiet šo izvēles rūtiņu, lai lietotājiem aktivizētu atkļūdošanas režīmu.

Cilne **Veidņu atļaujas** ļauj norādīt lietotāju grupas piekļuvi veidņu grupu (un līdz ar to arī veidņu) datiem:



Cilne **Hostu atļaujas** ļauj norādīt lietotāju grupas piekļuvi hostu grupu (un līdz ar to arī hostu) datiem:



Noklikšķiniet uz Add, lai izvēlētos veidņu/hostu grupas (gan vecākgrupu, gan ligzdotu grupu) un piešķirtu tām atļaujas. Sāciet rakstīt grupas nosaukumu (parādīsies nolaižamais saraksts ar atbilstošām grupām) vai noklikšķiniet uz *Atlasīt*, lai atvērtu uznirstošo logu ar visu grupu sarakstu.

Pēc tam izmantojiet opciju pogas, lai piešķirtu atļaujas izvēlētajām grupām. Iespējamās atļaujas ir šādas:

- **Lasīt-rakstīt** - lasīšanas un rakstīšanas piekļuve grupai
- **Lasīt** - tikai lasīšanas piekļuve grupai
- **Aizliegt** - piekļuve grupai ir liegta

Ja viena un tā pati veidņu/hostu grupa ir pievienota vairākās rindās ar atšķirīgām atļaujām, tiks piemērota visstingrākā atļauja.

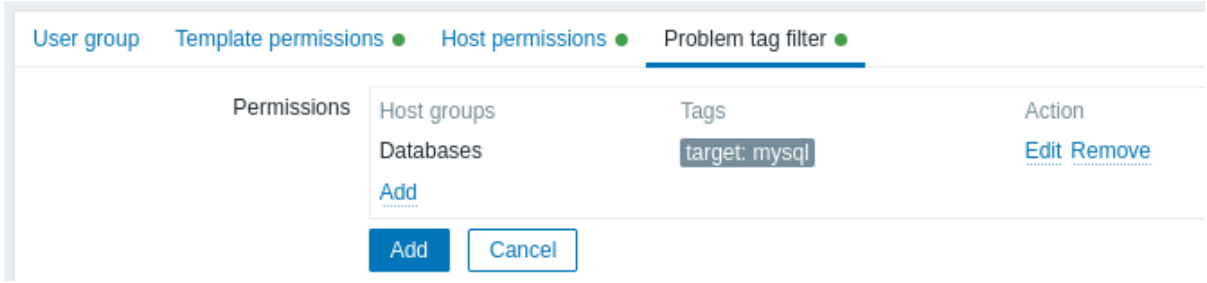
Ņemiet vērā, ka *Super admin* lietotājs var piespiest ligzdotajām grupām izmantot tādu pašu atļauju līmeni kā vecākgrupai; to var izdarīt hostu/veidņu grupu konfigurācijas formā.

Cilnes **Veidņu atļaujas** un **Hostu atļaujas** atbalsta vienu un to pašu parametru kopu.

Pašreizējās grupu atļaujas tiek parādītas blokā *Atļaujas*, un tās var mainīt vai noņemt.

Note:
Ja lietotāju grupai ir **Lasīt-rakstīt** atļaujas hostam un **Aizliegt** vai nav atļauju veidnei, kas ir piesaistīta šim hostam, šādas grupas lietotāji nevarēs rediģēt veidnētos vienumus hostā, un veidnes nosaukums tiks parādīts kā *Nepieejama veidne*.

Cilne **Problēmu birku filtrs** ļauj iestatīt uz birkām balstītas atļaujas lietotāju grupām, lai redzētu problēmas, kas filtrētas pēc birkas nosaukuma un vērtības:

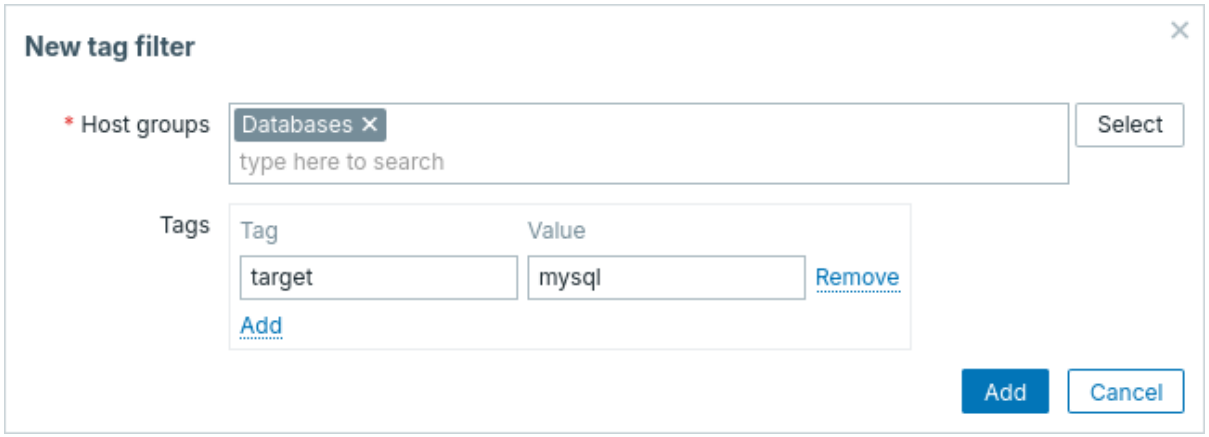


Noklikšķiniet uz Add, lai izvēlētos hostu grupas. Lai atlasītu hostu grupu, kurai piemērot birku filtru, noklikšķiniet uz *Atlasīt*, lai iegūtu pilnu esošo hostu grupu sarakstu, vai sāciet rakstīt hostu grupas nosaukumu, lai iegūtu nolaižamo sarakstu ar atbilstošām grupām. Tiks parādītas tikai hostu grupas, jo problēmu birku filtru nevar piemērot veidņu grupām.

Var pievienot birku nosaukumus bez vērtībām, bet vērtības bez nosaukumiem pievienot nevar. Blokā *Atļaujas* tiek parādītas tikai pirmās trīs birkas (ar vērtībām, ja tādas ir); ja to ir vairāk, tās var apskatīt, noklikšķinot uz ikonas ••• vai virzot kursoru virs tās.

Birku filtrs ļauj nodalīt piekļuvi hostu grupai no iespējas redzēt problēmas.

Piemēram, ja datubāzes administratoram jāredz tikai ”MySQL” datubāzes problēmas, vispirms ir jāizveido lietotāju grupa datubāzu administratoriem un pēc tam jānorāda birkas nosaukums ”target” un vērtība ”mysql”.



Ja ir norādīts birkas nosaukums ”target” un vērtības lauks ir atstāts tukšs, lietotāju grupa redzēs visas problēmas ar birkas nosaukumu ”target” atlasītajai hostu grupai.

Pārliecinieties, ka birkas nosaukums un birkas vērtība ir norādīti pareizi, pretējā gadījumā lietotāju grupa neredzēs nevienu problēmu.

Apskatīsim piemēru, kad lietotājs ir vairāku atlasītu lietotāju grupu dalībnieks. Šajā gadījumā filtrēšanā birkām tiks izmantots OR nosacījums.

Lietotāju grupa A			**Lietotāju grupa B**			**Redzamais rezultāts lietotājam (abu grupu dalībniekam)**
Birku filtrs						
Hostu grupa	*Birkas nosaukums*	*Birkas vērtība*	*Hostu grupa*	*Birkas nosaukums*	*Birkas vērtība*	
Datubāzes	target	mysql	Datubāzes	target	oracle	redzamas target:mysql vai target:oracle problēmas
Nav konfigurēts cilnē **Problēmu birku filtrs**			Datubāzes	target	oracle	redzamas target:oracle problēmas

Piekļuve no vairākām lietotāju grupām

Lietotājs var piederēt jebkuram skaitam lietotāju grupu. Šīm grupām var būt atšķirīgas piekļuves atļaujas hostiem vai veidnēm.

Tāpēc ir svarīgi zināt, kādām vienībām nepriviliģēts lietotājs rezultātā varēs piekļūt. Tālāk redzamajā piemērā aplūkots, kā dažādās situācijās tiks ietekmēta piekļuve hostam **X** (Hostu grupā 1) lietotājam, kurš ir lietotāju grupās A un B.

- Ja grupai A ir tikai *Lasīšanas* piekļuve Hostu grupai 1, bet grupai B ir *Lasīšanas-rakstīšanas* piekļuve Hostu grupai 1, lietotājs saņems **Lasīšanas-rakstīšanas** piekļuvi hostam ’X’.

Attention:
”Lasīšanas-rakstīšanas” atļaujām ir prioritāte pār ”Lasīšanas” atļaujām.

- Tajā pašā scenārijā kā iepriekš, ja ’X’ vienlaikus atrodas arī Hostu grupā 2, kas grupai A vai B ir **liegta**, piekļuve hostam ’X’ būs **nav pieejama**, neskatoties uz *Lasīšanas-rakstīšanas* piekļuvi Hostu grupai 1.

- Ja grupai A nav definētu atļauju un grupai B ir *Lasīšanas-rakstīšanas* piekļuve Hostu grupai 1, lietotājs saņems **Lasīšanas-rakstīšanas** piekļuvi hostam 'X'.
- Ja grupai A ir *Liegt* piekļuve Hostu grupai 1 un grupai B ir *Lasīšanas-rakstīšanas* piekļuve Hostu grupai 1, lietotājam piekļuve hostam 'X' tiks **liegta**.

Cita informācija

- Admin līmeņa lietotājs ar *Read-write* piekļuvi hostam nevarēs piesaistīt/atsaistīt veidnes, ja viņam nav piekļuves veidņu grupai, kurai tās pieder. Ja viņam ir *Read* piekļuve veidņu grupai, viņš varēs piesaistīt/atsaistīt veidnes hostam, taču neredzēs nevienu veidni veidņu sarakstā un nevarēs veikt darbības ar veidnēm citās vietās.
- Admin līmeņa lietotājs ar *Read* piekļuvi hostam neredzēs šo hostu konfigurācijas sadaļas hostu sarakstā; tomēr hosta trigeri būs pieejami IT pakalpojumu konfigurācijā.
- Jebkurš lietotājs, kas nav Super Admin (ieskaitot 'guest'), var redzēt tīkla kartes, kamēr karte ir tukša vai tajā ir tikai attēli. Kad kartei tiek pievienoti hosti, hostu grupas vai trigeri, tiek ievērotas piekļuves tiesības.
- Zabbix serveris nesūtīs paziņojumus lietotājiem, kas definēti kā darbības operāciju saņēmēji, ja piekļuve attiecīgajam hostam ir nepārprotami "denied".

13 Slepeno datu glabāšana

Pārskats

Zabbix var konfigurēt tā, lai izgūtu sensitīvu informāciju no drošas glabātuves. Tiek atbalstīti šādi noslēpumu pārvaldības pakalpojumi: HashiCorp Vault KV Secrets Engine - Version 2, CyberArk Vault CV12.

Noslēpumus var izmantot, lai izgūtu:

- lietotāja makro vērtības
- datubāzes piekļuves akreditācijas datus

Zabbix nodrošina tikai lasīšanas piekļuvi glabātuvē esošajiem noslēpumiem, pieņemot, ka noslēpumus pārvalda kāds cits.

Informāciju par konkrēta glabātuves nodrošinātāja konfigurāciju skatiet:

- HashiCorp konfigurācija
- CyberArk konfigurācija

Slepeno vērtību kešošana

Pēc noklusējuma glabātuves slepeno makro vērtības Zabbix serveris iegūst katrā konfigurācijas datu atsvaidzināšanas reizē un pēc tam saglabā konfigurācijas kešatmiņā. Zabbix starpniekserveris saņem glabātuves slepeno makro vērtības no Zabbix servera katrā konfigurācijas sinhronizācijas reizē un saglabā tās savā konfigurācijas kešatmiņā.

> **Attention:**
> Starp Zabbix serveri un starpniekserveri jābūt iespējotai šifrēšanai; pretējā gadījumā servera brīdinājuma ziņojums tiek ierakstīts žurnālā.

Ir iespējams arī konfigurēt, lai makro vērtības Zabbix serveris un Zabbix starpniekserveris iegūtu neatkarīgi.

Lai manuāli aktivizētu kešoto slepeno vērtību atsvaidzināšanu no glabātuves, izmantojiet komandrindas opciju 'secrets_reload'.

Zabbix lietotāja saskarnes datubāzes akreditācijas datu kešošana pēc noklusējuma ir atspējota, taču to var iespējot, iestatot opciju `$DB['VAULT_CACHE'] = true` failā zabbix.conf.php. Akreditācijas dati tiks saglabāti lokālajā kešatmiņā, izmantojot failu sistēmas pagaidu failu direktoriju. Tīmekļa serverim jāatļauj rakstīšana privātā pagaidu mapē (piemēram, Apache gadījumā jābūt iestatītai konfigurācijas opcijai `PrivateTmp=True`). Lai kontrolētu, cik bieži datu kešatmiņa tiek atsvaidzināta/invalidēta, izmantojiet konstanti ZBX_DATA_CACHE_TTL constant .

TLS konfigurācija

Lai konfigurētu TLS saziņai starp Zabbix komponentiem un glabātuvi, sistēmas mēroga noklusējuma CA krātuvē pievienojiet sertifikātu, ko parakstījusi sertifikācijas iestāde (CA). Lai izmantotu citu atrašanās vietu, norādiet direktoriju Zabbix serveris/starpniekserveris konfigurācijas parametrā SSLCALocation, ievietojiet sertifikāta failu šajā direktorijā un pēc tam palaidiet CLI komandu:

```
c_rehash .
```

1 CyberArk konfigurācija

Šajā sadaļā ir skaidrots, kā konfigurēt Zabbix, lai izgūtu noslēpumus no CyberArk Vault CV12.

Glabātuvei jābūt instalētai un konfigurētai, kā aprakstīts oficiālajā CyberArk dokumentācijā.

Lai uzzinātu par TLS konfigurēšanu Zabbix, skatiet *Noslēpumu glabāšana*.

Datubāzes akreditācijas dati

Piekļuve noslēpumam ar datubāzes akreditācijas datiem tiek konfigurēta katram Zabbix komponentam atsevišķi.

Serveris un starpniekserveri

Lai iegūtu datubāzes akreditācijas datus no glabātuves Zabbix serverim vai starpniekserverim, konfigurācijas failā norādiet šādus konfigurācijas parametrus:

- `Vault` - kurš glabātuves nodrošinātājs ir jāizmanto;
- `VaultURL` - glabātuves servera HTTP[S] URL;
- `VaultDBPath` - vaicājums uz glabātuves noslēpumu, kas satur datubāzes akreditācijas datus un tiks iegūts pēc atslēgām "Content" un "UserName" (šo opciju var izmantot tikai tad, ja nav norādīti DBUser un DBPassword);
- `VaultTLSCertFile`, `VaultTLSKeyFile` - SSL sertifikāta un atslēgas failu nosaukumi; šo opciju iestatīšana nav obligāta, taču ir ļoti ieteicama;
- `VaultPrefix` - pielāgots prefikss glabātuves ceļam vai vaicājumam atkarībā no glabātuves; ja tas nav norādīts, tiks izmantota vispiemērotākā noklusējuma vērtība.

Attention:
Konfigurācijas parametri `Vault`, `VaultURL`, `VaultTLSCertFile`, `VaultTLSKeyFile` un `VaultPrefix` tiek izmantoti arī glabātuves autentifikācijai, kad Zabbix serveris apstrādā slepenās glabātuves makrosus (un Zabbix starpniekserveris, ja tas ir konfigurēts). Zabbix serveris un Zabbix starpniekserveri neatvērs glabātuves slepenos makrosus, kas satur datubāzes akreditācijas datus no VaultDBPath.

Zabbix serveris un Zabbix starpniekserveris startēšanas laikā nolasa ar glabātuvi saistītos konfigurācijas parametrus no *zabbix_server.conf* un *zabbix_proxy.conf* failiem.

Piemērs

1. Failā *zabbix_server.conf* norādiet šādus parametrus:

```
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 nosūtīs uz glabātuvi šādu API pieprasījumu:

```
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. Glabātuves atbilde saturēs atslēgas "Content" un "UserName":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database": <Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

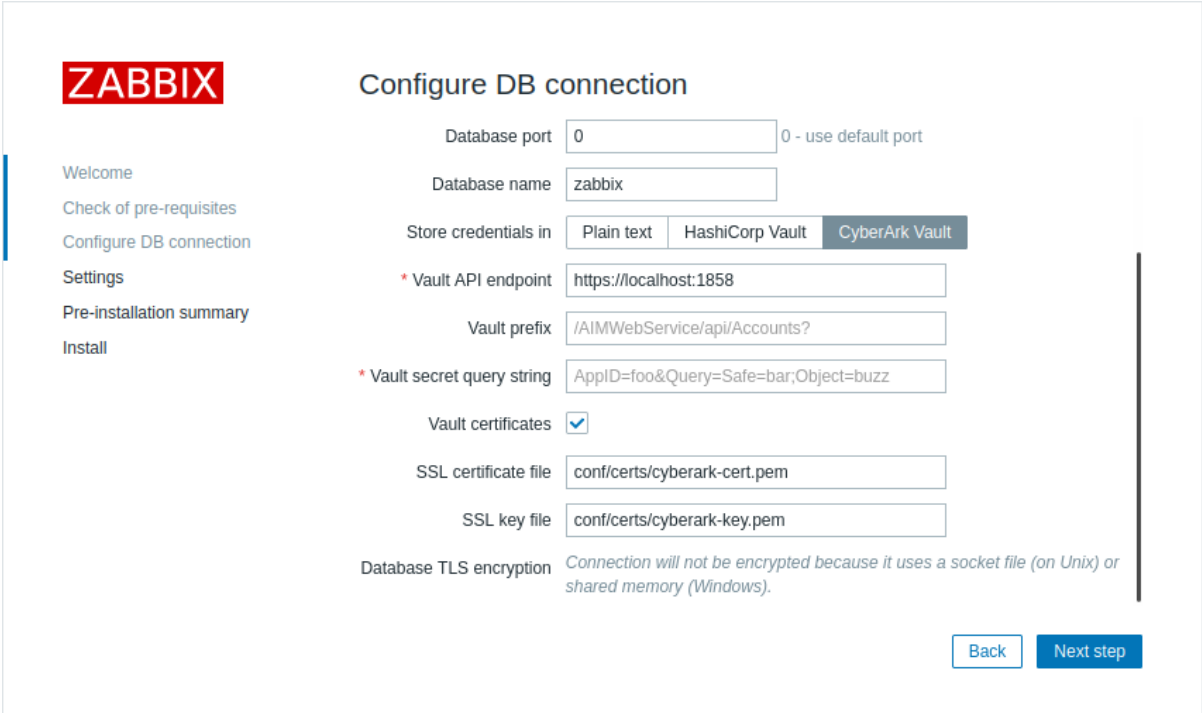
4. Rezultātā Zabbix datubāzes autentifikācijai izmantos šādus akreditācijas datus:

- Lietotājvārds: <username>
- Parole: <password>

Lietotāja saskarne

Lai iegūtu datubāzes akreditācijas datus no glabātuves Zabbix lietotāja saskarnei, lietotāja saskarnes instalēšanas laikā norādiet šādus parametrus.

1. Solī *Configure DB Connection* iestatiet parametru *Store credentials in* uz "CyberArk Vault".



2. Pēc tam aizpildiet papildu parametrus:

Parametrs	Obligāts	Noklusējuma vērtība	Apraksts
Vault API endpoint	jā	https://localhost:1858	Norādiet URL savienojumam ar glabātuvi formātā `scheme://host:port`
Vault prefix	nē	/AIMWebService/api/Accounts	Norādiet pielāgotu prefiksu glabātuves ceļam vai vaicājumam. Ja tas nav norādīts, tiek izmantota noklusējuma vērtība.
Vault secret query string	jā		Vaicājums, kas nosaka, no kurienes jāiegūst datubāzes akreditācijas dati. Piemērs: `AppID=foo&Query=Safe=bar;Object=buzz`
Vault certificates	nē		Atzīmējot izvēles rūtiņu, tiks parādīti papildu parametri, kas ļaus konfigurēt klienta autentifikāciju. Lai gan šis parametrs nav obligāts, ir ļoti ieteicams to iespējot saziņai ar CyberArk Vault.
SSL certificate file	nē	conf/certs/cyberark-cert.pem	Ceļš uz SSL sertifikāta failu. Failam jābūt PEM formātā. Ja sertifikāta fails satur arī privāto atslēgu, atstājiet parametru SSL key file tukšu.
SSL key file	nē	conf/certs/cyberark-key.pem	SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai. Failam jābūt PEM formātā.

Lietotāja makro vērtības

Lai izmantotu CyberArk Vault *Vault secret* lietotāja makro vērtību glabāšanai, pārliecinieties, ka:

- Zabbix serveris ir konfigurēts darbam ar CyberArk Vault;
- parametrs *Vault provider* sadaļā *Administrēšana → Vispārīgi → Citi* ir iestatīts uz "CyberArk Vault".

Note:
Zabbix serverim (un Zabbix starpniekserverim, ja tas ir konfigurēts) ir nepieciešama piekļuve *Vault secret* makro vērtībām no glabātuves. Zabbix lietotāja saskarnei šāda piekļuve nav nepieciešama.

Makro vērtībai jāsatur vaicājums (formātā `query:key`).

Detalizētu informāciju par makro vērtību apstrādi Zabbix skatiet sadaļā *Vault secret macros*.

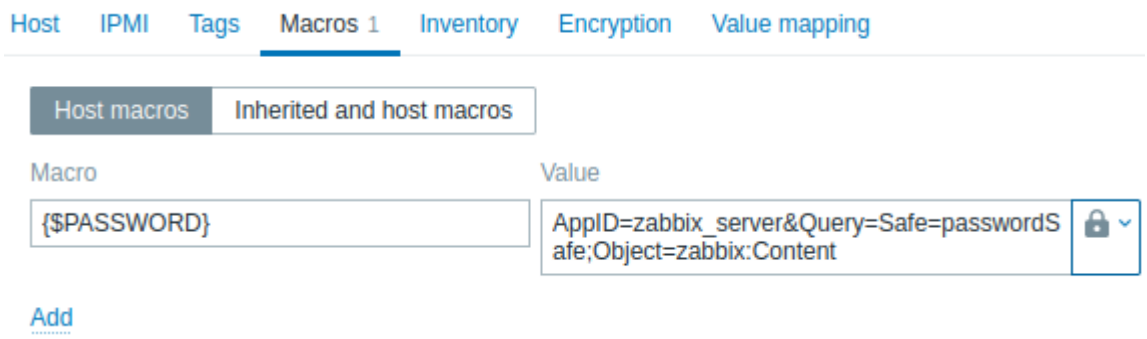
Vaicājuma sintakse

Kola simbols (":") ir rezervēts vaicājuma atdalīšanai no atslēgas.

Ja pats vaicājums satur slīpsvītru vai kolu, šie simboli ir jākodē URL formātā ("/" tiek kodēts kā "%2F", ":" tiek kodēts kā "%3A").

Piemērs

1. Zabbix pievienojiet lietotāja makrosu {$PASSWORD} ar tipu *Vault secret* un vērtību `AppID=zabbix_server&Query=Safe=passwordS`



2. Zabbix nosūtīs uz glabātuvi šādu API pieprasījumu:

```
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. Glabātuves atbilde saturēs atslēgu "Content":

```
{
    "Content": <password>,
    "UserName": <username>,
    "Address": <address>,
    "Database" :<Database>,
    "PasswordChangeInProcess":<PasswordChangeInProcess>
}
```

4. Rezultātā Zabbix atrisinās makrosu {$PASSWORD} uz vērtību - <password>

Esošās konfigurācijas atjaunināšana

Lai atjauninātu esošu konfigurāciju noslēpumu iegūšanai no CyberArk Vault:

1. Atjauniniet Zabbix servera vai starpniekservera konfigurācijas faila parametrus, kā aprakstīts sadaļā *Datubāzes akreditācijas dati*.
2. Atjauniniet DB savienojuma iestatījumus, pārkonfigurējot Zabbix lietotāja saskarni un norādot nepieciešamos parametrus, kā aprakstīts sadaļā *Lietotāja saskarne*. Lai pārkonfigurētu Zabbix lietotāja saskarni, atveriet lietotāja saskarnes iestatīšanas URL pārlūkprogrammā:

- Apache gadījumā: http://<server_ip_or_name>/zabbix/setup.php
- Nginx gadījumā: http://<server_ip_or_name>/setup.php

Alternatīvi, šos parametrus var iestatīt lietotāja saskarnes konfigurācijas failā (*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. Ja nepieciešams, konfigurējiet lietotāja makrosus, kā aprakstīts sadaļā *Lietotāja makrosu vērtības*.

Lai atjauninātu esošu konfigurāciju noslēpumu iegūšanai no HashiCorp Vault, skatiet *HashiCorp konfigurācija*.

2 HashiCorp konfigurācija

Pārskats

Šajā sadaļā ir skaidrots, kā konfigurēt Zabbix, lai iegūtu noslēpumus no HashiCorp Vault KV Secrets Engine — 2. versijas.

Vault jāizvieto un jākonfigurē, kā aprakstīts oficiālajā HashiCorp dokumentācijā.

Lai uzzinātu par TLS konfigurēšanu Zabbix, skatiet *Noslēpumu glabāšana*.

Datubāzes akreditācijas datu iegūšana

Lai veiksmīgi iegūtu noslēpumu ar datubāzes akreditācijas datiem, ir jākonfigurē abi šie komponenti:

- Zabbix serveris/starpniekserveris
- Zabbix lietotāja saskarne

Serveris/starpniekserveris

Lai konfigurētu Zabbix serveri vai starpniekserveri, konfigurācijas failā norādiet šādus konfigurācijas parametrus:

- `Vault` — kurš vault nodrošinātājs ir jāizmanto;
- `VaultToken` — vault autentifikācijas marķieris (sīkāku informāciju skatiet Zabbix servera/starpniekservera konfigurācijas failā);
- `VaultURL` — vault servera HTTP[S] URL;
- `VaultDBPath` — ceļš uz vault noslēpumu, kas satur datubāzes akreditācijas datus (šo opciju var izmantot tikai tad, ja `DBUser` un `DBPassword` nav norādīti); Zabbix serveris vai starpniekserveris iegūs akreditācijas datus pēc atslēgām "password" un "username";
- `VaultPrefix` — pielāgots prefikss vault ceļam vai vaicājumam atkarībā no vault; ja tas nav norādīts, tiks izmantota vispiemērotākā noklusējuma vērtība.

Attention:
Konfigurācijas parametri `Vault`, `VaultToken`, `VaultURL` un `VaultPrefix` tiek izmantoti arī vault autentifikācijai, kad Zabbix serveris apstrādā slepeno vault makrosu (un Zabbix starpniekserveris, ja tas ir konfigurēts). Zabbix serveris un starpniekserveri neatvērs vault slepenos makrosus, kas satur datubāzes akreditācijas datus no `VaultDBPath`.

Ir stingri ieteicams dažādiem starpniekserveriem izmantot atšķirīgus marķierus.

Zabbix serveris un Zabbix starpniekserveris ar vault saistītos konfigurācijas parametrus nolasa no *zabbix_server.conf* un *zabbix_proxy.conf* startēšanas laikā. Papildus tam Zabbix serveris un Zabbix starpniekserveris startēšanas laikā vienreiz nolasīs vides mainīgo `VAULT_TOKEN` un pēc tam to atiestatīs, lai tas nebūtu pieejams caur atzarotiem skriptiem; tā ir kļūda, ja gan `VaultToken`, gan `VAULT_TOKEN` parametri satur vērtību.

Piemērs

1. Failā *zabbix_server.conf* norādiet šādus parametrus:

```
Vault=HashiCorp
VaultToken=hvs.CAESIIG_PILmULFYOsEyWHxkZ2mF2a8VPKNLE8eHqd4autYGGh4KHGh2cy5aeTYONFNSaUp3ZnpWbDF1RUNjUkNTZEg
VaultURL=https://127.0.0.1:8200
VaultDBPath=database
VaultPrefix=/v1/secret/data/zabbix/
```

2. Izpildiet šādas CLI komandas, lai glabātuvē izveidotu nepieciešamo noslēpumu:

```
#### Iespējojiet "secret/" montēšanas punktu, ja tas vēl nav iespējots; ņemiet vērā, ka jāizmanto "kv-v2".
vault secrets enable -path=secret/ kv-v2

#### Pievienojiet jaunus noslēpumus ar atslēgām username un password zem montēšanas punkta "secret/" un ce
vault kv put -mount=secret zabbix/database username=zabbix password=<password>

#### Pārbaudiet, vai noslēpums ir veiksmīgi pievienots.
vault kv get secret/zabbix/database
```

```
#### Visbeidzot pārbaudiet ar Curl; ņemiet vērā, ka "data" manuāli jāpievieno pēc montēšanas punkta un "/v
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/database
```

3. Rezultātā Zabbix serveris iegūs šādus akreditācijas datus datubāzes autentifikācijai:

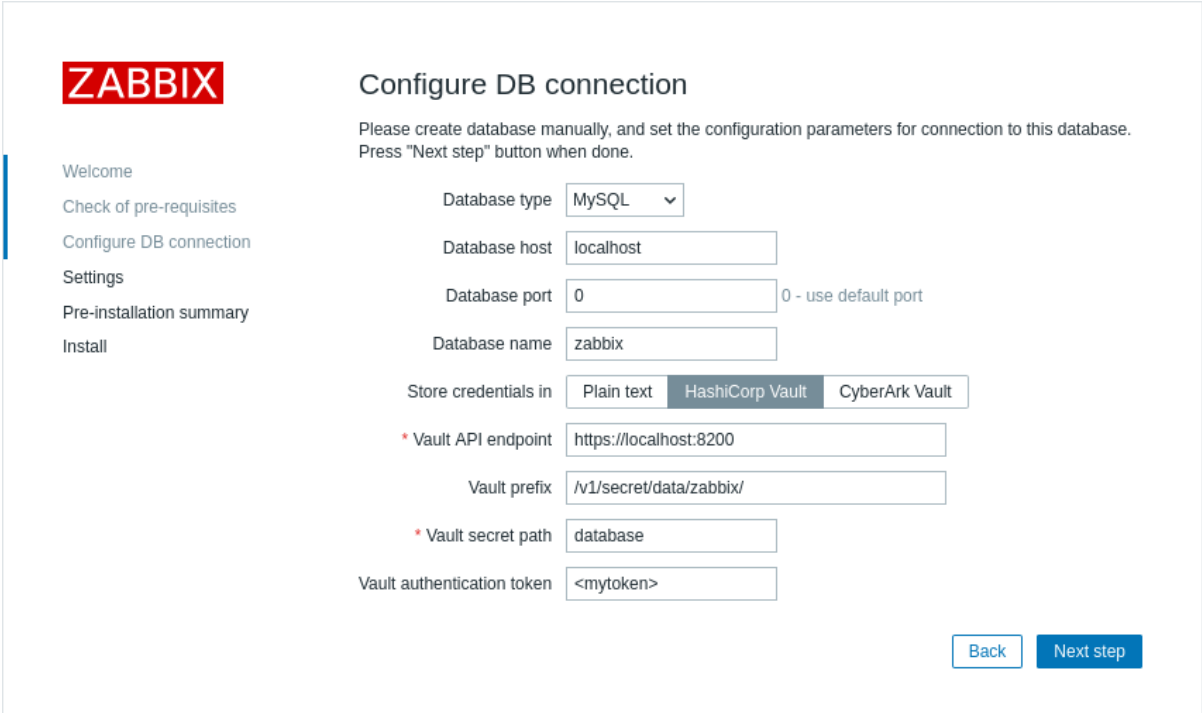
- Lietotājvārds: zabbix
- Parole: <password>

Lietotāja saskarne

Zabbix lietotāja saskarni var konfigurēt tā, lai tā iegūtu datubāzes akreditācijas datus no glabātuves vai nu lietotāja saskarnes instalēšanas laikā, vai atjauninot lietotāja saskarnes konfigurācijas failu (`zabbix.conf.php`).

Attention:
Ja kopš iepriekšējās lietotāja saskarnes instalēšanas glabātuves akreditācijas dati ir mainīti, atkārtoti palaidiet lietotāja saskarnes instalēšanu vai atjauniniet `zabbix.conf.php`. Skatiet arī: Esošās konfigurācijas atjaunināšana.

Lietotāja saskarnes instalēšanas laikā konfigurācijas parametri jānorāda solī *Configure DB Connection*:



- Iestatiet parametru *Store credentials in* uz "HashiCorp Vault".
- Norādiet savienojuma parametrus:

Parametrs	Obligāts	Noklusējuma vērtība	Apraksts
Vault API endpoint	jā	https://localhost:8200	Norādiet URL savienojumam ar glabātuvi formātā `scheme://host:port`
Vault prefix	nē	/v1/secret/data/	Norādiet pielāgotu prefiksu glabātuves ceļam vai vaicājumam. Ja tas nav norādīts, tiek izmantota noklusējuma vērtība. Piemērs: `/v1/secret/data/zabbix/`
Vault secret path	nē		Ceļš uz noslēpumu, no kura datubāzes akreditācijas dati tiks iegūti, izmantojot atslēgas "password" un "username". Piemērs: `database`
Vault authentication token	nē		Norādiet autentifikācijas marķieri tikai lasīšanas piekļuvei noslēpuma ceļam. Informāciju par marķieru un glabātuves politiku izveidi skatiet HashiCorp dokumentācijā.

Lietotāja makrosu vērtību izgūšana

Lai izmantotu HashiCorp Vault *Vault secret* lietotāja makrosu vērtību glabāšanai, pārliecinieties, ka:

- Zabbix serveris/starpniekserveris ir konfigurēts darbam ar HashiCorp Vault;
- parametrs *Vault provider* sadaļā *Administrēšana → Vispārīgi → Citi* ir iestatīts uz "HashiCorp Vault" (noklusējums);

Note:
Zabbix serverim (un Zabbix starpniekserverim, ja tas ir konfigurēts) ir nepieciešama piekļuve *Vault secret* makrosu vērtībām no glabātuves. Zabbix lietotāja saskarnei šāda piekļuve nav nepieciešama.

Makrosa vērtībai jāsatur atsauces ceļš (kā `path:key`, piemēram, `macros:password`). Autentifikācijas marķierim, kas norādīts Zabbix servera/starpniekservera konfigurācijas laikā (ar parametru `VaultToken`), jānodrošina tikai lasīšanas piekļuve šim ceļam.

Detalizētu informāciju par makrosa vērtību apstrādi Zabbix skatiet sadaļā *Vault secret macros*.

Ceļa sintakse

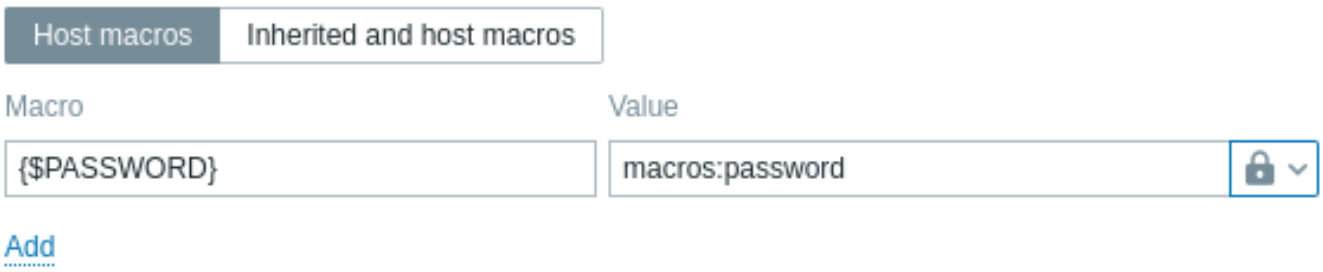
Simboli slīpsvītra uz priekšu ("/") un kols (":") ir rezervēti.

Slīpsvītru uz priekšu var izmantot tikai, lai atdalītu montēšanas punktu no ceļa (piemēram, *secret/zabbix*, kur montēšanas punkts ir "secret" un ceļš ir "zabbix"). Vault makrosu gadījumā kolu var izmantot tikai, lai atdalītu ceļu/vaicājumu no atslēgas.

Ir iespējams URL kodēt slīpsvītras uz priekšu un kola simbolus, ja ir nepieciešams izveidot montēšanas punktu ar nosaukumu, kas ir atdalīts ar slīpsvītru uz priekšu (piemēram, *foo/bar/zabbix*, kur montēšanas punkts ir "foo/bar" un ceļš ir "zabbix", var tikt kodēts kā "foo%2Fbar/zabbix"), un ja montēšanas punkta nosaukumā vai ceļā ir jāiekļauj kols.

Piemērs

1. Zabbix pievienojiet lietotāja makrosu {$PASSWORD} ar tipu "Vault secret" un vērtību `macros:password`



2. Palaidiet šādas CLI komandas, lai izveidotu nepieciešamo noslēpumu glabātuvē:

```
#### Iespējojiet "secret/" montēšanas punktu, ja tas vēl nav iespējots; ņemiet vērā, ka jāizmanto "kv-v2".
vault secrets enable -path=secret/ kv-v2

#### Ievietojiet jaunu noslēpumu ar atslēgu "password" zem montēšanas punkta "secret/" un ceļa "zabbix/mac
vault kv put -mount=secret zabbix/macros password=<password>

#### Pārbaudiet, vai noslēpums ir veiksmīgi pievienots.
vault kv get secret/zabbix/macros

#### Visbeidzot pārbaudiet ar Curl; ņemiet vērā, ka "data" manuāli jāpievieno aiz montēšanas punkta un "/
curl --header "X-Vault-Token: <VaultToken>" https://127.0.0.1:8200/v1/secret/data/zabbix/macros
```

3. Rezultātā Zabbix atrisinās makrosu {$PASSWORD} uz vērtību: <password>

Esošās konfigurācijas atjaunināšana

Lai atjauninātu esošu konfigurāciju slepeno datu iegūšanai no HashiCorp Vault:

1. Atjauniniet Zabbix servera vai starpniekservera konfigurācijas faila parametrus, kā aprakstīts sadaļā *Datubāzes akreditācijas dati*.

2. Atjauniniet DB savienojuma iestatījumus, pārkonfigurējot Zabbix lietotāja saskarne un norādot nepieciešamos parametrus, kā aprakstīts sadaļā *Lietotāja saskarne*. Lai pārkonfigurētu Zabbix lietotāja saskarne, atveriet lietotāja saskarnes iestatīšanas URL pārlūkprogrammā:

- Apache gadījumā: http://<server_ip_or_name>/zabbix/setup.php
- Nginx gadījumā: http://<server_ip_or_name>/setup.php

Alternatīvi, šos parametrus var iestatīt lietotāja saskarnes konfigurācijas failā (*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/';
```

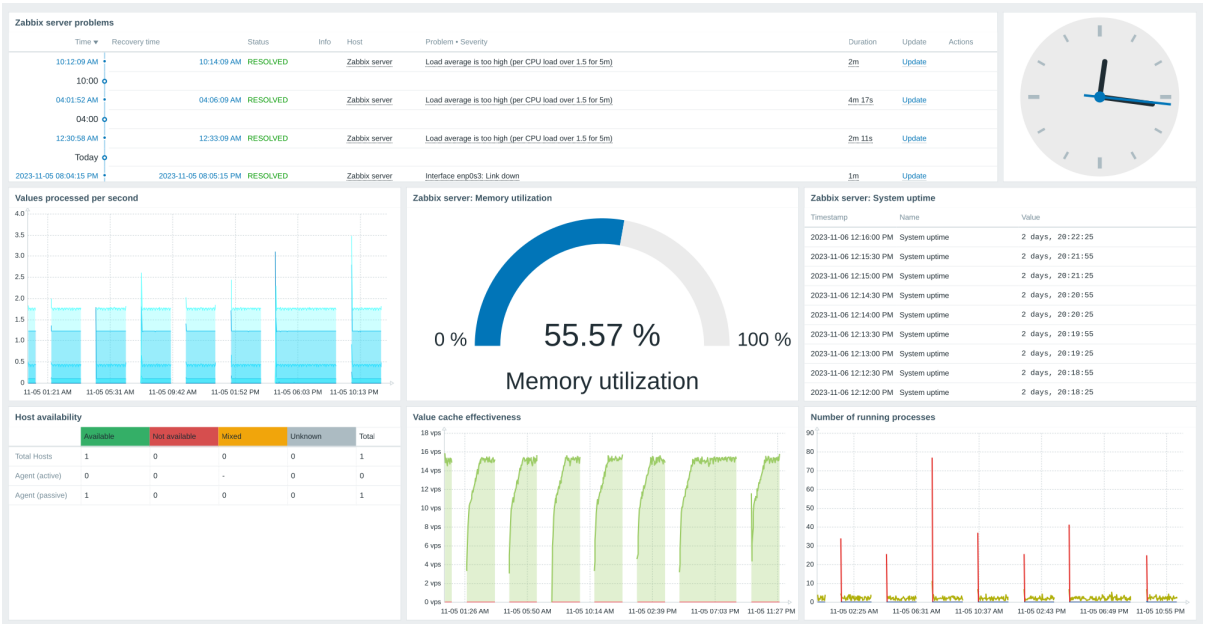
3. Ja nepieciešams, konfigurējiet lietotāja makrosus, kā aprakstīts sadaļā *Lietotāja makrosu vērtības*.

Lai atjauninātu esošu konfigurāciju slepeno datu iegūšanai no CyberArk Vault, skatiet *CyberArk konfigurācija*.

14 Plānotie pārskati

Pārskats

Izmantojot *Plānotie pārskati* funkciju, varat iestatīt, lai noteikta paneļa PDF versija tiktu nosūtīta norādītajiem saņēmējiem regulāros intervālos.



Priekšnosacījumi:

- Zabbix tīmekļa servisam jābūt instalētam un pareizi konfigurētam, lai iespējotu plānoto pārskatu ģenerēšanu — instrukcijas skatiet sadaļā Plānoto pārskatu iestatīšana.
- lietotājam jābūt ar lietotāja lomu tipa *Admin* vai *Super admin* ar šādām atļaujām:
  - *Scheduled reports* blokā *Access to UI elements* (lai skatītu pārskatu iestatījumus)
  - *Manage scheduled reports* blokā *Access to actions* (lai izveidotu/rediģētu pārskatus)

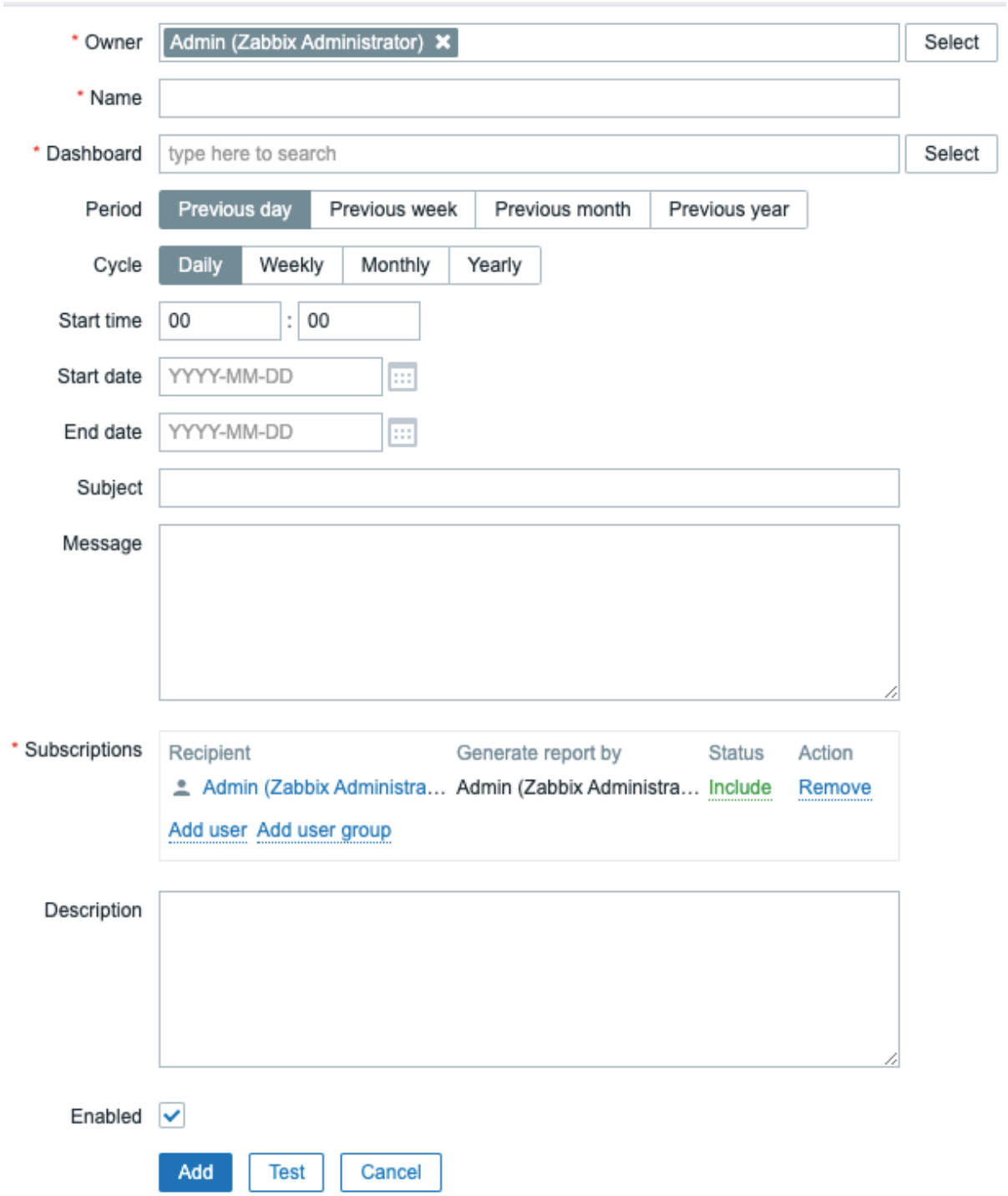
Lai izveidotu plānoto pārskatu Zabbix lietotāja saskarnē, rīkojieties šādi:

- Dodieties uz: *Reports* > *Scheduled reports*.
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Create report*.
- Veidlapā ievadiet pārskata parametrus.

Pārskatu varat izveidot arī, atverot esošu pārskatu, noklikšķinot uz pogas *Clone* un pēc tam saglabājot to ar citu nosaukumu.

Konfigurācija

Cilnē *Plānotie pārskati* ir ietverti pārskata vispārīgie atribūti.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Īpašnieks	Lietotājs, kas izveido pārskatu. Lietotājiem ar *Super admin* līmeni ir atļauts mainīt īpašnieku. Lietotājiem ar *Admin* līmeni šis lauks ir tikai lasāms.
Nosaukums	Pārskata nosaukums; tam jābūt unikālam.
Panelis	Panelis, uz kura balstīts pārskats; vienlaikus var atlasīt tikai vienu paneli. Lai atlasītu paneli, sāciet rakstīt nosaukumu — tiks parādīts atbilstošo paneļu saraksts; ritiniet uz leju, lai atlasītu. Varat arī noklikšķināt uz *Atlasīt* blakus laukam un atlasīt paneli no parādītā saraksta.
Periods	Laika periods, par kuru tiks sagatavots pārskats. Atlasiet iepriekšējo dienu, nedēļu, mēnesi vai gadu.
Cikls	Pārskata ģenerēšanas biežums. Pārskatus var sūtīt katru dienu, nedēļu, mēnesi vai gadu. Režīmā ”Katru nedēļu” ir iespējams atlasīt nedēļas dienas, kurās pārskats tiks nosūtīts.

Parametrs	Apraksts
Sākuma laiks	Diennakts laiks formātā hh:mm, kad pārskats tiks sagatavots. Ņemiet vērā, ka tiks izmantota Zabbix servera laika josla.
Atkārtot	Nedēļas dienas, kurās pārskats tiks nosūtīts. Šis lauks ir pieejams tikai tad, ja *Cikls* ir iestatīts uz "Katru nedēļu".
Sākuma datums	Datums, kad jāsāk regulāra pārskata ģenerēšana.
Beigu datums	Datums, kad jāpārtrauc regulāra pārskata ģenerēšana.
Temats	Pārskata e-pasta temats. Atbalstītie makrosi: {TIME}, {TIMESTAMP}.
Ziņojums	Pārskata e-pasta pamatteksts. Atbalstītie makrosi: {TIME}, {TIMESTAMP}.
Abonementi	Pārskata saņēmēju saraksts. Pēc noklusējuma tajā ir iekļauts tikai pārskata īpašnieks. Kā pārskata saņēmēju var norādīt jebkuru Zabbix lietotāju ar konfigurētu e-pasta mediju. Noklikšķiniet uz *Pievienot lietotāju* vai *Pievienot lietotāju grupu*, lai pievienotu vēl saņēmējus. Noklikšķiniet uz lietotājvārda, lai rediģētu iestatījumus: *Ģenerēt pārskatu pēc* — vai pārskata dati jāģenerē, pamatojoties uz pašreizējā lietotāja vai saņēmēja paneļa atļaujām. *Statuss* — atlasiet "Iekļaut", lai nosūtītu pārskatu lietotājam, vai "Izslēgt", lai novērstu pārskata nosūtīšanu šim lietotājam. Vismaz vienam lietotājam jābūt statusam "Iekļaut". Statusu "Izslēgt" var izmantot, lai izslēgtu konkrētus lietotājus no iekļautas lietotāju grupas. Ņemiet vērā, ka lietotāji ar nepietiekamām atļaujām (tas ir, lietotāji ar lomu, kas balstīta uz *Admin* lietotāja tipu un kuri nav tās pašas lietotāju grupas dalībnieki kā saņēmējs vai pārskata īpašnieks) lauku *Saņēmējs* un *Ģenerēt pārskatu pēc* vietā faktisko nosaukumu vietā redzēs "Nepieejams lietotājs" vai "Nepieejama lietotāju grupa"; lauki *Statuss* un *Darbība* tiks parādīti tikai lasīšanas režīmā.
Iespējots	Pārskata statuss. Noņemot atzīmi no šīs izvēles rūtiņas, pārskats tiks atspējots.
Apraksts	Neobligāts pārskata apraksts. Šis apraksts ir paredzēts iekšējai lietošanai un netiks nosūtīts pārskata saņēmējiem.

Veidlapas pogas

Pogas veidlapas apakšdaļā ļauj veikt vairākas darbības.

Add	Pievienot atskaiti. Šī poga ir pieejama tikai jaunām atskaitēm.
Update	Atjaunināt atskaites īpašības.
Clone	Izveidot citu atskaiti, pamatojoties uz pašreizējās atskaites īpašībām.
Test	Pārbaudīt, vai atskaites konfigurācija ir pareiza, nosūtot atskaiti pašreizējam lietotājam.
Delete	Dzēst atskaiti.
Cancel	Atcelt atskaites īpašību rediģēšanu.

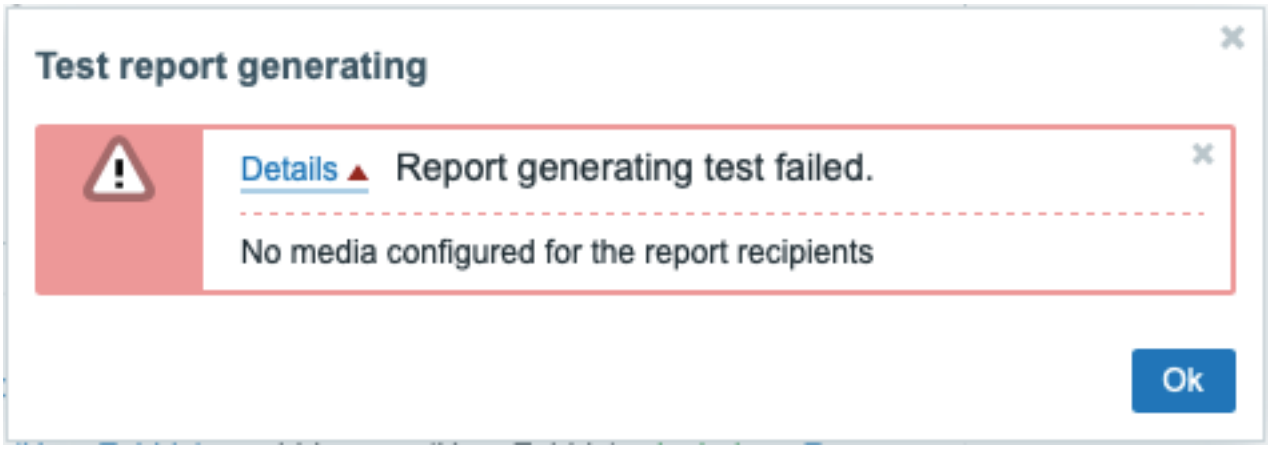
Testēšana

Lai pārbaudītu atskaiti, noklikšķiniet uz pogas *Test* atskaites konfigurācijas veidlapas apakšdaļā.

Note:
Poga *Test* nav pieejama, ja atskaites konfigurācijas veidlapa ir atvērta no paneļa darbību izvēlnes.

Ja konfigurācija ir pareiza, testa atskaite tiek nekavējoties nosūtīta pašreizējam lietotājam. Testa atskaitēm abonenti un lietotāja iestatījumi *Generate report by* tiek ignorēti.

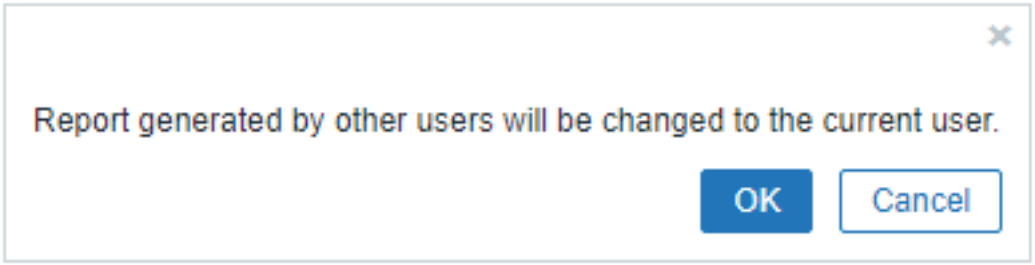
Ja konfigurācija nav pareiza, tiek parādīts kļūdas ziņojums ar iespējamā cēloņa aprakstu.



Atskaites atjaunināšana

Lai atjauninātu esošu atskaiti, noklikšķiniet uz atskaites nosaukuma, veiciet nepieciešamās konfigurācijas izmaiņas un pēc tam noklikšķiniet uz pogas *Update*.

Ja esošu atskaiti ir atjauninājis cits lietotājs un šis lietotājs maina paneli, tad, noklikšķinot uz pogas *Update*, tiks parādīts brīdinājuma ziņojums ”Report generated by other users will be changed to the current user”.



Noklikšķinot uz *OK* šajā solī, tiks veiktas šādas izmaiņas:

- Iestatījums *Generate report by* tiks atjaunināts, lai parādītu lietotāju, kurš atskaiti rediģēja pēdējais (ja vien *Generate report by* nav iestatīts uz saņēmēju).
- Lietotāji, kas tika parādīti kā ”Inaccessible user” vai ”Inaccessible user group”, tiks dzēsti no atskaites abonentu saraksta.

Noklikšķinot uz *Cancel*, konfigurācijas forma tiks aizvērta un atskaites atjaunināšana tiks atcelta.

Pārskata klonēšana

Lai ātri klonētu esošu pārskatu, noklikšķiniet uz pogas *Klonēt* esoša pārskata konfigurācijas formas apakšdaļā. Klonējot cita lietotāja izveidotu pārskatu, pašreizējais lietotājs kļūst par jaunā pārskata īpašnieku.

Pārskata iestatījumi tiks kopēti uz jaunā pārskata konfigurācijas formu, ievērojot lietotāja atļaujas:

- Ja lietotājam, kurš klonē pārskatu, nav atļauju informācijas panelim, lauks *Informācijas panelis* tiks notīrīts.
- Ja lietotājam, kurš klonē pārskatu, nav atļauju dažiem lietotājiem vai lietotāju grupām sarakstā *Abonementi*, nepieejamie saņēmēji netiks klonēti.
- Iestatījumi *Ģenerēt pārskatu kā* tiks atjaunināti, lai parādītu pašreizējo lietotāju (ja vien *Ģenerēt pārskatu kā* nav iestatīts uz saņēmēju).

Mainiet nepieciešamos iestatījumus un pārskata nosaukumu, pēc tam noklikšķiniet uz *Pievienot*.

15 Datu eksportēšana

Pārskats

Zabbix atbalsta datu eksportu reāllaikā divos veidos:

- eksportēšana uz failiem
- straumēšana uz ārējām sistēmām

Var eksportēt šādas entītijas:

- trigeru notikumi
- vienumu vērtības

- tendences (tikai eksportēšana uz failiem)

1 Eksportēšana uz failiem

Pārskats

Ir iespējams konfigurēt trigeru notikumu, vienumu vērtību un tendenču eksportēšanu reāllaikā JSON formātā ar rindām, kas atdalītas ar jaunu rindu.

Eksportēšana tiek veikta failos, kur katra eksporta faila rinda ir JSON objekts. Vērtību kartējumi netiek piemēroti.

Kļūdu gadījumā (datus nevar ierakstīt eksporta failā vai eksporta failu nevar pārdēvēt, vai pēc tā pārdēvēšanas nevar izveidot jaunu failu), datu vienums tiek atmests un nekad netiek ierakstīts eksporta failā. Tas tiek ierakstīts tikai Zabbix datubāzē. Datu rakstīšana eksporta failā tiek atsākta, kad rakstīšanas problēma ir novērsta.

Eksporta fails tiek izveidots ar lasīšanas un rakstīšanas atļaujām tikai faila īpašniekam. Papildus tam fails ir lasāms īpašnieka grupai. Visas pārējās atļaujas ir liegtas.

Precīzai informācijai par to, kāda informācija tiek eksportēta, skatiet eksporta protokola lapu.

Ņemiet vērā, ka hostam/vienumam var nebūt metadatu (hostu grupas, hosta nosaukums, vienuma nosaukums), ja hosts/vienums tika noņemts pēc datu saņemšanas, bet pirms serveris eksportēja datus.

Konfigurācija

Trigeru notikumu, vienumu vērtību un tendenču eksportēšana reāllaikā tiek konfigurēta, norādot direktoriju eksporta failiem — skatiet parametru ExportDir servera konfigurācijā.

Ir pieejami vēl divi parametri:

- `ExportFileSize` var izmantot, lai iestatītu maksimāli pieļaujamo atsevišķa eksporta faila lielumu. Kad procesam ir jāraksta failā, tas vispirms pārbauda faila lielumu. Ja tas pārsniedz konfigurēto lieluma ierobežojumu, fails tiek pārdēvēts, tā nosaukumam pievienojot .old, un tiek izveidots jauns fails ar sākotnējo nosaukumu.

Attention:
Katram procesam, kas rakstīs datus, tiks izveidots atsevišķs fails (t.i., aptuveni 4–30 faili). Tā kā noklusējuma lielums katram eksporta failam ir 1G, lielu eksporta failu glabāšana var ātri iztērēt diska vietu.

- `ExportType` ļauj norādīt, kuri entītiju tipi (notikumi, vēsture, tendences) tiks eksportēti.

2 Straumēšana uz ārējām sistēmām

Pārskats

Ir iespējams straumēt vienumu vērtības un notikumus no Zabbix uz ārējām sistēmām, izmantojot HTTP (skatiet protokola informāciju).

Birkas filtru var izmantot, lai straumētu vienumu vērtību vai notikumu apakškopas.

Par datu straumēšanu ir atbildīgi divi Zabbix servera procesi: `connector manager` un `connector worker`. Zabbix iekšējais vienums `zabbix[connector_queue]` ļauj uzraudzīt savienotāja rindā ievietoto vērtību skaitu.

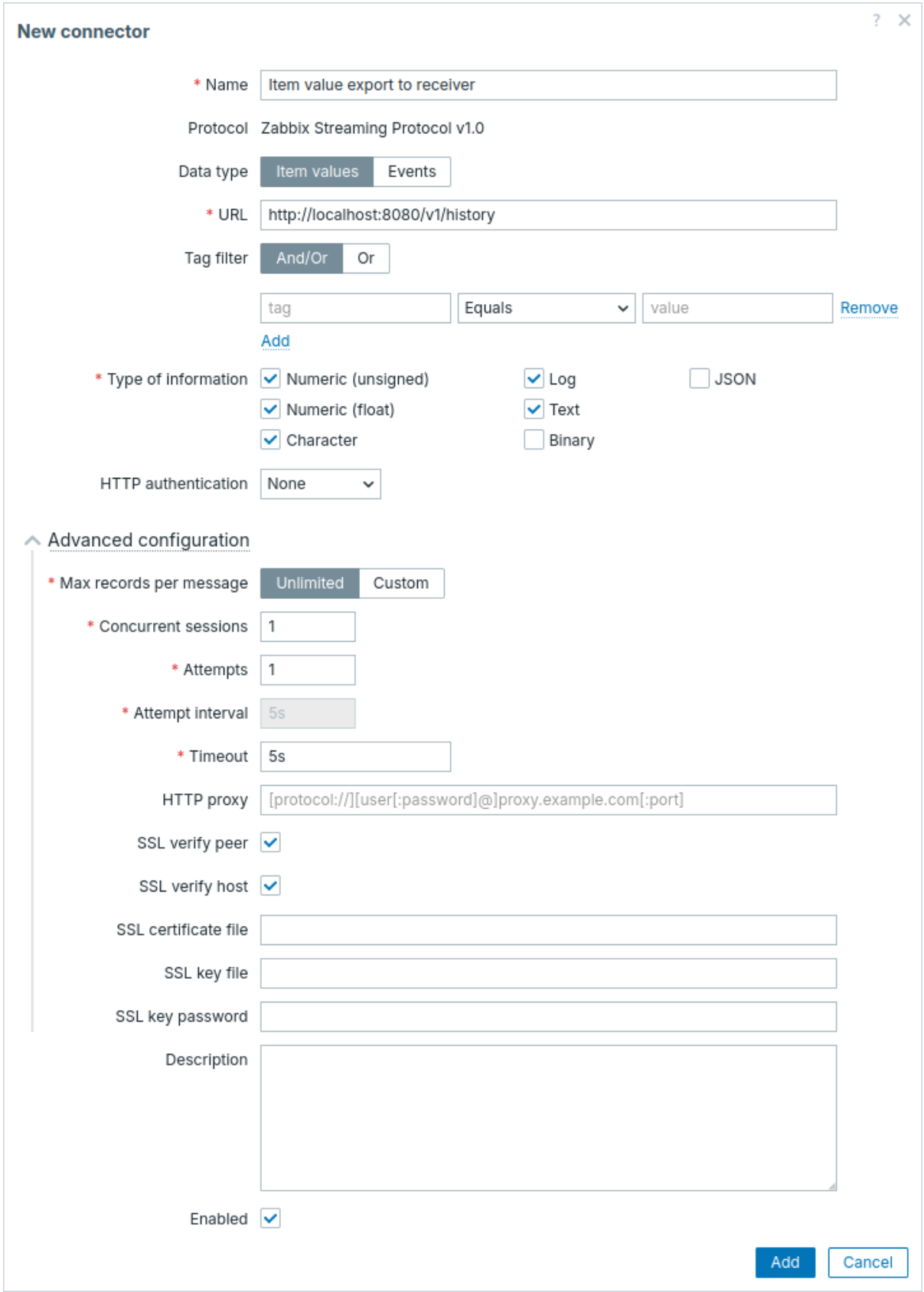
Konfigurācija

Lai konfigurētu datu straumēšanu uz ārēju sistēmu, ir jāveic šādas darbības:

1. Jābūt attālinātai sistēmai, kas ir iestatīta datu saņemšanai no Zabbix. Šim nolūkam ir pieejami šādi rīki:
   - Vienkārša uztvērēja piemērs, kas reģistrē saņemto informāciju failos `events.ndjson` un `history.ndjson`.
   - Kafka connector for Zabbix server - viegls serveris, kas rakstīts Go valodā un paredzēts vienumu vērtību un notikumu pārsūtīšanai no Zabbix servera uz Kafka brokeri.

2. Iestatiet nepieciešamo savienotāju darbinieku skaitu Zabbix, pielāgojot parametru `StartConnectors` failā `zabbix_server.conf`. Savienotāju darbinieku skaitam jāatbilst (vai jāpārsniedz, ja vienlaicīgo sesiju skaits ir lielāks par 1) konfigurētajam savienotāju skaitam Zabbix lietotāja saskarnē. Pēc tam restartējiet Zabbix serveri.

3. Konfigurējiet jaunu savienotāju Zabbix lietotāja saskarnē (*Administration* > *General* > *Connectors*) un pārlādējiet servera kešatmiņu ar komandu `zabbix_server -R config_cache_reload`.



Obligātie lauki ir atzīmēti ar zvaigznīti.

Parameter	Description
Name	Ievadiet savienotāja nosaukumu.
Data type	Atlasiet straumējamo datu tipu: **Item values** - straumēt vienumu vērtības no Zabbix uz ārējām sistēmām; **Events** - straumēt notikumus no Zabbix uz ārējām sistēmām.
URL	Ievadiet uztvērēja URL. Tiek atbalstīti lietotāja makrosi.

Parameter	Description
Tag filter	Eksportēt tikai tās vienumu vērtības vai notikumus, kas atbilst birku filtram. Ja tas nav iestatīts, tiek eksportēts viss. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir pieejami divi aprēķina tipi: **And/Or** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Or nosacījuma; **Or** - pietiek, ja ir izpildīts viens nosacījums.
Type of information	Atlasiet informācijas tipu (numeric (unsigned), numeric (float), character u.c.), pēc kura filtrēt vienumu vērtības, kuras savienotājam jāstraumē. Šis lauks ir pieejams, ja *Data type* ir iestatīts uz ”Item values”.
HTTP authentication	Atlasiet autentifikācijas opciju: **None** - autentifikācija netiek izmantota; **Basic** - tiek izmantota pamata autentifikācija; **NTLM** - tiek izmantota NTLM (Windows NT LAN Manager) autentifikācija; **Kerberos** - tiek izmantota Kerberos autentifikācija (skatiet arī: Configuring Kerberos with Zabbix); **Digest** - tiek izmantota Digest autentifikācija; **Bearer** - tiek izmantota Bearer autentifikācija.
Username	Ievadiet lietotājvārdu (līdz 255 rakstzīmēm). Tiek atbalstīti lietotāja makrosi. Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz ”Basic”, ”NTLM”, ”Kerberos” vai ”Digest”.
Password	Ievadiet lietotāja paroli (līdz 255 rakstzīmēm). Tiek atbalstīti lietotāja makrosi. Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz ”Basic”, ”NTLM”, ”Kerberos” vai ”Digest”.
Bearer token	Ievadiet Bearer tokenu. Tiek atbalstīti lietotāja makrosi. Šis lauks ir pieejams un obligāts, ja *HTTP authentication* ir iestatīts uz ”Bearer”.
Advanced configuration	Noklikšķiniet uz etiķetes *Advanced configuration*, lai parādītu paplašinātās konfigurācijas opcijas (skatiet tālāk).
Max records per message	Norādiet maksimālo vērtību vai notikumu skaitu, ko var straumēt vienā ziņojumā.
Concurrent sessions	Atlasiet sūtītāja procesu skaitu, kas jāpalaiž šim savienotājam. Var norādīt līdz 100 sesijām; noklusējuma vērtība ir ”1”.
Attempts	Datu straumēšanas mēģinājumu skaits. Var norādīt līdz 5 mēģinājumiem; noklusējuma vērtība ir ”1”.
Attempt interval	Norādiet, cik ilgi savienotājam jāgaida pēc neveiksmīga datu straumēšanas mēģinājuma. Var norādīt līdz 10s; noklusējuma vērtība ir ”5s”. Šis lauks ir pieejams, ja *Attempts* ir iestatīts uz ”2” vai vairāk. Neveiksmīgi mēģinājumi ir tie, kuros nav izdevies izveidot savienojumu vai kuros HTTP atbildes kods nav 200, 201, 202, 203, 204. Atkārtoti mēģinājumi tiek **aktivizēti**, ja ir sakaru kļūdas vai ja HTTP atbildes kods nav 200, 201, 202, 203, 204, 400, 401, 403, 404, 405, 415, 422. Pāradresācijas tiek sekotas, tāpēc 302 -> 200 ir pozitīva atbilde; savukārt 302 -> 503 izraisīs atkārtotu mēģinājumu.
Timeout	Norādiet ziņojuma noildzi (1-60 sekundes, noklusējums - 5 sekundes). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m. Tiek atbalstīti lietotāja makrosi.

Parameter	Description
HTTP proxy	Varat norādīt izmantojamo HTTP starpniekserveri šādā formātā: `[protocol://][username[:password]@]proxy.example.com[:port]` Tiek atbalstīti lietotāja makrosi. Neobligāto prefiksu `protocol://` var izmantot, lai norādītu alternatīvus starpniekservera protokolus (protokola prefiksa atbalsts tika pievienots cURL 7.21.7). Ja protokols nav norādīts, starpniekserveris tiks uzskatīts par HTTP starpniekserveri. Pēc noklusējuma tiks izmantots 1080 ports. Ja *HTTP proxy* ir norādīts, starpniekserveris pārrakstīs ar starpniekserveri saistītos vides mainīgos, piemēram, `http_proxy`, `HTTPS_PROXY`. Ja tas nav norādīts, starpniekserveris nepārrakstīs ar starpniekserveri saistītos vides mainīgos. Ievadītā vērtība tiek nodota tāda, kāda tā ir, bez jebkādas korektuma pārbaudes. Varat ievadīt arī SOCKS starpniekservera adresi. Ja norādīsiet nepareizu protokolu, savienojums neizdosies un vienums kļūs neatbalstīts. Ņemiet vērā, ka ar HTTP starpniekserveri tiek atbalstīta tikai vienkāršā autentifikācija.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiks automātiski paņemts no sistēmas sertifikācijas iestādes (CA) atrašanās vietas. Varat pārrakstīt CA failu atrašanās vietu, izmantojot Zabbix servera vai starpniekservera konfigurācijas parametru `SSLCALocation`.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta lauks *Common Name* vai lauks *Subject Alternate Name* atbilst. Tas iestata cURL opciju `CURLOPT_SSL_VERIFYHOST`.
SSL certificate file	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai. Sertifikāta failam jābūt PEM[1] formātā. Tiek atbalstīti lietotāja makrosi. Ja sertifikāta failā ir arī privātā atslēga, atstājiet lauku *SSL key file* tukšu. Ja atslēga ir šifrēta, norādiet paroli laukā *SSL key password*. Direktoriju, kurā atrodas šis fails, nosaka Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLCertLocation`.
SSL key file	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai. Privātās atslēgas failam jābūt PEM[1] formātā. Tiek atbalstīti lietotāja makrosi. Direktoriju, kurā atrodas šis fails, nosaka Zabbix servera vai starpniekservera konfigurācijas parametrs `SSLKeyLocation`.
SSL key password	SSL privātās atslēgas faila parole. Tiek atbalstīti lietotāja makrosi.
Description	Ievadiet savienotāja aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu savienotāju.

Ja Kafka savienotājs ir konfigurēts ar komatiem atdalītu bootstrap brokera adrešu sarakstu (piemēram, `Kafka.URL=kafka1.example.com:9`
Kafka klients izveido savienojumu ar brokeri(-iem), kas atbild pirmie, un izmanto to klastera metadatus. Ja sarakstā ir adreses no dažādiem Kafka klasteriem, tiks izmantots tikai visātrāk atbildējušais klasteris, bet pārējās adreses tiks reģistrētas kā nepieejamas; rezultātā startēšanas laikā var parādīties šādi brīdinājumi, pat ja savienotājs ir pieslēgts:

```
kafka cluster connected, but broker(s) "kafka1.example.com:9093, kafka2.example.com:9093" unavailable; wil
```

Dažās vidēs (privātie tīkli, konteineru tīkli vai nestandarta DNS/hosts iestatījumi) resursdatora nosaukumi vai IP adreses var tikt atrisinātas uz loopback adresēm (piemēram, `127.0.0.1/localhost`) vai klienta pusē normalizētas, kas šādus brīdinājumus var padarīt maldinošus. Lai mazinātu neskaidrības, pārliecinieties, ka visas `Kafka.URL` adreses pieder vienam un tam pašam Kafka klasterim, pārbaudiet DNS atrisināšanu no savienotāja hosta un brokeru `advertised.listeners`, kā arī dodiet priekšroku adresēm, kas tiek atrisinātas uz brokera izziņoto adresi.

Protokols

Saziņa starp serveri un saņēmēju notiek, izmantojot HTTP un REST API, NDJSON, ”Content-Type: application/x-ndjson”.

Lai iegūtu sīkāku informāciju, skatiet Ar jaunu rindu atdalīta JSON eksporta protokolu.

Servera pieprasījums

Piemērs vienumu vērtību straumēšanai:

```
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
```

Piemērs notikumu straumēšanai:

```
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}
```

Saņēmēja atbilde

Atbilde sastāv no HTTP atbildes statusa koda un JSON slodzes. HTTP atbildes statusa kodam jābūt ”200”, ”201”, ”202”, ”203” vai ”204” pieprasījumiem, kas tika veiksmīgi apstrādāti; citi kodi norāda uz neveiksmīgiem pieprasījumiem.

Veiksmīgas atbildes piemērs:

```
localhost:8080/v1/history": HTTP/1.1 200 OK
Date: Wed, 11 Jan 2023 16:40:30 GMT
Content-Length: 0
```

Piemērs ar kļūdām:

```
localhost:8080/v1/history": HTTP/1.1 422 Unprocessable Entity
Content-Type: application/json
Date: Wed, 11 Jan 2023 17:07:36 GMT
Content-Length: 55

{"error":"invalid character '{' after top-level value"}
```

3 SNMP vārteja

Pārskats

Zabbix SNMP gateway ir AgentX paplašinājums priekš snmpd, kas atbalsta gan SNMP aptaujāšanu, gan slazdus.

Ar Zabbix SNMP gateway ir iespējams izmantot SNMP protokolu, lai iegūtu:

- trigeru datus;
- problēmu trigeru datus;
- hostu grupu statusu (trigeru skaitu pēc trigeru stāvokļa katrā grupā)

Dati tiek iegūti pēc OID, kas ir kopīgas bāzes un konkrētas sufiksa daļas kombinācija. Kopīgā **bāze** tiek iestatīta SNMP gateway konfigurācijas failā, piemēram:

- `BaseOID=1.3.6.1.4.1.3043.7.55` - jebkuriem trigeru datiem;
- `ProblemBaseOID=1.3.6.1.4.1.3047.7.55` - problēmu trigeru datiem;
- `BaseOID=1.3.6.1.4.1.3046.7.55` - hostu grupu statusam.

OID **sufikss** tiek iestatīts hosta trigeru konfigurācijā kā **tags** (piemēram, `OIDSuffix:3`) lietotāja saskarnē.

Šajā gadījumā visa informācija par trigeri būs pieejama zem `OID=1.3.6.1.4.1.3043.7.55.X.3`. Šeit ”X” būs trigera datu lauku skaits (t.i., 1 - sufikss, 2 - ID, 3 - izteiksme, 4 - apraksts utt.).

Detalizētāku aprakstu un konfigurācijas faila piemēru skatiet SNMP gateway readme failā.

Instalēšana un iestatīšana

Norādījumus par tālāk minēto skatiet SNMP gateway readme failā:

- snmpd instalēšanu un konfigurēšanu;

- AgentX atbalsta iespējošanu;
- Zabbix SNMP gateway konfigurēšanu;
- SNMP slazdu konfigurēšanu trigera stāvokļa izmaiņām.

Datu iegūšana

Kad viss ir pareizi iestatīts, varat izmantot `snmpwalk` un `snmpget` komandas, lai iegūtu datus:

```
[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
...
```

Filtrēšanas opcijas

Jūs varat ierobežot problēmu trigeru informāciju SNMP gateway konfigurācijā:

- pēc svarīguma līmeņa (pēc noklusējuma `ProblemMinSeverity=-1`)
- paslēpjot apstiprinātās problēmas (pēc noklusējuma `ProblemHideAck=false`)

Jūs varat ierobežot problēmu skaitu katrā hostu grupā SNMP gateway konfigurācijā:

- pēc trigeriem ar nezināmu stāvokli (pēc noklusējuma `CountUnknown=false`)
- pēc trigeriem ar apstiprinātām/neapstiprinātām/visām problēmām (pēc noklusējuma `CountAcknowledgeStatus=all`)

6 Pakalpojumu uzraudzība

Pārskats Pakalpojumu uzraudzība ir biznesa līmeņa uzraudzība, ko var izmantot, lai iegūtu pārskatu par visu IT infrastruktūras pakalpojumu koku, identificētu infrastruktūras vājās vietas, aprēķinātu dažādu IT pakalpojumu SLA un skatītu citu informāciju augstākā līmenī. Pakalpojumu uzraudzība koncentrējas uz pakalpojuma kopējo pieejamību, nevis zema līmeņa detaļām, piemēram, diska vietas trūkumu, augstu procesora noslodzi u.c. Pakalpojumu uzraudzība nodrošina arī funkcionalitāti, lai noteiktu problēmas pamatcēloni, ja pakalpojums nedarbojas, kā paredzēts.

Pakalpojumu uzraudzība ļauj izveidot uzraudzīto datu hierarhisku attēlojumu.

Ļoti vienkārša pakalpojumu struktūra var izskatīties šādi:

```
Pakalpojums
|
|-Darbstacijas
| |
| |-Darbstacija1
| |
| |-Darbstacija2
|
|-Serveri
```

Katram struktūras mezglam ir statusa atribūts. Statuss tiek aprēķināts un izplatīts uz augstākiem līmeņiem saskaņā ar izvēlēto algoritmu. Atsevišķu mezglu statusu ietekmē kartēto problēmu statuss. Problēmu kartēšana tiek veikta ar atzīmēšanu.

Zabbix var sūtīt paziņojumus vai automātiski izpildīt skriptu Zabbix serveris, ja tiek konstatētas pakalpojuma stāvokļa izmaiņas. Ir iespējams definēt elastīgus noteikumus, vai vecākpakalpojumam jāpāriet uz "Problēmas stāvokli", pamatojoties uz bērnpakalpojumu statusiem. Pakalpojumu problēmu datus pēc tam var izmantot SLA aprēķināšanai un SLA atskaišu sūtīšanai, balstoties uz elastīgu nosacījumu kopu.

Pakalpojumu uzraudzība tiek konfigurēta izvēlnē Services, kas sastāv no šādām sadaļām:

- Services

Sadaļa Services ļauj izveidot jūsu uzraudzītās infrastruktūras hierarhiju, pievienojot vecākpakalpojumus un pēc tam — bērnpakalpojumus vecākpakalpojumiem.

Papildus pakalpojumu koka konfigurēšanai šī sadaļa sniedz visas infrastruktūras pārskatu un ļauj ātri identificēt problēmas, kas izraisījušas pakalpojuma stāvokļa izmaiņas.

- SLA

Šajā sadaļā varat definēt pakalpojumu līmeņa vienošanās un iestatīt pakalpojumu līmeņa mērķus konkrētiem pakalpojumiem.

- SLA report

Šajā sadaļā varat skatīt SLA atskaites.

Pakalpojumu darbības

Varat arī konfigurēt pakalpojumu darbības.

Pakalpojumu darbības nav obligātas un ļauj:

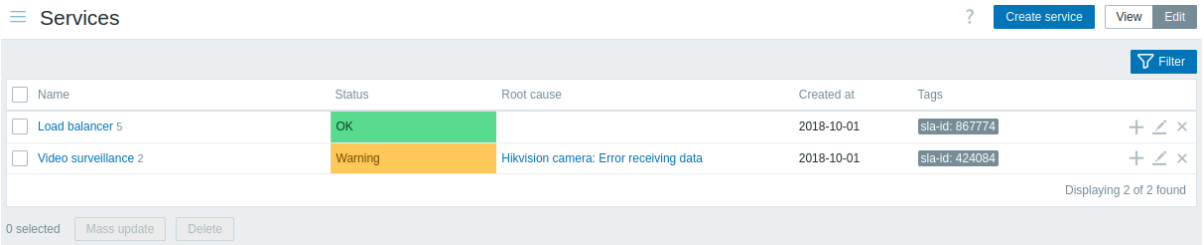
- nosūtīt paziņojumu, ka pakalpojums nav pieejams
- izpildīt attālo komandu Zabbix serveris, mainoties pakalpojuma stāvoklim
- nosūtīt atkopšanas paziņojumu, kad pakalpojums atkal ir pieejams.

Skatiet arī:

- SLA uzraudzības konfigurācijas piemēru
- Piezīmes par pakalpojumu jauninašanu no Zabbix versijām, kas ir zemākas par 6.0

1 Pakalpojumu koks

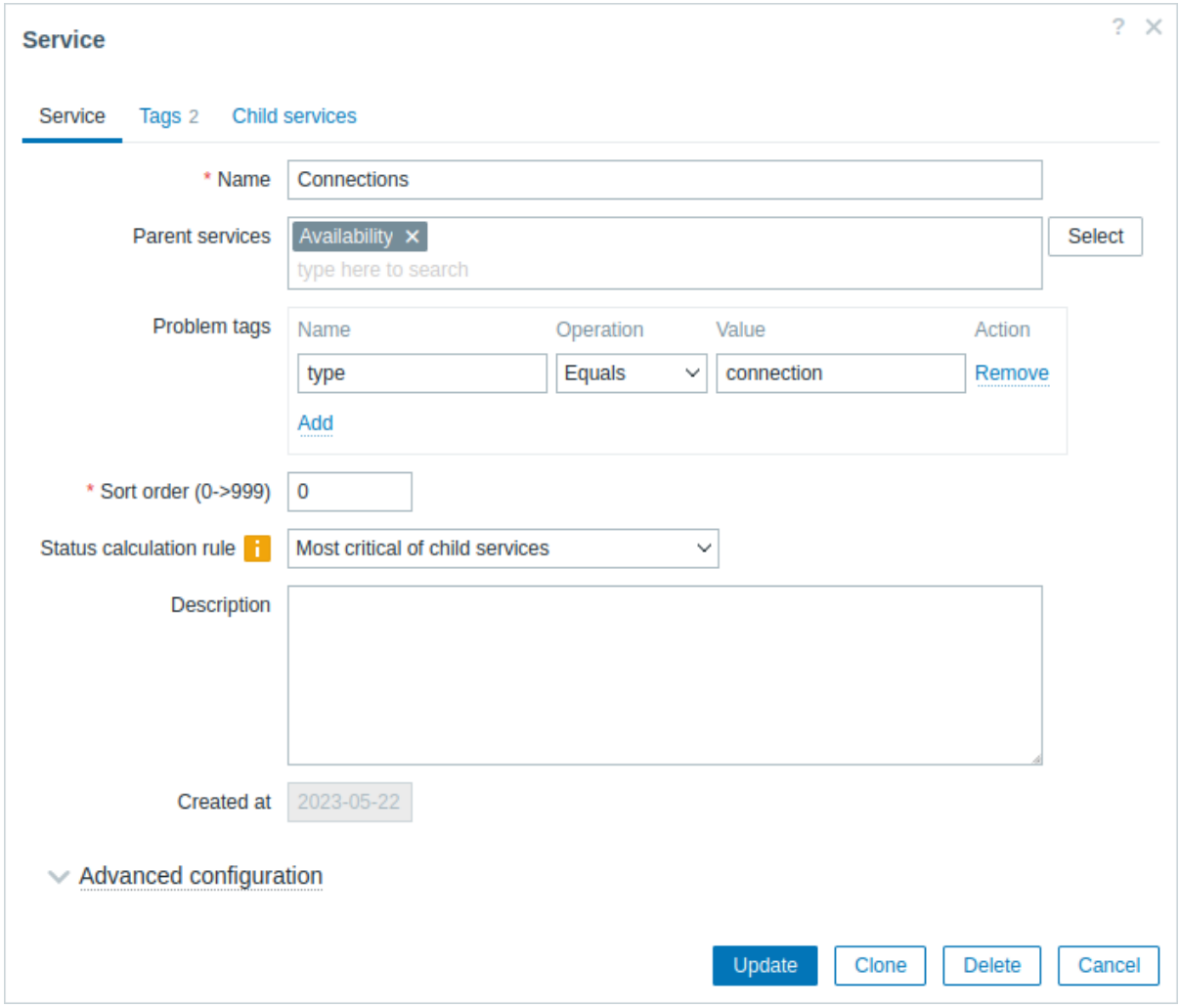
Pakalpojumu koks tiek konfigurēts izvēlnes sadaļā *Pakalpojumi -> Pakalpojumi*. Augšējā labajā stūrī pārslēdzieties no skatīšanas režīma uz rediģēšanas režīmu.



Lai konfigurētu jaunu pakalpojumu, noklikšķiniet uz pogas *Izveidot pakalpojumu* augšējā labajā stūrī.

Lai ātri pievienotu apakšpakalpojumu, varat arī nospiest pluszīmes ikonu blakus vecākpakalpojumam. Tādējādi tiks atvērta tā pati pakalpojuma konfigurācijas forma, taču parametrs *Vecākpakalpojumi* jau būs aizpildīts.

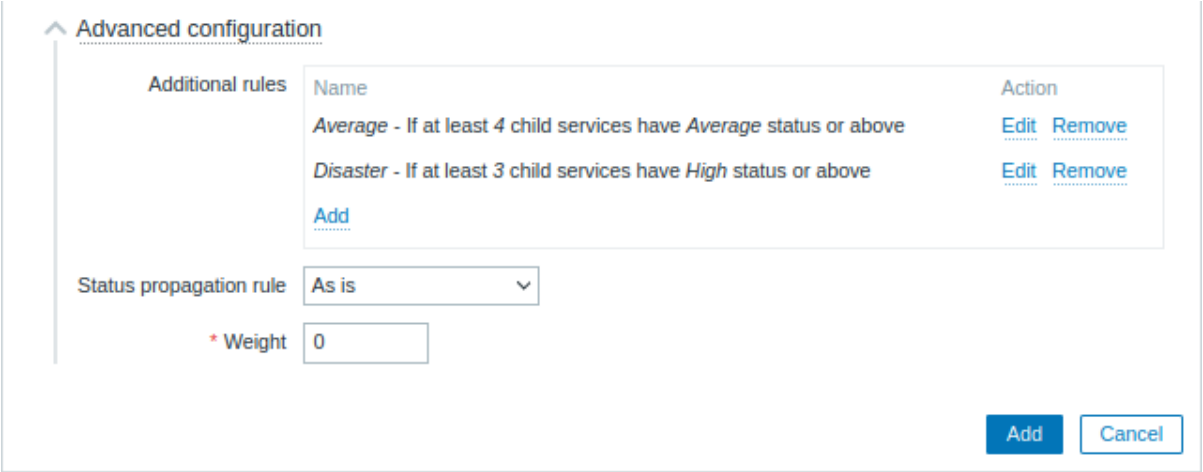
Pakalpojuma konfigurācija Cilnē **Pakalpojums** norādiet nepieciešamos pakalpojuma parametrus:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Nosaukums	Pakalpojuma nosaukums.
Vecākpakalpojumi	Vecākpakalpojumi, kuriem šis pakalpojums pieder. Atstājiet šo lauku tukšu, ja pievienojat augstākā līmeņa pakalpojumu. Vienam pakalpojumam var būt vairāki vecākpakalpojumi. Šādā gadījumā tas tiks attēlots pakalpojumu kokā zem katra no vecākpakalpojumiem.
Problēmu birkas	Norādiet birkas, lai piesaistītu problēmu datus pakalpojumam. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami divi operatori: **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutīga)
Kārtošanas secība	Attēlošanas kārtošanas secība; mazākā vērtība tiek rādīta pirmā.
Statusa aprēķināšanas noteikums	Noteikums pakalpojuma statusa aprēķināšanai: **Viskritiskākais, ja visiem bērnpakalpojumiem ir problēmas** — viskritiskākā problēma bērnpakalpojumos tiek izmantota pakalpojuma statusa iekrāsošanai, ja visiem bērnpakalpojumiem ir problēmas **Viskritiskākais no bērnpakalpojumiem** — viskritiskākā problēma bērnpakalpojumos tiek izmantota pakalpojuma statusa iekrāsošanai **Iestatīt statusu uz OK** — neaprēķināt pakalpojuma statusu Papildu statusa aprēķināšanas noteikumus var konfigurēt paplašinātās konfigurācijas opcijās.
Apraksts	Pakalpojuma apraksts.
Izveidots	Laiks, kad pakalpojums tika izveidots; tiek parādīts, rediģējot esošu pakalpojumu.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

Paplašinātā konfigurācija



Parametrs	Apraksts
Papildu noteikumi	Noklikšķiniet uz *Pievienot*, lai konfigurētu papildu statusa aprēķināšanas noteikumus.
Iestatīt statusu uz	Iestatiet pakalpojuma statusu uz *OK* (noklusējums), *Nav klasificēts*, *Informācija*, *Brīdinājums*, *Vidējs*, *Augsts* vai *Katastrofa*, ja nosacījums atbilst.
Nosacījums	Atlasiet nosacījumu tiešajiem bērnpakalpojumiem: **ja vismaz (N) bērnpakalpojumiem ir statuss (Status) vai augstāks** **ja vismaz (N%) bērnpakalpojumu ir statuss (Status) vai augstāks** **ja mazāk nekā (N) bērnpakalpojumiem ir statuss (Status) vai zemāks** **ja mazāk nekā (N%) bērnpakalpojumu ir statuss (Status) vai zemāks** **ja bērnpakalpojumu ar statusu (Status) vai augstāku svars ir vismaz (W)** **ja bērnpakalpojumu ar statusu (Status) vai augstāku svars ir vismaz (N%)** **ja bērnpakalpojumu ar statusu (Status) vai zemāku svars ir mazāks par (W)** **ja bērnpakalpojumu ar statusu (Status) vai zemāku svars ir mazāks par (N%)** Ja ir norādīti vairāki nosacījumi un situācija atbilst vairāk nekā vienam nosacījumam, tiks iestatīta augstākā būtiskuma pakāpe.
N (W)	Iestatiet N vai W vērtību (1-100000) vai N% (1-100) nosacījumā.
Status	Atlasiet *Status* vērtību nosacījumā: *OK* (noklusējums), *Nav klasificēts*, *Informācija*, *Brīdinājums*, *Vidējs*, *Augsts* vai *Katastrofa*.
Statusa izplatīšanas noteikums	Noteikums pakalpojuma statusa izplatīšanai uz vecākpakalpojumu: **Kā ir** - statuss tiek izplatīts bez izmaiņām **Palielināt par** - varat palielināt izplatīto statusu par 1 līdz 5 būtiskuma līmeņiem **Samazināt par** - varat samazināt izplatīto statusu par 1 līdz 5 būtiskuma līmeņiem **Ignorēt šo pakalpojumu** - statuss vispār netiek izplatīts uz vecākpakalpojumu **Fiksēts statuss** - statuss tiek izplatīts statiski, t.i., vienmēr kā viens un tas pats
Svars	Pakalpojuma svars (vesels skaitlis diapazonā no 0 (noklusējums) līdz 1000000).

Note:
Papildu statusa aprēķināšanas noteikumus var izmantot tikai, lai palielinātu būtiskuma līmeni virs līmeņa, kas aprēķināts saskaņā ar galveno parametru *Statusa aprēķināšanas noteikums*. Ja saskaņā ar papildu noteikumiem statusam jābūt *Brīdinājums*, bet saskaņā ar *Statusa aprēķināšanas noteikums* statuss ir *Katastrofa*, pakalpojumam būs statuss *Katastrofa*.

Cilne **Birkas** ir pakalpojuma līmeņa birkas. Pakalpojuma līmeņa birkas tiek izmantotas, lai identificētu pakalpojumu. Šāda veida birkas netiek izmantotas, lai piesaistītu problēmas pakalpojumam (tam izmantojiet *Problēmu birkas* no pirmās cilnes).

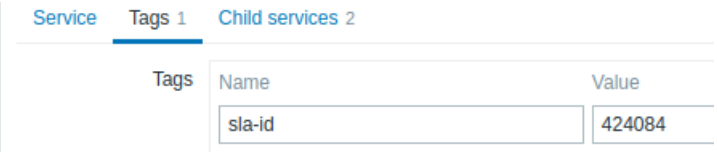
Cilne **Apakšpakalpojumi** ļauj norādīt atkarīgos pakalpojumus. Noklikšķiniet uz *Pievienot*, lai no esošo pakalpojumu saraksta pievienotu pakalpojumu. Ja vēlaties pievienot jaunu apakšpakalpojumu, vispirms saglabājiet šo pakalpojumu, pēc tam noklikšķiniet uz pluszīmes ikonas blakus tikko izveidotajam pakalpojumam.

Birkas Pakalpojumos ir divi dažādi birku veidi:

- Pakalpojumu birkas
- Problēmu birkas

Pakalpojumu birkas

Pakalpojumu birkas tiek izmantotas, lai sasaistītu pakalpojumus ar pakalpojumu darbībām un SLA. Šīs birkas tiek norādītas pakalpojuma konfigurācijas cilnē *Birkas*. SLA kartēšanai tiek izmantota *OR* loģika: pakalpojums tiks piesaistīts SLA, ja tam būs vismaz viena atbilstoša birka. Pakalpojumu darbībās kartēšanas noteikumi ir konfigurējami un var izmantot *AND*, *OR* vai *AND/OR* loģiku.

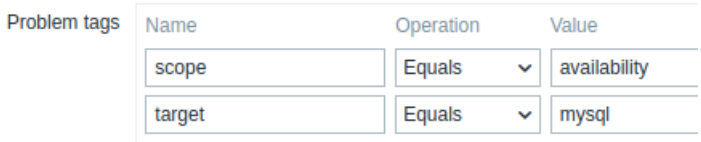


Problēmu birkas

Problēmu birkas tiek izmantotas, lai sasaistītu problēmas un pakalpojumus. Šīs birkas tiek norādītas primārā pakalpojuma konfigurācijas cilnē.

Tikai zemākā hierarhijas līmeņa apakšpakalpojumiem var būt definētas problēmu birkas un tie var tikt tieši korelēti ar problēmām. Ja problēmu birkas sakrīt, pakalpojuma statuss mainīsies uz tādu pašu statusu, kāds ir problēmai. Ja ir vairākas problēmas, pakalpojumam būs vissmagākās problēmas statuss. Vecākpakalpojuma statuss pēc tam tiek aprēķināts, pamatojoties uz apakšpakalpojumu statusiem, saskaņā ar statusa aprēķināšanas noteikumiem.

Ja ir norādītas vairākas birkas, tiek izmantota *AND* loģika: problēmai jābūt visām pakalpojuma konfigurācijā norādītajām birkām, lai tā tiktu piesaistīta pakalpojumam.



Note:
Problēma Zabbix pārmanto birkas no visas veidņu, hostu, vienumu, tīmekļa scenāriju un trigeru ķēdes. Jebkuru no šīm birkām var izmantot, lai sasaistītu problēmas ar pakalpojumiem.

Piemērs:

Problēmai *Web camera 3 is down* ir birkas `type:video-surveillance`, `floor:1` un `name:webcam-3`, un statuss *Warning*

Pakalpojumam **Web camera 3** ir norādīta tikai viena problēmas birka: `name:webcam-3`

Kad šī problēma tiks konstatēta, pakalpojuma statuss mainīsies no *OK* uz *Warning*.

Ja pakalpojumam **Web camera 3** būtu problēmas birkas `name:webcam-3` un `floor:2`, tā statuss nemainītos, kad problēma tiktu konstatēta, jo nosacījumi ir izpildīti tikai daļēji.

Esošo pakalpojumu modificēšana

Note:
Tālāk aprakstītās pogas ir redzamas tikai tad, kad sadaļa *Pakalpojumi* ir rediģēšanas režīmā.

Lai rediģētu esošu pakalpojumu, nospiediet zīmuļa ikonu blakus pakalpojumam.

Lai klonētu esošu pakalpojumu, nospiediet zīmuļa ikonu, lai atvērtu tā konfigurāciju, un pēc tam nospiediet pogu Clone. Kad pakalpojums tiek klonēts, tā vecāka saites tiek saglabātas, bet bērna saites netiek saglabātas.

Lai dzēstu pakalpojumu, nospiediet ikonu x blakus tam. Dzēšot vecāka pakalpojumu, tā bērna pakalpojumi netiks dzēsti un pārvietosies vienu līmeni augstāk pakalpojumu kokā (1. līmeņa bērni iegūs tādu pašu līmeni kā dzēstais vecāka pakalpojums).

Divas pogas zem pakalpojumu saraksta piedāvā dažas masveida rediģēšanas iespējas:

- *Mass update* - pakalpojumu īpašību masveida atjaunināšana
- *Delete* - dzēst pakalpojumus

Lai izmantotu šīs iespējas, atzīmējiet izvēles rūtiņas pie attiecīgajiem pakalpojumiem un pēc tam noklikšķiniet uz vajadzīgās pogas.

2 SLA

Pārskats Kad pakalpojumi ir izveidoti, varat sākt uzraudzīt, vai to veiktspēja atbilst pakalpojuma līmeņa vienošanās (SLA) prasībām.

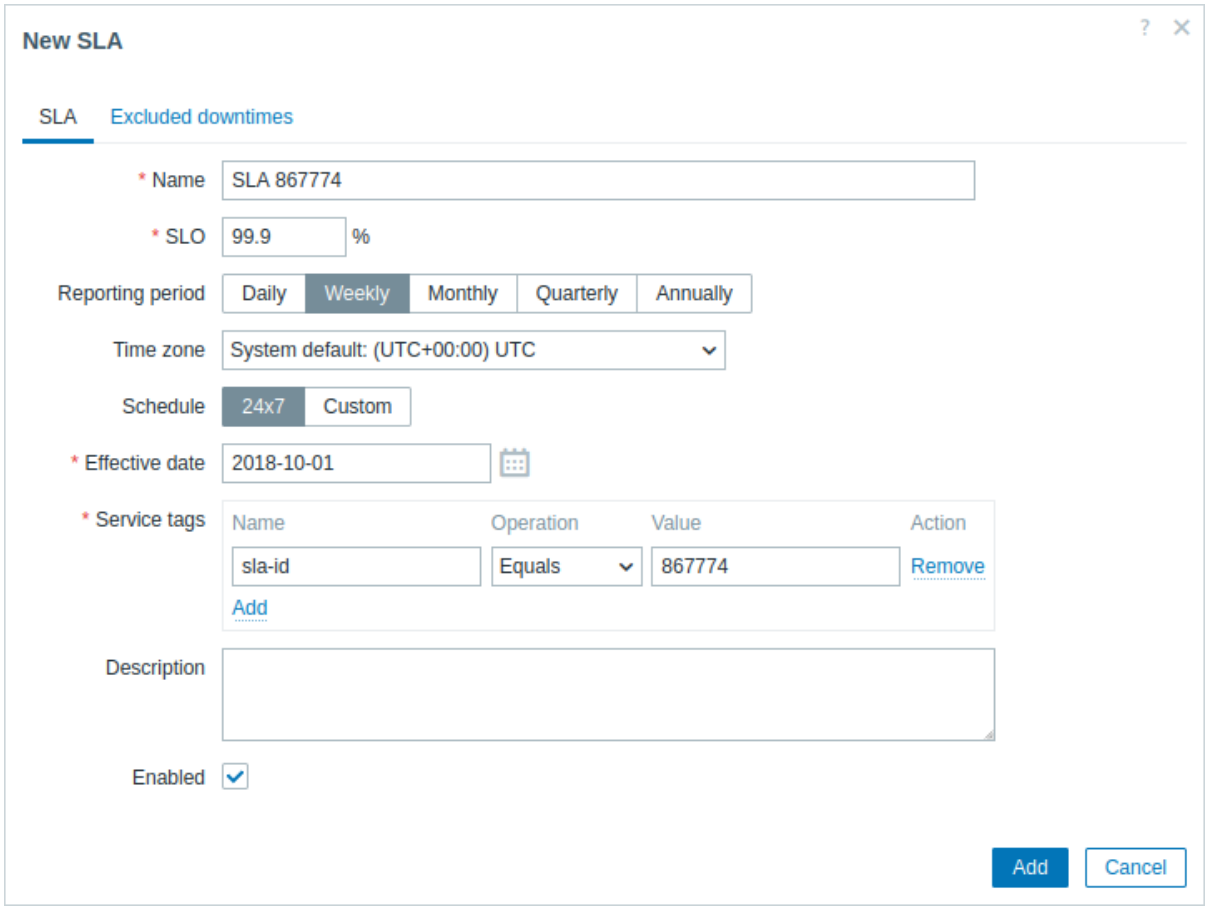
Izvēlnes sadaļa *Pakalpojumi->SLA* ļauj konfigurēt SLA dažādiem pakalpojumiem. SLA Zabbix sistēmā nosaka pakalpojuma līmeņa mērķi (SLO), paredzēto darbspējas grafiku un plānotās dīkstāves.

SLA un pakalpojumi tiek sasaistīti, izmantojot pakalpojumu birkas. To pašu SLA var piemērot vairākiem pakalpojumiem — veiktspēja tiks mērīta katram atbilstošajam pakalpojumam atsevišķi. Vienam pakalpojumam var būt piešķirti vairāki SLA — dati par katru SLA tiks attēloti atsevišķi.

SLA atskaitēs Zabbix nodrošina pakalpojuma līmeņa indikatora (SLI) datus, kas mēra faktisko pakalpojuma pieejamību. To, vai pakalpojums atbilst SLA mērķiem, nosaka, salīdzinot SLO (paredzētā pieejamība %) ar SLI (faktiskā pieejamība %).

Konfigurācija Lai izveidotu jaunu SLA, noklikšķiniet uz pogas *Create SLA*.

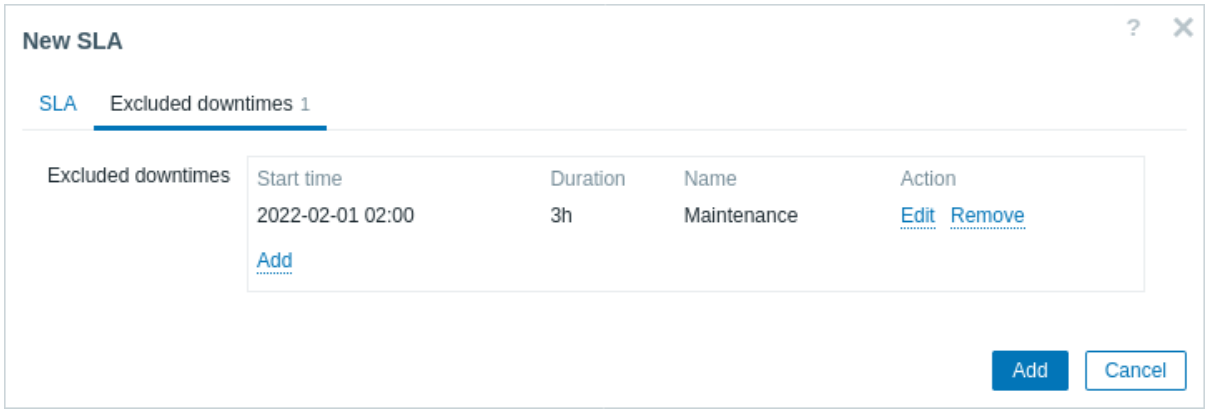
Cilne **SLA** ļauj norādīt vispārīgos SLA parametrus.



Parametrs	Apraksts
Name	Ievadiet SLA nosaukumu.
SLO	Ievadiet pakalpojuma līmeņa mērķi (SLO) procentos.
Reporting period	Atlasiet SLI pārskata periodu, kas tiek izmantots SLI datu aprēķināšanai SLA atskaitē: **Daily** — katru dienu no 00:00:00 līdz 23:59:59. **Weekly** — katru nedēļu no svētdienas 00:00:00 līdz sestdienai 23:59:59. **Monthly** — katru mēnesi no pirmās dienas 00:00:00 līdz pēdējai dienai 23:59:59. **Quarterly** — katru kalendāro ceturksni: Q1 (Jan–Mar), Q2 (Apr–Jun), Q3 (Jul–Sep), Q4 (Oct–Dec). **Annually** — katru kalendāro gadu no 1. janvāra 00:00:00 līdz 31. decembra 23:59:59.
Time zone	Atlasiet SLA laika joslu.
Schedule	Atlasiet SLA grafiku — 24x7 vai pielāgotu.
Effective date	Atlasiet SLA aprēķina sākuma datumu.
Service tags	Pievienojiet pakalpojumu tagus, lai identificētu pakalpojumus, kuriem šis SLA jāpiemēro. **Name** — pakalpojuma taga nosaukums; tam precīzi jāsakrīt, ievērojot reģistrjutību. **Operation** — atlasiet *Equals*, ja taga vērtībai precīzi jāsakrīt (ievērojot reģistrjutību), vai *Contains*, ja jāsakrīt daļai no taga vērtības (neievērojot reģistrjutību). **Value** — pakalpojuma taga vērtība, ko meklēt atbilstoši atlasītajai darbībai. SLA tiek piemērots pakalpojumam, ja sakrīt vismaz viens pakalpojuma tags.

Parametrs	Apraksts
Description	Pievienojiet SLA aprakstu.
Enabled	Atzīmējiet izvēles rūtiņu, lai iespējotu SLA aprēķinu.

Cilne **Excluded downtimes** ļauj norādīt dīkstāves periodus, kas ir izslēgti no SLA aprēķina.



Noklikšķiniet uz *Add*, lai konfigurētu izslēgtos dīkstāves periodus, pēc tam ievadiet perioda nosaukumu, sākuma datumu un ilgumu.

SLA atskaites Kā pakalpojums darbojas salīdzinājumā ar SLA, ir redzams SLA atskaitē. SLA atskaites var skatīt:

- sadaļā *SLA*, noklikšķinot uz SLA atskaites hipersaites;
- sadaļā *Pakalpojumi*, noklikšķinot uz SLA nosaukuma informācijas cilnē;
- paneļa logrīkā *SLA atskaite*.

Kad SLA ir konfigurēts, pakalpojumu sadaļas cilnē *Informācija* tiks parādīta arī noteikta informācija par pakalpojuma veiktspēju.

3 Iestatīšanas piemērs

Pārskats Šajā sadaļā ir aprakstīta vienkārša iestatīšana Zabbix augstas pieejamības klastera uzraudzībai kā pakalpojumam.

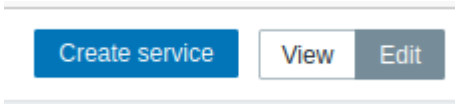
Priekšnosacījumi Pirms pakalpojumu uzraudzības konfigurēšanas jums ir jābūt konfigurētiem hostiem:

- *HA mezgls 1* ar vismaz vienu trigeri un birku (vēlams iestatītu trigera līmenī) `component:ha-node-1`
- *HA mezgls 2* ar vismaz vienu trigeri un birku (vēlams iestatītu trigera līmenī) `component:ha-node-2`

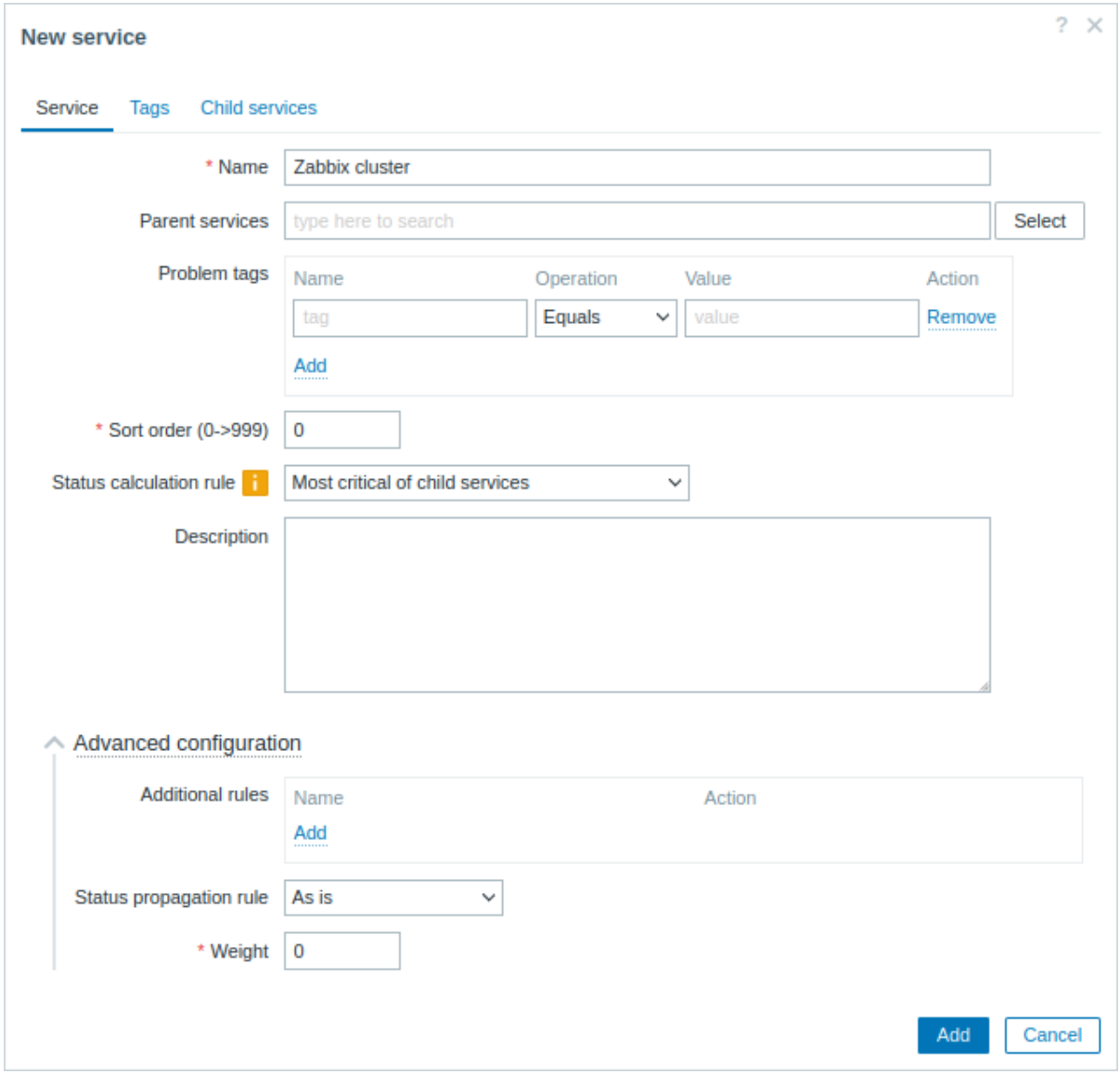
Pakalpojumu koks Nākamais solis ir izveidot pakalpojumu koku. Šajā piemērā infrastruktūra ir ļoti vienkārša un sastāv no trim pakalpojumiem: *Zabbix cluster* (vecāks) un diviem bērnpakalpojumiem *Zabbix server node 1* un *Zabbix server node 2*.

```
Zabbix cluster
|
|- Zabbix server node 1
|- Zabbix server node 2
```

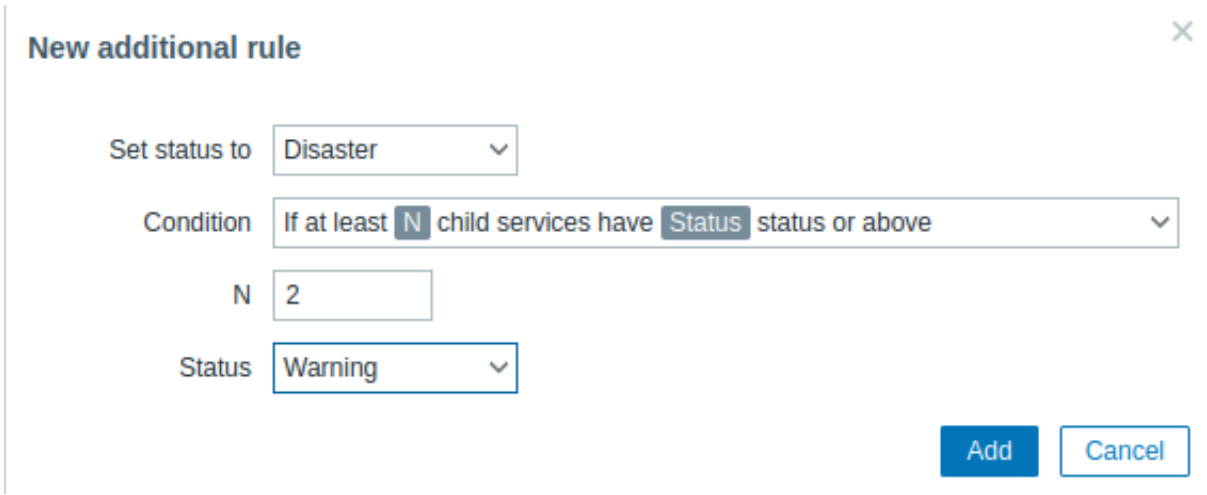
Lapas Services sadaļā ieslēdziet *Edit* režīmu un nospiediet Create service:



Pakalpojuma konfigurācijas logā ievadiet nosaukumu *Zabbix cluster* un noklikšķiniet uz *Advanced configuration* etiķetes, lai parādītu paplašinātās konfigurācijas opcijas.



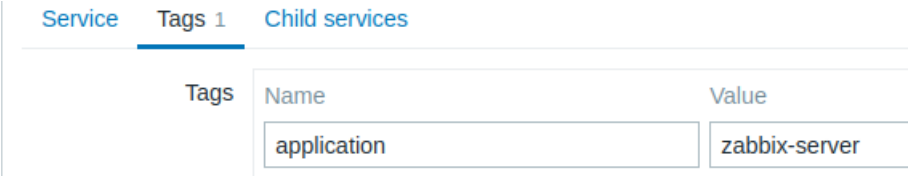
Konfigurējiet papildu nosacījumu:



Zabbix cluster būs divi bērnpakalpojumi — pa vienam katram HA mezglam. Ja abiem HA mezgliem ir problēmas ar vismaz *Warning* statusu, vecākā pakalpojuma statuss jāiestata uz *Disaster*. Lai to panāktu, papildu nosacījums jākonfigurē šādi:

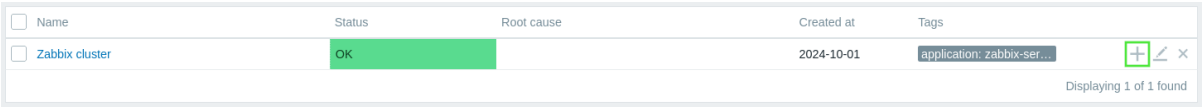
- Set status to: Disaster
- Condition: If at least `N` child services have `Status` status or above
- N: 2
- Status: Warning

Pārslēdzieties uz cilni *Tags* un pievienojiet birku `application:zabbix-server`. Šī birka vēlāk tiks izmantota pakalpojumu darbībām un SLA atskaitēm.



Saglabājiet jauno pakalpojumu.

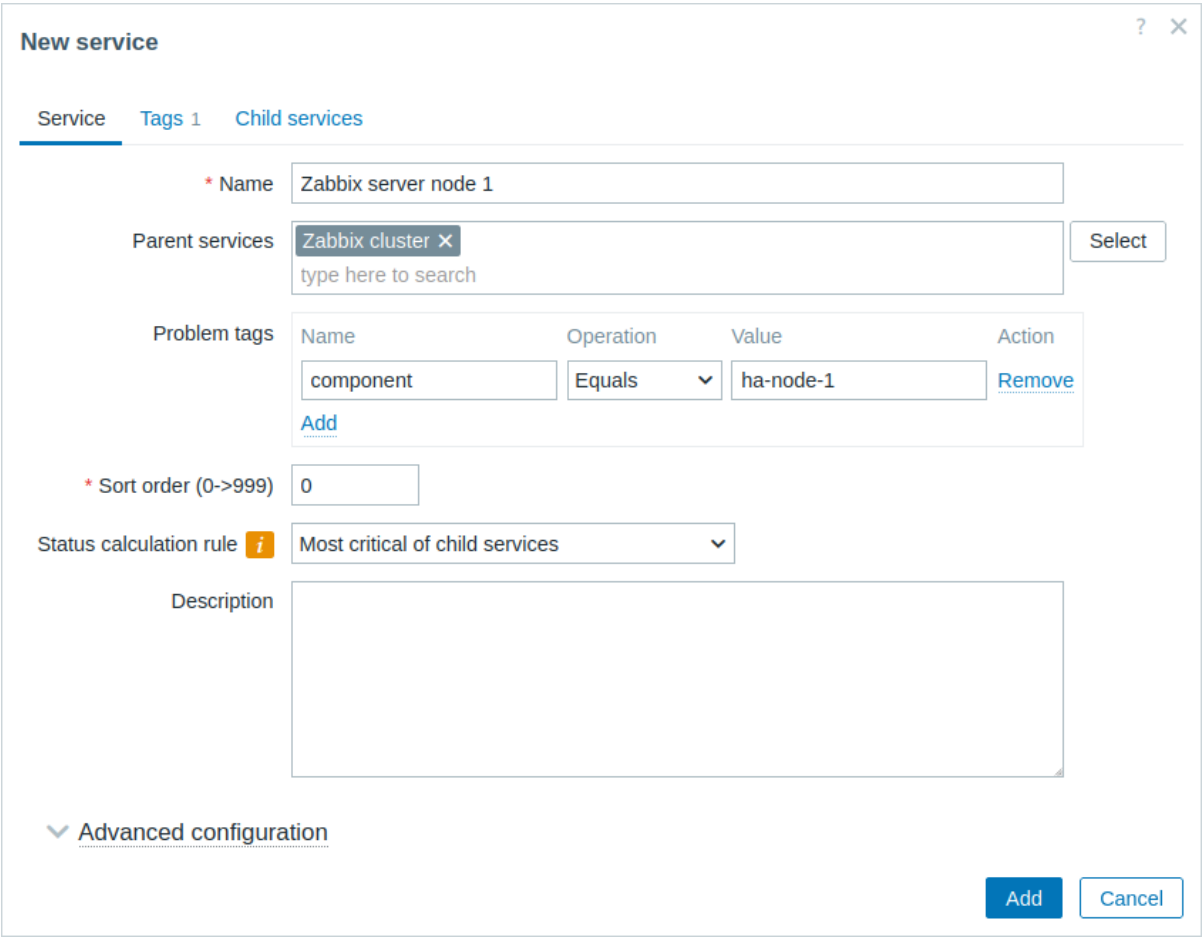
Lai pievienotu bērnservisu, nospiediet plusa ikonu blakus servisam Zabbix cluster (ikona ir redzama tikai *rediģēšanas* režīmā).



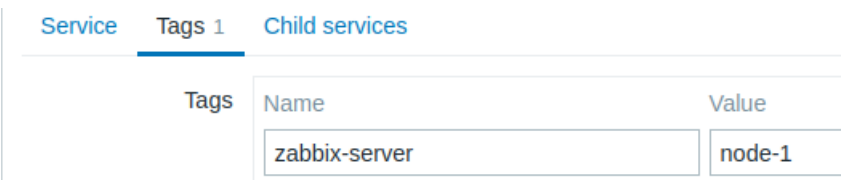
Servisa konfigurācijas logā ievadiet nosaukumu *Zabbix server node 1*. Ņemiet vērā, ka parametrs Parent services jau ir iepriekš aizpildīts ar *Zabbix cluster*.

Šī servisa pieejamību ietekmē problēmas hostā *HA node 1*, kas apzīmētas ar problēmas birku `component:ha-node-1`. Parametrā Problem tags ievadiet:

- Nosaukums: component
- Darbība: Equals
- Vērtība: ha-node-1



Pārslēdzieties uz cilni *Tags* un pievienojiet servisa birku: `zabbix-server:node-1`. Šī birka vēlāk tiks izmantota servisu darbībām un SLA atskaitēm.



Saglabājiet jauno servisu.

Izveidojiet vēl vienu Zabbix klastera bērnservisu ar nosaukumu ”Zabbix server node 2”.

Iestatiet problēmas birkas šādi:

- Nosaukums: component
- Darbība: Vienāds
- Vērtība: ha-node-2

Pārslēdzieties uz cilni *Birkas* un pievienojiet servisa birku: `zabbix-server:node-2`.

Saglabājiet jauno servisu.

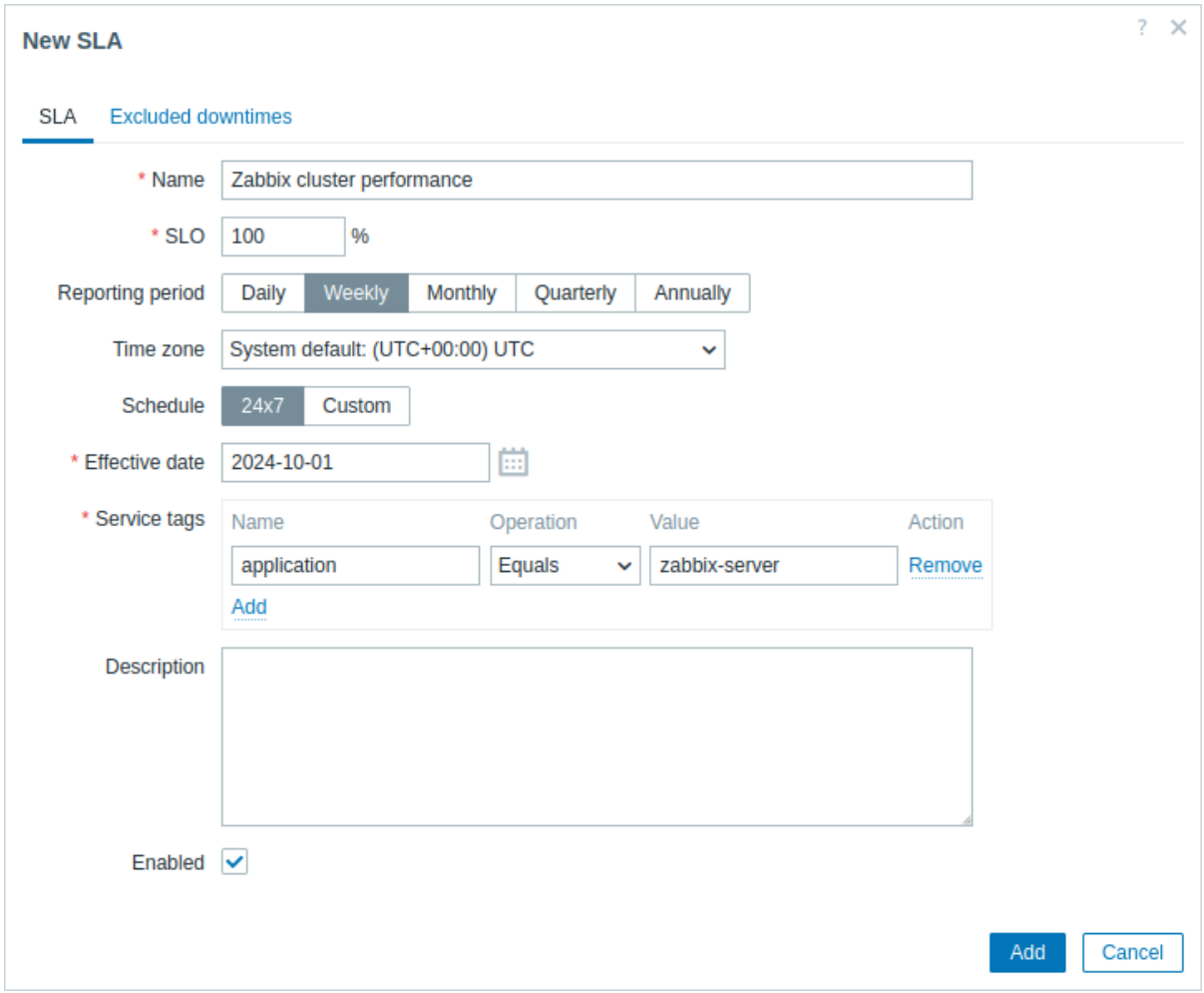
SLA Šajā piemērā paredzamā Zabbix klastera veiktspēja ir 100%, izņemot pusgada vienas stundas apkopes periodu.

Vispirms jums jāpievieno jauns pakalpojuma līmeņa līgums.

Dodieties uz izvēlnes sadaļu *Pakalpojumi->SLA* un nospiediet Create SLA. Ievadiet nosaukumu *Zabbix cluster performance* un iestatiet SLO uz 100%.

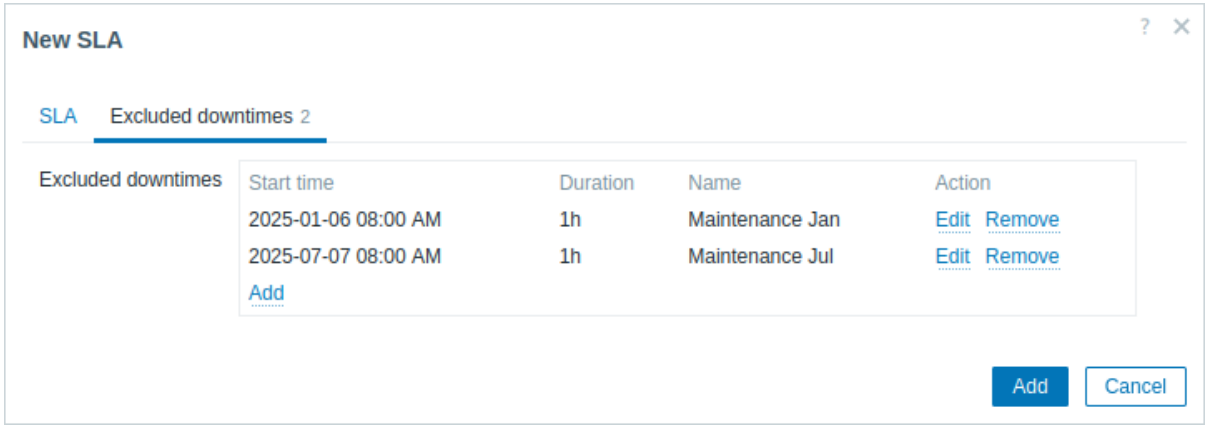
Pakalpojumam Zabbix cluster ir pakalpojuma tags `application:zabbix-server`. Lai izmantotu šo SLA Zabbix cluster veiktspējas mērīšanai, parametrā *Service tags* norādiet:

- Nosaukums: application
- Darbība: Equals
- Vērtība: zabbix-server



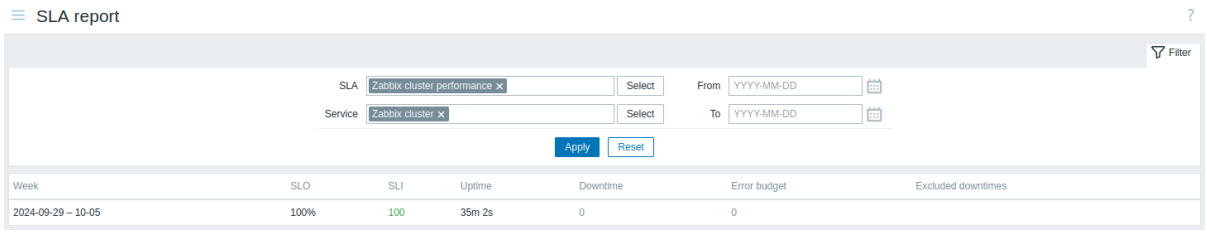
Reālā vidē varat arī atjaunināt vēlamo pārskatu periodu, laika joslu un sākuma datumu vai mainīt grafiku no 24/7 uz pielāgotu. Šajā piemērā pietiek ar noklusējuma iestatījumiem.

Pārslēdzieties uz cilni *Excluded downtimes* un pievienojiet dīkstāves plānotajiem apkopes periodiem, lai izslēgtu šos periodus no SLA aprēķina. Sadaļā Excluded downtimes nospiediet saiti Add, ievadiet dīkstāves nosaukumu, plānoto sākuma laiku un ilgumu.

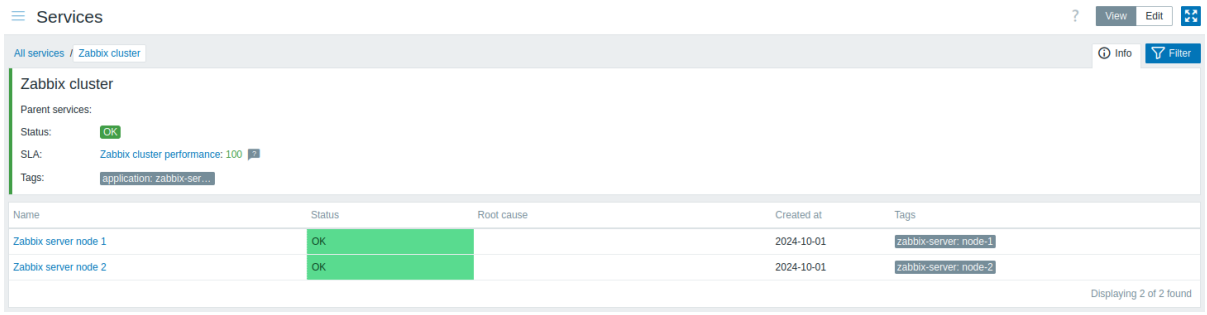


Nospiediet Add, lai saglabātu jauno SLA.

Pārslēdzieties uz SLA pārskatu sadaļu, lai skatītu Zabbix cluster SLA pārskatu.



SLA informāciju var pārbaudīt arī sadaļā *Pakalpojumi*.



7 Tīmekļa uzraudzība

Pārskats Ar Zabbix jūs varat pārbaudīt vairākus tīmekļa vietņu pieejamības aspektus.

Attention:
Lai veiktu tīmekļa uzraudzību, Zabbix serverim sākotnēji jābūt konfigurētam ar cURL (libcurl) atbalstu.

Lai aktivizētu tīmekļa uzraudzību, jums jādefinē tīmekļa scenāriji. Tīmekļa scenārijs sastāv no viena vai vairākiem HTTP pieprasījumiem jeb ”soļiem”. Soļi periodiski tiek izpildīti Zabbix serverī iepriekš noteiktā secībā. Ja hosts tiek uzraudzīts ar starpniekserveri, soļus izpilda starpniekserveris.

Tīmekļa scenāriji tiek piesaistīti hostiem/veidnēm tāpat kā vienumi, trigeri utt. Tas nozīmē, ka tīmekļa scenārijus var izveidot arī veidnes līmenī un pēc tam vienā darbībā piemērot vairākiem hostiem.

Jebkurā tīmekļa scenārijā tiek apkopota šāda informācija:

- vidējais lejupielādes ātrums sekundē visiem visa scenārija soļiem
- neveiksmīgā soļa numurs
- pēdējais kļūdas ziņojums

Jebkurā tīmekļa scenārija solī tiek apkopota šāda informācija:

- lejupielādes ātrums sekundē
- atbildes laiks
- atbildes kods

Plašāku informāciju skatiet sadaļā tīmekļa uzraudzības vienumi.

Dati, kas iegūti, izpildot tīmekļa scenārijus, tiek glabāti datubāzē. Šie dati tiek automātiski izmantoti grafikiem, trigeriem un paziņojumiem.

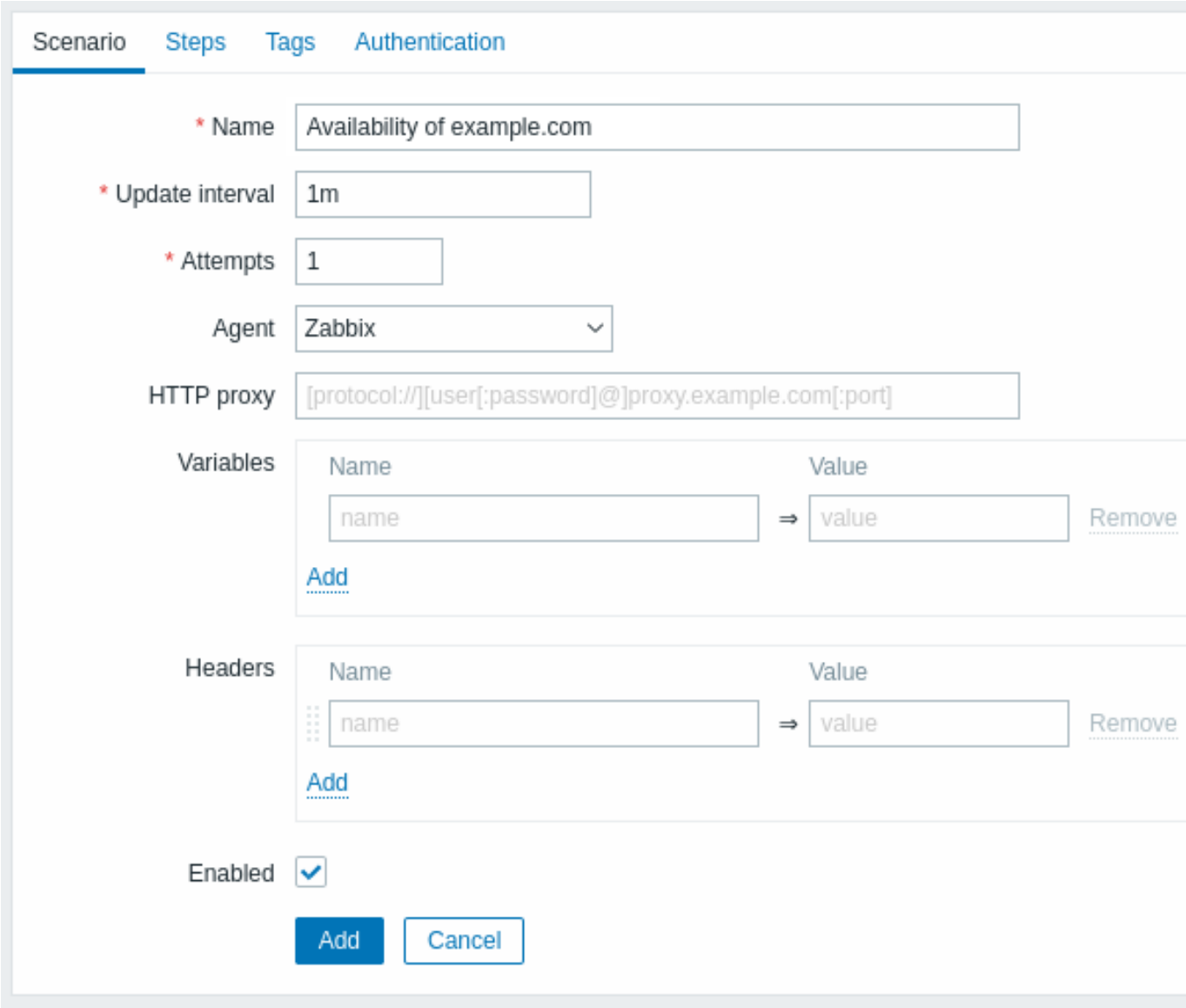
Zabbix var arī pārbaudīt, vai iegūtā HTML lapa satur iepriekš definētu virkni. Tas var izpildīt simulētu pieteikšanos un sekot simulētu peles klikšķu ceļam lapā.

Zabbix tīmekļa uzraudzība atbalsta gan HTTP, gan HTTPS. Izpildot tīmekļa scenāriju, Zabbix pēc izvēles sekos pāradresācijām (skatiet opciju *Sekot pāradresācijām* zemāk). Maksimālais pāradresāciju skaits ir stingri noteikts uz 10 (izmantojot cURL opciju CURLOPT_MAXREDIRS). Visas sīkdatnes tiek saglabātas viena scenārija izpildes laikā.

Tīmekļa scenārija konfigurēšana Lai konfigurētu tīmekļa scenāriju:

- Dodieties uz: *Datu ievākšana* → *Hosti* (vai *Veidnes*)
- Noklikšķiniet uz *Tīmeklis* attiecīgā hosta/veidnes rindā
- Noklikšķiniet uz *Izveidot tīmekļa scenāriju* labajā pusē (vai uz scenārija nosaukuma, lai rediģētu esošu scenāriju)
- Ievadiet scenārija parametrus formā

Cilne **Scenārijs** ļauj konfigurēt tīmekļa scenārija vispārīgos parametrus.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Scenārija parametri:

Parametrs	Apraksts
Nosaukums	Unikāls scenārija nosaukums. Tiek atbalstīti lietotāja makrosi. *Ņemiet vērā*, ka, ja tiek izmantoti lietotāja makrosi, šie makrosi tīmekļa uzraudzības vienumu nosaukumos paliks neatrisināti.
Atjaunināšanas intervāls	Cik bieži scenārijs tiks izpildīts. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Ņemiet vērā*, ka, ja tiek izmantots lietotāja makross un tā vērtība tiek mainīta (piemēram, 5m → 30s), nākamā pārbaude tiks izpildīta saskaņā ar iepriekšējo vērtību (piemērā norādīto vērtību gadījumā — vēlāk nākotnē). Jauni tīmekļa scenāriji tiks pārbaudīti 60 sekunžu laikā pēc to izveides.
Mēģinājumi	Mēģinājumu skaits tīmekļa scenārija soļu izpildei. Tīkla problēmu gadījumā (taimauts, nav savienojamības u.c.) Zabbix var atkārtot soļa izpildi vairākas reizes. Iestatītais skaits vienādi ietekmēs katru scenārija soli. Var norādīt līdz 10 mēģinājumiem, noklusējuma vērtība ir 1. *Piezīme*: Zabbix neatkārtos soli nepareiza atbildes koda vai prasītās virknes neatbilstības dēļ.
Aģents	Atlasiet klienta aģentu. Zabbix uzdosies par atlasīto pārlūku. Tas ir noderīgi, ja vietne dažādiem pārlūkiem atgriež atšķirīgu saturu. Šajā laukā var izmantot lietotāja makrosus.
HTTP starpniekserveris	Varat norādīt izmantojamo HTTP starpniekserveri, izmantojot formātu `[protocol://][username[:password]@]proxy.example.com[:port]`. Tas iestata cURL opciju CURLOPT_PROXY. Neobligāto prefiksu `protocol://` var izmantot, lai norādītu alternatīvus starpniekservera protokolus (protokola prefiksa atbalsts tika pievienots cURL 7.21.7). Ja protokols nav norādīts, starpniekserveris tiks uzskatīts par HTTP starpniekserveri. Pēc noklusējuma tiks izmantots 1080. ports. Ja norādīts, starpniekserveris pārrakstīs ar starpniekserveri saistītos vides mainīgos, piemēram, http_proxy, HTTPS_PROXY. Ja nav norādīts, starpniekserveris nepārrakstīs ar starpniekserveri saistītos vides mainīgos. Ievadītā vērtība tiek nodota "kā ir", bez jebkādas korektuma pārbaudes. Varat ievadīt arī SOCKS starpniekservera adresi. Ja norādīsiet nepareizu protokolu, savienojums neizdosies un vienums kļūs neatbalstīts. *Ņemiet vērā*, ka ar HTTP starpniekserveri tiek atbalstīta tikai vienkāršā autentifikācija. Šajā laukā var izmantot lietotāja makrosus.
Mainīgie	Mainīgie, kurus var izmantot scenārija soļos (URL, post mainīgie). Tiem ir šāds formāts: **{macro1}**=value1 **{macro2}**=value2 **{macro3}**=regex:<regular expression> **{macro4}**=jsonpath:<jsonpath> **{macro5}**=xmlxpath:<xmlxpath> **{macro6}**={{macro}.function()} (skatiet makrosu funkcijas) Piemēram: {username}=Alexei {password}=kj3h5kJ34bd {hostid}=regex:hostid is ([0-9]+) {url}=jsonpath:$.host_url {status}=xmlxpath://host/response/status {newvar}={{myvar}.btoa()} Pēc tam uz makrosiem soļos var atsaukties kā uz {username}, {password}, {hostid} utt. Zabbix automātiski aizstās tos ar faktiskajām vērtībām. Ņemiet vērā, ka mainīgajiem ar `regex:` ir nepieciešams viens solis, lai iegūtu regulārās izteiksmes vērtību, tāpēc iegūto vērtību var piemērot tikai nākamajam solim. Ja vērtības daļa sākas ar `regex:`, tad daļa pēc tās tiek uzskatīta par regulāro izteiksmi, kas meklē tīmekļa lapā un, ja tiek atrasta, saglabā atbilstību mainīgajā. Jābūt vismaz vienai apakšgrupai, lai varētu iegūt atbilstošo vērtību. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi. Mainīgie tiek automātiski URL kodēti, ja tos izmanto vaicājuma laukos vai formas datos post mainīgajiem, taču tie ir manuāli URL jākodē, ja tos izmanto neapstrādātā post saturā vai tieši URL.

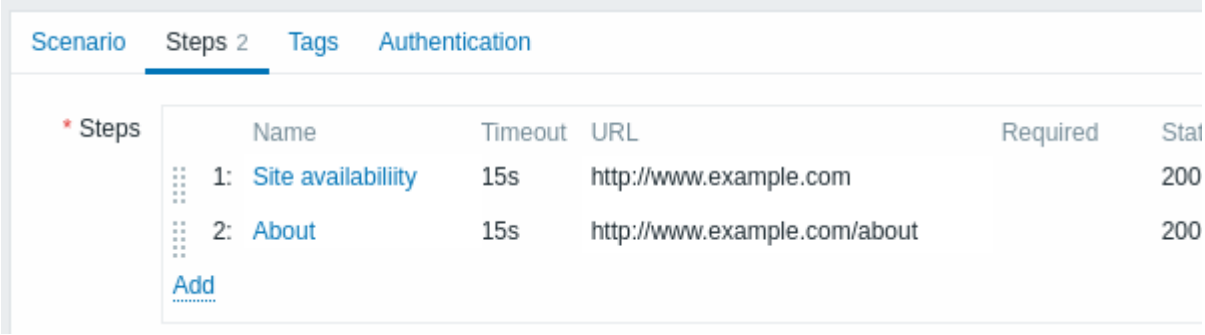
Parametrs	Apraksts
Galvenes	Veicot pieprasījumu, tiek izmantotas HTTP galvenes. Var izmantot noklusējuma un pielāgotās galvenes. Galvenes tiks piešķirtas, izmantojot noklusējuma iestatījumus atkarībā no aģenta tipa, kas atlasīts scenārija līmenī nolaižamajā sarakstā, un tās tiks piemērotas visiem soļiem, ja vien tās nav pielāgoti definētas soļa līmenī. **Jāņem vērā, ka galvenes definēšana soļa līmenī automātiski atmet visas iepriekš definētās galvenes, izņemot noklusējuma galveni, kas tiek piešķirta, scenārija līmenī nolaižamajā sarakstā atlasot 'User-Agent'.** Tomēr pat noklusējuma galveni 'User-Agent' var pārrakstīt, to norādot soļa līmenī. Lai noņemtu galveni scenārija līmenī, galvene soļa līmenī ir jānorāda ar nosaukumu, bet bez vērtības. Galvenes jāuzskaita, izmantojot to pašu sintaksi, kādā tās parādītos HTTP protokolā, pēc izvēles izmantojot dažas papildu iespējas, ko atbalsta cURL opcija CURLOPT_HTTPHEADER. Piemēram: Accept-Charset=utf-8 Accept-Language=en-US Content-Type=application/xml; charset=utf-8 Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi.
Iespējots	Scenārijs ir aktīvs, ja šī izvēles rūtiņa ir atzīmēta, pretējā gadījumā — atspējots.

Ņemiet vērā, ka, rediģējot esošu scenāriju, formā ir pieejamas divas papildu pogas:

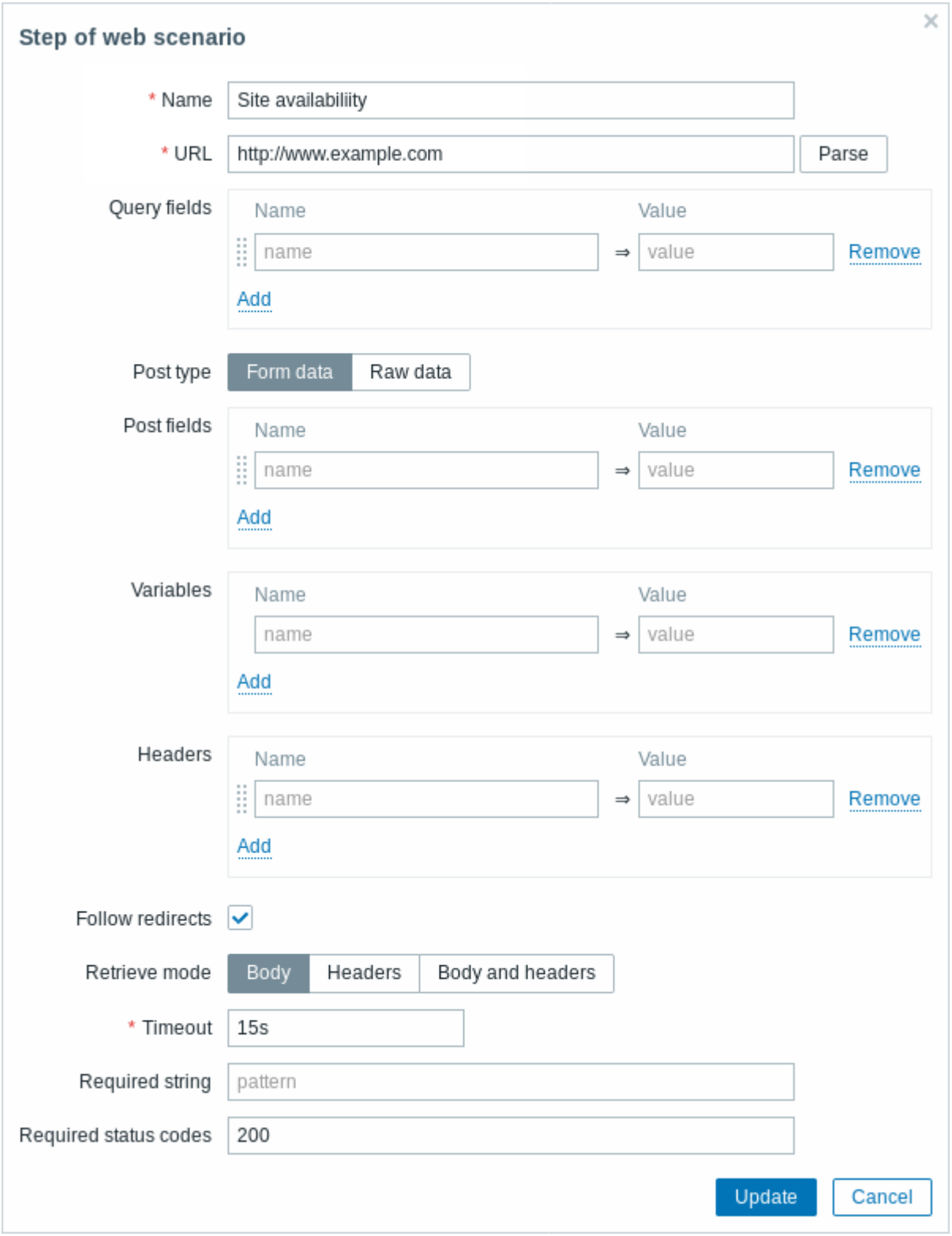
Clone	Izveidot citu scenāriju, pamatojoties uz esošā scenārija īpašībām.
Clear history and trends	Dzēst scenārija vēstures un tendenču datus. Tas liks serveris izpildīt scenāriju uzreiz pēc datu dzēšanas.

Note:
Ja lauks *HTTP starpniekserveris* ir atstāts tukšs, vēl viens veids, kā izmantot HTTP starpniekserveri, ir iestatīt ar starpniekserveri saistītos vides mainīgos.
HTTP pārbaudēm — iestatiet vides mainīgo **http_proxy** Zabbix servera lietotājam. Piemēram, `http_proxy=http://proxy_ip:proxy_port`.
HTTPS pārbaudēm — iestatiet vides mainīgo **HTTPS_PROXY**. Piemēram, `HTTPS_PROXY=http://proxy_ip:proxy_port`.
Papildu informācija ir pieejama, izpildot čaulas komandu: *# man curl*.

Cilne **Soļi** ļauj konfigurēt tīmekļa scenārija soļus. Lai pievienotu tīmekļa scenārija soli, noklikšķiniet uz *Pievienot* blokā *Soļi*.



Note:
Slepenos lietotāja makrosus nedrīkst izmantot URL, jo tie tiks atrisināti kā "******".



Soļu konfigurēšana

Soļa parametri:

Parametrs	Apraksts
Nosaukums	Unikāls soļa nosaukums. Tiek atbalstīti lietotāja makrosi. *Ņemiet vērā*, ka, ja tiek izmantoti lietotāja makrosi, šie makrosi tīmekļa uzraudzības vienuma nosaukumos paliks neatrisināti.

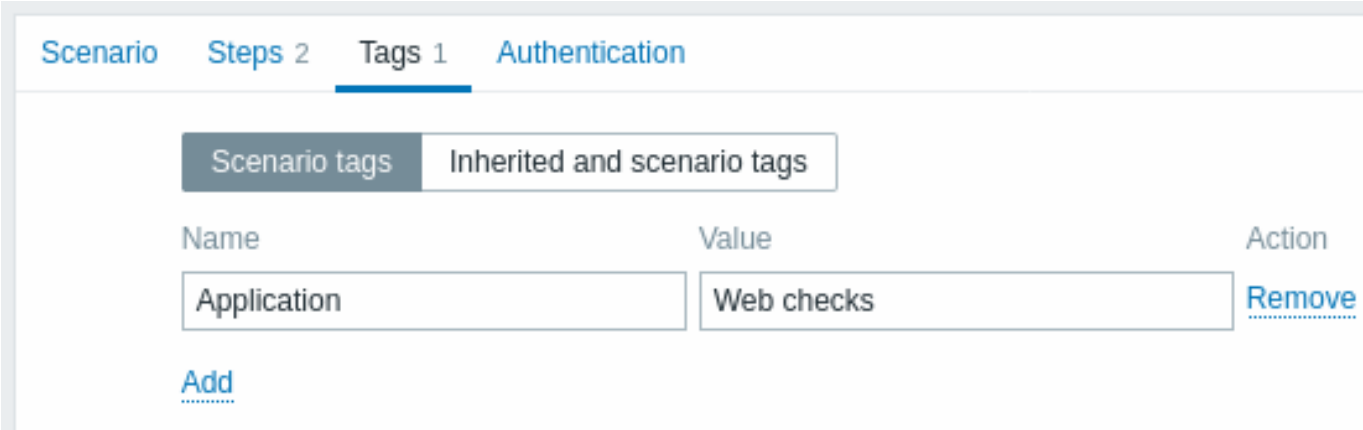
Parametrs	Apraksts
URL	URL, ar kuru izveidot savienojumu un no kura iegūt datus. Piemēram: https://www.example.com http://www.example.com/download Domēna nosaukumus var norādīt Unicode rakstzīmēs. Izpildot tīmekļa scenārija soli, tie automātiski tiek pārveidoti no punycode uz ASCII. Pogu *Parse* var izmantot, lai no URL atdalītu neobligātos vaicājuma laukus (piemēram, ?name=Admin&password=mypassword), pārvietojot atribūtus un vērtības uz *Query fields* automātiskai URL kodēšanai. URL var izmantot mainīgos, izmantojot sintaksi {macro}. Mainīgos var manuāli URL kodēt, izmantojot sintaksi {{macro}.urlencode()}. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi. Ierobežots līdz 2048 rakstzīmēm.
Query fields	URL HTTP GET mainīgie. Norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski URL kodētas. Vērtības no scenārija mainīgajiem, lietotāja makrosiem vai {HOST.*} makrosiem tiek atrisinātas un pēc tam automātiski URL kodētas. Izmantojot sintaksi {{macro}.urlencode()}, tās tiks URL kodētas divreiz. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi.
Post	HTTP POST mainīgie. Režīmā **Form data** tie tiek norādīti kā atribūtu un vērtību pāri. Vērtības tiek automātiski URL kodētas. Vērtības no scenārija mainīgajiem, lietotāja makrosiem vai {HOST.*} makrosiem tiek atrisinātas un pēc tam automātiski URL kodētas. Režīmā **Raw data** atribūti/vērtības tiek parādīti vienā rindā un savienoti ar simbolu **&**. Neapstrādātas vērtības var manuāli URL kodēt/dekodēt, izmantojot sintaksi {{macro}.urlencode()} vai {{macro}.urldecode()}. Piemēram: id=2345&userid={user} Ja {user} ir definēts kā tīmekļa scenārija mainīgais, izpildot soli tas tiks aizstāts ar savu vērtību. Ja vēlaties mainīgo URL kodēt, aizstājiet {user} ar {{user}.urlencode()}. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi.
Variables	Soļa līmeņa mainīgie, kurus var izmantot GET un POST funkcijām. Norādīti kā atribūtu un vērtību pāri. Soļa līmeņa mainīgie pārraksta scenārija līmeņa mainīgos vai iepriekšējā soļa mainīgos. Tomēr soļa līmeņa mainīgā vērtība ietekmē tikai nākamo soli (nevis pašreizējo soli). Tiem ir šāds formāts: **{macro}**=value **{macro}**=regex:<regular expression> Papildinformāciju skatiet mainīgo aprakstā scenārija līmenī. Mainīgie tiek automātiski URL kodēti, ja tos izmanto vaicājuma laukos vai formas datos POST mainīgajiem, bet tie ir manuāli URL jākodē, ja tos izmanto neapstrādātā POST vai tieši URL.
Headers	Pielāgotas HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu. Norādītas kā atribūtu un vērtību pāri. Soļa līmenī definēta galvene tiks izmantota tikai attiecīgajam solim. **Jāņem vērā, ka galvenes definēšana soļa līmenī automātiski atmet visas iepriekš definētās galvenes, izņemot noklusējuma galveni, kas tiek piešķirta, scenārija līmenī nolaižamajā sarakstā izvēloties 'User-Agent'.** Tomēr pat 'User-Agent' noklusējuma galveni var pārrakstīt, norādot to soļa līmenī. Piemēram, piešķirot galvenei nosaukumu, bet nenorādot vērtību, scenārija līmeņa noklusējuma galvene tiks noņemta. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi. Tas iestata cURL opciju CURLOPT_HTTPHEADER.
Follow redirects	Atzīmējiet izvēles rūtiņu, lai sekotu HTTP pāradresācijām. Tas iestata cURL opciju CURLOPT_FOLLOWLOCATION.
Retrieve mode	Atlasiet iegūšanas režīmu: **Body** - iegūt tikai HTTP atbildes pamattekstu **Headers** - iegūt tikai HTTP atbildes galvenes **Body and headers** - iegūt HTTP atbildes pamattekstu un galvenes
Timeout	Zabbix URL apstrādei netērēs vairāk par iestatīto laiku (no vienas sekundes līdz maksimāli 1 stundai). Faktiski šis parametrs nosaka maksimālo laiku savienojuma izveidei ar URL un maksimālo laiku HTTP pieprasījuma izpildei. Tādēļ Zabbix šim solim netērēs vairāk par **2 x Timeout** sekundēm. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 1h. Tiek atbalstīti lietotāja makrosi.

Parametrs	Apraksts
Required string	Obligāts regulārās izteiksmes raksts. Ja vien iegūtais saturs (HTML) neatbilst obligātajam rakstam, solis neizdosies. Ja lauks ir tukšs, obligātās virknes pārbaude netiek veikta. Piemēram: Homepage of Zabbix Welcome.*admin *Piezīme*: Šajā laukā netiek atbalstīta atsauce uz regulārajām izteiksmēm, kas izveidotas Zabbix lietotāja saskarnē. Tiek atbalstīti lietotāja makrosi un {HOST.*} makrosi.
Required status codes	Sagaidāmo HTTP statusa kodu saraksts. Ja Zabbix saņem kodu, kas nav sarakstā, solis neizdosies. Ja lauks ir tukšs, statusa kodu pārbaude netiek veikta. Piemēram: 200,201,210-299 Tiek atbalstīti lietotāja makrosi.

Note:
Jebkādas izmaiņas tīmekļa scenārija soļos tiks saglabātas tikai tad, kad tiks saglabāts viss scenārijs.

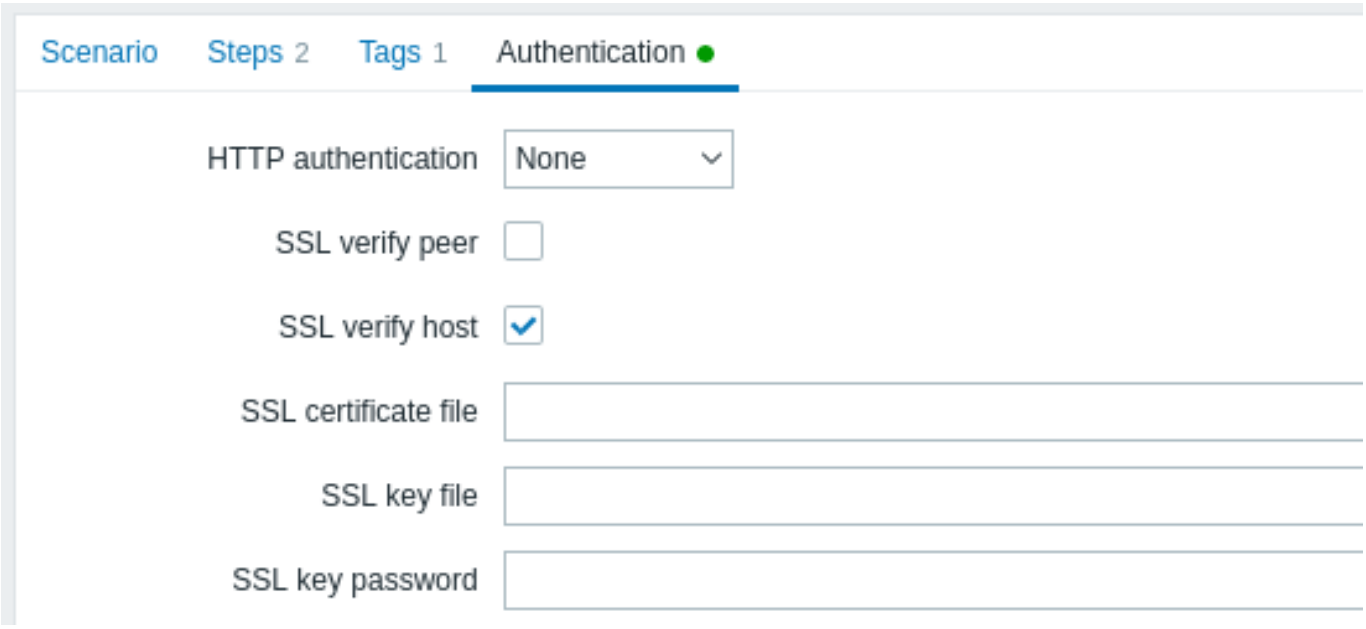
Skatiet arī reālas dzīves piemēru, kā var konfigurēt tīmekļa uzraudzības soļus.

Birku konfigurēšana Cilne **Tags** ļauj definēt scenārija līmeņa birkas.



Birku izmantošana ļauj filtrēt tīmekļa scenārijus un tīmekļa uzraudzības vienumus.

Autentifikācijas konfigurēšana Cilne **Authentication** ļauj konfigurēt scenārija autentifikācijas opcijas. Zaļš punkts blakus cilnes nosaukumam norāda, ka ir iespējots kāds HTTP autentifikācijas veids.



Autentifikācijas parametri:

Parametrs	Apraksts
HTTP authentication	Izvēlieties autentifikācijas opciju: **None** - autentifikācija netiek izmantota; **Basic** - tiek izmantota pamata autentifikācija; **NTLM** - tiek izmantota NTLM (Windows NT LAN Manager) autentifikācija; **Kerberos** - tiek izmantota Kerberos autentifikācija (skatiet arī: Configuring Kerberos with Zabbix); **Digest** - tiek izmantota Digest autentifikācija.
User	Ievadiet lietotājvārdu (līdz 255 rakstzīmēm). Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz Basic, NTLM, Kerberos vai Digest. Tiek atbalstīti lietotāja makrosi.
Password	Ievadiet lietotāja paroli (līdz 255 rakstzīmēm). Šis lauks ir pieejams, ja *HTTP authentication* ir iestatīts uz Basic, NTLM, Kerberos vai Digest. Tiek atbalstīti lietotāja makrosi.
SSL verify peer	Atzīmējiet izvēles rūtiņu, lai pārbaudītu tīmekļa servera SSL sertifikātu. Servera sertifikāts tiks automātiski ņemts no sistēmas sertifikātu iestādes (CA) atrašanās vietas. CA failu atrašanās vietu varat pārrakstīt, izmantojot Zabbix serveris vai starpniekserveris konfigurācijas parametru SSLCALocation. Tas iestata CURLOPT_SSL_VERIFYPEER cURL opciju.
SSL verify host	Atzīmējiet izvēles rūtiņu, lai pārbaudītu, vai tīmekļa servera sertifikāta lauks *Common Name* vai lauks *Subject Alternate Name* atbilst. Tas iestata CURLOPT_SSL_VERIFYHOST cURL opciju.
SSL certificate file	SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai. Sertifikāta failam jābūt PEM[1] formātā. Ja sertifikāta fails satur arī privāto atslēgu, atstājiet lauku *SSL key file* tukšu. Ja atslēga ir šifrēta, norādiet paroli laukā *SSL key password*. Direktoriju, kurā atrodas šis fails, nosaka Zabbix serveris vai starpniekserveris konfigurācijas parametrs SSLCertLocation. Šajā laukā var izmantot `HOST.*` makrosus un lietotāja makrosus. Tas iestata CURLOPT_SSLCERT cURL opciju.
SSL key file	SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai. Privātās atslēgas failam jābūt PEM[1] formātā. Direktoriju, kurā atrodas šis fails, nosaka Zabbix serveris vai starpniekserveris konfigurācijas parametrs SSLKeyLocation. Šajā laukā var izmantot `HOST.*` makrosus un lietotāja makrosus. Tas iestata CURLOPT_SSLKEY cURL opciju.
SSL key password	SSL privātās atslēgas faila parole. Šajā laukā var izmantot lietotāja makrosus. Tas iestata CURLOPT_KEYPASSWD cURL opciju.

Attention:
[1] Zabbix atbalsta tikai sertifikātu un privātās atslēgas failus PEM formātā. Ja jūsu sertifikāta un privātās atslēgas dati ir PKCS #12 formāta failā (parasti ar paplašinājumu *.p12 vai *.pfx), no tā varat ģenerēt PEM failu, izmantojot šādas komandas:

```
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 serveris uztver izmaiņas sertifikātos bez restartēšanas.

Note:
Ja klienta sertifikāts un privātā atslēga ir vienā failā, vienkārši norādiet to laukā "SSL certificate file" un atstājiet lauku "SSL key file" tukšu. Sertifikātam un atslēgai joprojām jābūt PEM formātā. Sertifikāta un atslēgas apvienošana ir vienkārša:

```
cat client.crt client.key > client.pem
```

Attēlošana Lai skatītu hostam konfigurētos tīmekļa scenārijus, dodieties uz *Uzraudzība* → *Hosti*, atrodiet hostu sarakstā un pēdējā kolonnā noklikšķiniet uz *Web* hipersaites. Noklikšķiniet uz scenārija nosaukuma, lai iegūtu detalizētu informāciju.

Tīmekļa scenāriju pārskatu var attēlot arī sadaļā *Paneļi* ar tīmekļa uzraudzības logrīku.

Jaunākie tīmekļa scenārija izpildes rezultāti ir pieejami sadaļā *Uzraudzība* → *Jaunākie dati*.

Paplašinātā uzraudzība Dažreiz ir nepieciešams reģistrēt saņemtās HTML lapas saturu. Tas ir īpaši noderīgi, ja kāds tīmekļa scenārija solis neizdodas. Šim nolūkam kalpo 5. atkļūdošanas līmenis (izsekošana). Šo līmeni var iestatīt serveris un starpniekserveris konfigurācijas failos vai izmantojot izpildlaika vadības opciju (`-R log_level_increase="http poller,N"`, kur N ir procesa numurs). Tālāk sniegtie piemēri parāda, kā var palaist paplašināto uzraudzību, ja 4. atkļūdošanas līmenis jau ir iestatīts:

```
# Palielināt žurnāla līmeni visiem http poller procesiem:
zabbix_server -R log_level_increase="http poller"

# Palielināt žurnāla līmeni otrajam http poller procesam:
zabbix_server -R log_level_increase="http poller,2"
```

Ja paplašinātā tīmekļa uzraudzība nav nepieciešama, to var apturēt, izmantojot opciju `-R log_level_decrease`.

1 Tīmekļa uzraudzības vienumi

Pārskats

Daži jauni vienumi tiek automātiski pievienoti uzraudzībai, kad tiek izveidoti tīmekļa scenāriji.

Visi vienumi manto birkas no tīmekļa scenārija.

Scenārija vienumi

Tiklīdz scenārijs ir izveidots, Zabbix automātiski pievieno šādus vienumus uzraudzībai.

Vienums	Apraksts
Scenārija <Scenario> lejupielādes ātrums	Šis vienums apkopos informāciju par visa scenārija lejupielādes ātrumu (baiti sekundē), t.i., vidējo rādītāju visiem soļiem. Vienuma atslēga: web.test.in[Scenario„bps] Tips: *Numeric(float)*
Scenārija <Scenario> neveiksmīgais solis	Šis vienums parādīs tā soļa numuru, kurā scenārijā radās kļūda. Ja visi soļi tiek izpildīti veiksmīgi, tiek atgriezts 0. Vienuma atslēga: web.test.fail[Scenario] Tips: *Numeric(unsigned)*
Scenārija <Scenario> pēdējais kļūdas ziņojums	Šis vienums atgriež scenārija pēdējā kļūdas ziņojuma tekstu. Jauna vērtība tiek saglabāta tikai tad, ja scenārijā ir neveiksmīgs solis. Ja visi soļi ir kārtībā, jauna vērtība netiek ievākta. Vienuma atslēga: web.test.error[Scenario] Tips: *Character*

”Scenario” vietā tiks izmantots faktiskais scenārija nosaukums.

Note:
Ja scenārija nosaukums satur lietotāja makrosus, šie makrosi tīmekļa uzraudzības vienumu nosaukumos paliks neatrisināti.

 Ja scenārija nosaukums sākas ar dubultpēdiņu vai satur komatu vai kvadrātiekavas, tas tiks pareizi ievietots pēdiņās vienumu atslēgās. Citos gadījumos papildu pēdiņu pievienošana netiks veikta.

Note:
Tīmekļa uzraudzības vienumi tiek pievienoti ar 30 dienu vēstures un 90 dienu tendenču glabāšanas periodu.

Šos vienumus var izmantot, lai izveidotu trigerus un definētu paziņojumu nosacījumus.

Piemērs 1

Lai izveidotu ”Tīmekļa scenārijs neizdevās” trigeri, varat definēt trigera izteiksmi:

```
last(/host/web.test.fail[Scenario])<>0
```

Pārliecinieties, ka aizstājat ’Scenario’ ar sava scenārija īsto nosaukumu.

Piemērs 2

Lai izveidotu trigeri ”Web scenario failed” ar noderīgu problēmas aprakstu trigera nosaukumā, varat definēt trigeri ar nosaukumu:

```
Web scenario "Scenario" failed: {ITEM.VALUE}
```

un trigera izteiksmi:

```
length(last(/host/web.test.error[Scenario]))>0 and last(/host/web.test.fail[Scenario])>0
```

Noteikti aizstājiet ’Scenario’ ar jūsu scenārija īsto nosaukumu.

Piemērs 3

Lai izveidotu trigeri ”Tīmekļa lietojumprogramma ir lēna”, varat definēt trigera izteiksmi:

```
last(/host/web.test.in[Scenario,,bps])<10000
```

Pārliecinieties, ka aizstājat ’Scenario’ ar sava scenārija īsto nosaukumu.

Scenārija soļu vienumi

Tiklīdz solis ir izveidots, Zabbix automātiski pievieno šādus vienumus uzraudzībai.

Vienums	Apraksts
Lejupielādes ātrums solim <Step> scenārijā <Scenario>	Šis vienums apkopos informāciju par soļa lejupielādes ātrumu (baiti sekundē). Vienuma atslēga: web.test.in[Scenario,Step,bps] Tips: *Numeric(float)*
Atbildes laiks solim <Step> scenārijā <Scenario>	Šis vienums apkopos informāciju par soļa atbildes laiku sekundēs. Atbildes laiks tiek skaitīts no pieprasījuma sākuma līdz brīdim, kad visa informācija ir pārsūtīta. Vienuma atslēga: web.test.time[Scenario,Step,resp] Tips: *Numeric(float)*
Atbildes kods solim <Step> scenārijā <Scenario>	Šis vienums apkopos soļa atbildes kodus. Vienuma atslēga: web.test.rspcode[Scenario,Step] Tips: *Numeric(unsigned)*

”Scenario” un ”Step” vietā attiecīgi tiks izmantoti faktiskie scenārija un soļa nosaukumi.

Note:
Tīmekļa uzraudzības vienumi tiek pievienoti ar 30 dienu vēsturi un 90 dienu tendenču glabāšanas periodu.

Note:
Ja scenārija nosaukums sākas ar dubultpēdiņu vai satur komatu vai kvadrātiekavas, tas tiks pareizi ievietots pēdiņās vienumu atslēgās. Citos gadījumos papildu pēdiņu pievienošana netiks veikta.

Šos vienumus var izmantot, lai izveidotu trigerus un definētu paziņojumu nosacījumus. Piemēram, lai izveidotu trigeri ”Zabbix GUI pieteikšanās ir pārāk lēna”, var definēt šādu trigera izteiksmi:

```
last(/zabbix/web.test.time[ZABBIX GUI,Login,resp])>3
```

2 Reālas dzīves scenārijs

Pārskats

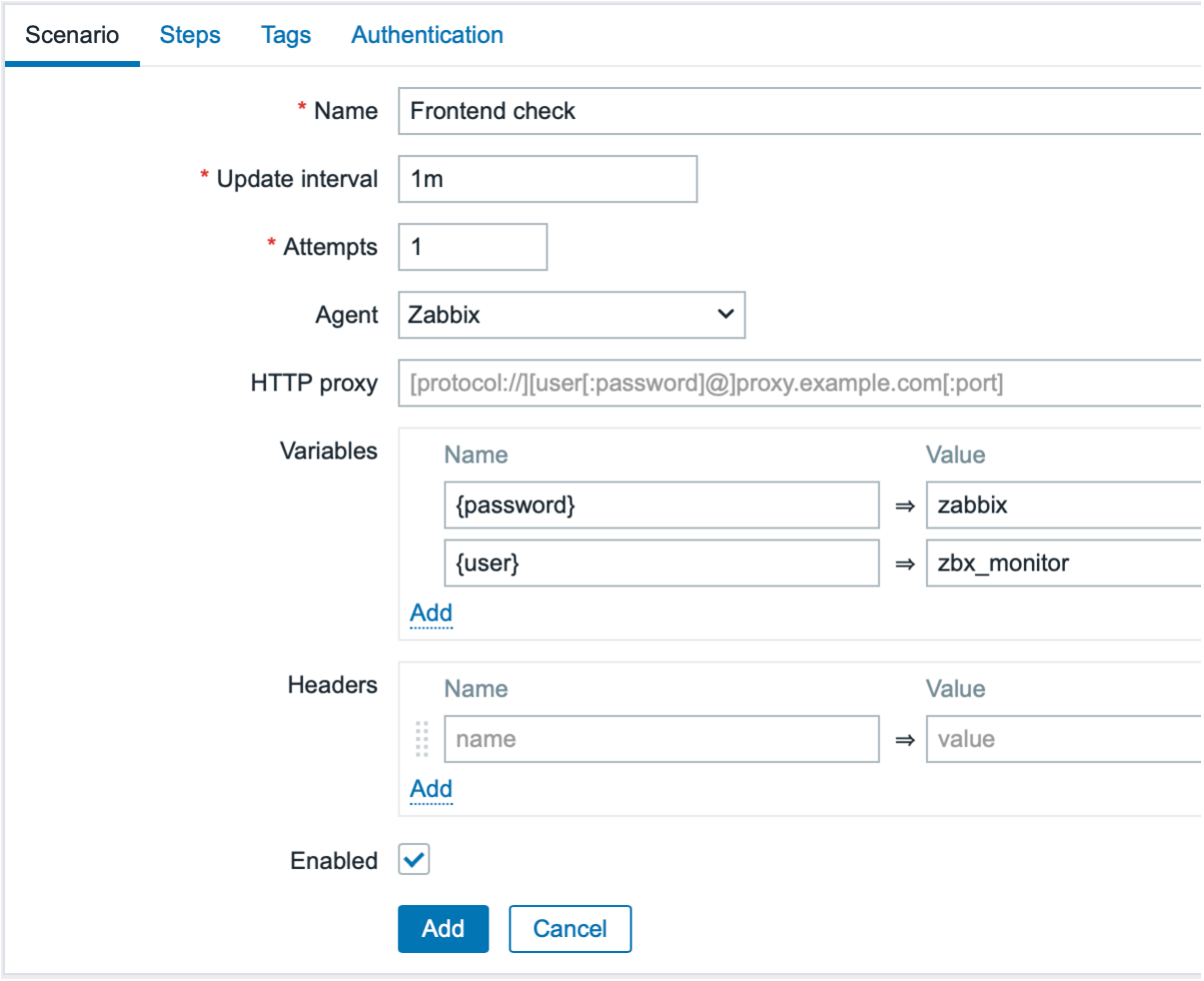
Šajā sadaļā ir sniegts soli pa solim reāls piemērs tam, kā var izmantot tīmekļa uzraudzību.

Zabbix tīmekļa uzraudzība tiks izmantota, lai uzraudzītu Zabbix lietotāja saskarne. Mērķis ir noteikt, vai tā ir pieejama, nodrošina pareizo saturu un cik ātri tā darbojas. Lai to panāktu, ir nepieciešami vairāki soļi, tostarp pirmās lapas pieejamības pārbaude, pieteikšanās ar lietotājvārdu un paroli, veiksmīgas pieteikšanās pārbaude, atteikšanās un atteikšanās apstiprināšana.

Scenārijs

Pievienojiet jaunu tīmekļa scenāriju

Dodieties uz *Datu ievākšana → Hosts*, izvēlieties hostu un šī hosta rindā noklikšķiniet uz *Web*.
Pēc tam noklikšķiniet uz *Create web scenario*.



Jaunā scenārija formā aizpildiet šādus laukus:

- **Name** - Lietotāja saskarnes pārbaude
- **Update interval** - 1m
- **Attempts** - 1
- **Agent** - Zabbix

Sadaļā *Variables* pievienojiet divus mainīgos: *{password}* un *{user}*.
Ievadiet savu esošo Zabbix lietotāja akreditācijas informāciju kā vērtības.

Drošības apsvērumu dēļ ieteicams izveidot atsevišķu lietotāju ar minimālām atļaujām, ko izmantot uzraudzības vajadzībām.

Pēc izvēles pārslēdzieties uz cilni *Tags* un pievienojiet tīmekļa scenārija birkas.

Note:
Kad šis tīmekļa scenārijs būs pilnībā konfigurēts, tas automātiski pievienos hostam Zabbix trapper vienumu.
Varat izmantot tīmekļa scenārija birkas, lai ātri identificētu saistītos vienumus un trigerus vai meklētu savāktos datus.
Piemēram, šai pamācībai piemērotas birkas ir `component: web-scenario` un/vai `target: frontend`.

Tīmekļa scenārija soļu konfigurēšana

Pārslēdzieties uz cilni *Soļi* un definējiet scenārija soļus. Noklikšķiniet uz pogas *Pievienot*, lai pievienotu atsevišķu soli.

Kopīgie lauki

Katram tālāk aprakstītajam solim papildus konkrētajam solim paredzētajiem laukiem aizpildiet šādus laukus:

- **URL** - Zabbix lietotāja saskarnes URL
- **Timeout** - 15s
- **Required status codes** - 200

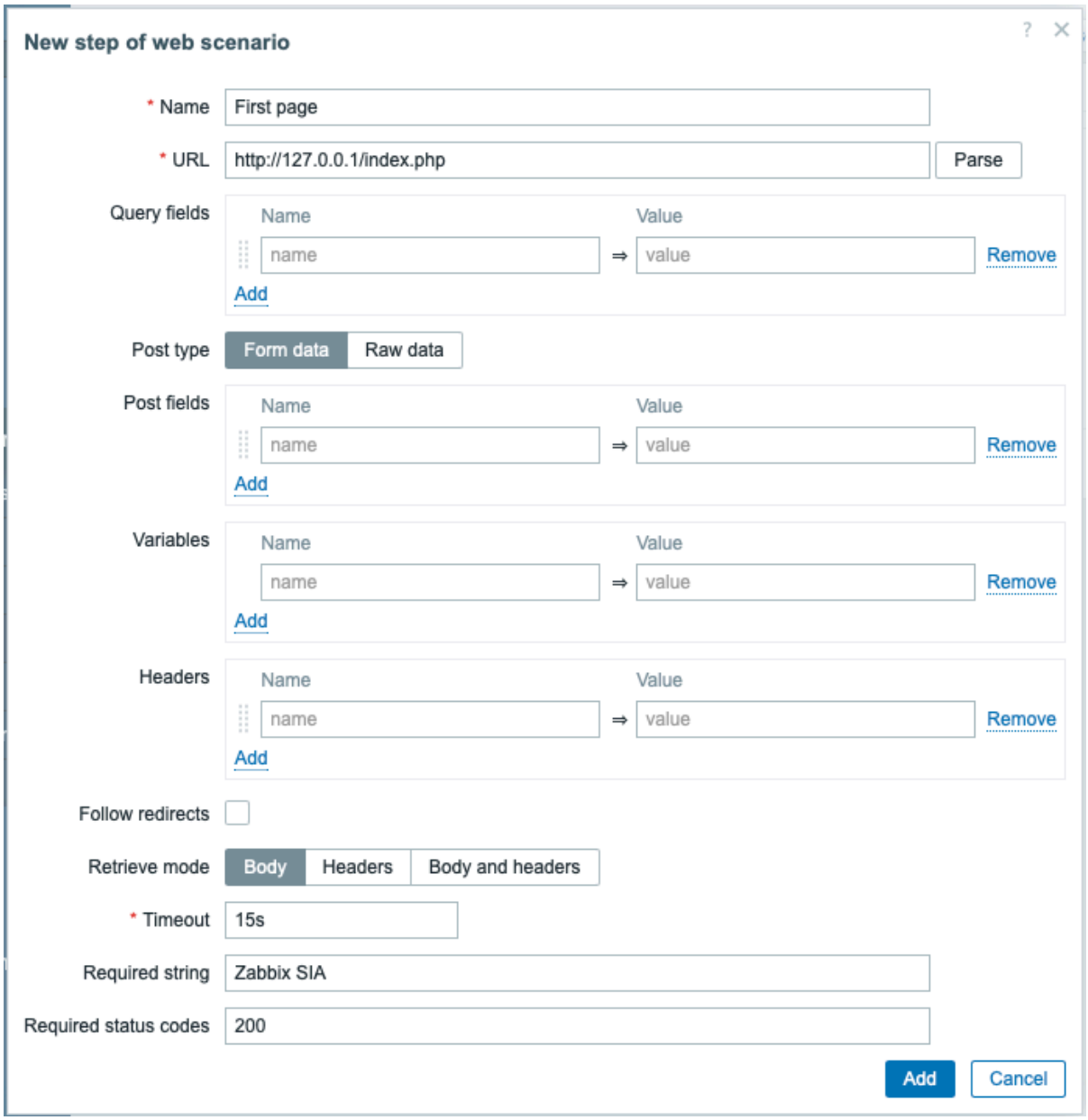
Tīmekļa scenārija 1. solis

Pārbaudiet, vai pirmā lapa atbild pareizi, atgriež HTTP atbildes kodu 200 un satur tekstu ”Zabbix SIA”.

- Laukā **Name** ievadiet *Pirmā lapa*.
- Laukā **Required string** ievadiet *Zabbix SIA*.

- Aizpildiet kopīgos laukus.

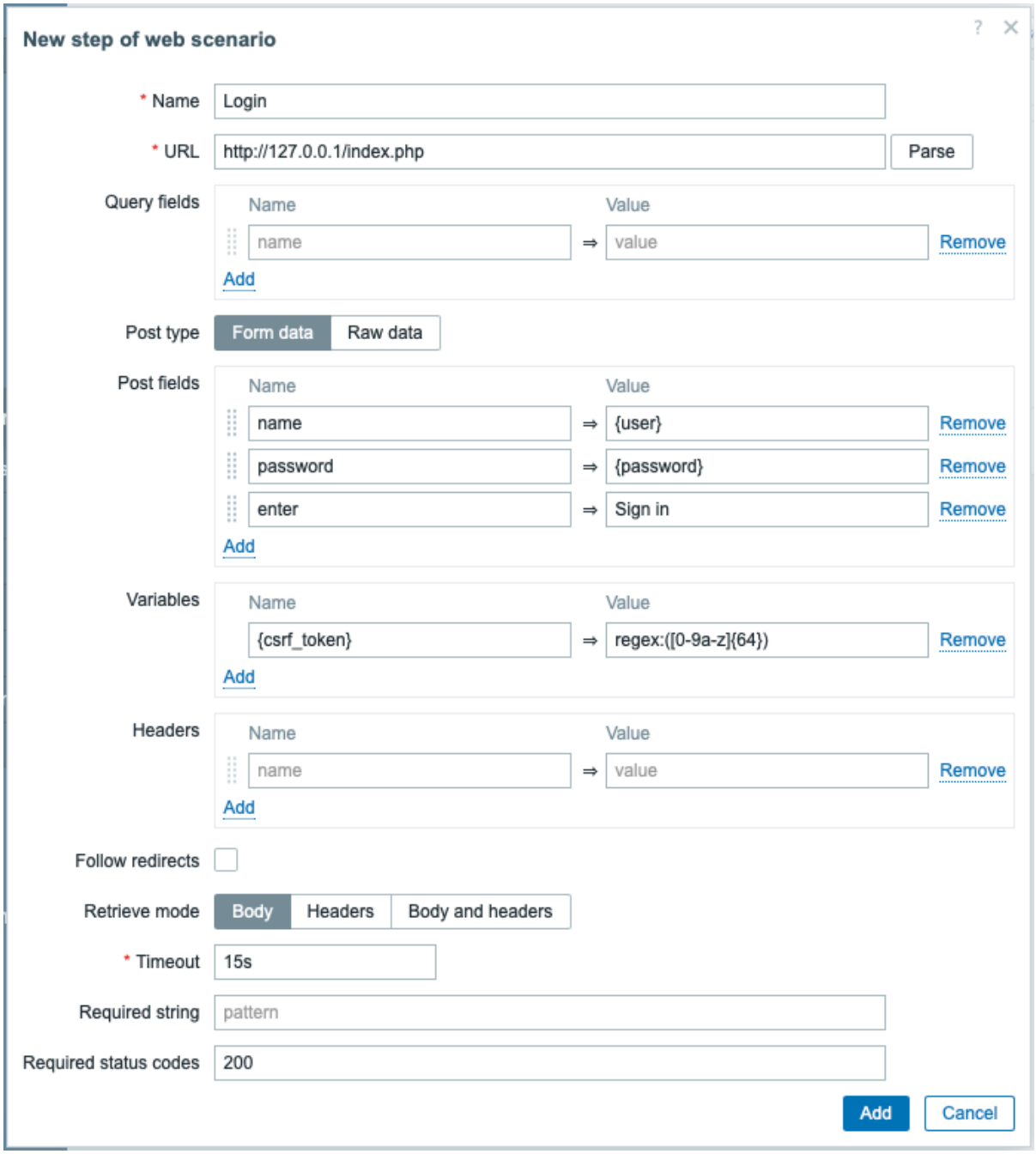
Kad soļa konfigurēšana ir pabeigta, nospiediet pogu *Add*.



Tīmekļa scenārija 2. solis

Piesakieties Zabbix lietotāja saskarnē, izmantojot scenārija līmenī definētos makrosus (mainīgos) — *{user}* un *{password}*.

- Laukā **Name** ievadiet *Login*.
- Sadaļā **Post fields** pievienojiet trīs post laukus:
  - *name* ar vērtību *{user}*
  - *password* ar vērtību *{password}*
  - *enter* ar vērtību *Sign in*
- Sadaļā **Variables** pievienojiet jaunu mainīgo *{csrf_token}* ar vērtību *regex:([0-9a-z]{64})*. Šis mainīgais uztvers piešķirtā CSRF pilnvaras marķiera vērtību, lai to atkārtoti izmantotu 4. solī.
- Aizpildiet kopīgos laukus.

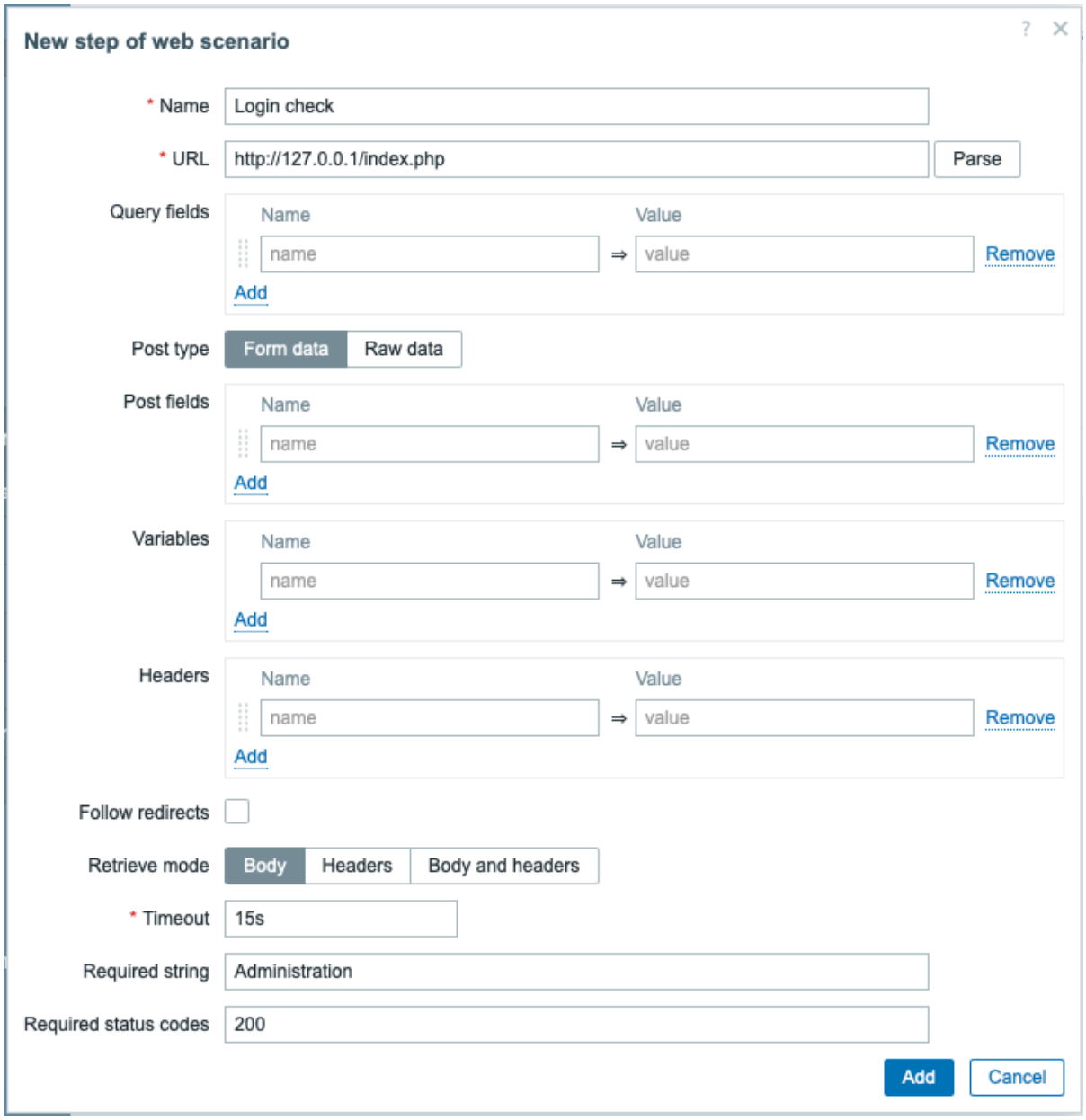


Attention:
Ņemiet vērā, ka Zabbix lietotāja saskarne, piesakoties, izmanto JavaScript pāradresāciju, tāpēc pieteikšanās jāveic vispirms, un pieteikušos lietotāju funkcijas var pārbaudīt tikai turpmākajos soļos. Turklāt pieteikšanās solī jāizmanto pilns **index.php** faila URL.

Tīmekļa scenārija 3. solis

Pēc pieteikšanās pārbaudiet, vai pieteikšanās ir veiksmīga, meklējot virkni, kas ir redzama tikai pēc pieteikšanās, piemēram, *Administration*.

- Laukā **Name** ievadiet *Login check*.
- Laukā **Required string** ievadiet *Administration*.
- Aizpildiet kopīgos laukus.

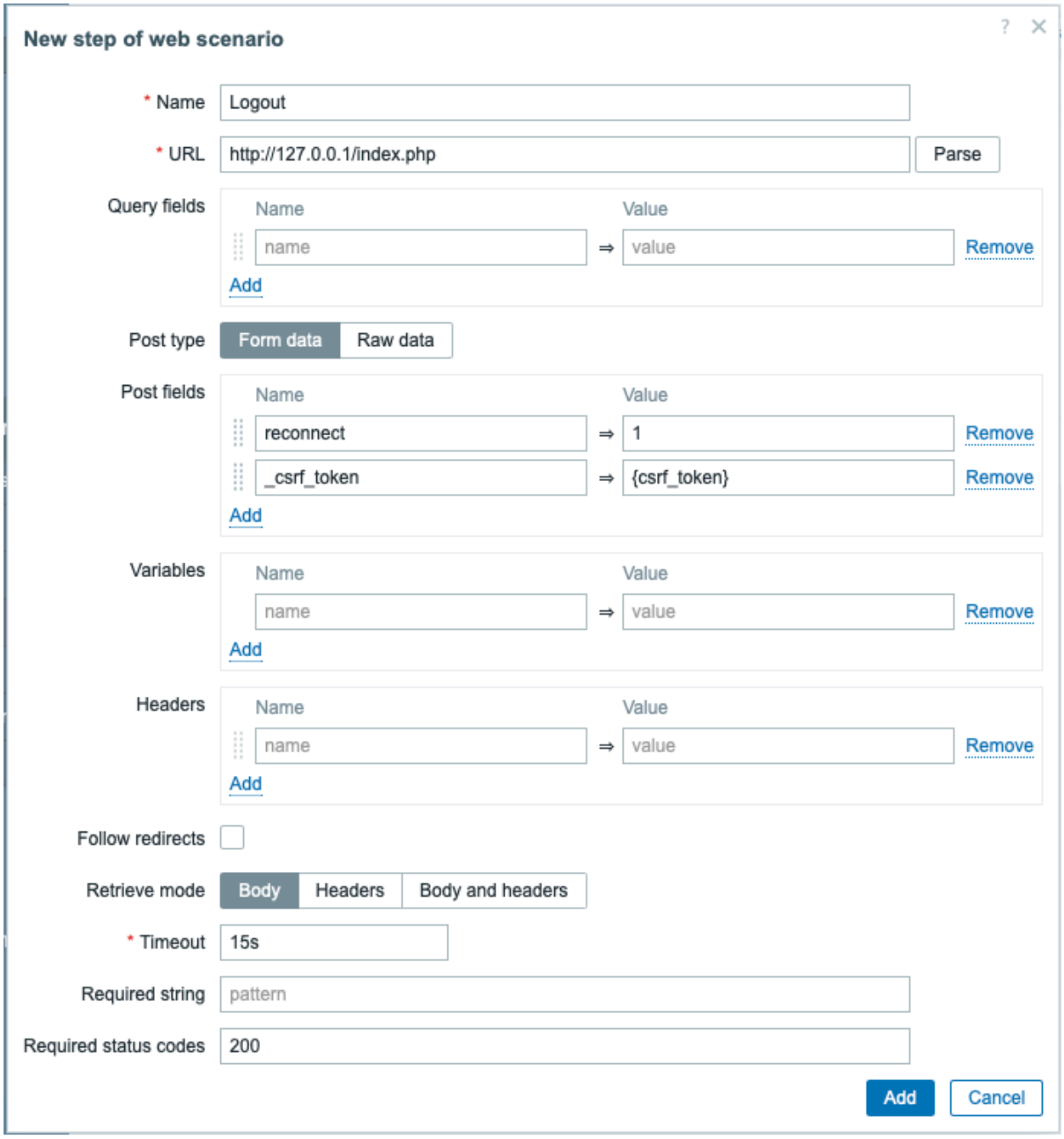


Tīmekļa scenārija 4. solis

Kad lietotāja saskarnes pieejamība un pieteikšanās ir pārbaudīta, pievienojiet atteikšanās soli — pretējā gadījumā Zabbix datubāze tiks pārblīvēta ar daudziem atvērtu sesiju ierakstiem.

- Laukā **Name** ievadiet *Logout*.
- Sadaļā **Post fields** pievienojiet divus post laukus:
  - *reconnect* ar vērtību *1*
  - *_csrf_token* ar vērtību *{csrf_token}*.
- Aizpildiet kopīgos laukus.

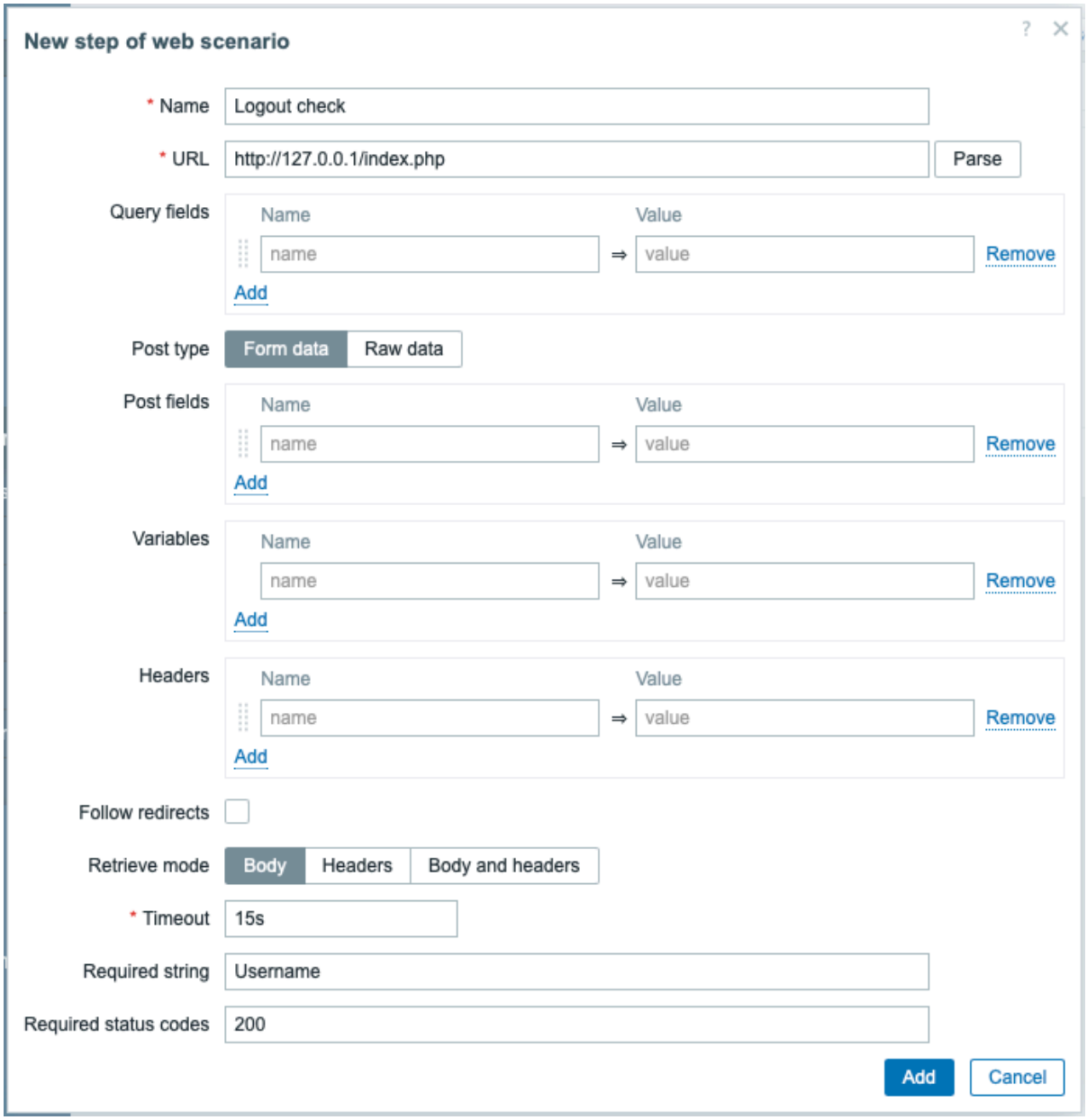
Šajā solī tiek izmantots mainīgais {csrf_token}, kas iegūts 2. solī



Tīmekļa scenārija 5. solis

Lai apstiprinātu atteikšanos, pārbaudiet virkni **Username**.

- Laukā **Name** ievadiet *Logout check*.
- Laukā **Required string** ievadiet *Username*.
- Aizpildiet kopīgos laukus.



Pilna soļu konfigurācija

Pilnīgai tīmekļa scenārija soļu konfigurācijai vajadzētu izskatīties šādi:

	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

Pārbaudiet rezultātus

Saglabājiet pabeigto tīmekļa uzraudzības scenāriju.

Scenārijs tiks pievienots hostam. Lai skatītu tīmekļa scenārija informāciju, dodieties uz *Uzraudzība → Hosti*, sarakstā atrodiet hostu un pēdējā kolonnā noklikšķiniet uz hipersaites Web.

Noklikšķiniet uz scenārija nosaukuma, lai skatītu detalizētāku statistiku:

8 Virtuālo mašīnu uzraudzība

Pārskats Zabbix var izmantot zema līmeņa atklāšanas noteikumus, lai automātiski atklātu VMware hipervizorus un virtuālās mašīnas un izveidotu hostus to uzraudzībai, pamatojoties uz iepriekš definētiem hostu prototipiem.

Zabbix ietver arī lietošanai gatavas veidnes VMware vCenter vai ESXi hipervizoru uzraudzībai.

Minimālā nepieciešamā VMware vCenter vai vSphere versija ir 5.1.

Datu vākšana Virtuālo mašīnu uzraudzība sastāv no diviem soļiem:

1. Zabbix *vmware collector* procesi apstrādā virtuālo mašīnu datus — procesi iegūst nepieciešamo informāciju no VMware tīmekļa pakalpojumiem, izmantojot SOAP protokolu, to iepriekš apstrādā un saglabā Zabbix servera koplietojamajā atmiņā.
2. Zabbix *poller* procesi izgūst datus, izmantojot Zabbix vienkāršo pārbaudi VMware monitoring item keys.

Zabbix sadala savāktos datus VMware konfigurācijas datos un VMware veiktspējas skaitītāju datos. Abu veidu datus neatkarīgi vāc *vmware collector* procesi.

Pamatojoties uz VMware veiktspējas skaitītāju informāciju, ir pieejama šāda statistika:

- Datu krātuve
- Diska ierīce
- CPU
- Barošana
- Tīkla saskarne
- Pielāgotie veiktspējas skaitītāju vienumi

Pilnu vienumu sarakstu, kas iegūst datus no VMware veiktspējas skaitītājiem, skatiet *VMware monitoring item keys*.

Ņemiet vērā, ka VMware notikumu izgūšanas biežums ir atkarīgs no vmware.eventlog aptaujas intervāla, taču tas nevar būt mazāks par 5 sekundēm.

Konfigurācija Ja Zabbix serveris ir kompilēts no avotiem, tas jākompilē ar konfigurācijas opcijām `--with-libcurl --with-libxml2`, lai iespējotu virtuālo mašīnu uzraudzību. Zabbix pakotnes jau ir kompilētas ar iespējotām šīm opcijām.

Virtuālo mašīnu uzraudzībai var mainīt šādus Zabbix servera konfigurācijas faila parametrus:

- `StartVMwareCollectors`

Note:
Ieteicams iespējot vairāk kolektoru nekā uzraudzīto VMware servisu skaits; pretējā gadījumā VMware veiktspējas skaitītāju statistikas iegūšana var aizkavēties VMware konfigurācijas datu iegūšanas dēļ (kas lielās instalācijās aizņem ilgāku laiku).

 Parasti `StartVMwareCollectors` vērtībai nevajadzētu būt mazākai par 2 un tai nevajadzētu pārsniegt divkāršu uzraudzīto VMware servisu skaitu: servisu skaits < `StartVMwareCollectors` < (servisu skaits * 2). Piemēram, uzraugot vienu VMware servisu, iestatiet `StartVMwareCollectors` uz 2; uzraugot trīs servisus, iestatiet `StartVMwareCollectors` uz 5.

 Ņemiet vērā, ka nepieciešamais kolektoru skaits ir atkarīgs arī no VMware vides apjoma un konfigurācijas parametriem `VMwareFrequency` un `VMwarePerfFrequency`.

- `VMwareCacheSize`
- `VMwareFrequency`
- `VMwarePerfFrequency`
- `VMwareTimeout`

Attention:
Lai atbalstītu datu krātuves ietilpības metriku, pārliecinieties, ka VMware atslēgas `vpxd.stats.maxQueryMetrics` vērtība ir iestatīta vismaz uz 64. Papildinformāciju skatiet *VMware Knowledge Base* rakstā.

Atklāšana

Zabbix var izmantot zema līmeņa atklāšanas noteikumus (piemēram, *vmware.hv.discovery[{$VMWARE.URL}]*) , lai automātiski atklātu VMware hipervizorus un virtuālās mašīnas. Turklāt Zabbix var izmantot hostu prototipus, lai automātiski ģenerētu reālus hostus atklātajām entītijām. Plašāku informāciju skatiet *Hostu prototipi*.

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*.

Paplašināta žurnalēšana Datus, ko apkopo *vmware collector* procesi, var reģistrēt detalizētai atkļūdošanai, izmantojot 5. atkļūdošanas līmeni. Atkļūdošanas līmeni var konfigurēt serveris un starpniekserveris konfigurācijas failos vai izmantojot izpildlaika vadības opciju `-R log_level_increase="vmware collector,N"`, kur "N" ir procesa numurs.

Piemēram, lai palielinātu atkļūdošanas līmeni no 4 līdz 5 visiem *vmware collector* procesiem, izpildiet šādu komandu:

```
zabbix_server -R log_level_increase="vmware collector"
```

Lai palielinātu atkļūdošanas līmeni no 4 līdz 5 otrajam *vmware collector* procesam, izpildiet šādu komandu:

```
zabbix_server -R log_level_increase="vmware collector,2"
```

Kad VMware collector datu paplašinātā žurnalēšana vairs nav nepieciešama, ieteicams samazināt atkļūdošanas līmeni līdz noklusējuma vērtībai (3), izpildot komandu `-R log_level_decrease`.

Problēmu novēršana

- Ja metriskie dati nav pieejami, lūdzu, pārliecinieties, ka tie nav padarīti nepieejami vai pēc noklusējuma izslēgti jaunākajās VMware vSphere versijās, un ka veiktspējas metriku datubāzes vaicājumiem nav noteikti ierobežojumi. Plašāku informāciju skatiet ZBX-12094.
- Ja rodas kļūda, ka `config.vpxd.stats.maxQueryMetrics` nav derīgs vai pārsniedz maksimāli atļauto rakstzīmju skaitu, pievienojiet parametru `config.vpxd.stats.maxQueryMetrics` vCenter Server iestatījumiem. Šī parametra vērtībai jābūt tādai pašai kā `maxQuerysize` vērtībai VMware *web.xml* failā. Plašāku informāciju skatiet *VMware Knowledge Base* rakstā.
- Ja jums ir aizdomas, ka jūsu Zabbix instalācija izmanto pārāk daudz atmiņas, skatiet Pārmērīga atmiņas lietojuma profilēšana ar tcmalloc.

1 VMware uzraudzības vienumu atslēgas

Pārskats Šajā lapā ir sniegta informācija par vienkāršajām pārbaudēm, kuras var izmantot VMware vides uzraudzībai. Metrikas ir sagrupētas pēc uzraudzības mērķa.

Atbalstītās vienumu atslēgas Vienumu atslēgas ir uzskaitītas bez parametriem un papildu informācijas. Noklikšķiniet uz vienuma atslēgas, lai skatītu pilnu informāciju.

Vienuma atslēga	Apraksts	Vienumu grupa
vmware.eventlog	VMware notikumu žurnāls.	Vispārīgs serviss
vmware.fullname	VMware servisa pilnais nosaukums.	
vmware.version	VMware servisa versija.	
vmware.cl.perfcounter	VMware klastera veiktspējas skaitītāja metrikas.	Klasteris
vmware.cluster.alarms.get	VMware klastera trauksmju dati.	
vmware.cluster.discovery	VMware klasteru atklāšana.	
vmware.cluster.property	VMware klastera īpašība.	
vmware.cluster.status	VMware klastera statuss.	
vmware.cluster.tags.get	VMware klastera birku masīvs.	
vmware.datastore.alarms.get	VMware datu krātuves trauksmju dati.	Datu krātuve
vmware.datastore.discovery	VMware datu krātuvju atklāšana.	
vmware.datastore.hv.list	Datu krātuves hipervizoru saraksts.	
vmware.datastore.perfcounter	VMware datu krātuves veiktspējas skaitītāja vērtība.	
vmware.datastore.property	VMware datu krātuves īpašība.	
vmware.datastore.read	Laiks, kas nepieciešams lasīšanas operācijai no datu krātuves.	
vmware.datastore.size	VMware datu krātuves vieta baitos vai procentos no kopējā apjoma.	
vmware.datastore.tags.get	VMware datu krātuves birku masīvs.	
vmware.datastore.write	Laiks, kas nepieciešams rakstīšanas operācijai datu krātuvē.	
vmware.dc.alarms.get	VMware datu centra trauksmju dati.	Datu centrs
vmware.dc.discovery	VMware datu centru atklāšana.	
vmware.dc.tags.get	VMware datu centra birku masīvs.	
vmware.dvswitch.discovery	VMware vSphere Distributed Switch atklāšana.	vSphere Distributed Switch
vmware.dvswitch.fetchportsget	VMware vSphere Distributed Switch portu dati.	
vmware.hv.alarms.get	VMware hipervizora trauksmju dati.	Hipervizors
vmware.hv.cluster.name	VMware hipervizora klastera nosaukums.	
vmware.hv.connectionstate	VMware hipervizora savienojuma stāvoklis.	
vmware.hv.cpu.usage	VMware hipervizora procesora lietojums (Hz).	
vmware.hv.cpu.usage.perf	VMware hipervizora procesora lietojums procentos intervāla laikā.	
vmware.hv.cpu.utilization	VMware hipervizora procesora lietojums procentos intervāla laikā, atkarīgs no enerģijas pārvaldības vai HT.	

Vienuma atslēga	Apraksts	Vienumu grupa
vmware.hv.datacenter.name	VMware hipervizora datu centra nosaukums.	
vmware.hv.datastore.discovery	VMware hipervizora datu krātuvju atklāšana.	
vmware.hv.datastore.list	VMware hipervizora datu krātuvju saraksts.	
vmware.hv.datastore.multipath	Pieejamo datu krātuves ceļu skaits.	
vmware.hv.datastore.read	Vidējais laiks, kas nepieciešams lasīšanas operācijai no datu krātuves.	
vmware.hv.datastore.size	VMware datu krātuves vieta baitos vai procentos no kopējā apjoma.	
vmware.hv.datastore.write	Vidējais laiks, kas nepieciešams rakstīšanas operācijai datu krātuvē.	
vmware.hv.discovery	VMware hipervizoru atklāšana.	
vmware.hv.diskinfo.get	VMware hipervizora disku dati.	
vmware.hv.fullname	Pilns produkta nosaukums, ieskaitot versijas informāciju.	
vmware.hv.hw.cpu.freq	VMware hipervizora procesora frekvence.	
vmware.hv.hw.cpu.model	VMware hipervizora procesora modelis.	
vmware.hv.hw.cpu.num	Procesora kodolu skaits VMware hipervizorā.	
vmware.hv.hw.cpu.threads	Procesora pavedienu skaits VMware hipervizorā.	
vmware.hv.hw.memory	VMware hipervizora kopējais atmiņas apjoms.	
vmware.hv.hw.model	VMware hipervizora modelis.	
vmware.hv.hw.sensors.get	VMware hipervizora aparatūras sensoru vērtība.	
vmware.hv.hw.serialnumber	VMware hipervizora sērijas numurs.	
vmware.hv.hw.uuid	VMware hipervizora BIOS UUID.	
vmware.hv.hw.vendor	VMware hipervizora piegādātāja nosaukums.	
vmware.hv.maintenance	VMware hipervizora apkopes statuss.	
vmware.hv.memory.size.ballooned	VMware hipervizora ballooned atmiņas apjoms.	
vmware.hv.memory.used	VMware hipervizora izmantotās atmiņas apjoms.	
vmware.hv.net.if.discovery	VMware hipervizora tīkla saskarņu atklāšana.	
vmware.hv.network.in	VMware hipervizora tīkla ienākošās plūsmas statistika.	
vmware.hv.network.linkspeed	VMware hipervizora tīkla saskarnes ātrums.	
vmware.hv.network.out	VMware hipervizora tīkla izejošās plūsmas statistika.	
vmware.hv.perfcounter	VMware hipervizora veiktspējas skaitītāja vērtība.	
vmware.hv.property	VMware hipervizora īpašība.	
vmware.hv.power	VMware hipervizora enerģijas patēriņš.	
vmware.hv.sensor.health.state	VMware hipervizora veselības stāvokļa kopsavilkuma sensors.	
vmware.hv.sensors.get	VMware hipervizora aparatūras piegādātāja stāvokļa sensori.	
vmware.hv.status	VMware hipervizora statuss.	
vmware.hv.tags.get	VMware hipervizora birku masīvs.	
vmware.hv.uptime	VMware hipervizora darbspējas laiks.	
vmware.hv.version	VMware hipervizora versija.	
vmware.hv.vm.num	Virtuālo mašīnu skaits VMware hipervizorā.	
vmware.rp.cpu.usage	CPU lietojums hercos intervāla laikā VMware resursu pūlā.	Resursu pūls
vmware.rp.memory	VMware resursu pūla atmiņas metrikas.	
vmware.alarms.get	VMware virtual center trauksmju dati.	Virtuālais centrs
vmware.vm.alarms.get	VMware virtuālās mašīnas trauksmju dati.	Virtuālā mašīna
vmware.vm.attribute	VMware virtuālās mašīnas pielāgotā atribūta vērtība.	
vmware.vm.cluster.name	VMware virtuālās mašīnas nosaukums.	
vmware.vm.consolidationneeded	VMware virtuālās mašīnas diskam nepieciešama konsolidācija.	
vmware.vm.cpu.latency	Laika procentuālā daļa, kurā virtuālā mašīna nevar darboties, jo konkurē par piekļuvi fiziskajam CPU(-iem).	
vmware.vm.cpu.num	Procesoru skaits VMware virtuālajā mašīnā.	
vmware.vm.cpu.readiness	Laika procentuālā daļa, kurā virtuālā mašīna bija gatava darbam, bet nevarēja tikt ieplānota darbībai uz fiziskā CPU.	
vmware.vm.cpu.ready	Laiks, kurā virtuālā mašīna bija gatava darbam, bet nevarēja tikt ieplānota darbībai uz fiziskā CPU.	
vmware.vm.cpu.swapwait	CPU laika procentuālā daļa, kas pavadīta, gaidot swap-in.	
vmware.vm.cpu.usage	VMware virtuālās mašīnas procesora lietojums (Hz).	
vmware.vm.cpu.usage.perf	VMware virtuālās mašīnas procesora lietojums procentos intervāla laikā.	
vmware.vm.datacenter.name	VMware virtuālās mašīnas datu centra nosaukums.	
vmware.vm.discovery	VMware virtuālo mašīnu atklāšana.	
vmware.vm.guest.memory.size.swapped	Viesa fiziskās atmiņas apjoms, kas ir pārvietots uz maiņvietu.	
vmware.vm.guest.osuptime	Kopējais laiks, kas pagājis kopš pēdējās operētājsistēmas palaišanas.	

Vienuma atslēga	Apraksts	Vienumu grupa
vmware.vm.hv.maintenance	VMware virtuālās mašīnas hipervizora apkopes statuss.	
vmware.vm.hv.name	VMware virtuālās mašīnas hipervizora nosaukums.	
vmware.vm.memory.size	VMware virtuālās mašīnas kopējais atmiņas apjoms.	
vmware.vm.memory.size.ballooned	VMware virtuālās mašīnas ballooned atmiņas apjoms.	
vmware.vm.memory.size.compressed	VMware virtuālās mašīnas saspiestās atmiņas apjoms.	
vmware.vm.memory.size.consumed	Hosta fiziskās atmiņas apjoms, kas patērēts viesa fiziskās atmiņas lapu uzturēšanai.	
vmware.vm.memory.size.private	VMware virtuālās mašīnas privātās atmiņas apjoms.	
vmware.vm.memory.size.shared	VMware virtuālās mašīnas koplietotās atmiņas apjoms.	
vmware.vm.memory.size.swapped	VMware virtuālās mašīnas swapotās atmiņas apjoms.	
vmware.vm.memory.size.usage.guest	VMware virtuālās mašīnas viesa atmiņas lietojums.	
vmware.vm.memory.size.usage.host	VMware virtuālās mašīnas hosta atmiņas lietojums.	
vmware.vm.memory.usage	Hosta fiziskās atmiņas procentuālā daļa, kas ir patērēta.	
vmware.vm.net.if.discovery	VMware virtuālās mašīnas tīkla saskarņu atklāšana.	
vmware.vm.net.if.in	VMware virtuālās mašīnas tīkla saskarnes ienākošās plūsmas statistika.	
vmware.vm.net.if.out	VMware virtuālās mašīnas tīkla saskarnes izejošās plūsmas statistika.	
vmware.vm.net.if.usage	VMware virtuālās mašīnas tīkla izmantojums intervāla laikā.	
vmware.vm.perfcounter	VMware virtuālās mašīnas veiktspējas skaitītāja vērtība.	
vmware.vm.powerstate	VMware virtuālās mašīnas barošanas stāvoklis.	
vmware.vm.property	VMware virtuālās mašīnas īpašība.	
vmware.vm.snapshot.get	VMware virtuālās mašīnas momentuzņēmuma stāvoklis.	
vmware.vm.state	VMware virtuālās mašīnas stāvoklis.	
vmware.vm.storage.committed	VMware virtuālās mašīnas rezervētā krātuves vieta.	
vmware.vm.storage.readoio	Vidējais neizpildīto lasīšanas pieprasījumu skaits uz virtuālo disku datu vākšanas intervāla laikā.	
vmware.vm.storage.totalreadlatency	Vidējais laiks, kas nepieciešams lasīšanai no virtuālā diska.	
vmware.vm.storage.totalwritelatency	Vidējais laiks, kas nepieciešams rakstīšanai uz virtuālo disku.	
vmware.vm.storage.uncommitted	VMware virtuālās mašīnas nerezervētā krātuves vieta.	
vmware.vm.storage.unshared	VMware virtuālās mašīnas nekoplietotā krātuves vieta.	
vmware.vm.storage.writeoio	Vidējais neizpildīto rakstīšanas pieprasījumu skaits uz virtuālo disku datu vākšanas intervāla laikā.	
vmware.vm.tags.get	VMware virtuālās mašīnas birku masīvs.	
vmware.vm.tools	VMware virtuālās mašīnas viesa rīku stāvoklis vai versija.	
vmware.vm.uptime	VMware virtuālās mašīnas darbspējas laiks.	
vmware.vm.vfs.dev.discovery	VMware virtuālās mašīnas disku ierīču atklāšana.	
vmware.vm.vfs.dev.read	VMware virtuālās mašīnas diska ierīces lasīšanas statistika.	
vmware.vm.vfs.dev.write	VMware virtuālās mašīnas diska ierīces rakstīšanas statistika.	
vmware.vm.vfs.fs.discovery	VMware virtuālās mašīnas failu sistēmu atklāšana.	
vmware.vm.vfs.fs.size	VMware virtuālās mašīnas failu sistēmas statistika.	

Vienuma atslēgas detaļas Parametri bez leņķiekavām ir obligāti. Parametri, kas atzīmēti ar leņķiekavām **< >**, ir neobligāti.

vmware.eventlog[url,<mode>,<severity>]

 VMware notikumu žurnāls.
 Atgrieztā vērtība: *Log*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **mode** - *all* (noklusējums) vai *skip* - izlaist vecāku datu apstrādi;
- **severity** - filtrēt pēc svarīguma: *error*, *warning*, *info* vai *user*. Šim parametram jābūt pēdiņās, ja ir norādīts vairāk nekā viens svarīguma līmenis ar komatiem atdalītā sarakstā (piem.,`"error,warning,info,user"`). Pēc noklusējuma atspējots.

Komentāri:

- Katram URL drīkst būt tikai viena `vmware.eventlog` vienuma atslēga;
- Skatiet arī VMware notikumu žurnāla ierakstu filtrēšanas piemēru.

vmware.fullnameurl

 VMware servisa pilnais nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL.

vmware.versionurl

 VMware servisa versija.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL.

vmware.cl.perfcounter[url,id,path,<instance>]

 VMware klastera veiktspējas skaitītāja metrikas.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **id** - VMware klastera ID. `id` var saņemt no `vmware.cluster.discovery[]` kā {#CLUSTER.ID}.
- **path** - veiktspējas skaitītāja ceļš[1];
- **instance** - veiktspējas skaitītāja instance.

vmware.cluster.alarms.get[url,id]

 VMware klastera trauksmju dati.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **id** - VMware klastera ID.

vmware.cluster.discoveryurl

 VMware klasteru atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.cluster.property[url,id,prop]

 VMware klastera īpašība.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL;
- **id** - VMware klastera ID;
- **prop** - īpašības ceļš, kas ir VM objekta īpašības nosaukums, kā definēts VMware SDK.

Piemēri:

```
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},overallStatus]
vmware.vm.property[{$VMWARE.URL},{$VMWARE.VM.UUID},runtime.powerState]
```

vmware.cluster.status[url,name]

 VMware klastera statuss.
 Atgrieztā vērtība: *0* - pelēks; *1* - zaļš; *2* - dzeltens; *3* - sarkans.

Parametri:

- **url** - VMware pakalpojuma URL;
- **name** - VMware klastera nosaukums.

vmware.cluster.tags.get[url,id]

 VMware klastera birku masīvs.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **id** - VMware klastera ID.

Šis vienums darbojas ar vSphere 6.5 un jaunākām versijām.

vmware.datastore.alarms.get[url,uuid]

 VMware datu glabātuves trauksmju dati.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware datu glabātuves globāli unikālais identifikators.

vmware.datastore.discovery[url,<filter_uuid>]

 VMware datu krātuvju atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **filter_uuid** - viens unikāls HV vai VM identifikators (noklusējums: tukšs). Ja norādīts, tiks atklātas tikai piesaistītās datu krātuves.

vmware.datastore.hv.list[url,datastore]

 Datu krātuves hipervizoru saraksts.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL;
- **datastore** - datu krātuves UUID vai nosaukums.

Izvades piemērs:

```
esx7-01-host.zabbix.sandbox
esx7-02-host.zabbix.sandbox
```

vmware.datastore.perfcounter[url,uuid,path,<instance>]

 VMware datu krātuves veiktspējas skaitītāja vērtība.
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware datu krātuves globāli unikālais identifikators;
- **path** - veiktspējas skaitītāja ceļš[1];
- **instance** - veiktspējas skaitītāja instance. Apkopotajām vērtībām izmantojiet tukšu instanci (noklusējums). `instance` var iegūt no `vmware.datastore.discovery[]` kā daļu no masīva {#DATASTORE.EXTENT}.

vmware.datastore.property[url,uuid,prop]

 VMware datu krātuves īpašība.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware datu krātuves globāli unikālais identifikators;
- **prop** - īpašības ceļš.

vmware.datastore.read[url,datastore,<mode>]

 Laiks, kas nepieciešams lasīšanas operācijai no datu krātuves (milisekundēs).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - *latency* (vidējā vērtība, noklusējums) vai *maxlatency* (maksimālā vērtība).

vmware.datastore.size[url,datastore,<mode>]

 VMware datu krātuves vieta baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.

Parametri:

- **url** - VMware servisa URL;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - iespējamās vērtības: *total* (noklusējums), *free*, *pfree* (brīvās vietas procents), *uncommitted*.

vmware.datastore.tags.get[url,uuid]

 VMware datu krātuves tagu masīvs.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware datu krātuves globālais unikālais identifikators.

Šis vienums darbojas ar vSphere 6.5 un jaunākām versijām.

vmware.datastore.write[url,datastore,<mode>]

 Laiks, kas nepieciešams rakstīšanas operācijai datu krātuvē (milisekundēs).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - *latency* (vidējā vērtība, noklusējums) vai *maxlatency* (maksimālā vērtība).

vmware.dc.alarms.get[url,id]

 VMware datu centra trauksmju dati.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **id** - VMware datu centra ID.

vmware.dc.discoveryurl

 VMware datu centru atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.dc.tags.get[url,id]

 VMware datu centra birku masīvs.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **id** - VMware datu centra ID.

Šis vienums darbojas ar vSphere 6.5 un jaunākām versijām.

vmware.dvswitch.discoveryurl

 VMware vSphere Distributed Switches atklāšana.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.dvswitch.fetchports.get[url,uuid,<filter>,<mode>]

 VMware vSphere Distributed Switch portu dati.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware DVSwitch globāli unikālais identifikators.
- **filter** - viena virkne ar kritērijiem, kas atdalīti ar komatiem, portu atlasei;
- **mode** - *state* (viss XML bez ”config” XML mezgliem, pēc noklusējuma) vai *full*.

Parametrs **filter** atbalsta kritērijus, kas pieejami VMware datu objektā DistributedVirtualSwitchPortCriteria.

Piemērs:

```
vmware.dvswitch.fetchports.get[{$VMWARE.URL},{$VMWARE.DVS.UUID},"connected:true,active:true,uplinkPort:fal
```

vmware.hv.alarms.get[url,uuid]

 VMware hipervizora trauksmju dati.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.cluster.name[url,uuid]

 VMware hipervizora klastera nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;

- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.connectionstate[url,uuid]

 VMware hipervizora savienojuma stāvoklis.
 Atgrieztā vērtība: *String*: *connected*, *disconnected* vai *notResponding*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.cpu.usage[url,uuid]

 VMware hipervizora procesora lietojums (Hz).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.cpu.usage.perf[url,uuid]

 VMware hipervizora procesora izmantojums procentos intervāla laikā.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.cpu.utilization[url,uuid]

 VMware hipervizora procesora izmantošana procentos intervāla laikā; atkarīga no enerģijas pārvaldības vai HT.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.datacenter.name[url,uuid]

 VMware hipervizora datu centra nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

vmware.hv.datastore.discovery[url,uuid]

 VMware hipervizora datu krātuvju atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.datastore.list[url,uuid]

 VMware hipervizora datu krātuvju saraksts.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Izvades piemērs:

```
SSD-RAID1-VAULT1
SSD-RAID1-VAULT2
SSD-RAID10
```

vmware.hv.datastore.multipath[url,uuid,<datastore>,<partitionid>]

 Pieejamo datu krātuves ceļu skaits.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators;

- **datastore** - datu krātuves UUID vai nosaukums;
- **partitionid** - fiziskās ierīces iekšējais ID no `vmware.hv.datastore.discovery`.

vmware.hv.datastore.read[url,uuid,datastore,<mode>]

 Vidējais laiks lasīšanas operācijai no datu krātuves (milisekundēs).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - *latency* (noklusējums).

vmware.hv.datastore.size[url,uuid,datastore,<mode>]

 VMware datu krātuves vieta baitos vai procentos no kopējā apjoma.
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - iespējamās vērtības: *total* (noklusējums), *free*, *pfree* (brīvās vietas procents), *uncommitted*.

vmware.hv.datastore.write[url,uuid,datastore,<mode>]

 Vidējais laiks rakstīšanas operācijai datu krātuvē (milisekundēs).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **datastore** - datu krātuves UUID vai nosaukums;
- **mode** - *latency* (noklusējums).

vmware.hv.discoveryurl

 VMware hipervizoru atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.hv.diskinfo.get[url,uuid]

 VMware hipervizora diska dati.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.fullname[url,uuid]

 Pilns produkta nosaukums, ieskaitot versijas informāciju.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

vmware.hv.hw.cpu.freq[url,uuid]

 VMware hipervizora procesora frekvence (Hz).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.hw.cpu.model[url,uuid]

 VMware hipervizora procesora modelis.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;

- **uuid** - VMware hipervizora globālais unikālais identifikators.

vmware.hv.hw.cpu.num[url,uuid]

 Procesora kodolu skaits VMware hipervizorā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.hw.cpu.threads[url,uuid]

 Procesora pavedienu skaits VMware hipervizorā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.hw.memory[url,uuid]

 VMware hipervizora kopējais atmiņas apjoms (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.hw.model[url,uuid]

 VMware hipervizora modelis.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

vmware.hv.hw.sensors.get[url,uuid]

 VMware hipervizora aparatūras sensoru vērtība.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.hw.serialnumber[url,uuid]

 VMware hipervizora sērijas numurs.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Šis vienums darbojas ar vSphere API 6.7 un jaunāku versiju.

vmware.hv.hw.uuid[url,uuid]

 VMware hipervizora BIOS UUID.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

Šis vienums darbojas ar vSphere API 6.7 un jaunāku.

vmware.hv.hw.vendor[url,uuid]

 VMware hipervizora piegādātāja nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Šis vienums darbojas ar vSphere API 6.7 un jaunāku.

vmware.hv.maintenance[url,uuid]

 VMware hipervizora uzturēšanas statuss.
 Atgrieztā vērtība: *0* - nav uzturēšanas režīmā; *1* - uzturēšanas režīmā.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.memory.size.ballooned[url,uuid]

 VMware hipervizora ballooned atmiņas izmērs (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

vmware.hv.memory.used[url,uuid]

 VMware hipervizora izmantotās atmiņas apjoms (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.net.if.discovery[url,uuid]

 VMware hipervizora tīkla saskarņu atklāšana.
 Atgrieztā vērtība: *JSON objekts*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.network.in[url,uuid,<mode>]

 VMware hipervizora tīkla ienākošās statistikas dati (baiti sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **mode** - *bps* (noklusējums), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.network.linkspeed[url,uuid,ifname]

 VMware hipervizora tīkla saskarnes ātrums.
 Atgrieztā vērtība: *Integer*. Atgriež *0*, ja tīkla saskarne nedarbojas, pretējā gadījumā — saskarnes ātruma vērtību.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **ifname** - saskarnes nosaukums.

vmware.hv.network.out[url,uuid,<mode>]

 VMware hipervizora tīkla izejošās plūsmas statistika (baiti sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **mode** - *bps* (noklusējums), *packets*, *dropped*, *errors*, *broadcast*.

vmware.hv.perfcounter[url,uuid,path,<instance>]

 VMware hipervizora veiktspējas skaitītāja vērtība.
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **path** - veiktspējas skaitītāja ceļš [1];
- **instance** - veiktspējas skaitītāja instance. Izmantojiet tukšu instanci apkopotajām vērtībām (noklusējums).

vmware.hv.property[url,uuid,prop]

 VMware hipervizora īpašība.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **prop** - īpašības ceļš.

vmware.hv.power[url,uuid,<max>]

 VMware hipervizora enerģijas patēriņš (W).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators;
- **max** - maksimāli pieļaujamais enerģijas patēriņš.

vmware.hv.sensor.health.state[url,uuid]

 VMware hipervizora veselības stāvokļa apkopojuma sensors.
 Atgrieztā vērtība: *Integer*: *0* - pelēks; *1* - zaļš; *2* - dzeltens; *3* - sarkans.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Ņemiet vērā, ka vienums var nedarboties VMware vSphere 6.5 un jaunākās versijās, jo VMware ir novecinājis *VMware Rollup Health State* sensoru.

vmware.hv.sensors.get[url,uuid]

 VMware hipervizora aparatūras piegādātāja stāvokļa sensori.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.status[url,uuid]

 VMware hipervizora statuss.
 Atgrieztā vērtība: *Integer*: *0* - pelēks; *1* - zaļš; *2* - dzeltens; *3* - sarkans.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Šis vienums izmanto hosta sistēmas vispārējā statusa īpašību.

vmware.hv.tags.get[url,uuid]

 VMware hipervizora birku masīvs.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

Šis vienums darbojas ar vSphere 6.5 un jaunākām versijām.

vmware.hv.uptime[url,uuid]

 VMware hipervizora darbības laiks (sekundēs).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware hipervizora globālais unikālais identifikators.

Šis vienums izmanto hosta sistēmas vispārējā statusa īpašību.

vmware.hv.version[url,uuid]

 VMware hipervizora versija.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.hv.vm.num[url,uuid]

 Virtuālo mašīnu skaits VMware hipervizorā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware hipervizora globāli unikālais identifikators.

vmware.rp.cpu.usage[url,rpid]

 CPU lietojums hercos intervāla laikā VMware resursu pūlā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **rpid** - VMware resursu pūla ID.

vmware.rp.memory[url,rpid,<mode>]

 VMware resursu kopas atmiņas metriskie dati.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **rpid** - VMware resursu kopas ID;
- **mode** - iespējamās vērtības:
consumed (noklusējums) - hosta fiziskās atmiņas apjoms, kas patērēts viesa fiziskās atmiņas lapu nodrošināšanai
 ballooned - viesa fiziskās atmiņas apjoms, ko no virtuālās mašīnas atguvis viesa balloon draiveris
overhead - hosta fiziskā atmiņa, ko patērē ESXi datu struktūras virtuālo mašīnu darbināšanai

vmware.alarms.geturl

 VMware Virtual Center trauksmju dati.
 Atgrieztā vērtība: JSON object.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.vm.alarms.get[url,uuid]

 VMware virtuālās mašīnas trauksmju dati.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.attribute[url,uuid,name]

 VMware virtuālās mašīnas pielāgotā atribūta vērtība.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **name** - pielāgotā atribūta nosaukums.

vmware.vm.cluster.name[url,uuid]

 VMware virtuālās mašīnas nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.consolidationneeded[url,uuid]

 VMware virtuālās mašīnas diskam ir nepieciešama konsolidācija.
 Atgrieztā vērtība: *String*: *true* - konsolidācija ir nepieciešama; *false* - konsolidācija nav nepieciešama.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.cpu.latency[url,uuid]

 Laika procentuālā daļa, kurā virtuālā mašīna nevar darboties, jo tā konkurē par piekļuvi fiziskajam(-iem) CPU.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.cpu.num[url,uuid]

 Procesoru skaits VMware virtuālajā mašīnā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.cpu.readiness[url,uuid,<instance>]

 Laika procentuālā daļa, kurā virtuālā mašīna bija gatava, bet nevarēja tikt ieplānota izpildei uz fiziskā CPU.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - CPU instance.

vmware.vm.cpu.ready[url,uuid]

 Laiks (milisekundēs), kurā virtuālā mašīna bija gatava darbam, bet nevarēja tikt ieplānota izpildei uz fiziskā CPU. CPU gatavības laiks ir atkarīgs no virtuālo mašīnu skaita hostā un to CPU slodzes (%).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.cpu.swapwait[url,uuid,<instance>]

 CPU laika procentuālā daļa, kas pavadīta, gaidot ielādi no mijmaiņas atmiņas.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators;
- **instance** - CPU instance.

vmware.vm.cpu.usage[url,uuid]

 VMware virtuālās mašīnas procesora lietojums (Hz).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.cpu.usage.perf[url,uuid]

 VMware virtuālās mašīnas procesora izmantojums procentos intervāla laikā.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.datacenter.name[url,uuid]

 VMware virtuālās mašīnas datu centra nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.discoveryurl

 VMware virtuālo mašīnu atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL.

vmware.vm.guest.memory.size.swapped[url,uuid]

 Viesa fiziskās atmiņas apjoms, kas ir izmainīts uz mijmaiņas vietu (KB).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.guest.osuptime[url,uuid]

 Kopējais laiks, kas pagājis kopš pēdējās operētājsistēmas palaišanas (sekundēs).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.hv.maintenance[url,uuid]

 VMware virtuālās mašīnas hipervizora uzturēšanas statuss.
 Atgrieztā vērtība: *0* - nav uzturēšanas režīmā; *1* - uzturēšanas režīmā.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.hv.name[url,uuid]

 VMware virtuālās mašīnas hipervizora nosaukums.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size[url,uuid]

 VMware virtuālās mašīnas kopējais atmiņas apjoms (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.ballooned[url,uuid]

 VMware virtuālās mašīnas ballooned atmiņas lielums (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.compressed[url,uuid]

 VMware virtuālās mašīnas saspiestās atmiņas izmērs (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.consumed[url,uuid]

 Hosta fiziskās atmiņas apjoms, kas patērēts viesa fiziskās atmiņas lapu dublēšanai (KB).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.private[url,uuid]

 VMware virtuālās mašīnas privātās atmiņas izmērs (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.memory.size.shared[url,uuid]

 VMware virtuālās mašīnas koplietotās atmiņas lielums (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.swapped[url,uuid]

 VMware virtuālās mašīnas samainītās atmiņas lielums (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.usage.guest[url,uuid]

 VMware virtuālās mašīnas viesa atmiņas lietojums (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.size.usage.host[url,uuid]

 VMware virtuālās mašīnas hosta atmiņas lietojums (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.memory.usage[url,uuid]

 Hosta fiziskās atmiņas izmantotais procentuālais apjoms.
 Atgrieztā vērtība: *Float*.

Parametri:

- **url** — VMware pakalpojuma URL;
- **uuid** — VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.net.if.discovery[url,uuid]

 VMware virtuālās mašīnas tīkla saskarņu atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.net.if.in[url,uuid,instance,<mode>]

 VMware virtuālās mašīnas tīkla saskarnes ienākošās statistikas dati (baiti/paketes sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - tīkla saskarnes instance;
- **mode** - *bps* (noklusējums) vai *pps* - baiti vai paketes sekundē.

vmware.vm.net.if.out[url,uuid,instance,<mode>]

 VMware virtuālās mašīnas tīkla saskarnes izejošās statistikas dati (baiti/paketes sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - tīkla saskarnes instance;
- **mode** - *bps* (noklusējums) vai *pps* - baiti vai paketes sekundē.

vmware.vm.net.if.usage[url,uuid,<instance>]

 VMware virtuālās mašīnas tīkla izmantojums (apvienotie pārraides un saņemšanas ātrumi) intervāla laikā (KBps).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - tīkla saskarnes instance.

vmware.vm.perfcounter[url,uuid,path,<instance>]

 VMware virtuālās mašīnas veiktspējas skaitītāja vērtība.
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **path** - veiktspējas skaitītāja ceļš [1];
- **instance** - veiktspējas skaitītāja instance. Apkopotajām vērtībām izmantojiet tukšu instanci (noklusējums).

vmware.vm.powerstate[url,uuid]

 VMware virtuālās mašīnas barošanas stāvoklis.
 Atgrieztā vērtība: *0* - poweredOff; *1* - poweredOn; *2* - suspended.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.property[url,uuid,prop]

 VMware virtuālās mašīnas īpašība.
 Atgrieztā vērtība: *String*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **prop** - īpašības ceļš.

vmware.vm.snapshot.get[url,uuid]

 VMware virtuālās mašīnas momentuzņēmuma stāvoklis.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.state[url,uuid]

 VMware virtuālās mašīnas stāvoklis.
 Atgrieztā vērtība: *String*: *notRunning*, *resetting*, *running*, *shuttingDown*, *standby* vai *unknown*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.storage.committed[url,uuid]

 VMware virtuālās mašīnas aizņemtā krātuves vieta (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.storage.readoio[url,uuid,instance]

 Vidējais neizpildīto lasīšanas pieprasījumu skaits virtuālajam diskam datu vākšanas intervāla laikā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance.

vmware.vm.storage.totalreadlatency[url,uuid,instance]

 Vidējais laiks, kas nepieciešams lasīšanai no virtuālā diska (milisekundēs).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance.

vmware.vm.storage.totalwritelatency[url,uuid,instance]

 Vidējais laiks, kas nepieciešams rakstīšanai virtuālajā diskā (milisekundēs).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance.

vmware.vm.storage.uncommitted[url,uuid]

 VMware virtuālās mašīnas neapstiprinātā krātuves vieta (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.storage.unshared[url,uuid]

 VMware virtuālās mašīnas nekoplietotās krātuves vieta (baitos).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globālais unikālais identifikators.

vmware.vm.storage.writeoio[url,uuid,instance]

 Vidējais neizpildīto rakstīšanas pieprasījumu skaits uz virtuālo disku datu vākšanas intervāla laikā.
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance.

vmware.vm.tags.get[url,uuid]

 VMware virtuālās mašīnas tagu masīvs.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

Šis vienums darbojas ar vSphere 6.5 un jaunākām versijām.

vmware.vm.tools[url,uuid,<mode>]

 VMware virtuālās mašīnas viesa rīku stāvoklis.
 Atgrieztā vērtība: *String*.
 status režīmā: *guestToolsExecutingScripts* - VMware Tools tiek palaists; *guestToolsNotRunning* - VMware Tools nedarbojas; *guestToolsRunning* - VMware Tools darbojas.
 version režīmā: versija.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **mode** - *version*, *status* (noklusējums).

vmware.vm.uptime[url,uuid]

 VMware virtuālās mašīnas darbības laiks (sekundēs).
 Atgrieztā vērtība: *Integer*.

Parametri:

- **url** - VMware pakalpojuma URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.vfs.dev.discovery[url,uuid]

 VMware virtuālās mašīnas disku ierīču atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

vmware.vm.vfs.dev.read[url,uuid,instance,<mode>]

 VMware virtuālās mašīnas diska ierīces lasīšanas statistika (baiti/operācijas sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance;
- **mode** - *bps* (noklusējums) vai *ops* - baiti vai operācijas sekundē.

vmware.vm.vfs.dev.write[url,uuid,instance,<mode>]

 VMware virtuālās mašīnas diska ierīces rakstīšanas statistika (baiti/operācijas sekundē).
 Atgrieztā vērtība: *Integer* [2].

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **instance** - diska ierīces instance;
- **mode** - *bps* (noklusējums) vai *ops* - baiti vai operācijas sekundē.

vmware.vm.vfs.fs.discovery[url,uuid]

 VMware virtuālās mašīnas failu sistēmu atklāšana.
 Atgrieztā vērtība: JSON objekts.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators.

Lai šis vienums darbotos, viesu virtuālajā mašīnā jābūt instalētiem VMware Tools.

vmware.vm.vfs.fs.size[url,uuid,fsname,<mode>]

 VMware virtuālās mašīnas failu sistēmas statistika (baiti/procenti).
 Atgrieztā vērtība: *Integer* - baitiem; *Float* - procentiem.

Parametri:

- **url** - VMware servisa URL;
- **uuid** - VMware virtuālās mašīnas globāli unikālais identifikators;
- **fsname** - failu sistēmas nosaukums;
- **mode** - *total*, *free*, *used*, *pfree* vai *pused*.

Lai šis vienums darbotos, viesu virtuālajā mašīnā jābūt instalētiem VMware Tools.

Piezīmes

[1] Skatiet Pielāgotu VMware veiktspējas skaitītāju nosaukumu izveide.

[2] Šo vienumu vērtība tiek iegūta no VMware veiktspējas skaitītājiem, un VMwarePerfFrequency parametrs tiek izmantots, lai atsvaidzinātu to datus Zabbix VMware kešatmiņā:

- 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

Papildu informācija

Skatiet Virtuālo mašīnu uzraudzība, lai iegūtu detalizētu informāciju par to, kā konfigurēt Zabbix VMware vides uzraudzībai.

2 Virtuālās mašīnas atklāšanas atslēgu lauki

Šajā tabulā ir uzskaitīti lauki, ko atgriež ar virtuālajām mašīnām saistītās atklāšanas atslēgas.

Vienuma atslēga		
Apraksts	**Lauks**	**Iegūtais saturs**
vmware.cluster.discovery		
Veic klasteru atklāšanu.	{#CLUSTER.ID}	Klastera identifikators.
	{#CLUSTER.NAME}	Klastera nosaukums.

	”resource_pool”	Masīvs, kas satur resursu pūla datus, tostarp resursu grupas ID, tagu masīvu, resursu pūla ceļu un virtuālo mašīnu skaitu.
		
		Masīva struktūra:
		`[{`
		`"rpid":"resursu grupas id",`
		`"tags":[{}],`
		`"rpath":"resursu grupas ceļš",`
		`"vm_count":0`
		`}]`

	”tags”	`"tags"` masīva struktūru skatiet laukā ”tags”.
		Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju.
		
		Masīva struktūra:
		`[{`
		`"tag":"taga nosaukums",`
		`"tag_description":"taga apraksts",`
		`"category":"taga kategorija"`
		`}]`

vmware.datastore.discovery		
Veic datu krātuvju atklāšanu.	{#DATASTORE}	Datu krātuves nosaukums.

	{#DATASTORE.EXTENT}	Masīvs, kas satur datu krātuves extent nodalījuma ID un instances nosaukumu.
		
		Masīva struktūra:
		`[{`
		`"partitionid":1,`
		`"instance":"nosaukums"`
		`}]`

	{#DATASTORE.TYPE}	Datu krātuves tips. Vērtību piemēri: `VMFS`, `NFS`, `vsan` u.c.
	{#DATASTORE.UUID}	Datu krātuves identifikators.

Vienuma atslēga		

	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju.
		
		Masīva struktūra:
		`[{`
		`"tag":"taga nosaukums",`
		`"tag_description":"taga apraksts",`
		`"category":"taga kategorija"`
		`}]`

vmware.dc.discovery		
Veic datu centru atklāšanu.	{#DATACENTER}	Datu centra nosaukums.
	{#DATACENTERID}	Datu centra identifikators.

	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju.
		
		Masīva struktūra:
		`[{`
		`"tag":"taga nosaukums",`
		`"tag_description":"taga apraksts",`
		`"category":"taga kategorija"`
		`}]`

vmware.dvswitch.discovery		
Veic vSphere izkliedēto komutatoru atklāšanu.	{#DVS.NAME}	Komutatora nosaukums.
	{#DVS.UUID}	Komutatora identifikators.
vmware.hv.discovery		
Veic hipervizoru atklāšanu.	{#HV.UUID}	Unikāls hipervizora identifikators.
	{#HV.ID}	Hipervizora identifikators (HostSystem pārvaldītā objekta nosaukums).
	{#HV.NAME}	Hipervizora nosaukums.
	{#HV.NETNAME}	Hipervizora tīkla hosta nosaukums.
	{#HV.IP}	Hipervizora IP adrese; var būt tukša. HA konfigurācijas gadījumā ar vairākām tīkla saskarnēm tiek ievērota šāda saskarnes atlases prioritāte: - priekšroka tiek dota IP adresei, kas atrodas tajā pašā IP apakštīklā kā vCenter IP; - priekšroka tiek dota IP adresei no IP apakštīkla ar noklusējuma vārteju; - priekšroka tiek dota IP adresei no saskarnes ar mazāko ID.
	{#CLUSTER.NAME}	Klastera nosaukums; var būt tukšs.
	{#DATACENTER.NAME}	Datu centra nosaukums.
	{#PARENT.NAME}	Konteinera nosaukums, kurā tiek glabāts hipervizors.
	{#PARENT.TYPE}	Konteinera tips, kurā tiek glabāts hipervizors. Vērtības var būt `Datacenter`, `Folder`, `ClusterComputeResource`, `VMware`, kur ”VMware” apzīmē nezināmu konteinera tipu.

	”resource_pool”	Masīvs, kas satur resursu pūla datus, tostarp resursu grupas ID, tagu masīvu, resursu pūla ceļu un virtuālo mašīnu skaitu.
		
		Masīva struktūra:
		`[{`
		`"rpid":"resursu grupas id",`
		`"tags":[{}],`
		`"rpath":"resursu grupas ceļš",`
		`"vm_count":0`
		`}]`
		
		`"tags"` masīva struktūru skatiet laukā ”tags”.

Vienuma atslēga		
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"taga nosaukums", "tag_description":"taga apraksts", "category":"taga kategorija" }]
vmware.hv.datastore.discovery		
Veic hipervizora datu krātuvju atklāšanu. Ņemiet vērā, ka vienu un to pašu datu krātuvi var izmantot vairāki hipervizori.	{#DATASTORE}	Datu krātuves nosaukums.
	{#DATASTORE.TYPE}	Datu krātuves tips. Vērtību piemēri: VMFS, NFS, vsan u.c.
	{#DATASTORE.UUID}	Datu krātuves identifikators.
	{#MULTIPATH.COUNT}	Reģistrētais datu krātuves ceļu skaits.
	{#MULTIPATH.PARTITION.COUNT}	Diska nodalījumu skaits.
	”datastore_extent”	Masīvs, kas satur datu krātuves extent instances nosaukumu un nodalījuma ID. Masīva struktūra: [{ "partitionid":1, "instance":"nosaukums" }]
	”tags”	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju. Masīva struktūra: [{ "tag":"taga nosaukums", "tag_description":"taga apraksts", "category":"taga kategorija" }]
vmware.hv.net.if.discovery		
Veic hipervizora tīkla saskarņu atklāšanu.	{#IFNAME}	Saskarnes nosaukums.
	{#IFDRIVER}	Saskarnes draiveris.
	{#IFDUPLEX}	Saskarnes dupleksa iestatījumi.
	{#IFSPEED}	Saskarnes ātrums.
	{#IFMAC}	Saskarnes MAC adrese.
vmware.vm.discovery		
Veic virtuālo mašīnu atklāšanu.	{#VM.UUID}	Unikāls virtuālās mašīnas identifikators.
	{#VM.ID}	Virtuālās mašīnas identifikators (VirtualMachine pārvaldītā objekta nosaukums).
	{#VM.NAME}	Virtuālās mašīnas nosaukums.
	{#HV.NAME}	Hipervizora nosaukums.
	{#HV.UUID}	Unikāls hipervizora identifikators.
	{#HV.ID}	Hipervizora identifikators (HostSystem pārvaldītā objekta nosaukums).
	{#CLUSTER.NAME}	Klastera nosaukums; var būt tukšs.
	{#DATACENTER.NAME}	Datu centra nosaukums.
	{#DATASTORE.NAME}	Datu krātuves nosaukums.
	{#DATASTORE.UUID}	Datu krātuves identifikators.
	{#VM.IP}	Virtuālās mašīnas IP adrese; var būt tukša.
	{#VM.DNS}	Virtuālās mašīnas DNS nosaukums; var būt tukšs.
	{#VM.GUESTFAMILY}	Viesa virtuālās mašīnas OS saime; var būt tukša.
	{#VM.GUESTFULLNAME}	Pilns viesa virtuālās mašīnas OS nosaukums; var būt tukšs.

Vienuma atslēga		
	{#VM.FOLDER}	Virtuālās mašīnas vecākmapju ķēde; to var izmantot kā vērtību ligzdotām grupām; mapju nosaukumi tiek apvienoti ar "/". Var būt tukšs.
	{#VM.TOOLS.STATUS}	VMware virtuālās mašīnas rīku stāvoklis.
	{#VM.POWERSTATE}	VMware virtuālās mašīnas barošanas stāvoklis (`poweredOff`, `poweredOn` vai `suspended`).
	{#VM.RPOOL.ID}	Resursu pūla identifikators.
	{#VM.RPOOL.PATH}	Pilns resursu pūla ceļš, neiekļaujot "root" nosaukumu "Resources". Mapju nosaukumi tiek apvienoti ar "/".
	{#VM.SNAPSHOT.COUNT}	VM momentuzņēmumu skaits.

	"tags"	Masīvs, kas satur tagus ar taga nosaukumu, aprakstu un kategoriju.
		
		Masīva struktūra:
		`[{`
		`"tag":"taga nosaukums",`
		`"tag_description":"taga apraksts",`
		`"category":"taga kategorija"`
		`}]`

	"vm_customattribute"	Virtuālās mašīnas pielāgoto atribūtu masīvs (ja definēti).
		
		Masīva struktūra:
		`[{`
		`"name":"pielāgotā lauka nosaukums",`
		`"value":"pielāgotā lauka vērtība"`
		`}]`

	"net_if"	Virtuālās mašīnas tīkla saskarņu masīvs.
		
		Masīva struktūra:
		`[{`
		`"ifname": "saskarnes nosaukums",`
		`"ifdesc": "saskarnes apraksts",`
		`"ifmac": "00:00:00:00:00:00",`
		`"ifconnected": true,`
		`"iftype": "saskarnes tips",`
		`"ifbackingdevice": "saskarnes piesaistītā ierīce",`
		`"ifdvswitch_uuid": "saskarnes komutatora uuid",`
		`"ifdvswitch_portgroup": "saskarnes komutatora portu grupa",`
		`"ifdvswitch_port": "saskarnes komutatora ports",`
		`"ifip": ["saskarnes ip adreses"]`
		`}]`
		
		Atgriezto datu aprakstu skatiet vienuma atslēgā "vmware.vm.net.if.discovery".

vmware.vm.net.if.discovery		
Veic virtuālās mašīnas tīkla saskarnes atklāšanu.	{#IFNAME}	Tīkla saskarnes nosaukums.
	{#IFDESC}	Saskarnes apraksts.
	{#IFMAC}	Saskarnes MAC adrese.
	{#IFCONNECTED}	Saskarnes savienojuma statuss (`false` - atvienota; `true` - savienota).
	{#IFTYPE}	Saskarnes tips.
	{#IFBACKINGDEVICE}	Piesaistītās ierīces nosaukums.
	{#IFDVSWITCH.UUID}	Unikāls vSphere Distributed Switch identifikators.
	{#IFDVSWITCH.PORTGROUP}	Izkliedētā portu grupa.
	{#IFDVSWITCH.PORT}	vSphere Distributed Switch ports.
	"ifip"	Saskarnes adrešu masīvs.
vmware.vm.vfs.dev.discovery		
Veic virtuālās mašīnas diska ierīču atklāšanu.	{#DISKNAME}	Diska ierīces nosaukums.
vmware.vm.vfs.fs.discovery		

Vienuma atslēga		
Veic virtuālās mašīnas failu sistēmas atklāšanu.	{#FSNAME}	Failu sistēmas nosaukums.

3 JSON piemēri VMware vienumiem

Pārskats Šajā sadaļā ir sniegta papildu informācija par JSON objektiem, ko atgriež dažādi VMware vienumi.

vmware.*.alarms.get Vienumi **vmware.alarms.get[]**, **vmware.cluster.alarms.get[]**, **vmware.datastore.alarms.get[]**, **vmware.dc.alarms.get[]**, **vmware.hv.alarms.get[]**, **vmware.vm.alarms.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
{
    "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
        },
        {
            // other alarms
        }
    ]
}
```

vmware.*.tags.get Vienumi **vmware.cluster.tags.get[]**, **vmware.datastore.tags.get[]**, **vmware.dc.tags.get[]**, **vmware.hv.tags.get[]**, **vmware.vm.tags.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
{
    "tags": [
        {
            "name": "Windows",
        "description": "tags OS tipa kategorijai",
        "category": "OS tips"
    },
    {
            "name": "SQL Server",
        "description": "tags lietojumprogrammas nosaukuma kategorijai",
        "category": "lietojumprogrammas nosaukums"
        },
    {
            // citi tagi
        }
```

```
    ]
}
```

vmware.hv.diskinfo.get Vienums **vmware.hv.diskinfo.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
[
  {
    "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": {}
  },
  {
    // citas instances
  }
]
```

vmware.dvswitch.fetchports.get Vienums **vmware.dvswitch.fetchports.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
{
    "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"
            },
            {
                //citas atslēgas
            }
        ]
    }
}
```


vmware.hv.hw.sensors.get Vienums **vmware.hv.hw.sensors.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):


```
{
    "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"
                }
            },
            {
                // citi hv hw sensori
            }
        ]
        }
}
```

vmware.hv.sensors.get Vienums **vmware.hv.sensors.get[]** atgriež JSON objektus ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
{
    "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"
        },
                {
                    // citi hv sensori
                }
        ]
        }
}
```

vmware.vm.snapshot.get Ja eksistē kādi momentuzņēmumi, vienums **vmware.snapshot.get[]** atgriež JSON objektu ar šādu struktūru (vērtības ir sniegtas kā piemērs):

```
{
  "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
}
```


Ja neeksistē neviens momentuzņēmums, vienums **vmware.snapshot.get[]** atgriež JSON objektu ar tukšām vērtībām:


```
{
  "snapshot": [],
  "count": 0,
  "latestdate": null,
  "latestage": 0,
  "oldestdate": null,
  "oldestage": 0,
  "size": 0,
  "uniquesize": 0
}
```


4 VMware uzraudzības iestatīšanas piemērs

Pārskats

Šajā piemērā ir aprakstīts, kā iestatīt Zabbix VMware virtuālo mašīnu uzraudzībai. Tas ietver:

- hosta izveidi, kas attēlo jūsu VMware vidi;
- zema līmeņa atklāšanas kārtulas izveidi, kas atklāj virtuālās mašīnas jūsu VMware vidē;
- hosta prototipa izveidi, uz kura pamata Zabbix ģenerēs reālus hostus virtuālajām mašīnām, ko atklāj zema līmeņa atklāšanas kārtula.

Priekšnosacījumi

> **Note:**
> Šis piemērs neaptver VMware konfigurēšanu. Tiek pieņemts, ka VMware jau ir konfigurēts.

Pirms turpināt, iestatiet parametru `StartVMwareCollectors` Zabbix servera konfigurācijas failā uz 2 vai vairāk (noklusējuma vērtība ir 0).

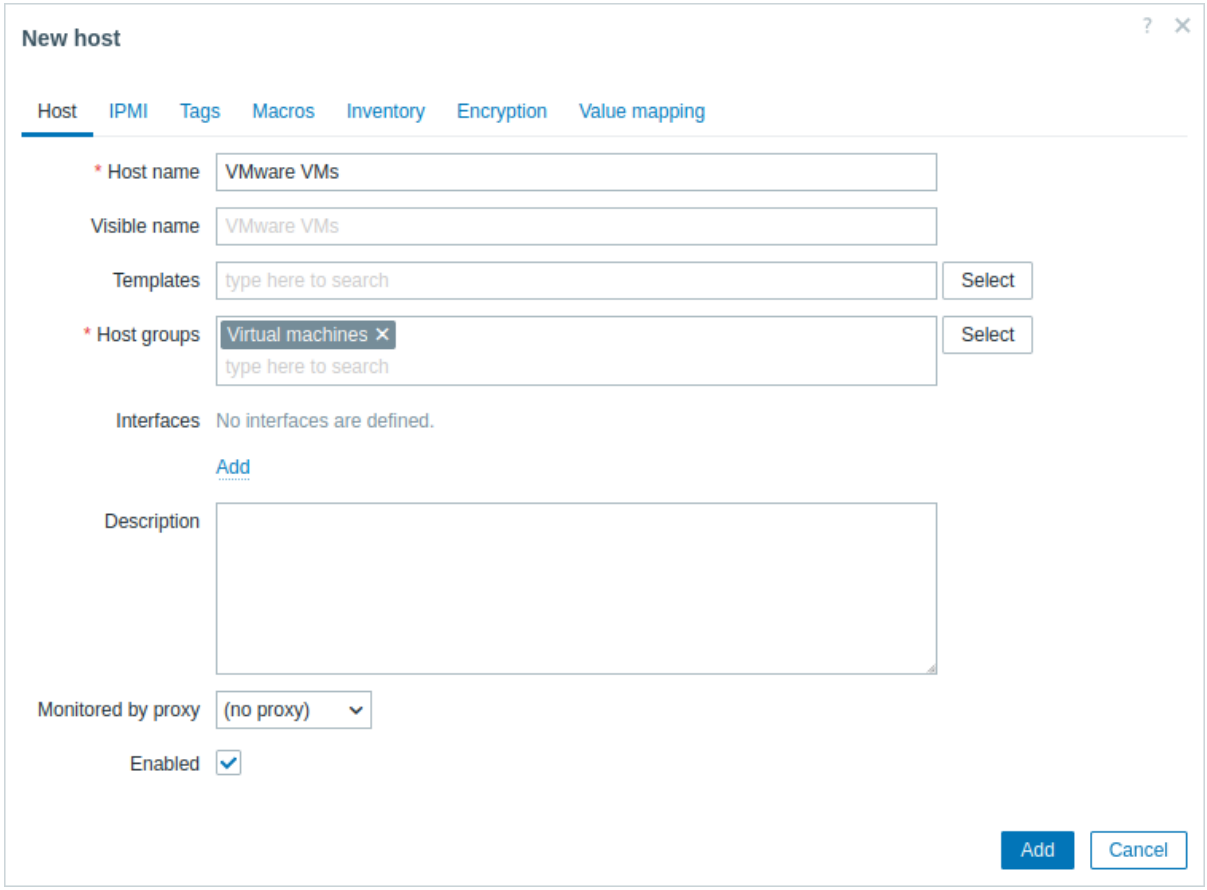
Izveidojiet hostu

1. Dodieties uz *Datu ievākšana* → *Hosti*.

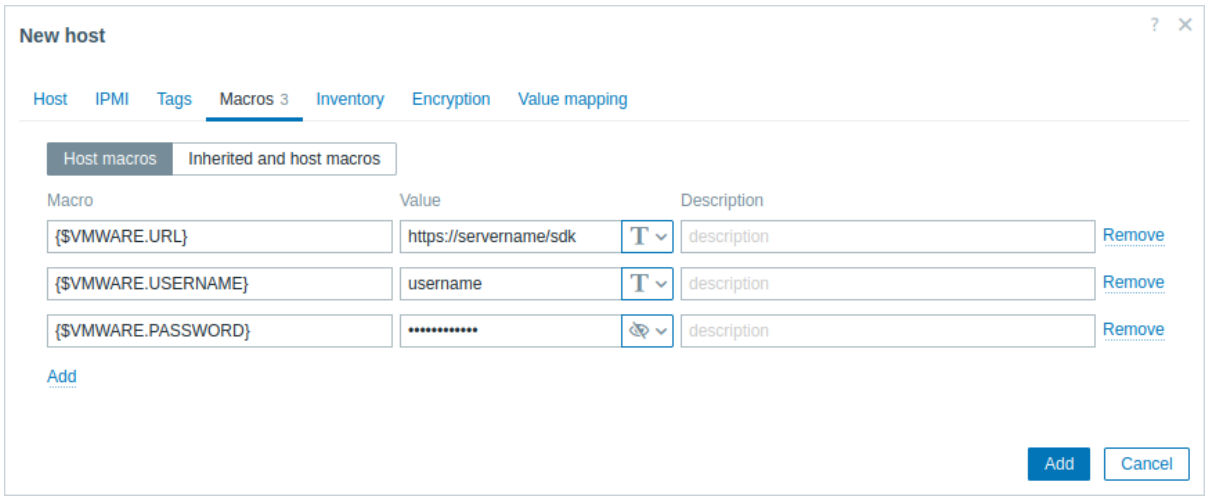
2. Izveidojiet hostu:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, "VMware VMs").

- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”Virtual machines”).



- Cilnē *Macros* iestatiet šādus hosta makrosus:
  - {$VMWARE.URL} - VMware pakalpojuma (ESXi hipervizora) SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware pakalpojuma lietotājvārds
  - {$VMWARE.PASSWORD} - VMware pakalpojuma lietotāja {$VMWARE.USERNAME} parole



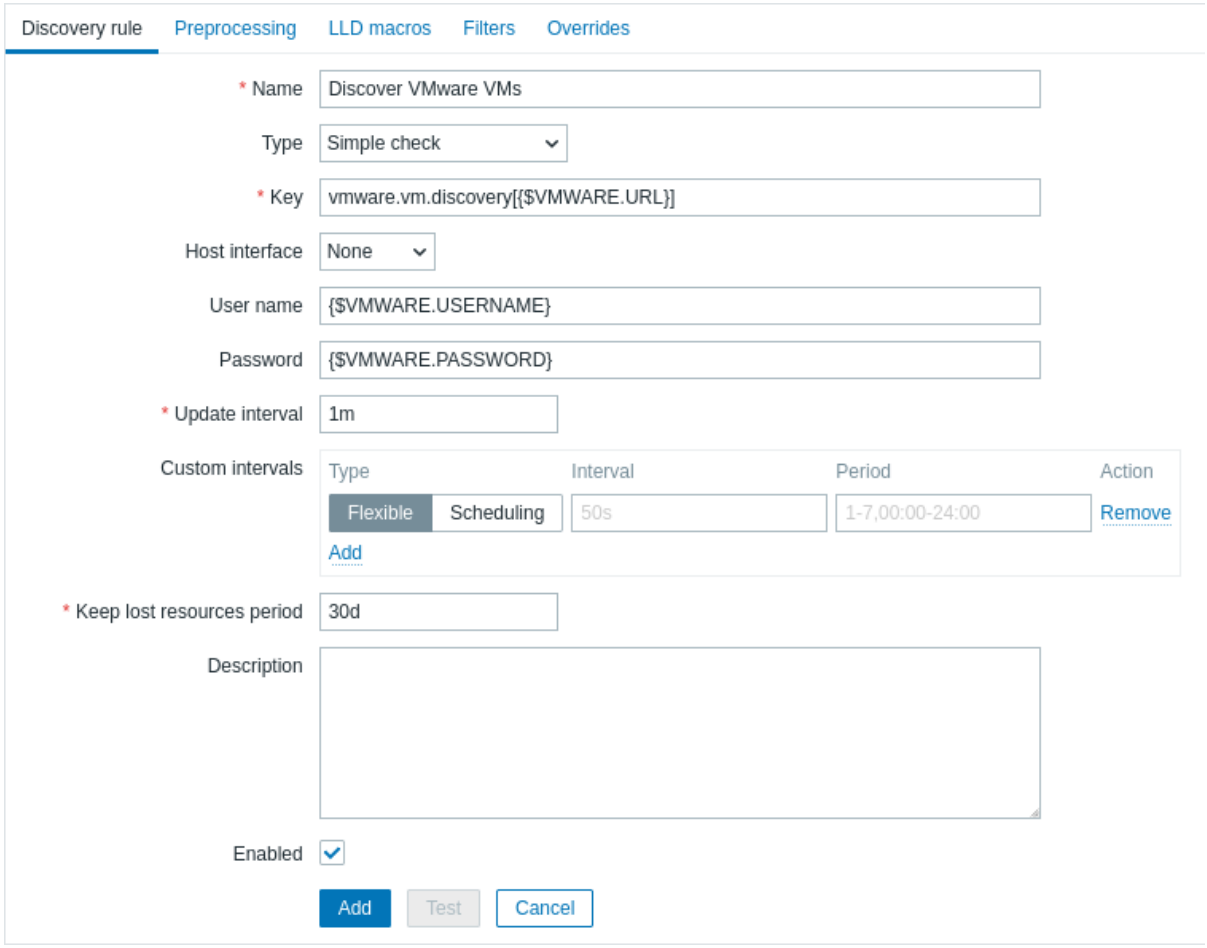
3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts attēlos jūsu VMware vidi.

Izveidojiet zema līmeņa atklāšanas kārtulu

1. Noklikšķiniet uz *Discovery* izveidotajam hostam, lai pārietu uz šī hosta zema līmeņa atklāšanas kārtulu sarakstu.

2. Izveidojiet zema līmeņa atklāšanas kārtulu:

- Laukā *Name* ievadiet zema līmeņa atklāšanas kārtulas nosaukumu (piemēram, ”Atklāt VMware VM”).
- Laukā *Type* atlasiet ”Simple check”.
- Laukā *Key* ievadiet iebūvēto vienuma atslēgu VMware virtuālo mašīnu atklāšanai: `vmware.vm.discovery[{$VMWARE.URL}]`
- Laukos *User name* un *Password* ievadiet atbilstošos makrosus, kas iepriekš konfigurēti hostā.



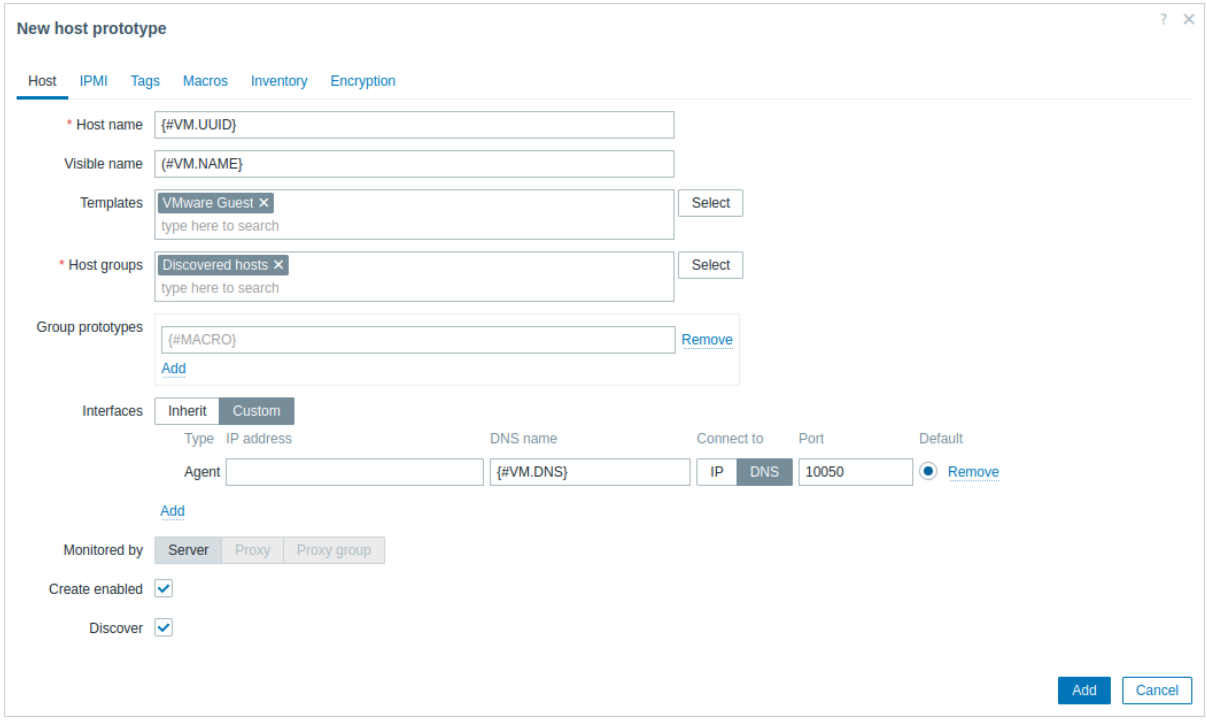
3. Noklikšķiniet uz pogas *Add*, lai izveidotu zema līmeņa atklāšanas kārtulu. Šī atklāšanas kārtula atklās virtuālās mašīnas jūsu VMware vidē.

Izveidot hosta prototipu

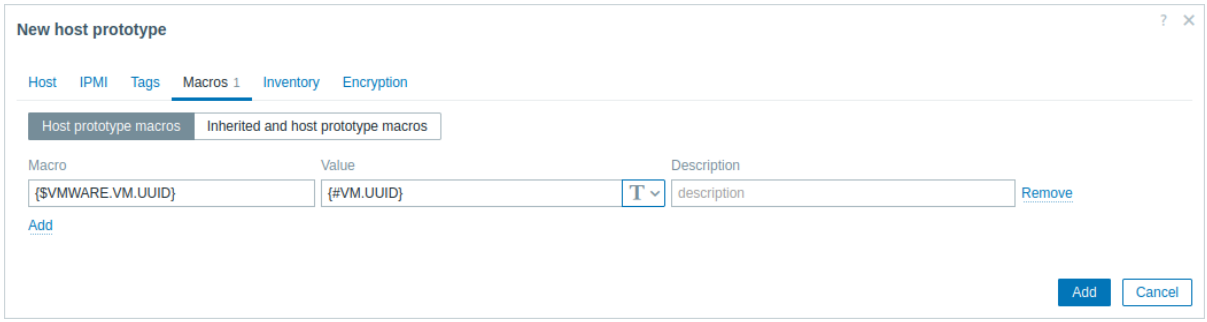
1. Zema līmeņa atklāšanas noteikumu sarakstā noklikšķiniet uz *Host prototypes* iepriekš izveidotajam zema līmeņa atklāšanas noteikumam.

2. Izveidojiet hosta prototipu. Tā kā hostu prototipi ir paraugi hostu izveidei, izmantojot zema līmeņa atklāšanas noteikumus, lielākajā daļā lauku būs zema līmeņa atklāšanas makrosi. Tas nodrošina, ka hosti tiek izveidoti ar īpašībām, kuru pamatā ir iegūtais saturs, ko atgrieza iepriekš izveidotais zema līmeņa atklāšanas noteikums.

- Laukā *Host name* ievadiet makrosu {#VM.UUID}.
- Laukā *Visible name* ievadiet makrosu {#VM.NAME}.
- Laukā *Templates* ievadiet vai atlasiet veidni "VMware Guest". Šī veidne satur VMware vienumus un atklāšanas noteikumus virtuālās mašīnas barošanas stāvokļa, CPU lietojuma, atmiņas lietojuma, tīkla ierīču u.c. uzraudzībai.
- Laukā *Host groups* ievadiet vai atlasiet hostu grupu (piemēram, "Discovered hosts").
- Laukā *Interfaces* pievienojiet pielāgotu hosta saskarni. Pēc tam laukā *DNS name* ievadiet makrosu {#VM.DNS} vai laukā *IP address* ievadiet makrosu {#VM.IP}. Alternatīvi, ja jūsu VMware vides virtuālajām mašīnām ir vairākas saskarnes, pārejiet uz sadaļu *Paplašinātā hosta saskarnes konfigurācija*. Pielāgotas hosta saskarnes konfigurēšana ir nepieciešama veidnes *VMware Guest* pareizai darbībai.



- Cilnē *Macros* iestatiet makrosu {$VMWARE.VM.UUID} ar vērtību {#VM.UUID}. Tas ir nepieciešams veidnes *VMware Guest* pareizai darbībai, jo tā izmanto šo makrosu kā hosta līmeņa lietotāja makrosu vienumu parametros (piemēram, *vmware.vm.net.if.discovery[{$VMWARE.URL}, {$VMWARE.VM.UUID}]*).

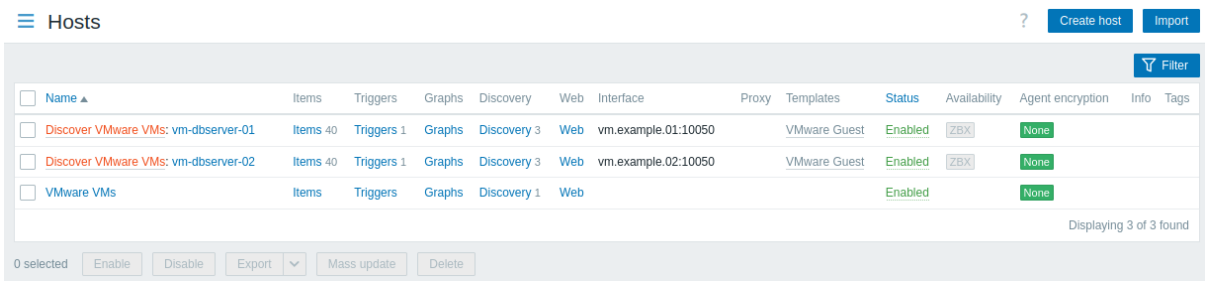


3. Noklikšķiniet uz pogas *Add*, lai izveidotu hosta prototipu. Šis hosta prototips tiks izmantots, lai izveidotu hostus virtuālajām mašīnām, ko atklāja iepriekš izveidotais zema līmeņa atklāšanas noteikums.

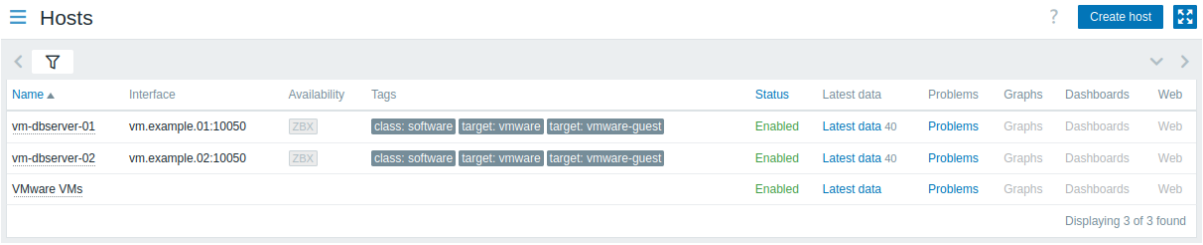
Skatīt hostus un metriku

Pēc hosta prototipa izveides zema līmeņa atklāšanas kārtula izveidos hostus atrastajām VMware virtuālajām mašīnām, un Zabbix sāks tās uzraudzīt. Ņemiet vērā, ka hostu atklāšanu un izveidi vajadzības gadījumā var arī izpildīt manuāli.

Lai skatītu izveidotos hostus, dodieties uz izvēlnes sadaļu *Datu ievākšana → Hosti*.



Lai skatītu ievākto metriku, dodieties uz izvēlnes sadaļu *Uzraudzība → Hosti* un vienam no hostiem noklikšķiniet uz *Jaunākie dati*.



Paplašināta hosta saskarnes konfigurācija

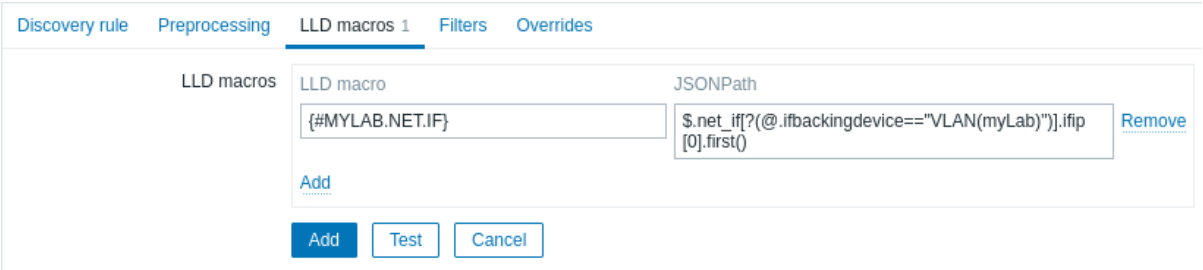
`vmware.vm.discovery[{$VMWARE.URL}]` vienuma atslēga, kas konfigurēta sadaļā *Izveidot zema līmeņa atklāšanas kārtulu*, atgriež tīkla saskarņu datus laukā "net_if":

```
"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": []
      }
    ]
```

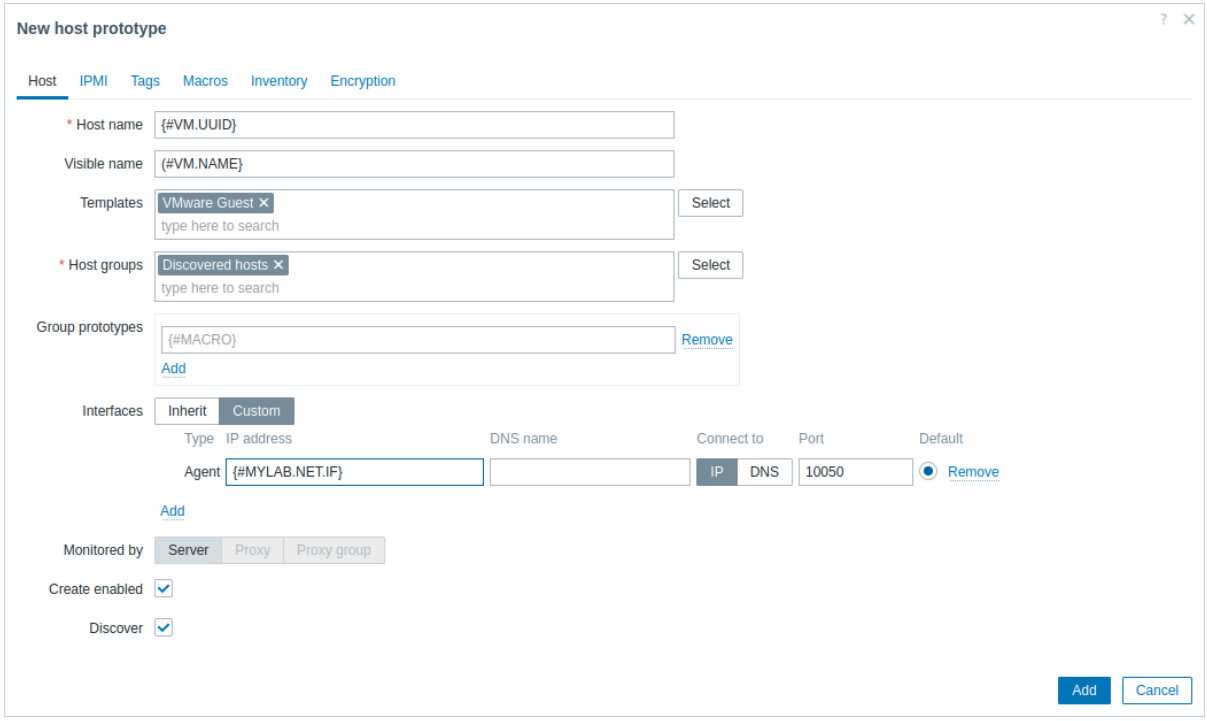
Šos datus var izmantot pielāgotas hosta saskarnes konfigurēšanai.

1. Veidojot zema līmeņa atklāšanas kārtulu, papildus konfigurējiet zema līmeņa atklāšanas makrosu. Cilnē *LLD macros* izveidojiet pielāgotu LLD makrosu ar JSONPath vērtību. Piemēram:

- `{#MYLAB.NET.IF}` - `$.net_if[?(@.ifbackingdevice=="VLAN(myLab)")].ifip[0].first()`



2. Veidojot hosta prototipu, pievienojiet pielāgotu hosta saskarni un ievadiet LLD makrosu laukā *DNS name* vai *IP address*.



9 Uzturēšana

Pārskats Jūs varat definēt uzturēšanas periodus hostiem un hostu grupām Zabbix.

Turklāt ir iespējams definēt uzturēšanu tikai vienam trigerim (vai trigeru apakškopai), norādot trigeru tagus. Šajā gadījumā uzturēšana tiks aktivizēta tikai šiem trigeriem; visi pārējie hosta vai hostu grupas trigeri nebūs uzturēšanas režīmā.

Ir divi uzturēšanas veidi — ar datu vākšanu un bez datu vākšanas.

Uzturēšanas laikā "ar datu vākšanu" trigeri tiek apstrādāti kā parasti, un notikumi tiek izveidoti, kad tas ir nepieciešams. Tomēr problēmu eskalācijas hostiem/trigeriem uzturēšanas režīmā tiek apturētas, ja darbības konfigurācijā ir atzīmēta opcija *Pause operations for suppressed problems*. Šajā gadījumā eskalācijas soļi, kas var ietvert paziņojumu sūtīšanu vai attālinātās komandas, tiks ignorēti tik ilgi, kamēr turpinās uzturēšanas periods. Ņemiet vērā, ka problēmu atkopšanas un atjaunināšanas darbības uzturēšanas laikā netiek apspiestas, tiek apturētas tikai eskalācijas.

Piemēram, ja eskalācijas soļi ir ieplānoti 0, 30 un 60 minūtes pēc problēmas sākuma un ir pusstundu ilgs uzturēšanas periods no 10. līdz 40. minūtei pēc reālas problēmas rašanās, otrais un trešais solis tiks izpildīts pusstundu vēlāk, tas ir, pēc 60 un 90 minūtēm (ja problēma joprojām pastāv). Līdzīgi, ja problēma rodas uzturēšanas laikā, eskalācija sāksies pēc uzturēšanas beigām.

Lai uzturēšanas laikā saņemtu problēmu paziņojumus parastajā veidā (bez aizkaves), darbības konfigurācijā ir jānoņem atzīme opcijai *Pause operations for suppressed problems*.

Note:
Ja vismaz viens hosts (izmantots trigera izteiksmē) nav uzturēšanas režīmā, Zabbix nosūtīs problēmas paziņojumu.

Zabbix serverim uzturēšanas laikā ir jādarbojas. Uzturēšanas periodi tiek pārrēķināti katru minūti vai arī uzreiz pēc konfigurācijas kešatmiņas pārlādes, ja uzturēšanas periodā ir veiktas izmaiņas.

Taimera procesi pārbauda, vai hosta statuss ir jāmaina uz/no uzturēšanas režīma katras minūtes 0. sekundē. Papildus tam taimera process katru sekundi pārbauda, vai kādi uzturēšanas periodi ir jāsāk/jāpārtrauc, pamatojoties uz to, vai pēc konfigurācijas atjaunināšanas ir izmaiņas [maintenance periods]. Tādējādi uzturēšanas periodu sākšanas/pārtraukšanas ātrums ir atkarīgs no konfigurācijas update interval (pēc noklusējuma 10 sekundes). Ņemiet vērā, ka uzturēšanas perioda izmaiņas neietver iestatījumus *Active since/Active till*. Tāpat, ja hosts/hostu grupa tiek pievienota esošam aktīvam uzturēšanas periodam, izmaiņas taimera process aktivizēs tikai nākamās minūtes sākumā.

Ņemiet vērā, ka, hostam ieejot uzturēšanas režīmā, Zabbix servera taimera procesi nolasīs visas atvērtās problēmas, lai pārbaudītu, vai tās ir jāapspiež. Tas var ietekmēt veiktspēju, ja ir daudz atvērtu problēmu. Zabbix serveris startēšanas laikā arī nolasīs visas atvērtās problēmas, pat ja tajā brīdī nav konfigurētu uzturēšanas periodu.

Ņemiet vērā, ka Zabbix serveris (vai starpniekserveris) vienmēr vāc datus neatkarīgi no uzturēšanas veida (ieskaitot uzturēšanu ”bez datiem”). Vēlāk serveris šos datus ignorē, ja ir iestatīta opcija ”bez datu vākšanas”.

Kad beidzas uzturēšana ”bez datiem”, trigeri, kas izmanto funkciju nodata(), neaktivizēsies pirms nākamās pārbaudes tajā periodā, kuru tie pārbauda.

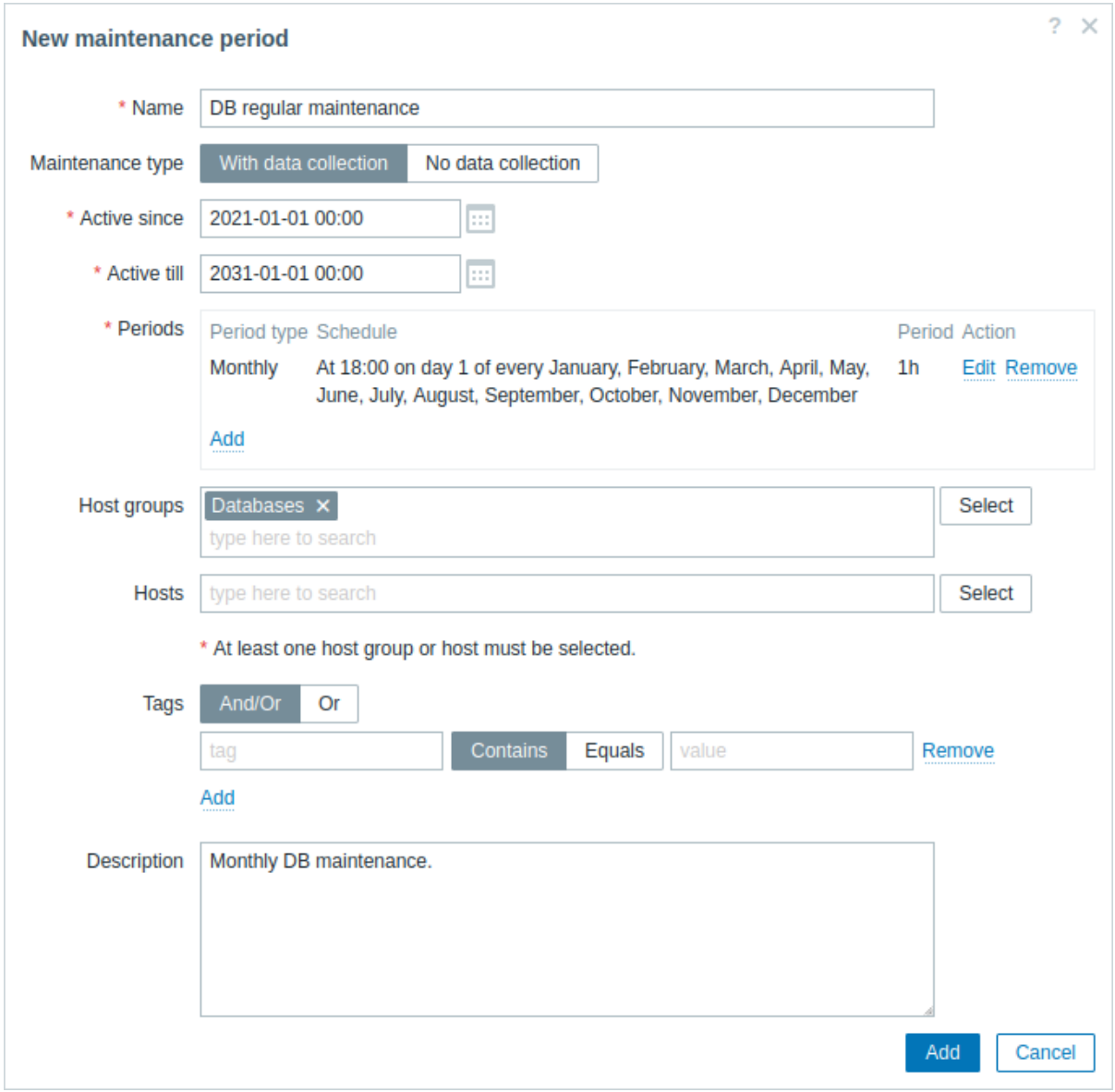
Ja žurnāla vienums tiek pievienots, kamēr hosts ir uzturēšanas režīmā, un uzturēšana beidzas, tiks savākti tikai jaunie žurnālfaila ieraksti kopš uzturēšanas beigām.

Ja hostam, kas atrodas uzturēšanas veidā “bez datiem”, tiek nosūtīta vērtība ar laika zīmogu (piemēram, izmantojot Zabbix sender), tad šī vērtība tiks atmesta, tomēr ir iespējams nosūtīt vērtību ar laika zīmogu uzturēšanas periodam, kas jau ir beidzies, un tā tiks pieņemta.

Ja lietotājs maina uzturēšanas periodu, hostus, grupas vai tagus, izmaiņas stāsies spēkā tikai pēc konfigurācijas kešatmiņas sinhronizācijas.

Konfigurācija Lai konfigurētu apkopes periodu:

1. Dodieties uz: *Datu ievākšana* > *Apkope*.
2. Noklikšķiniet uz *Izveidot apkopes periodu* (vai uz esoša apkopes perioda nosaukuma).
3. Ievadiet apkopes parametrus formā.



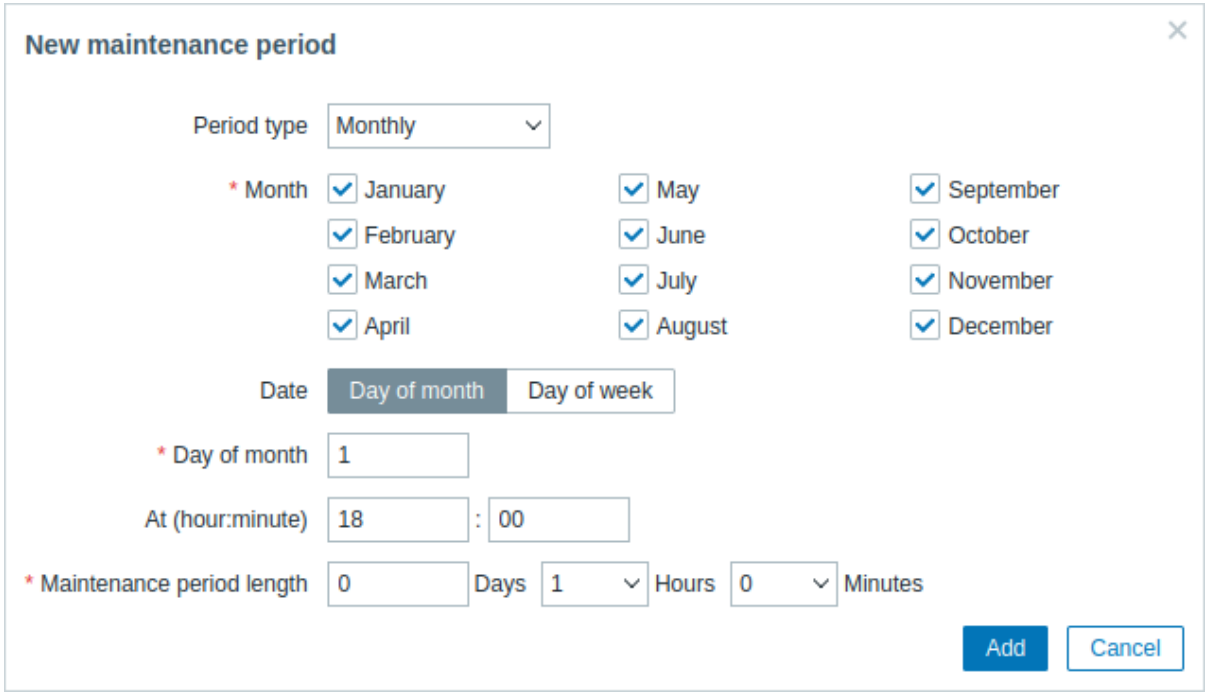
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Apkopes perioda nosaukums.

Parametrs	Apraksts
Maintenance type	Var iestatīt divus apkopes veidus: **Ar datu ievākšanu** — datus apkopes laikā ievāks serveris, trigeri tiks apstrādāti; **Bez datu ievākšanas** — datus apkopes laikā serveris neievāks. Skatiet Apkopes periodu ietekme, lai uzzinātu, kā katrs veids ietekmē pieejamības atskaiti.
Active since	Datums un laiks, no kura apkopes periodu izpilde kļūst aktīva. *Piezīme:* tikai šī laika iestatīšana neaktivizē apkopes periodu; apkopes periodi ir jākonfigurē sadaļā *Periods* (skatiet zemāk).
Active till	Datums un laiks, līdz kuram apkopes periodu izpilde ir aktīva.
Periods	Šis bloks ļauj definēt precīzas dienas un stundas, kad notiek apkope. Noklikšķinot uz Add, tiek atvērts uznirstošais logs ar elastīgu *Maintenance period* formu, kurā var definēt apkopes grafiku. Detalizētu aprakstu skatiet sadaļā Apkopes periodi.
Host groups	Atlasiet hostu grupas, kurām tiks aktivizēta apkope. Apkope tiks aktivizēta visiem hostiem no norādītās(-ajām) hostu grupas(-ām). Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot tajā rakstīt, tiks parādīts nolaižamais saraksts ar visām pieejamajām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Tādējādi apkope tiks aktivizēta arī hostiem no apakšgrupām.
Hosts	Atlasiet hostus, kuriem tiks aktivizēta apkope. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot tajā rakstīt, tiks parādīts nolaižamais saraksts ar visiem pieejamajiem hostiem.
Tags	Norādiet birkas, lai apspiestu problēmas hostiem apkopes režīmā ar atbilstošām birkām. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami divi operatori: **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, reģistrjutīga); **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi). Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Or nosacījuma; **Or** — pietiek, ja ir izpildīts viens nosacījums. Birkas var norādīt tikai tad, ja ir atlasīts režīms *Ar datu ievākšanu*.
Description	Apkopes perioda apraksts.

Uzturēšanas periodi

Uzturēšanas perioda logs ir paredzēts laika plānošanai periodiskai vai vienreizējai uzturēšanai. Forma ir dinamiska, un pieejamie lauki mainās atkarībā no izvēlētā *Period type*.



Period type	Apraksts
One time only	Konfigurējiet vienreizēju uzturēšanas periodu: *Date* - uzturēšanas perioda datums un laiks; *Maintenance period length* - cik ilgi uzturēšana būs aktīva.
Daily	Konfigurējiet ikdienas uzturēšanas periodu: *Every day(s)* - uzturēšanas biežums (1 - *(noklusējums)* katru dienu, 2 - ik pēc divām dienām utt.); *At (hour:minute)* - diennakts laiks, kad uzturēšana sākas; *Maintenance period length* - cik ilgi uzturēšana būs aktīva. Ja parametrs *Every day(s)* ir lielāks par ”1”, sākuma diena ir tā diena, kurā ietilpst *Active since* laiks. Piemēri: - ja *Active since* ir iestatīts uz ”2021-01-01 12:00”, *Every day(s)* ir iestatīts uz ”2” un *At (hour:minute)* ir iestatīts uz ”23:00”, tad pirmais uzturēšanas periods sāksies 1. janvārī plkst. 23:00, bet otrais uzturēšanas periods sāksies 3. janvārī plkst. 23:00; - ja *Active since* ir iestatīts uz ”2021-01-01 12:00”, *Every day(s)* ir iestatīts uz ”2” un *At (hour:minute)* ir iestatīts uz ”01:00”, tad pirmais uzturēšanas periods sāksies 3. janvārī plkst. 01:00, bet otrais uzturēšanas periods sāksies 5. janvārī plkst. 01:00.
Weekly	Konfigurējiet iknedēļas uzturēšanas periodu: *Every week(s)* - uzturēšanas biežums (1 - *(noklusējums)* katru nedēļu, 2 - ik pēc divām nedēļām utt.); *Day of week* - kurā nedēļas dienā uzturēšanai jānotiek; *At (hour:minute)* - diennakts laiks, kad uzturēšana sākas; *Maintenance period length* - cik ilgi uzturēšana būs aktīva. Ja parametrs *Every week(s)* ir lielāks par ”1”, sākuma nedēļa ir tā nedēļa, kurā ietilpst *Active since* laiks. Piemērus skatiet iepriekš parametra *Daily* aprakstā.
Monthly	Konfigurējiet ikmēneša uzturēšanas periodu: *Month* - atlasiet visus mēnešus, kuros tiek veikta regulārā uzturēšana; *Date:* ***Day of month*** - atlasiet šo opciju, ja uzturēšanai jānotiek vienā un tajā pašā datumā katru mēnesi (piemēram, katra mēneša 1. datumā), un pēc tam parādītajā laukā *Day of month* atlasiet nepieciešamo dienu; *Date:* ***Day of week*** - atlasiet šo opciju, ja uzturēšanai jānotiek tikai noteiktās dienās (piemēram, katra mēneša pirmajā pirmdienā), pēc tam nolaižamajā sarakstā atlasiet nepieciešamo mēneša nedēļu (pirmo, otro, trešo, ceturto vai pēdējo) un pēc tam atzīmējiet uzturēšanas dienu(-as); *At (hour:minute)* - diennakts laiks, kad uzturēšana sākas; *Maintenance period length* - cik ilgi uzturēšana būs aktīva.

Attention:
Veidojot uzturēšanas periodu, tiek izmantota tā lietotāja laika josla, kurš to izveido. Tomēr, plānojot periodiski atkārtojošos uzturēšanas periodus (*Katru dienu*, *Katru nedēļu*, *Katru mēnesi*), tiek izmantota Zabbix servera laika josla. Lai nodrošinātu paredzamu periodiski atkārtojošos uzturēšanas periodu darbību, visās Zabbix daļās ir jāizmanto vienota laika josla.

Kad pabeigts, nospiediet *Pievienot*, lai pievienotu uzturēšanas periodu blokam *Periodi*.

Ņemiet vērā, ka pāreja uz vasaras/ziemas laiku (DST) neietekmē uzturēšanas ilgumu. Piemēram, pieņemsim, ka ir konfigurēta divu stundu uzturēšana, kas parasti sākas plkst. 01:00 un beidzas plkst. 03:00:

- ja pēc vienas uzturēšanas stundas (plkst. 02:00) notiek pāreja uz vasaras/ziemas laiku un pašreizējais laiks mainās no 02:00 uz 03:00, uzturēšana turpināsies vēl vienu stundu (līdz 04:00);
- ja pēc divām uzturēšanas stundām (plkst. 03:00) notiek pāreja uz vasaras/ziemas laiku un pašreizējais laiks mainās no 03:00 uz 02:00, uzturēšana tiks apturēta, jo būs pagājušas divas stundas;
- ja uzturēšanas periods sākas stundā, kas tiek izlaista pārejas uz vasaras/ziemas laiku dēļ, uzturēšana nesāksies.

Ja uzturēšanas periods ir iestatīts uz ”1 day” (faktiskais uzturēšanas periods ir 24 stundas, jo Zabbix dienas aprēķina stundās), sākas plkst. 00:00 un beidzas nākamajā dienā plkst. 00:00:

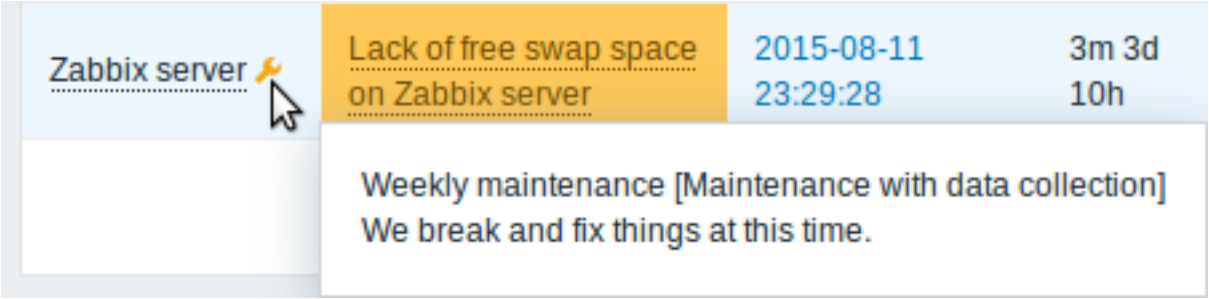
- uzturēšana tiks apturēta nākamajā dienā plkst. 01:00, ja pašreizējais laiks tiek pārbīdīts vienu stundu uz priekšu;
- uzturēšana tiks apturēta tajā pašā dienā plkst. 23:00, ja pašreizējais laiks tiek pārbīdīts vienu stundu atpakaļ.

Attēlošana Hostu attēlošana uzturēšanas režīmā

Oranža uzgriežņu atslēgas ikona blakus hosta nosaukumam norāda, ka šis hosts atrodas uzturēšanas režīmā šeit:

- *Paneļi*

- *Uzraudzība* > *Problēmas*
- *Inventārs* > *Hosti* > *Hosta inventāra informācija*
- *Datu ievākšana* > *Hosti* (skatiet kolonnu 'Statuss')



Informācija par uzturēšanu tiek parādīta, kad peles rādītājs tiek novietots virs ikonas.

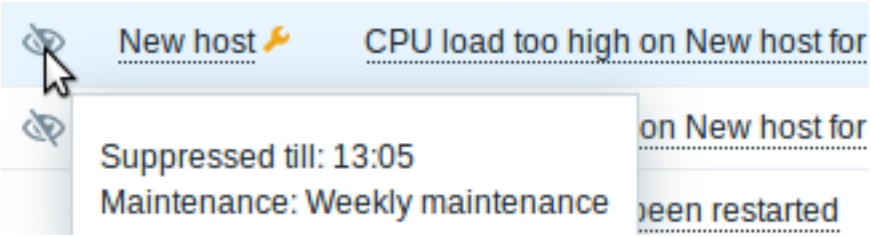
Papildus tam hosti uzturēšanas režīmā tiek attēloti ar oranžu fonu sadaļā *Uzraudzība* > *Kartes*.

Apspiesto problēmu attēlošana

Parasti problēmas hostiem apkopes režīmā tiek apspiestas, t.i., netiek attēlotas lietotāja saskarnē. Tomēr ir iespējams arī konfigurēt, lai apspiestās problēmas tiktu rādītas, atlasot opciju *Rādīt apspiestās problēmas* šajās vietās:

- *Paneļi* (*Problēmu hosti*, *Problēmas*, *Problēmas pēc svarīguma*, *Trigeru pārskats* logrīka konfigurācijā)
- *Uzraudzība* > *Problēmas* (filtrā)
- *Uzraudzība* > *Kartes* (kartes konfigurācijā)
- Globālie paziņojumi (lietotāja profila konfigurācijā)

Kad apspiestās problēmas tiek attēlotas, tiek rādīta šāda ikona: . Novietojot peles kursoru virs ikonas, tiek parādīta papildu informācija.



Rindu aprēķināšana apkopes laikā Rindas, kas tiek parādītas Zabbix lietotāja saskarnē (Administration > Queue), aprēķina Zabbix serveris. Tās neietver vienumus, kuriem apkopes laikā netiek vākoti dati — šo vienumu rindas garums vienmēr ir nulle, pat ja to vērtības kavējas. Kavētie vienumi apkopes laikā ar datu vākšanu joprojām tiek ieskaitīti rindā.

Zabbix starpniekserveris nezina par apkopes periodiem, jo starp Zabbix serveri un starpniekserveri netiek sinhronizēta apkopes konfigurācija. Iekšējās pārbaudes, kas tiek aprēķinātas Zabbix starpniekserveros (piemēram, `zabbix[queue,,]` un `zabbix[stats,,,queue,,]`), ziņo par kavētiem vienumiem neatkarīgi no apkopes statusa Zabbix serverī.

Rezultātā vieniem un tiem pašiem vienumiem apkopes laikā bez datu vākšanas Zabbix lietotāja saskarne un iekšējās pārbaudes Zabbix starpniekserveros var ziņot atšķirīgu rindas garumu.

10 Regulārās izteiksmes

Pārskats Zabbix atbalsta Perl Compatible Regular Expressions (PCRE, PCRE2).

Ir divi veidi, kā izmantot regulārās izteiksmes Zabbix:

- manuāli ievadīt regulāro izteiksmi
- izmantot Zabbix izveidotu globālo regulāro izteiksmi

Regulārās izteiksmes Atbalstītajās vietās varat manuāli ievadīt regulāro izteiksmi. Ņemiet vērā, ka izteiksme nedrīkst sākties ar @, jo šis simbols tiek izmantots Zabbix globālo regulāro izteiksmju atsaucēm.

Warning:
Izmantojot regulārās izteiksmes, ir iespējams izsmelt steku. Plašāku informāciju skatiet pcrestack man lapā.

Ņemiet vērā, ka vairākrindu atbilstības gadījumā enkuri ^ un $ atbilst attiecīgi katras rindas sākumam/beigām, nevis visas virknes sākumam/beigām.

Skatiet arī pareizas ekrānēšanas piemērus dažādos kontekstos.

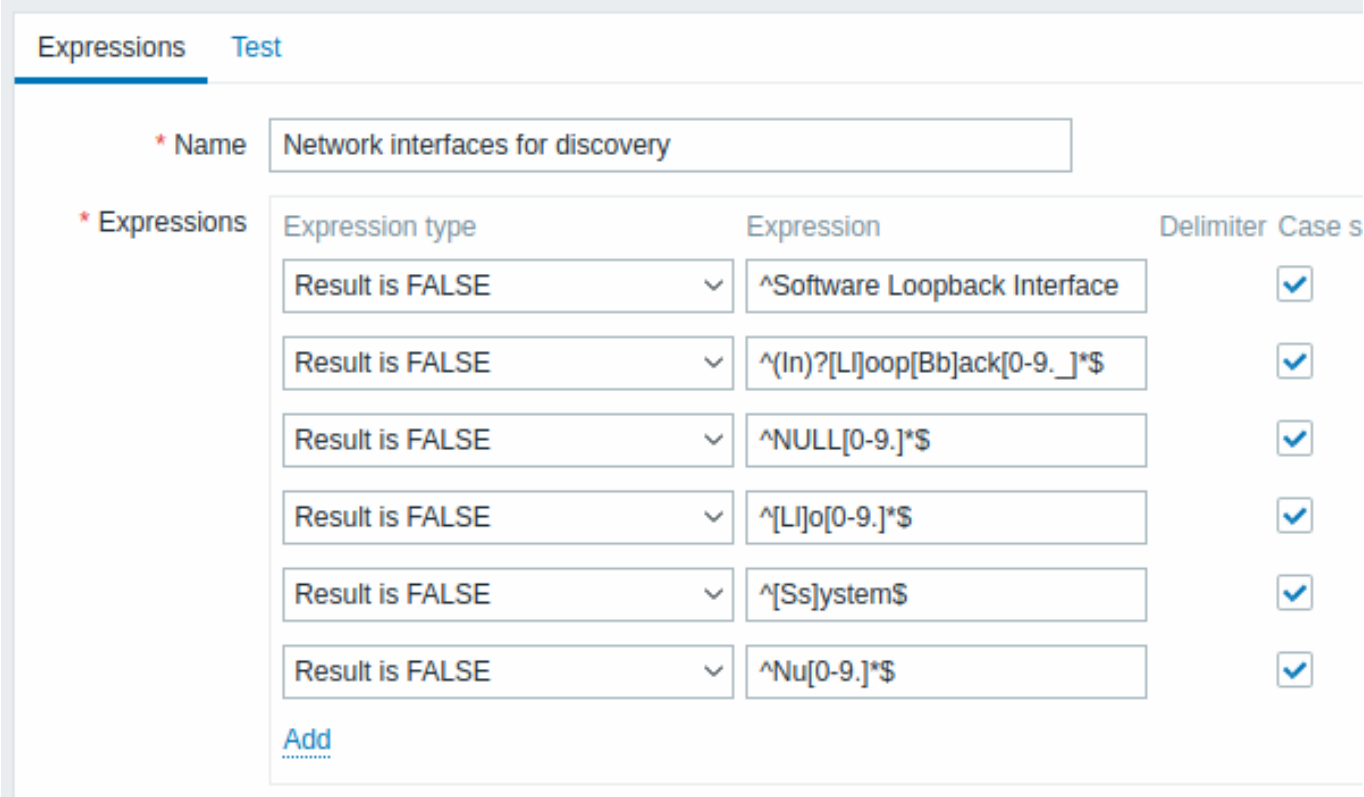
Globālās regulārās izteiksmes Zabbix lietotāja saskarnē ir pieejams paplašināts redaktors sarežģītu regulāro izteiksmju izveidei un testēšanai.

Kad regulārā izteiksme šādā veidā ir izveidota, to var izmantot vairākās vietās lietotāja saskarnē, atsaucoties uz tās nosaukumu ar prefiksu @, piemēram, *@mycustomregexp*.

Lai izveidotu globālu regulāro izteiksmi:

- Dodieties uz: *Administration* → *General*
- Nolaižamajā sarakstā atlasiet *Regular expressions*
- Noklikšķiniet uz *New regular expression*

Cilne **Expressions** ļauj iestatīt regulārās izteiksmes nosaukumu un pievienot apakšizteiksmes.



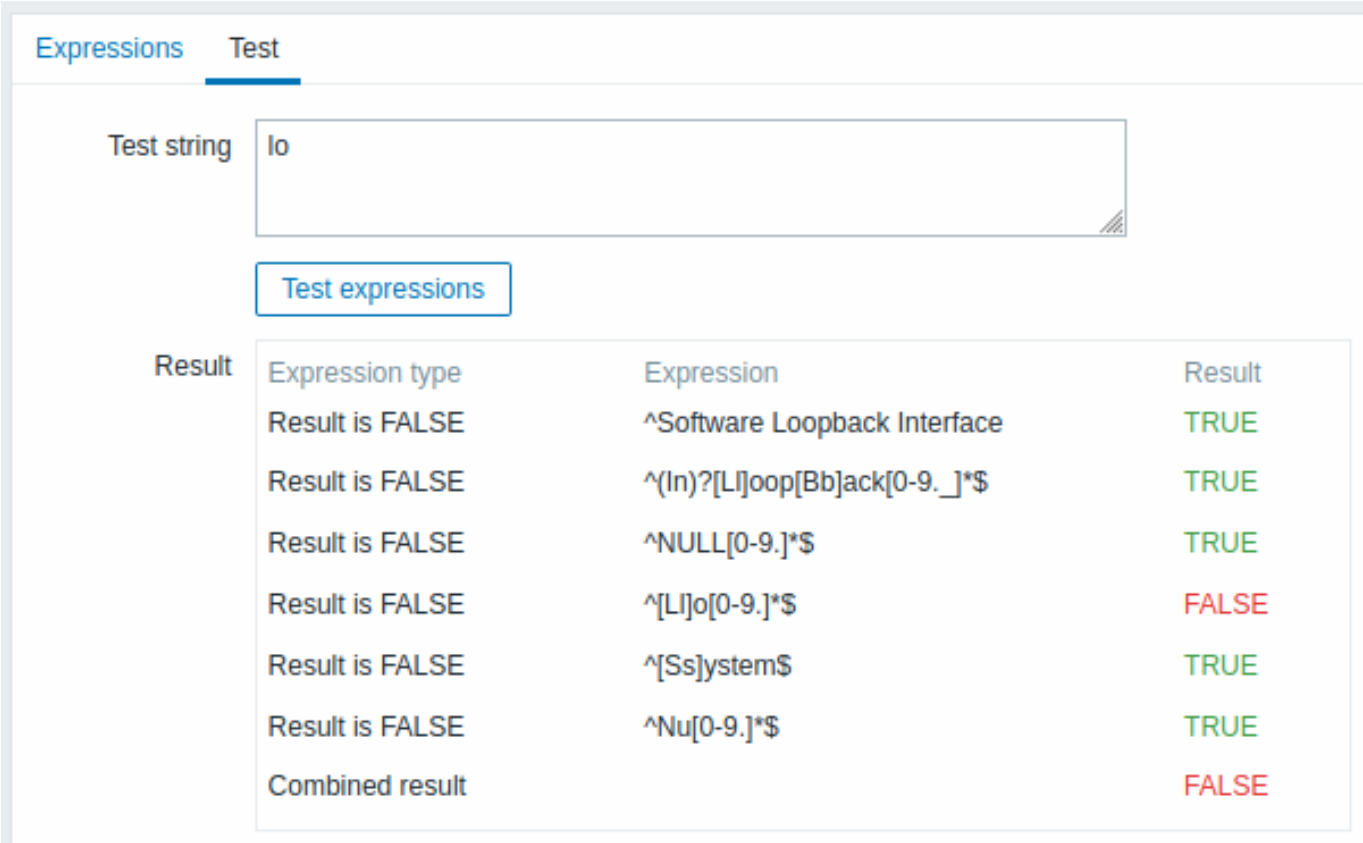
Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Iestatiet regulārās izteiksmes nosaukumu. Ir atļautas jebkuras Unicode rakstzīmes.
Expressions	Noklikšķiniet uz *Add* blokā Expressions, lai pievienotu jaunu apakšizteiksmi.
Expression type	Atlasiet izteiksmes tipu: **Character string included** - atbilst apakšvirknei **Any character string included** - atbilst jebkurai apakšvirknei no atdalīta saraksta. Atdalītais saraksts ietver komatu (,), punktu (.) vai slīpsvītru (/). **Character string not included** - atbilst jebkurai virknei, izņemot apakšvirkni **Result is TRUE** - atbilst regulārajai izteiksmei **Result is FALSE** - neatbilst regulārajai izteiksmei
Expression	Ievadiet apakšvirkni/regulāro izteiksmi.
Delimiter	Komats (,), punkts (.) vai slīpsvītra (/), lai atdalītu teksta virknes regulārajā izteiksmē. Šis parametrs ir aktīvs tikai tad, ja ir atlasīts izteiksmes tips *"Any character string included"*.
Case sensitive	Izvēles rūtiņa, lai norādītu, vai regulārā izteiksme ir jutīga pret burtu lielajiem un mazajiem burtiem.

Slīpsvītra (/) izteiksmē tiek apstrādāta burtiski, nevis kā atdalītājs. Tādējādi ir iespējams saglabāt izteiksmes, kas satur slīpsvītru, bez kļūdām.

Attention:
Pielāgotas regulārās izteiksmes nosaukums Zabbix var saturēt komatus, atstarpes utt. Gadījumos, kad tas var radīt nepareizu interpretāciju atsaucē (piemēram, komats vienuma atslēgas parametrā), visu atsauci var ievietot pēdiņās šādi: *"@My custom regexp for purpose1, purpose2"*.
 Regulāro izteiksmju nosaukumus citās vietās nedrīkst likt pēdiņās (piemēram, LLD kārtulas īpašībās).

Cilnē **Test** regulāro izteiksmi un tās apakšizteiksmes var pārbaudīt, norādot testa virkni.



Rezultāti parāda katras apakšizteiksmes statusu un kopējo pielāgotās izteiksmes statusu.

Pielāgotās izteiksmes kopējo statusu nosaka *Combined result*. Ja ir definētas vairākas apakšizteiksmes, Zabbix izmanto loģisko operatoru AND, lai aprēķinātu *Combined result*. Tas nozīmē, ka, ja vismaz vienam Result ir False, arī *Combined result* būs False statuss.

Noklusējuma globālās regulārās izteiksmes Zabbix noklusējuma datu kopā ir iekļautas vairākas globālās regulārās izteiksmes.

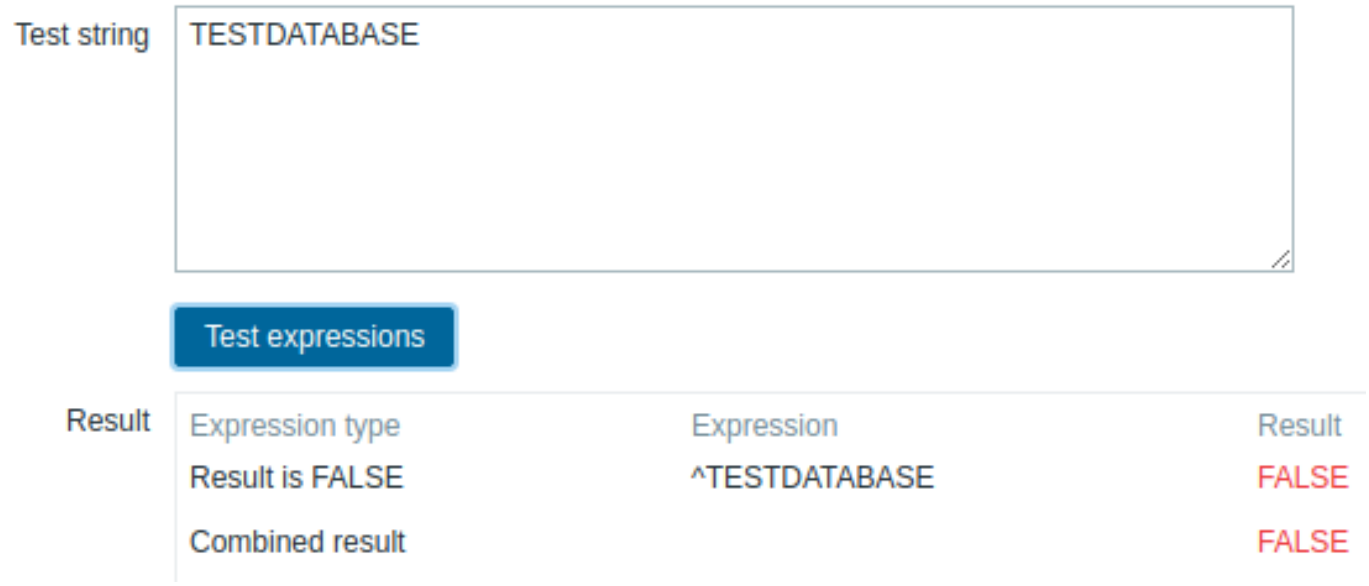
Name	Expression	Matches
Failu sistēmas atklāšanai	`^(btrfs|ext2|ext3|ext4|jfs|reiser|xfs|ffs|ufs|jfs|jfs2|vxfs|hfs|refs|apfs|ntfs|fat32|zfs)`	"btrfs" vai "ext2" vai "ext3" vai "ext4" vai "jfs" vai "reiser", vai "xfs", vai "ffs", vai "ufs", vai "jfs", vai "jfs2", vai "vxfs", vai "hfs", vai "refs", vai "apfs", vai "ntfs", vai "fat32", vai "zfs"
Tīkla saskarnes atklāšanai	`^Software Loopback Interface`	Virknes, kas sākas ar "Software Loopback Interface".
	`^lo$`	"lo"
	`^(In)?[Ll]oop[Bb]ack[0-9._]*$`	Virknes, kas pēc izvēles sākas ar "In", pēc tam satur "L" vai "l", tad "oop", tad "B" vai "b", tad "ack", kam pēc izvēles var sekot jebkurš ciparu, punktu vai pasvītrojuma rakstzīmju skaits.
	`^NULL[0-9.]*$`	Virknes, kas sākas ar "NULL", kam pēc izvēles seko jebkurš ciparu vai punktu skaits.
	`^[Ll]o[0-9.]*$`	Virknes, kas sākas ar "Lo" vai "lo" un kam pēc izvēles seko jebkurš ciparu vai punktu skaits.

Name	Expression	Matches
	`^[Ss]ystem$` `^Nu[0-9.]*$`	”System” vai ”system” Virknes, kas sākas ar ”Nu”, kam pēc izvēles seko jebkurš ciparu vai punktu skaits.
Krātuves ierīces SNMP atklāšanai	`^(Physical memory|Virtual memory|Memory buffers|Cached memory|Swap space)$`	”Physical memory”, vai ”Virtual memory”, vai ”Memory buffers”, vai ”Cached memory”, vai ”Swap space”
Windows pakalpojumu nosaukumi atklāšanai	`^(MMCSS|gupdate|SysmonLog|clr_optimization_v2.0.50727_32|clr_optimization_v4.0.30319_32)$`	”MMCSS”, vai ”gupdate”, vai ”SysmonLog”, vai virknes, piemēram, ”clr_optimization_v2.0.50727_32” un ”clr_optimization_v4.0.30319_32”, kur punktu vietā var ievietot jebkuru rakstzīmi, izņemot jaunu rindu.
Windows pakalpojumu startēšanas stāvokļi atklāšanai	`^(automatic|automatic delayed)$`	”automatic” vai ”automatic delayed”

Piemēri Piemērs 1

Šādas izteiksmes izmantošana zema līmeņa atklāšanā, lai atklātu datubāzes, izņemot datubāzi ar noteiktu nosaukumu:

`^TESTDATABASE$`

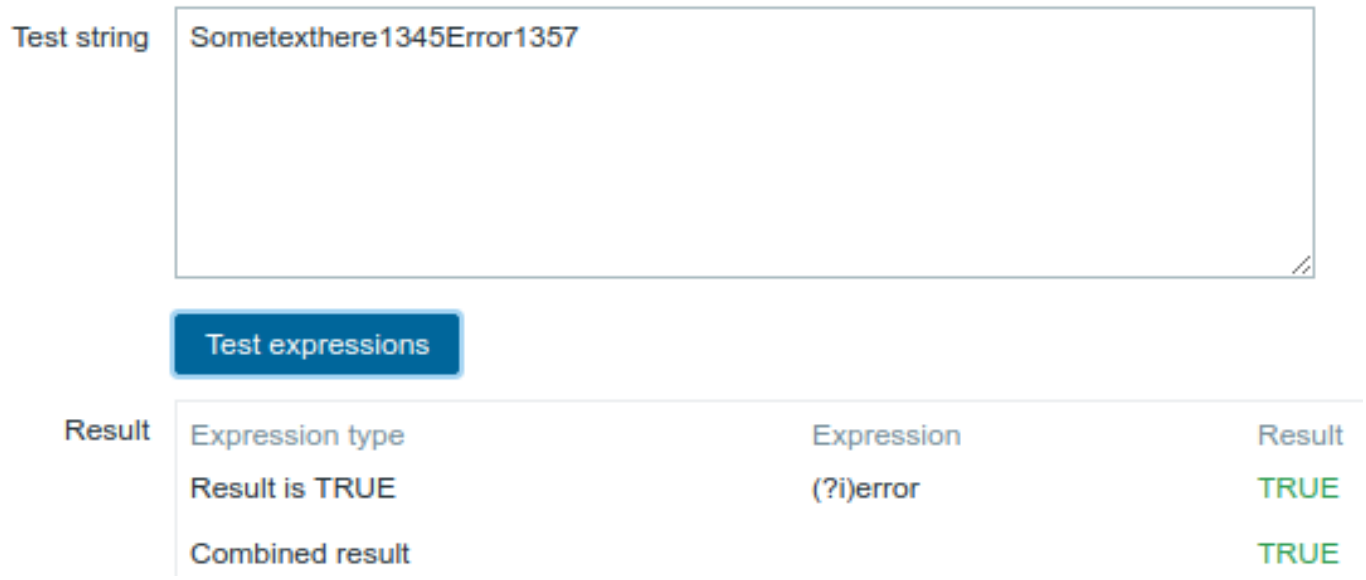


Izvēlētais *Izteiksmes tips*: ”Rezultāts ir FALSE”. Neatbilst nosaukumam, kas satur virkni ”*TESTDATABASE*”.

Piemērs ar iekļautu regex modifikatoru

Šādas regulārās izteiksmes izmantošana, kas ietver iekļautu modifikatoru (?i), lai atbilstu rakstzīmēm ”error”:

`(?i)error`



Izvēlētais *Izteiksmes tips*: "Rezultāts ir TRUE". Rakstzīmes "error" tiek atbilstinātas.

Vēl viens piemērs ar iekļautu regex modifikatoru

Tālāk norādītās regulārās izteiksmes izmantošana, kas ietver vairākus iekļautos modifikatorus, lai atbilstu rakstzīmēm pēc noteiktas rindas:

```
(?<=match (?i)everything(?-i) after this line\n)(?sx).*# we add s modifier to allow . match newline charac
```

Izvēlētais izteiksmes tips: "Rezultāts ir TRUE". Rakstzīmes pēc noteiktas rindas tiek atbilstas.

Attention:
g modifikatoru nevar norādīt rindā. Pieejamo modifikatoru sarakstu var atrast pcresyntax man lapā. Plašāku informāciju par PCRE sintaksi skatiet PCRE HTML dokumentācijā.

Regulāro izteiksmju atbalsts pēc atrašanās vietas

Atrašanās vieta	Regulārā izteiksme	Globālā regulārā izteiksme	Vairākrindu atbilstība	Komentāri
Aģenta vienumi				
eventlog[]	Jā	Jā	Jā	`regexp`, `severity`, `source`, `eventid` parametri
eventlog.count[]				`regexp`, `severity`, `source`, `eventid` parametri
log[]				`regexp` parametrs
log.count[]				
logrt[]		Jā/Nē		`regexp` parametrs atbalsta abus, `file_regexp` parametrs atbalsta tikai neglobālās izteiksmes

Atrašanās vieta	Regulārā izteiksme	Globālā regulārā izteiksme	Vairākrindu atbilstība	Komentāri
logrt.count[]				
proc.cpu.util[]		Nē	Nē	`cmdline` parametrs
proc.get[]				
proc.mem[]				
proc.num[]				
sensor[]				`device` un `sensor` parametri Linux 2.4
system.hw.macaddr[]				`interface` parametrs
system.sw.packages[]				`regexp` parametrs
system.sw.packages.get[]				`regexp` parametrs
vfs.dir.count[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parametri
vfs.dir.get[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parametri
vfs.dir.size[]				`regex_incl`, `regex_excl`, `regex_excl_dir` parametri
vfs.file.regexp[]			Jā	`regexp` parametrs
vfs.file.regmatch[]				
web.page.regexp[]				
SNMP trapi				
snmptrap[]	Jā	Jā	Nē	`regexp` parametrs
Vienuma vērtības priekšapstrāde	Jā	Nē	Nē	`pattern` parametrs
Funkcijas trigeriem/aprēķinātajiem vienumiem				
count()	Jā	Jā	Jā	`pattern` parametrs, ja `operator` parametrs ir *regexp* vai *iregexp*
countunique()	Jā	Jā		
find()	Jā	Jā		
logeventid()	Jā	Jā	Nē	`pattern` parametrs
logsource()				
Zema līmeņa atklāšana				
Filtri	Jā	Jā	Nē	Lauks *Regular expression*
Aizstāšanas	Jā	Nē		Opcijās *matches*, *does not match* nosacījumiem *Operation*
Darbību nosacījumi	Jā	Nē	Nē	Opcijās *matches*, *does not match* automātiskās reģistrācijas nosacījumiem *Host name* un *Host metadata*
Skripti	Jā	Jā	Nē	Lauks *Input validation rule*
Tīmekļa uzraudzība	Jā	Nē	Jā	*Variables* ar prefiksu **regex:** Lauks *Required string*
Lietotāja makro kontekstā	Jā	Nē	Nē	Makro kontekstā ar prefiksu **regex:**
Makro funkcijas				
regsub()	Jā	Nē	Nē	`pattern` parametrs

Atrašanās vieta	Regulārā izteiksme	Globālā regulārā izteiksme	Vairākrindu atbilstība	Komentāri
iregsub()				
Saišu indikatori kartēs	Jā	Nē	Nē	Lauks *Pattern* (teksta vienumiem)
Ikonu kartēšana	Jā	Jā	Nē	Lauks *Expression*
Vērtību kartēšana	Jā	Nē	Nē	Lauks *Value*, ja kartēšanas tips ir `regexp`

11 Problēmas apstiprināšana

Pārskats Problēmu notikumus Zabbix sistēmā lietotāji var apstiprināt.

Ja lietotājs saņem paziņojumu par problēmas notikumu, viņš var atvērt Zabbix lietotāja saskarni, atvērt šīs problēmas problēmas atjaunināšanas uznirstošo logu, izmantojot kādu no tālāk norādītajiem veidiem, un apstiprināt problēmu. Veicot apstiprināšanu, viņš var ievadīt savu komentāru, norādot, ka strādā pie tās vai pievienojot jebkādu citu informāciju, ko uzskata par nepieciešamu.

Tādējādi, ja cits sistēmas lietotājs pamana to pašu problēmu, viņš uzreiz redz, vai tā ir apstiprināta, kā arī līdzšinējos komentārus.

Tādējādi problēmu risināšanas darbplūsma, kurā iesaistīts vairāk nekā viens sistēmas lietotājs, var notikt koordinēti.

Apstiprināšanas statuss tiek izmantots arī, definējot darbību operācijas. Piemēram, var definēt, ka paziņojums tiek nosūtīts augstāka līmeņa vadītājam tikai tad, ja notikums kādu laiku nav apstiprināts.

Lai apstiprinātu notikumus un pievienotu tiem komentārus, lietotājam jābūt vismaz lasīšanas tiesībām uz atbilstošajiem trigeriem. Lai mainītu problēmas būtiskumu vai aizvērtu problēmu, lietotājam jābūt lasīšanas-rakstīšanas tiesībām uz atbilstošajiem trigeriem.

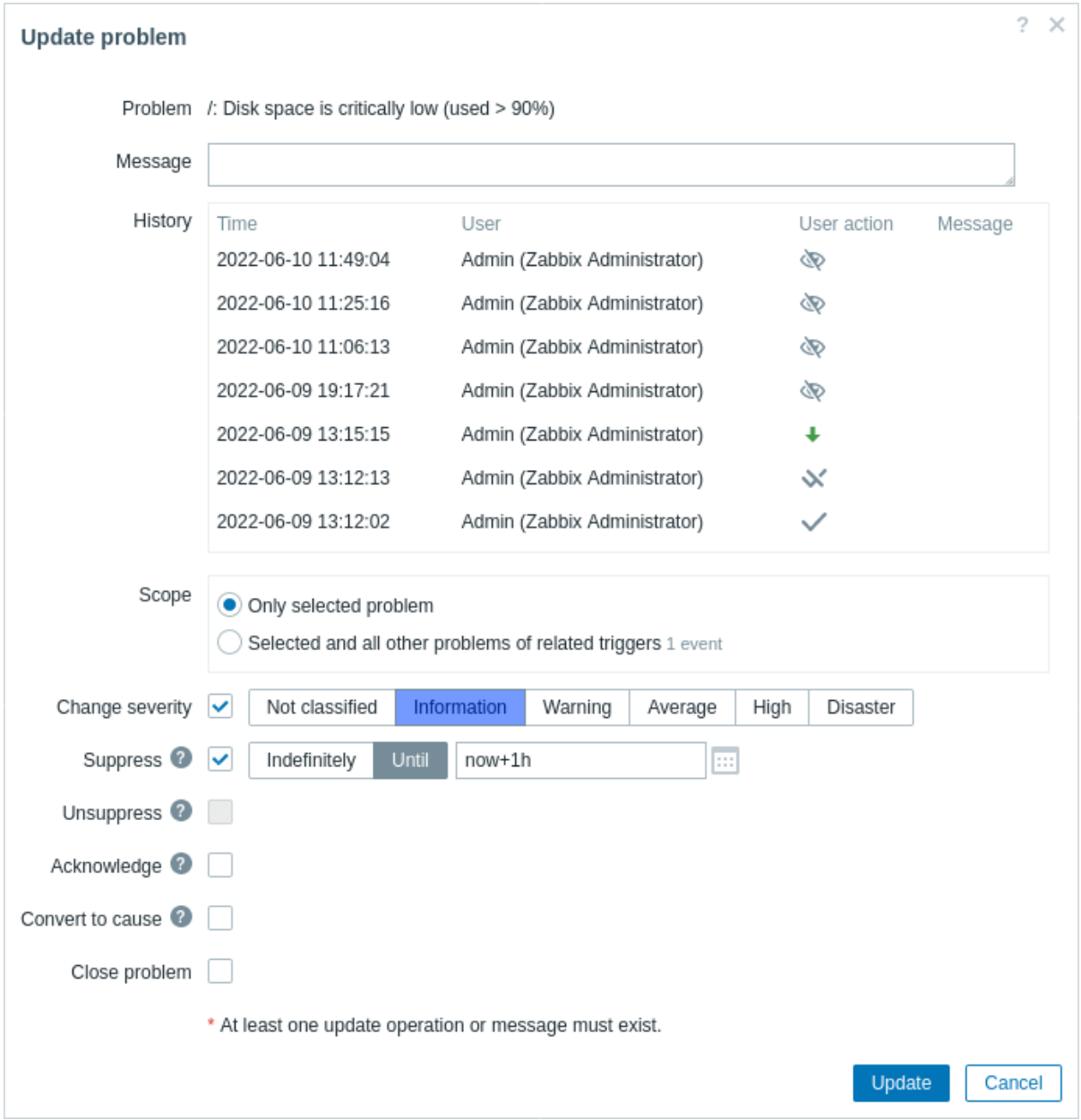
Ir **vairāki** veidi, kā piekļūt problēmas atjaunināšanas uznirstošajam logam, kas ļauj apstiprināt problēmu.

- Varat atlasīt problēmas sadaļā *Uzraudzība* → *Problēmas* un pēc tam noklikšķināt uz *Masveida atjaunināšana* zem saraksta
- Varat noklikšķināt uz *Atjaunināt* problēmas kolonnā *Atjaunināt* šeit:
  - *Paneļi* (*Problēmas* un *Problēmas pēc būtiskuma* logrīki)
  - *Uzraudzība* → *Problēmas*
  - *Uzraudzība* → *Problēmas* → *Notikuma informācija*
- Varat noklikšķināt uz neatrisinātas problēmas šūnas šeit:
  - *Paneļi* (*Trigeru pārskata* logrīks)

Uznirstošajā izvēlnē ir opcija *Atjaunināt*, kas atver problēmas atjaunināšanas logu.

Problēmu atjaunināšana Problēmas atjaunināšanas uznirstošais logs ļauj:

- komentēt problēmu
- skatīt līdzšinējos komentārus un darbības
- mainīt problēmas būtiskumu
- apspiest/noņemt apspiešanu problēmai
- apstiprināt/noņemt apstiprinājumu problēmai
- mainīt simptomu problēmu uz cēloņa problēmu
- manuāli aizvērt problēmu



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Problem	Ja ir atlasīta tikai viena problēma, tiek parādīts problēmas nosaukums. Ja ir atlasītas vairākas problēmas, tiek parādīts *N problems selected*.
Message	Ievadiet tekstu, lai komentētu problēmu (maksimums 2048 rakstzīmes).
History	Tiek uzskaitītas iepriekšējās darbības un komentāri par problēmu, kā arī laika un lietotāja informācija. Par lietotāju darbību apzīmēšanai izmantoto ikonu nozīmi skatiet lapā event detail. Ņemiet vērā, ka vēsture tiek parādīta tikai tad, ja atjaunināšanai ir atlasīta viena problēma.
Scope	Definējiet tādu darbību tvērumu kā būtiskuma maiņa, apstiprināšana vai problēmu manuāla aizvēršana: **Only selected problem** - ietekmēs tikai šo notikumu **Selected and all other problems of related triggers** - problēmas apstiprināšanas/aizvēršanas gadījumā ietekmēs šo notikumu un visas pārējās problēmas, kas līdz šim vēl nav apstiprinātas/aizvērtas. Ja tvērumā ir problēmas, kas jau ir apstiprinātas vai aizvērtas, tās netiks atkārtoti apstiprinātas/aizvērtas. Savukārt ziņojumu un būtiskuma maiņas darbību skaits nav ierobežots.

Parametrs	Apraksts
Change severity	Atzīmējiet izvēles rūtiņu un noklikšķiniet uz būtiskuma pogas, lai atjauninātu problēmas būtiskumu. Izvēles rūtiņa būtiskuma maiņai ir pieejama, ja vismaz vienai no atlasītajām problēmām ir lasīšanas-rakstīšanas tiesības. Noklikšķinot uz *Update*, tiks atjauninātas tikai tās problēmas, kurām ir lasīšanas-rakstīšanas tiesības. Ja nevienam no atlasītajiem trigeriem nav lasīšanas-rakstīšanas tiesību, izvēles rūtiņa ir atspējota.
Suppress	Atzīmējiet izvēles rūtiņu, lai apspiestu problēmu: **Indefinitely** - apspiest uz nenoteiktu laiku **Until** - apspiest līdz norādītajam laikam. Tiek atbalstīti gan absolūtie, gan relatīvie laika formāti, piemēram: `now+1d` - uz vienu dienu no pašreizējā brīža (noklusējums) `now/w` - līdz pašreizējās nedēļas beigām `2022-05-28 12:00:00` - līdz absolūtajam datumam/laikam Ņemiet vērā, ka vienkāršs periods (piem., `1d`, `1w`) netiek atbalstīts. Šīs opcijas pieejamība ir atkarīga no lietotāja lomas iestatījuma "Suppress problems". Skatiet arī: Problem suppression
Unsuppress	Atzīmējiet izvēles rūtiņu, lai noņemtu problēmas apspiešanu. Šī izvēles rūtiņa ir aktīva tikai tad, ja vismaz viena no atlasītajām problēmām ir apspiesta. Šīs opcijas pieejamība ir atkarīga no lietotāja lomas iestatījuma "Suppress problems".
Acknowledge	Atzīmējiet izvēles rūtiņu, lai apstiprinātu problēmu. Šī izvēles rūtiņa ir pieejama, ja starp atlasītajām ir vismaz viena neapstiprināta problēma. Jau apstiprinātai problēmai nav iespējams pievienot vēl vienu apstiprinājumu (tomēr ir iespējams pievienot vēl vienu komentāru).
Unacknowledge	Atzīmējiet izvēles rūtiņu, lai noņemtu problēmas apstiprinājumu. Šī izvēles rūtiņa ir pieejama, ja starp atlasītajām ir vismaz viena apstiprināta problēma.
Convert to cause	Atzīmējiet izvēles rūtiņu, lai pārveidotu simptomu problēmu(-as) par cēloņa problēmu(-ām).
Close problem	Atzīmējiet izvēles rūtiņu, lai manuāli aizvērtu atlasīto(-ās) problēmu(-as). Izvēles rūtiņa problēmas aizvēršanai ir pieejama, ja vismaz vienai no atlasītajām problēmām trigera konfigurācijā ir atzīmēta opcija *Allow manual close*. Noklikšķinot uz *Update*, tiks aizvērtas tikai tās problēmas, kuras ir atļauts aizvērt. Ja nevienu problēmu nevar aizvērt manuāli, izvēles rūtiņa ir atspējota. Jau aizvērtas problēmas netiks aizvērtas atkārtoti.

Attēlošana Pamatojoties uz apstiprināšanas informāciju, ir iespējams konfigurēt, kā informācijas panelī vai kartēs tiek attēlots problēmu skaits. Lai to izdarītu, jāveic izvēles opcijā *Problem display*, kas ir pieejama gan kartes konfigurācijā, gan *Problems by severity* informācijas paneļa logrīkā. Ir iespējams attēlot kopējo problēmu skaitu, neapstiprināto problēmu skaitu atsevišķi no kopējā skaita vai tikai neapstiprināto problēmu skaitu.

Pamatojoties uz problēmu atjaunināšanas informāciju (apstiprināšana u.c.), ir iespējams konfigurēt atjaunināšanas darbības — nosūtīt ziņojumu vai izpildīt attālās komandas.

1 Problēmu apspiešana

Pārskats

Problēmu apspiešana piedāvā veidu, kā uz laiku paslēpt problēmu, ar kuru var tikt galā vēlāk. Tas ir noderīgi, lai sakārtotu problēmu sarakstu, piešķirot visaugstāko prioritāti vissteidzamākajām problēmām. Piemēram, dažreiz problēma var rasties nedēļas nogalē un nebūt tik steidzama, lai to risinātu nekavējoties, tāpēc to var "atlikt" līdz pirmdienas rītam.

Problēmu apspiešana ļauj paslēpt *vienu* problēmu, atšķirībā no problēmu apspiešanas, izmantojot hosta uzturēšanu, kad tiek paslēptas visas uzturēšanā esošā hosta problēmas.

Darbības trigeru darbībām apspiestām problēmām tiks apturētas tāpat, kā tas tiek darīts hosta uzturēšanā.

Konfigurācija

Problēmu var apspiest, izmantojot **problēmas atjaunināšanas** logu, kur apspiešana ir viena no problēmas atjaunināšanas opcijām līdzās komentēšanai, svarīguma līmeņa maiņai, apstiprināšanai u.c.

Problēmai var arī noņemt apspiešanu, izmantojot to pašu problēmas atjaunināšanas logu.

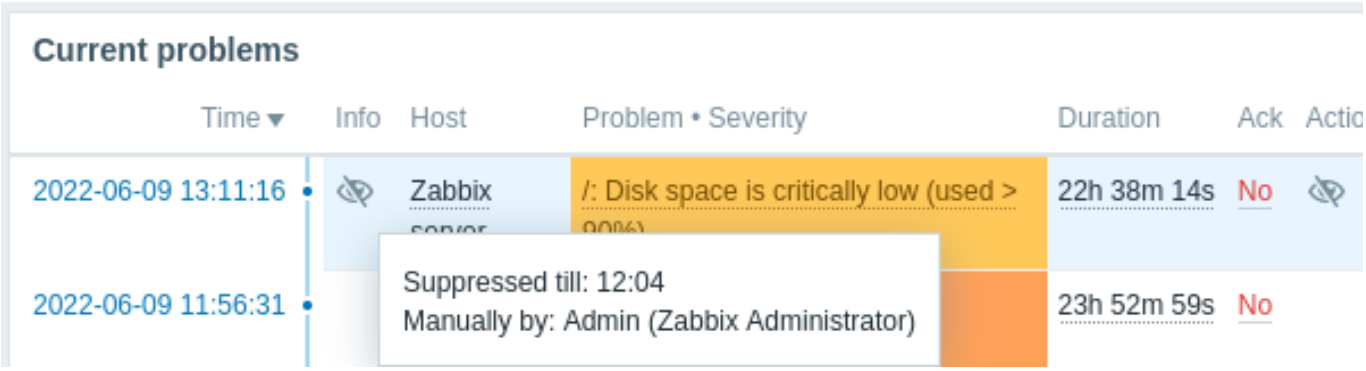
Attēlošana

Pēc apspiešanas problēma tiek atzīmēta ar mirgojošu apspiešanas ikonu kolonnā *Info*, pirms tā tiek paslēpta.

Apspiešanas ikona mirgo, kamēr apspiešanas uzdevums atrodas gaidīšanas sarakstā. Kad uzdevumu pārvaldnieks ir apspiedis problēmu, ikona pārstās mirgot. Ja apspiešanas ikona turpina mirgot ilgu laiku, tas var norādīt uz servera problēmu, piemēram, ja serveris nedarbojas un uzdevumu pārvaldnieks nevar pabeigt uzdevumu. Tāda pati loģika attiecas arī uz apspiešanas atcelšanu. Īsajā periodā pēc uzdevuma iesniegšanas, kamēr serveris to vēl nav pabeidzis, apspiešanas atcelšanas ikona mirgo.

Apspiesta problēma var būt gan paslēpta, gan parādīta atkarībā no problēmu filtra/logrīka iestatījumiem.

Ja apspiesta problēma tiek rādīta problēmu sarakstā, tā tiek atzīmēta ar apspiešanas ikonu, un, novietojot peles kursoru virs tās, tiek parādīta informācija par apspiešanu:



Informācija par apspiešanu tiek parādīta arī uznirstošajā logā, novietojot peles kursoru uz apspiešanas ikonas kolonnā *Actions*.

12 Konfigurācijas eksportēšana/importēšana

Pārskats Zabbix eksporta/importa funkcionalitāte ļauj apmainīties ar dažādām konfigurācijas entītijām starp vienu Zabbix sistēmu un citu.

Tipiski šīs funkcionalitātes lietošanas gadījumi:

- koplietot veidnes vai tīkla kartes — Zabbix lietotāji var koplietot savus konfigurācijas parametrus
- augšupielādēt veidni uz Zabbix Community templates. Pēc tam citi var lejupielādēt veidni un importēt failu Zabbix.
- integrēties ar trešo pušu rīkiem — universālie YAML, XML un JSON formāti nodrošina integrāciju un datu importu/eksportu ar trešo pušu rīkiem un lietotnēm

Ko var eksportēt/importēt

Objekti, kurus var eksportēt/importēt, ir:

- Hostu grupas (*tikai caur Zabbix API*)
- Veidņu grupas (*tikai caur Zabbix API*)
- Veidnes
- Hosti
- Tīkla kartes
- Mediju tipi
- Attēli

Eksporta formāts

Datus var eksportēt, izmantojot Zabbix lietotāja saskarni vai Zabbix API. Atbalstītie eksporta formāti ir YAML, XML un JSON.

Informācija par eksportēšanu

- Visi atbalstītie elementi tiek eksportēti vienā failā.
- No saistītajām veidnēm mantotās hostu un veidņu entītijas (vienumi, trigeri, grafiki, atklāšanas kārtulas) netiek eksportētas. Visas izmaiņas, kas veiktas šajās entītijās hosta līmenī (piemēram, mainīts vienuma intervāls, modificēta regulārā izteiksme vai zema līmeņa atklāšanas kārtulai pievienoti prototipi), eksportēšanas laikā tiks zaudētas; importēšanas laikā visas entītijas no saistītajām veidnēm tiek izveidotas no jauna tāpat kā sākotnējā saistītajā veidnē.
- Zema līmeņa atklāšanas izveidotās entītijas un visas no tām atkarīgās entītijas netiek eksportētas. Piemēram, trigeris, kas izveidots vienumam, kuru ģenerējusi ZLA kārtula, netiks eksportēts.

- Ja eksportētais hosts/veidne satur entītijas, kas atbalsta noildzes, noildzes vērtības tiks eksportētas, ja šīm entītijām ir konfigurētas savas noildzes.

Informācija par importēšanu

- Importēšana tiek pārtraukta pie pirmās kļūdas.
- Atjauninot esošos attēlus attēlu importēšanas laikā, lauks "imagetype" tiek ignorēts, t.i., attēla tipu nav iespējams mainīt, izmantojot importēšanu.
- Importējot hosts/veidnes, izmantojot opciju "Delete missing", hosts/veidnes makrosi, kas nav iekļauti importēšanas failā, pēc importēšanas tiks dzēsti no hosts/veidnes.
- Tukšas birkas vienumiem, trigeriem, grafikiem, discoveryRules, itemPrototypes, triggerPrototypes, graphPrototypes ir bezjēdzīgas, t.i., tas ir tas pats, it kā tās nebūtu norādītas.
- Ja importētā hosts/veidnes entītijām ir konfigurēti savi noildzes laiki, tie tiks piemēroti; pretējā gadījumā tiks piemēroti starpniekservera/globālie noildzes laiki.
- Importēšana atbalsta YAML, XML un JSON; importēšanas failam jābūt ar pareizu faila paplašinājumu: .yaml un .yml YAML failiem, .xml XML failiem un .json JSON failiem. Skatiet saderības informāciju par atbalstītajām XML versijām.
- Importēšana atbalsta tikai konfigurācijas failus UTF-8 kodējumā (ar vai bez BOM); citi kodējumi (UTF16LE, UTF16BE, UTF32LE, UTF32BE u.c.) izraisīs importēšanas konvertēšanas kļūdu.

YAML pamatformāts YAML eksporta formāts satur šādus mezglus:

- Saknes mezgls Zabbix YAML eksportam
- Eksporta versija

```
zabbix_export:
  version: '8.0'
```

Pārējie mezgli ir atkarīgi no eksportētajiem objektiem.

XML formāts XML eksporta formāts satur šādus tagus:

- Noklusējuma galveni XML dokumentiem
- Saknes tagu Zabbix XML eksportam
- Eksporta versiju

```
<?xml version="1.0" encoding="UTF-8"?>
<zabbix_export>
    <version>8.0</version>
</zabbix_export>
```

Pārējie tagi ir atkarīgi no eksportētajiem objektiem.

JSON formāts JSON eksporta formāts satur šādus objektus:

- Zabbix JSON eksporta saknes objekts
- Eksporta versija

```
{
    "zabbix_export": {
        "version": "8.0"
    }
}
```

Pārējie objekti ir atkarīgi no eksportētajiem objektiem.

1 Veidņu grupas

Pārskats

Lietotāja saskarnē veidņu grupas var eksportēt tikai kopā ar veidņu eksportu. Kad veidne tiek eksportēta, visas grupas, kurām tā pieder, tiek automātiski eksportētas kopā ar to.

API ļauj eksportēt veidņu grupas neatkarīgi no veidnēm.

Eksporta formāts

```
  template_groups:
    - uuid: 36bff6c29af64692839d077febfc7079
      name: 'Tīkla ierīces'
```

Eksportētie elementi

Element	Type	Description
uuid	string	Unikāls identifikators šai veidņu grupai.
name	string	Grupas nosaukums.

2 Hostu grupas

Pārskats

Lietotāja saskarnē hostu grupas var eksportēt tikai kopā ar hostu eksportu. Kad hosts tiek eksportēts, visas grupas, kurām tas pieder, tiek automātiski eksportētas kopā ar to.

API ļauj eksportēt hostu grupas neatkarīgi no hostiem.

Eksporta formāts

```
  host_groups:
    - uuid: 6f6799aa69e844b4b3918f779f2abf08
      name: 'Zabbix serveri'
```

Eksportētie elementi

Elements	Tips	Apraksts
uuid	string	Unikāls identifikators šai hostu grupai.
name	string	Grupas nosaukums.

3 Veidnes

Pārskats

Veidnes tiek eksportētas kopā ar daudziem saistītiem objektiem un objektu attiecībām.

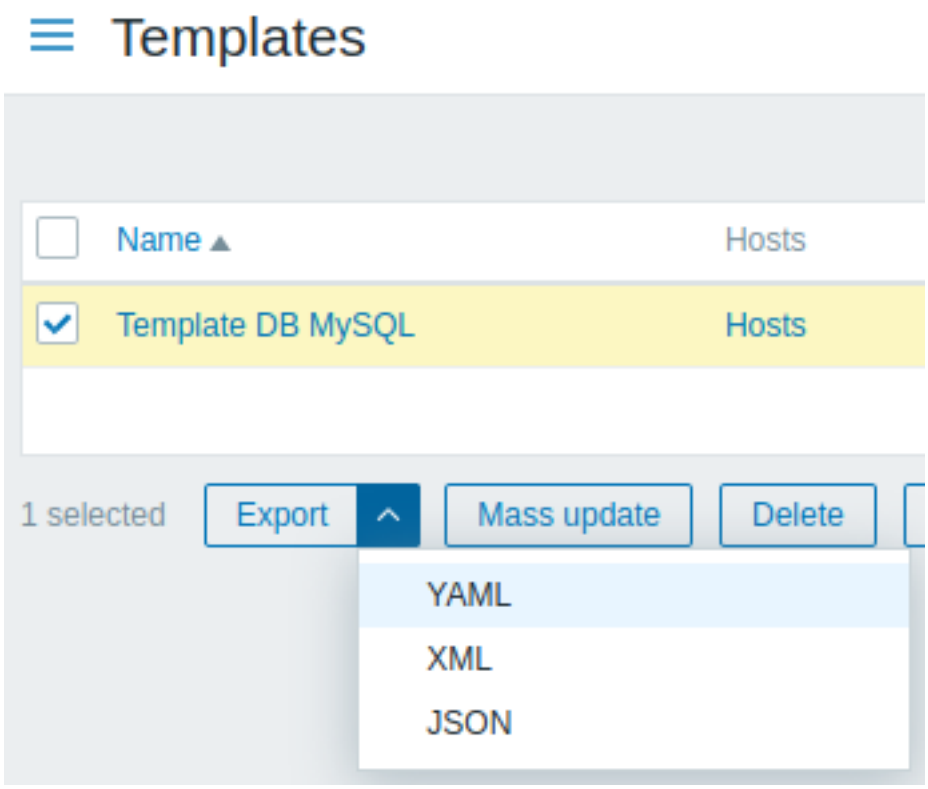
Veidnes eksports satur:

- Saistītās veidņu grupas
- Saistītās hostu grupas (ja tiek izmantotas hosta prototipa konfigurācijā)
- Veidnes datus
- Saisti ar citām veidnēm
- Saisti ar veidņu grupām
- Tieši saistītos vienumus
- Tieši saistītos trigerus
- Tieši saistītos grafikus
- Tieši saistītos paneļus
- Tieši saistītos atklāšanas noteikumus ar visiem prototipiem
- Tieši saistītos tīmekļa scenārijus
- Vērtību kartes

Eksportēšana

Lai eksportētu veidnes, rīkojieties šādi:

1. Dodieties uz *Datu ievākšana* → *Veidnes*.
2. Atzīmējiet eksportējamo veidņu izvēles rūtiņas.
3. Zem saraksta noklikšķiniet uz *Eksportēt*.



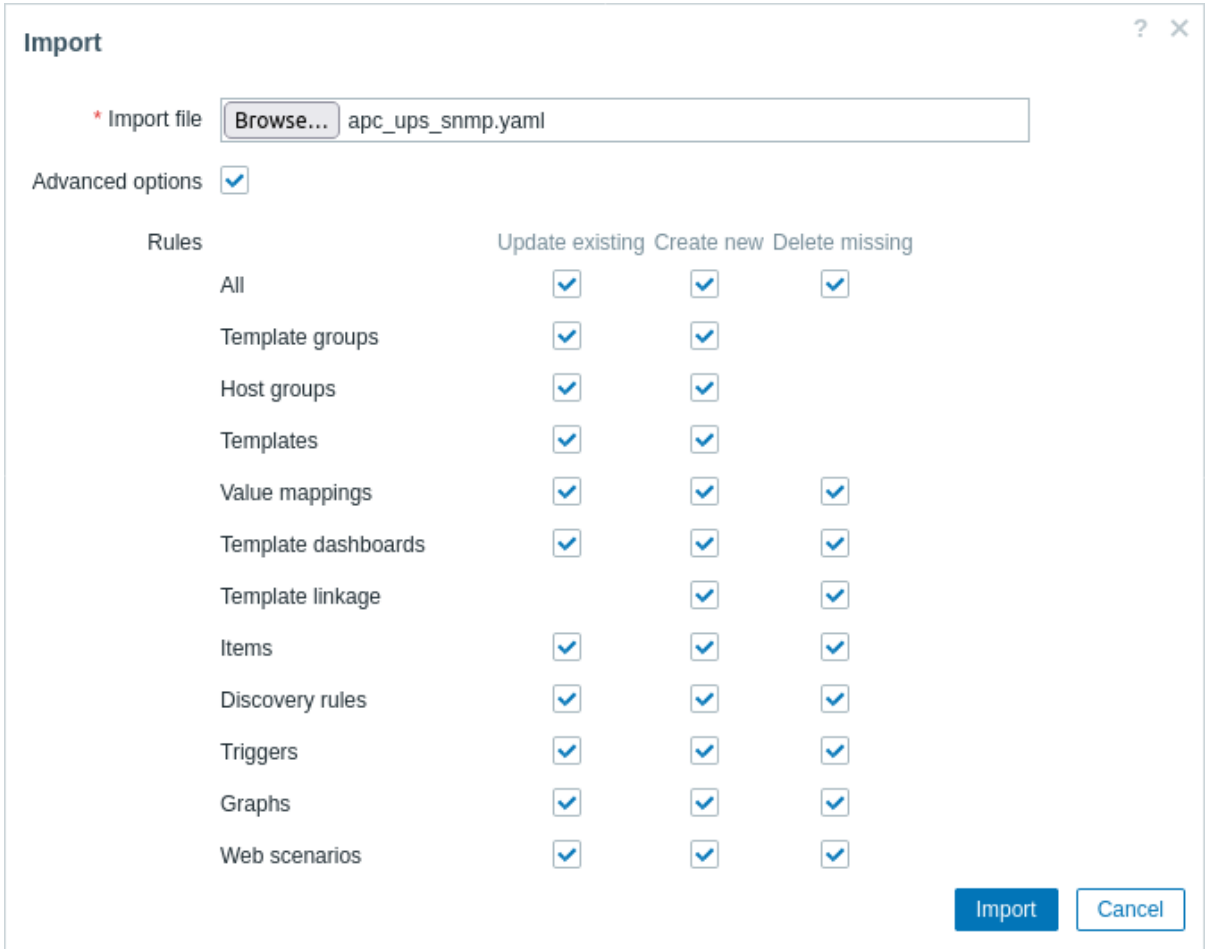
Atkarībā no izvēlētā formāta veidnes tiek eksportētas uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_templates.yaml` - YAML eksportā (noklusējuma eksporta opcija);
- `zabbix_export_templates.xml` - XML eksportā;
- `zabbix_export_templates.json` - JSON eksportā.

Importēšana

Lai importētu veidnes, rīkojieties šādi:

1. Dodieties uz *Datu vākšana* → *Veidnes*.
2. Augšējā labajā stūrī noklikšķiniet uz *Importēt*.
3. Atlasiet importēšanas failu.
4. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Importēt*.



Ja atzīmēsiet izvēles rūtiņu *Papildu opcijas*, tiks parādīts detalizēts visu importējamo elementu saraksts — atzīmējiet vai noņemiet atzīmi katram importēšanas noteikumam pēc nepieciešamības.

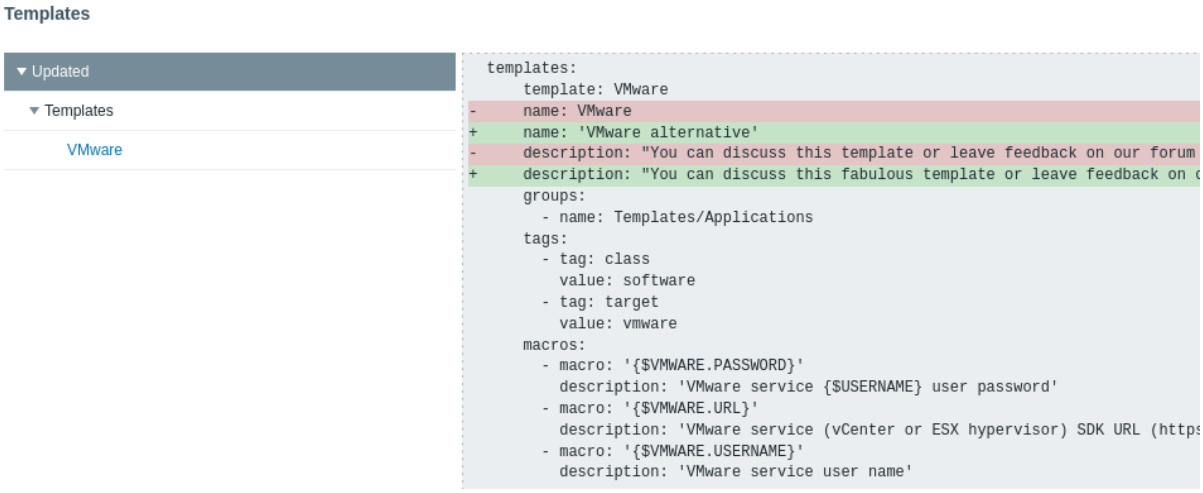
Ja noklikšķināsiet uz izvēles rūtiņas rindā *Visi*, visi zem tās esošie elementi tiks atzīmēti/noņemta to atzīme.

Importēšanas noteikumi:

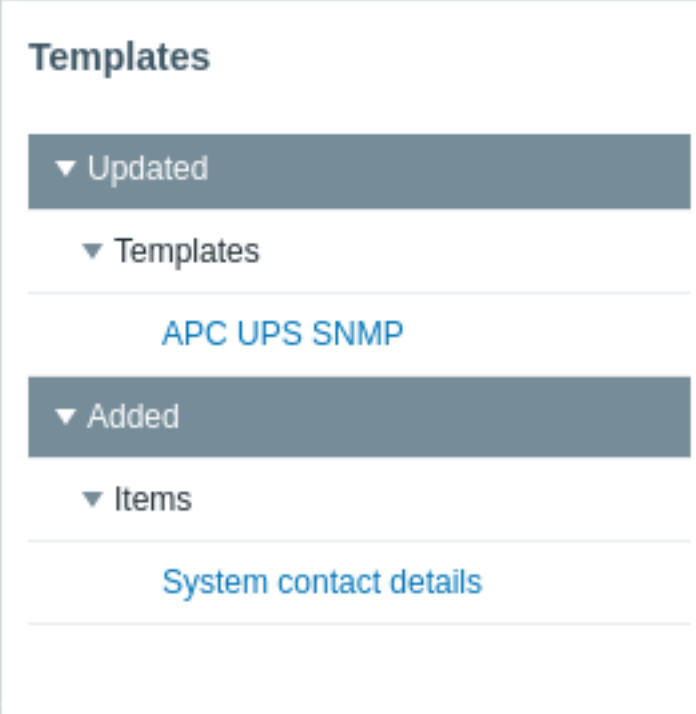
Noteikums	Apraksts
Atjaunināt esošos	Esošie elementi tiks atjaunināti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks atjaunināti.
Izveidot jaunus	Jauni elementi tiks izveidoti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks izveidoti.
Dzēst trūkstošos	Esošie elementi, kas nav iekļauti importēšanas failā, tiks noņemti. Pretējā gadījumā tie netiks noņemti. Ja opcija *Dzēst trūkstošos* ir atzīmēta vienumam *Veidņu sasaiste*, pašreizējā veidņu sasaiste, kas nav iekļauta importēšanas failā, tiks atsaistīta. Entītijas (vienumi, trigeri, grafiki u.c.), kas mantotas no atsaistītajām veidnēm, netiks noņemtas (ja vien opcija *Dzēst trūkstošos* nav atlasīta arī katrai entītijai).

Attention:
Ja veidnes ar tādiem pašiem nosaukumiem jau pastāv, importēšanas laikā ir jāatzīmē opcijas *Dzēst trūkstošos*, lai panāktu tīru importēšanu. Tādējādi vecie vienumi, kas vairs nav atjauninātajā veidnē, tiks noņemti (ņemiet vērā, ka tas nozīmēs šo veco vienumu vēstures zaudēšanu).

Nākamajā ekrānā varēsiet apskatīt importējamās veidnes saturu. Ja tā ir jauna veidne, visi elementi būs uzskaitīti zaļā krāsā. Ja tiek atjaunināta esoša veidne, jaunie veidnes elementi būs izcelti zaļā krāsā; noņemtie veidnes elementi būs izcelti sarkanā krāsā; elementi, kas nav mainījušies, būs uzskaitīti uz pelēka fona.



Izvēlni kreisajā pusē var izmantot, lai pārvietotos pa izmaiņu sarakstu. Sadaļā *Atjaunināts* ir izceltas visas izmaiņas, kas veiktas esošajos veidnes elementos. Sadaļā *Pievienots* ir uzskaitīti jaunie veidnes elementi. Elementi katrā sadaļā ir sagrupēti pēc elementa tipa; noklikšķiniet uz pelēkās bultiņas, lai izvērstu vai sakļautu elementu grupu.



Pārskatiet veidnes izmaiņas un pēc tam noklikšķiniet uz *Importēt*, lai veiktu veidnes importēšanu. Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts Kad veidne tiek eksportēta, Zabbix ģenerē strukturētu YAML, JSON vai XML formātu. Eksports ietver veidnes elementus, piemēram, veidnes metadatus, vienumus, makrosus, trigerus, paneļus un citus.

Katrs elements kalpo noteiktam mērķim un var saturēt ligzdotus elementus.

Tālākajās sadaļās ir aprakstīts katrs elements eksporta formātā. Piemēros tiek izmantota veidne Linux by Zabbix agent.

Daudzpunkte (`...`) norāda uz elementiem, kas īsuma labad ir izlaisti. Piezīme (`skatiet tabulu zemāk`) tiek izmantota, ja elements ir sīkāk izskaidrots turpmākajās sadaļās.

```
zabbix_export:
  version: '8.0'
  template_groups:
    - uuid: 846977d1dfed4968bc5f8bdb363285bc
      name: 'Templates/Operating systems'
  templates: (skatiet tabulu zemāk)
```

```
  triggers: (skatiet tabulu zemāk)
  graphs: (skatiet tabulu zemāk)
```

Veidnes metadati

Elements	Type	Apraksts
version	string	(obligāts) Eksportēšanas versija.
template_groups		(obligāts) Saknes elements veidņu grupām.
uuid	string	(obligāts) Unikāls identifikators šai veidņu grupai.
name	string	(obligāts) Veidņu grupas nosaukums.
host_groups		Saknes elements hostu grupām, kuras izmanto hostu prototipi.
uuid	string	(obligāts) Unikāls identifikators šai hostu grupai.
name	string	(obligāts) Hostu grupas nosaukums.
templates		Saknes elements veidnēm.
triggers		Saknes elements veidnes trigeru elementiem, kas ir tādi paši kā veidnes vienumu trigeriem.
graphs		Saknes elements veidnes grafiku elementiem, kas ir tādi paši kā hostu grafikiem.

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      template: 'Linux by Zabbix agent'
      name: 'Linux by Zabbix agent'
      description: |
        Šī ir oficiāla Linux veidne. Tai ir nepieciešams Zabbix aģents 8.0 vai jaunāks. (...)
      wizard_ready: 'YES'
      readme: |
        ## Pārskats

        Šī ir oficiāla Linux veidne. Tai ir nepieciešams Zabbix aģents 8.0 vai jaunāks. (...)
      vendor:
        name: Zabbix
        version: 8.0-2
      groups:
        - name: 'Templates/Operating systems'
      items: (skatiet tabulu zemāk)
      discovery_rules: (skatiet tabulu zemāk)
      tags:
        - tag: class
          value: os
        - tag: target
          value: linux
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          value: 3m
          description: 'Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu. Darbojas tikai aģentiem,
          config: (skatiet tabulu zemāk)
        - macro: '{$CPU.UTIL.CRIT}'
          value: '90'
          description: 'CPU izmantošanas kritiskais slieksnis, izteikts %.'
          config: (skatiet tabulu zemāk)
        (...)
      dashboards: (skatiet tabulu zemāk)
      valuemaps: (skatiet tabulu zemāk)
    (...)
```

Veidnes

Elements	Type	Apraksts
uuid	string	(obligāts) Unikāls šīs veidnes identifikators.
template	string	(obligāts) Unikāls veidnes nosaukums.
name	string	Redzamais veidnes nosaukums.
description	text	Veidnes apraksts.
wizard_ready	text	Vai veidne ir pieejama atlasei Host Wizard. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1). Skatiet arī: Template object (`wizard_ready`).
readme	text	Veidnei specifiskas konfigurācijas instrukcijas, kas tiek parādītas Host Wizard. Atbalsta Markdown formatējumu.
vendor		Saknes elements veidnes piegādātājam (ir pieejams, ja eksportētā veidne satur piegādātāja datus).
name	string	(obligāts) Veidnes piegādātāja nosaukums.
version	string	(obligāts) Veidnes versija. Standarta veidnēm versija tiek attēlota šādi: Zabbix galvenā versija, atdalītājs (”-”), revīzijas numurs (palielinās ar katru jaunu veidnes versiju un tiek atiestatīts ar katru jaunu Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
templates		Saknes elements saistītajām veidnēm.
name	string	(obligāts) Veidnes nosaukums.
groups		Saknes elements veidņu grupām.
name	string	(obligāts) Veidņu grupas nosaukums.
items		Saknes elements veidnes vienumiem.
discovery_rules		Saknes elements veidnes zema līmeņa atklāšanas kārtulām.
httptests		Saknes elements veidnes tīmekļa scenārijiem.
tags		Saknes elements veidnes tagiem.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
macros		Saknes elements veidnes lietotāja makrosi.
macro	string	(obligāts) Lietotāja makrosa nosaukums.
type	string	Lietotāja makrosa tips. Iespējamās vērtības:[1] TEXT (0, noklusējums), SECRET_TEXT (1), VAULT (2). Skatiet arī: User macro object (`type`).
value	string	Lietotāja makrosa vērtība.
description	string	Lietotāja makrosa apraksts.
config		Saknes elements veidnes makrosa konfigurācijai, kas nosaka, kā makross tiek attēlots Host Wizard.
dashboards		Saknes elements veidnes paneļiem.
valuemaps		Saknes elements veidnes vērtību kartēm.

Veidnes vienumi

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        - uuid: f94f9f4699e94c369e6c98b2a2f485ce
          name: 'Zabbix aģenta ping'
          key: agent.ping
          description: 'Aģents šim vienumam vienmēr atgriež "1". Var izmantot kopā ar `nodata()` pieejamīb
          valuemap:
            name: 'Zabbix aģenta ping statuss'
          tags:
            - tag: component
              value: system
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          name: 'Instalēto pakotņu skaits'
          key: system.sw.packages.get
          delay: 1h
          preprocessing: (skatiet tabulu zemāk)
          tags:
```

```
        - tag: component
          value: os
      triggers: (skatiet tabulu zemāk)
    - uuid: 403cebed115441369e94d35d070ca7b8
      name: 'Atmiņas izmantojums'
      type: DEPENDENT
      key: vm.memory.utilization
      value_type: FLOAT
      units: '%'
      description: 'Izmantotās atmiņas procentuālā daļa tiek aprēķināta kā `100-pavailable`.'
      preprocessing: (skatiet tabulu zemāk)
      master_item:
        key: 'vm.memory.size[pavailable]'
      tags:
        - tag: component
          value: memory
      triggers: (skatiet tabulu zemāk)
    (...)
```

Element	Type	Apraksts
uuid	string	(obligāts) Unikāls šī vienuma identifikators.
name	string	(obligāts) Vienuma nosaukums.
type	string	Vienuma tips. Iespējamās vērtības:[1] ZABBIX_PASSIVE (0, noklusējums), 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	(obligāts SNMP_AGENT vienumiem) SNMP objekta ID.
key	string	(obligāts) Vienuma atslēga.
delay	string	Vienuma atjaunināšanas intervāls. Noklusējums: 1m. TRAP vienumiem šī vērtība vienmēr būs 0.
history	string	Laika periods (izmantojot laika sufiksu, lietotāja makrosu vai LLD makrosu), cik ilgi jāsaglabā vēstures dati. Noklusējums: 31d.
trends	string	Laika periods (izmantojot laika sufiksu, lietotāja makrosu vai LLD makrosu), cik ilgi jāsaglabā tendenču dati. Noklusējums: 365d.
status	string	Vienuma statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
value_type	string	Saņemtās vērtības tips. Iespējamās vērtības:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, noklusējums), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Ar komatiem atdalīts hostu IP adrešu saraksts, kuriem ir atļauts sūtīt datus šim vienumam. Atbalstīts TRAP un HTTP_AGENT vienumiem.
units	string	Saņemtās vērtības mērvienības (bps, B utt.).
params	text	Papildu parametri atkarībā no vienuma tipa (izpildāmais skripts SSH un TELNET vienumiem; SQL vaicājums ODBC vienumiem; formula CALCULATED vienumiem; skripts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem).
ipmi_sensor	string	IPMI sensors. Atbalstīts IPMI vienumiem.
authtype	string	Autentifikācijas tips. Atbalstīts SSH un HTTP_AGENT vienumiem. Iespējamās vērtības SSH vienumiem:[1] PASSWORD (0, noklusējums), PUBLIC_KEY (1). Iespējamās vērtības HTTP_AGENT vienumiem:[1] NONE (0, noklusējums), BASIC (1), NTLM (2), Kerberos (3) vai Digest (4).
username	string	(obligāts SSH un TELNET vienumiem) Lietotājvārds autentifikācijai. Atbalstīts SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Ja tiek izmantots JMX vienumiem, jānorāda arī `password` (skatiet zemāk), vai arī abi elementi jāatstāj tukši.

Element	Type	Apraksts
password	string	(obligāts SSH un TELNET vienumiem) Parole autentifikācijai. Atbalstīts SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Ja tiek izmantots JMX vienumiem, jānorāda arī `username` (skatiet iepriekš), vai arī abi elementi jāatstāj tukši.
publickey	string	(obligāts SSH vienumiem) Publiskās atslēgas faila nosaukums.
privatekey	string	(obligāts SSH vienumiem) Privātās atslēgas faila nosaukums.
description	text	Vienuma apraksts.
inventory_link	string	Hosta inventarizācijas lauks, ko aizpilda šis vienums. Iespējamās vērtības:[1] NONE (0), ALIAS (4) utt. (atbalstītos laukus skatiet Host inventory).
valuemap		Saknes elements vienuma vērtību kartēm.
name	string	(obligāts) Vienumam izmantojamās vērtību kartes nosaukums.
logtimefmt	string	Laika formāts žurnāla ierakstos. Atbalstīts LOG vērtības tipa vienumiem.
preprocessing		Saknes elements vienuma vērtību priekšapstrādei.
step		Saknes elements veidnes vienuma vērtību priekšapstrādes soļiem.
jmx_endpoint	string	JMX galapunkts. Atbalstīts JMX vienumiem.
master_item		(obligāts DEPENDENT vienumiem) Saknes elements atkarīgā vienuma galvenajam vienumam.
key	string	(obligāts) Atkarīgā vienuma galvenā vienuma atslēga.
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Atbalstīts vienumu tipu sarakstam Timeouts.
url	string	(obligāts HTTP_AGENT vienumiem) URL virkne.
query_fields		Saknes elements vaicājuma parametriem. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība. Atbalstīts HTTP_AGENT vienumiem.
parameters		Saknes elements lietotāja definētiem parametriem. Atbalstīts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem.
name	string	(obligāts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem) Lietotāja definēta parametra nosaukums.
value	string	Lietotāja definēta parametra vērtība. Atbalstīts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem.
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Atbalstīts HTTP_AGENT vienumiem.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalstīts HTTP_AGENT vienumiem.
follow_redirects	string	Sekot atbildes pāradresācijām datu aptaujas laikā. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
post_type	string	POST datu pamatteksta tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējums), JSON (2), XML (3).
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Atbalstīts HTTP_AGENT vienumiem.
headers		Saknes elements HTTP(S) pieprasījuma galvenēm. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Galvenes nosaukums.
value	string	(obligāts HTTP_AGENT vienumiem) Galvenes vērtība.
retrieve_mode	string	Kura atbildes daļa jāsaglabā. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] BODY (0, noklusējums), HEADERS (1), BOTH (2).
request_method	string	Pieprasījuma metodes tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] GET (0, noklusējums), POST (1), PUT (2), HEAD (3).
output_format	string	Kā apstrādāt atbildi. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējums), JSON (1).

Element	Type	Apraksts
allow_traps	string	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Atbalstīts HTTP_AGENT vienumiem.
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Atbalstīts HTTP_AGENT vienumiem.
ssl_key_password	string	SSL atslēgas faila parole. Atbalstīts HTTP_AGENT vienumiem.
verify_peer	string	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
verify_host	string	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
tags		Saknes elements vienuma tagiem.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
triggers		Saknes elements veidnes vienuma trigeriem.

Note:
Skatiet arī: Item object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes vienuma vērtības priekšapstrādes soļi

```
zabbix_export:
  (...)
  templates:
    (...)
      items:
        (...)
        - uuid: 58818005e76d46dda14d6592f601ab00
          (...)
          preprocessing:
            - type: JSONPATH
              parameters:
                - $.length()
            - type: DISCARD_UNCHANGED_HEARTBEAT
              parameters:
                - 12h
          (...)
```

Elements	Type	Apraksts
type	string	(obligāti) Vienuma vērtības priekšapstrādes soļa tips. Iespējamās vērtības:[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, aprēķināts: saņemtā vērtība - iepriekšējā vērtība), CHANGE_PER_SECOND (10, aprēķināts: (saņemtā vērtība - iepriekšējā vērtība)/(pašreizējais laiks - pēdējās pārbaudes laiks)), 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		(obligāti) Saknes elements vienuma vērtības priekšapstrādes soļa parametriem.
parameter	string	Atsevišķs vienuma vērtības priekšapstrādes soļa parametrs.
error_handler	string	Darbības tips, ko izmanto priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības:[1] ORIGINAL_ERROR (0, pēc noklusējuma), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).

Elements	Type	Apraksts
error_handler_params	string	Kļūdu apstrādātāja parametri.

Note:
Skatiet arī: Vienuma priekšapstrādes objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes vienumu trigeri

```
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
        (...)
```

Elements	Type	Apraksts
uuid	string	(obligāts) Unikāls šī trigera identifikators.
expression	string	(obligāts) Trigera izteiksme.
recovery_mode	string	Pamats OK notikumu ģenerēšanai. Iespējamās vērtības:[1] EXPRESSION (0, noklusējums), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Trigera atkopšanas izteiksme.
correlation_mode	string	Korelācijas režīms (bez notikumu korelācijas vai notikumu korelācija pēc taga). Iespējamās vērtības:[1] DISABLED (0, noklusējums), TAG_VALUE (1).
correlation_tag	string	Taga nosaukums, kas tiks izmantots notikumu korelācijai.
name	string	(obligāts) Trigera nosaukums.
event_name	string	Notikuma nosaukums.
opdata	string	Operatīvie dati.
url_name	string	Ar trigeri saistītā URL etiķete.
url	string	Ar trigeri saistītais URL.

Elements	Type	Apraksts
status	string	Trigera statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
priority	string	Trigera nopietnība. Iespējamās vērtības:[1] NOT_CLASSIFIED (0, noklusējums), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Trigera apraksts.
type	string	Notikumu ģenerēšanas tips (viens problēmas notikums vai vairāki problēmas notikumi). Iespējamās vērtības:[1] SINGLE (0, noklusējums), MULTIPLE (1).
manual_close	string	Problēmu notikumu manuāla aizvēršana. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
dependencies		Saknes elements atkarībām.
name	string	(obligāts) Atkarīgā trigera nosaukums.
expression	string	(obligāts) Atkarīgā trigera izteiksme.
recovery_expression	string	Atkarīgā trigera atkopšanas izteiksme.
tags		Saknes elements trigera tagiem.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Trigger object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidņu zema līmeņa atklāšanas noteikumi

```
zabbix_export:
  (...)
  templates:
    (...)
      discovery_rules:
        - uuid: acfdea9c46ef48c68e6636f43b8f96a2
          name: 'Tīkla saskarņu atklāšana'
          key: net.if.discovery
          delay: 1h
          filter: (skatiet tabulu zemāk)
          description: 'Tīkla saskarņu atklāšana.'
          item_prototypes: (skatiet tabulu zemāk)
          trigger_prototypes: (skatiet tabulu zemāk)
          graph_prototypes: (skatiet tabulu zemāk)
        (...)
```

Attention:
Lielākā daļa veidnes zema līmeņa atklāšanas noteikumu elementu ir tādi paši kā veidnes vienumiem. Tālāk esošajā tabulā ir aprakstīti tie elementi, kas atšķiras no veidnes vienumiem.

Element	Type	Description
type	string	Vienuma tips. Iespējamās vērtības:[1] ZABBIX_PASSIVE (0, noklusējums), 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	(obligāti) Zema līmeņa atklāšanas noteikuma atslēga.
filter		Saknes elements veidnes zema līmeņa atklāšanas noteikumu filtriem.
lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makrosu), pēc kura vairs neatklātie resursi tiks dzēsti. Noklusējums: 7d.
lifetime_type	string	Scenārijs zaudēto LLD resursu dzēšanai. Iespējamās vērtības: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makrosu), pēc kura vairs neatklātie resursi tiks atspējoti.

Element	Type	Description
enabled_lifetime_type	string	Scenārijs zaudēto LLD resursu atspējošanai. Iespējamās vērtības: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Saknes elements veidnes vienumu prototipu elementiem, kas ir tādi paši kā veidnes vienumiem.
trigger_prototypes		Saknes elements veidnes trigeru prototipu elementiem, kas ir tādi paši kā veidnes vienumu trigeriem.
graph_prototypes		Saknes elements veidnes grafiku prototipu elementiem, kas ir tādi paši kā hostu grafikiem.
host_prototypes		Saknes elements veidnes hostu prototipu elementiem, kas ir tādi paši kā hostiem.
parent_discovery_rule		Saknes elements zema līmeņa atklāšanas noteikuma prototipa vecākajam zema līmeņa atklāšanas noteikumam (vai noteikuma prototipam). Šī īpašība norāda, ka tas ir LLD noteikuma prototips, tiešs atsauktā noteikuma/noteikuma prototipa bērnelements.
key	string	(obligāti) Vecākā zema līmeņa atklāšanas noteikuma (vai noteikuma prototipa) atslēga.
master_item	string	(obligāti DEPENDENT noteikumiem) Saknes elements atkarīgā noteikuma galvenajam vienumam.
lld_macro_paths		Saknes elements zema līmeņa atklāšanas noteikuma makrosu ceļiem.
lld_macro	string	(obligāti) Zema līmeņa atklāšanas noteikuma makrosa nosaukums.
path	string	(obligāti) Selektors vērtībai, kas tiks piešķirta atbilstošajam makrosam.
preprocessing		Saknes elements zema līmeņa atklāšanas noteikuma vērtības priekšapstrādei.
step		Saknes elements zema līmeņa atklāšanas noteikuma vērtības priekšapstrādes soļu elementiem, kas ir tādi paši kā veidnes vienuma vērtības priekšapstrādes soļiem, izņemot ar mazāku iespējamo vērtību skaitu. Skatiet arī: LLD rule preprocessing object.
overrides		Saknes elements zema līmeņa atklāšanas noteikuma pārrakstīšanas noteikumiem.
name	string	(obligāti) Unikāls pārrakstīšanas nosaukums.
step	string	(obligāti) Unikāls pārrakstīšanas secības numurs.
stop	string	Pārtraukt nākamo pārrakstīšanas noteikumu apstrādi, ja ir atbilstība.
filter		Saknes elements veidnes zema līmeņa atklāšanas noteikuma pārrakstīšanas noteikuma filtra elementiem, kas ir tādi paši kā veidnes zema līmeņa atklāšanas noteikumu filtriem.
operations		Saknes elements veidnes zema līmeņa atklāšanas noteikuma pārrakstīšanas darbībām.

Note:
Skatiet arī: LLD rule object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes zema līmeņa atklāšanas kārtulas filtri

```
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
        (...)
```

Elements	Type	Apraksts
evaltype	string	Filtra nosacījumu novērtēšanas metodes pārrakstīšana. Iespējamās vērtības:[1] AND_OR (0, noklusējums), AND (1), OR (2), FORMULA (3).
formula	string	Pielāgota filtra nosacījumu aprēķina formula.
conditions		Saknes elements filtra nosacījumiem.
macro	string	(obligāti) Zema līmeņa atklāšanas makrosa nosaukums, kuram veikt pārbaudi.
value	string	Vērtība, ar ko salīdzināt.
operator	string	Nosacījuma operators. Iespējamās vērtības:[1] MATCHES_REGEX (8, noklusējums), NOT_MATCHES_REGEX (9).
formulaid	string	(obligāti) Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu no pielāgotās izteiksmes. Var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču tas tiks ģenerēts no jauna, pieprasot tos vēlāk.

Note:
Skatiet arī: LLD kārtulas filtra objektu (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes zema līmeņa atklāšanas kārtulas ignorēšanas operācijas

Elements	Type	Apraksts
operationobject	string	Objekts, kuram tiek piemērota operācija. Iespējamās vērtības:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3).
operator	string	Ignorēšanas nosacījuma operators. Iespējamās vērtības:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Regulārā izteiksme vai virkne ignorēšanas nosacījuma operatoram.
status	string	Objekta statuss pēc ignorēšanas operācijas.
discover	string	Vai objekts tiek pievienots atklāšanas rezultātā.
delay	string	Atjaunināšanas intervāls, kas iestatīts vienuma prototipam ignorēšanas operācijas laikā.
history	string	Vēstures glabāšanas periods, kas iestatīts vienuma prototipam ignorēšanas operācijas laikā.
trends	string	Tendenču glabāšanas periods, kas iestatīts vienuma prototipam ignorēšanas operācijas laikā.
severity	string	Trigera prototipa būtiskuma līmenis, kas iestatīts ignorēšanas operācijas laikā.
tags		Saknes elements birkām, kas iestatītas objektam ignorēšanas operācijas laikā.
tag	string	(obligāti) Birkas nosaukums.
value	string	Birkas vērtība.
templates		Saknes elements veidnēm, kas saistītas ar hosta prototipu ignorēšanas operācijas laikā.
name	string	(obligāti) Veidnes nosaukums.
inventory_mode	string	Hosta prototipa inventarizācijas režīms, kas iestatīts ignorēšanas operācijas laikā.

Note:
Skatiet arī: LLD rule override operation object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes tīmekļa scenāriji

Element	Type	Description
uuid	string	(obligāti) Unikāls šī tīmekļa scenārija identifikators.
name	string	(obligāti) Tīmekļa scenārija nosaukums.
delay	string	Tīmekļa scenārija izpildes biežums (izmantojot sekundes, laika sufiksu vai lietotāja makrosu). Noklusējums: 1m.
attempts	integer	Tīmekļa scenārija soļu izpildes mēģinājumu skaits. Iespējamās vērtības:[1] 1-10 (noklusējums: 1).

Element	Type	Description
agent	string	Klienta aģents. Zabbix uzdosies par izvēlēto pārlūku. Tas ir noderīgi, ja vietne dažādām pārlūkprogrammām atgriež atšķirīgu saturu. Noklusējums: Zabbix.
http_proxy	string	Starpniekserveris, kas tiks izmantots tīmekļa scenārijam, norādot to šādā formā: `http://[username[:password]@]proxy.example.com[:port]`
variables		Saknes elements tīmekļa scenārija mainīgajiem, kurus var izmantot scenārija soļos.
name	string	(obligāti) Mainīgā nosaukums.
value	text	(obligāti) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas tiks nosūtītas, izpildot pieprasījumu. Galvenes jānorāda, izmantojot to pašu sintaksi, kāda tiktu izmantota HTTP protokolā.
name	string	(obligāti) Galvenes nosaukums.
value	text	(obligāti) Galvenes vērtība.
status	string	Tīmekļa scenārija statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
authentication	string	Autentifikācijas metode. Iespējamās vērtības:[1] NONE (0, noklusējums), BASIC (1), NTLM (2), Kerberos (3) vai Digest (4).
http_user	string	Lietotājvārds, kas tiek izmantots BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
http_password	string	Parole, kas tiek izmantota BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
verify_peer	string	Pārbaudīt tīmekļa servera SSL sertifikātu. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
verify_host	string	Pārbaudīt, vai tīmekļa servera sertifikāta lauks *Common Name* vai lauks *Subject Alternate Name* atbilst. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
ssl_cert_file	string	SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai (jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai (jābūt PEM formātā).
ssl_key_password	string	SSL privātās atslēgas faila parole.
steps		(obligāti) Saknes elements veidnes tīmekļa scenārija soļiem.
tags		Saknes elements tīmekļa scenārija tagiem.
tag	string	(obligāti) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Tīmekļa scenārija objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes tīmekļa scenārija soļi

Elements	Type	Apraksts
name	string	(obligāti) Tīmekļa scenārija soļa nosaukums.
url	string	(obligāti) URL uzraudzībai.
query_fields		Saknes elements vaicājuma parametriem (HTTP lauku masīvs, kas tiek pievienots URL, izpildot pieprasījumu).
name	string	(obligāti) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība.
posts		Saknes elements HTTP POST mainīgajiem (virkne (neapstrādāti POST dati) vai HTTP lauku masīvs (veidlapas lauku dati)).
name	string	(obligāti) POST lauka nosaukums.
value	string	(obligāti) POST lauka vērtība.
variables		Soļa līmeņa mainīgo (makrosu) saknes elements, kas jāpiemēro pēc šī soļa. Ja mainīgā vērtībai ir prefikss 'regex:', tad tā vērtība tiek iegūta no šī soļa atgrieztajiem datiem atbilstoši regulārās izteiksmes šablonam, kas seko aiz prefiksa 'regex:'
name	string	(obligāti) Mainīgā nosaukums.
value	text	(obligāti) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas jānosūta, izpildot pieprasījumu.
name	string	(obligāti) Galvenes nosaukums.
value	text	(obligāti) Galvenes vērtība.

Elements	Type	Apraksts
follow_redirects	string	Sekot HTTP pāradresācijām. Iespējamās vērtības:[1] NO (0), YES (1, pēc noklusējuma).
retrieve_mode	string	HTTP atbildes iegūšanas režīms. Iespējamās vērtības:[1] BODY (0, pēc noklusējuma), HEADERS (1), BOTH (2).
timeout	string	Soļa izpildes noildze (izmantojot sekundes, laika sufiksu vai lietotāja makrosu). Pēc noklusējuma: 15s.
required	string	Teksts, kam jābūt atbildē (tiek ignorēts, ja tukšs).
status_codes	string	Ar komatiem atdalīts pieņemamo HTTP statusa kodu saraksts (piemēram, 200-201,210-299; tiek ignorēts, ja tukšs).

Note:
Skatiet arī: Web scenario step object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes makro konfigurācija

```
zabbix_export:
  (...)
  templates:
    - uuid: f8f7908280354f2abeed07dc788c3747
      (...)
      macros:
        - macro: '{$AGENT.TIMEOUT}'
          (...)
          config:
            type: TEXT
            priority: '1'
            label: 'Sekundes kopš pēdējās redzētās Zabbix aģenta aktivitātes'
            description: 'Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu.'
        - macro: '{$CPU.UTIL.CRIT}'
          (...)
          config:
            type: TEXT
            priority: '2'
            section_name: Sliekšņi
            label: 'CPU izmantošanas sliekšņa izteiksme'
            description: 'Kritiskais CPU izmantošanas slieksnis, izteikts %. Diapazonā no 0 līdz 100 ieska
            regex: '^-?([0-9]+|(([0-9]+)\.([0-9]+)))$'
        (...)
        - macro: '{$IFCONTROL}'
          (...)
          config:
            type: CHECKBOX
            priority: '19'
            label: 'Saskarņu kontrole'
            description: 'Aktivizēt trigeri, kad saskarnes darbības statuss mainās uz "Link down".'
            options:
              - checked: '1'
                unchecked: '0'
        (...)
```

Elements	Type	Apraksts
type	string	(obligāts) Makro ievades lauka tips. Iespējamās vērtības:[1] NOCONF (0), TEXT (1), LIST (2), CHECKBOX (3).
priority	string	Makro pozīcija makro sarakstā.
section_name	string	Sakļaujamās sadaļas etiķete, kurā makro ir grupēts.
label	string	(obligāts TEXT, LIST, CHECKBOX makro) Makro etiķete.
description	text	Makro palīdzības teksts. Atbalsta Markdown formatējumu.
required	string	Atzīmē makro kā obligātu. Iespējamās vērtības:[1] NO (0), YES (1). Atbalstīts TEXT un LIST makro.

Elements	Type	Apraksts
regex	string	Regulārā izteiksme lietotāja ievades validēšanai teksta lauka laukā. Atbalstīts TEXT makro.
options		Saknes elements LIST vienumiem vai CHECKBOX vērtībām.
value	string	(obligāts LIST makro) LIST vienuma vērtība.
text	string	(obligāts LIST makro) LIST vienuma teksts.
checked	string	(obligāts CHECKBOX makro) Vērtība, kas apzīmē atzīmētu stāvokli. Iespējamās vērtības:[1] 0 (false), 1 (true). Atbalstīts CHECKBOX makro.
unchecked	string	(obligāts CHECKBOX makro) Vērtība, kas apzīmē neatzīmētu stāvokli. Iespējamās vērtības:[1] 0 (false), 1 (true). Atbalstīts CHECKBOX makro.

Note:
Skatiet arī: Macro configuration object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidņu paneļi

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets: (skatiet tabulu zemāk)
        (...)
```

Elements	Tips	Apraksts
uuid	string	(obligāts) Unikāls šī paneļa identifikators.
name	string	(obligāts) Veidnes paneļa nosaukums.
display pe-riod	integer	Paneļa lapu attēlošanas periods.
auto_start	string	Slīdrādes automātiska palaišana. Iespējamās vērtības:[1] NO (0), YES (1, pēc noklusējuma).
pages		Veidnes paneļa lapu saknes elements.
name	string	Lapas nosaukums.
display period	integer	Lapas attēlošanas periods.
sortorder	integer	Lapas kārtošanas secība.
widgets		Veidnes paneļa logrīku saknes elements.

Note:
Skatiet arī: Veidnes paneļa objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidnes paneļa logrīki

```
zabbix_export:
  (...)
  templates:
    (...)
      dashboards:
        - uuid: c689ad3115fd46a4b927d1f70ee2e5a4
          name: Filesystems
          pages:
            - name: Overview
              widgets:
```

```
              - type: graphprototype
                width: '72'
                height: '5'
                fields:
                  - type: INTEGER
                    name: columns
                    value: '3'
                  - type: GRAPH_PROTOTYPE
                    name: graphid.0
                    value:
                      host: 'Linux by Zabbix agent'
                      name: 'FS [{#FSTYPE}({#FSNAME})]: Space utilization chart (relative to total)'
                  - type: STRING
                    name: reference
                    value: AAAEB
              - type: graphprototype
                'y': '5'
                width: '72'
                height: '5'
                fields:
                  - type: INTEGER
                    name: columns
                    value: '1'
                  - type: GRAPH_PROTOTYPE
                    name: graphid.0
                    value:
                      host: 'Linux by Zabbix agent'
                      name: 'FS [{#FSTYPE}({#FSNAME})]: Space usage graph, in % (relative to max availab
                  - type: STRING
                    name: reference
                    value: AAAEC
        (...)
```

Element	Type	Description
type	string	(obligāti) Logrīka tips.
name	string	Logrīka nosaukums.
x	integer	Horizontālā pozīcija no veidnes paneļa kreisās malas. Iespējamās vērtības:[1] 0-71.
y	integer	Vertikālā pozīcija no veidnes paneļa augšējās malas. Iespējamās vērtības:[1] 0-63.
width	integer	Logrīka platums. Iespējamās vērtības:[1] 1-72.
height	integer	Logrīka augstums. Iespējamās vērtības:[1] 1-64.
hide_header	string	Paslēpt logrīka galveni. Iespējamās vērtības:[1] NO (0, pēc noklusējuma), YES (1).
fields		Saknes elements veidnes paneļa logrīka laukiem.
type	string	(obligāti) Logrīka lauka tips. Iespējamās vērtības:[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	(obligāti) Logrīka lauka nosaukums.
value	mixed	(obligāti) Logrīka lauka vērtība atkarībā no lauka tipa.

Note:
Skatiet arī: Template dashboard widget object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Veidņu vērtību kartes

```
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
      (...)
```

Elements	Type	Apraksts
uuid	string	(obligāts) Unikāls šīs vērtību kartes identifikators.
name	string	(obligāts) Vērtību kartes nosaukums.
mapping		Saknes elements kartējumiem.
type	string	Kartējuma atbilstības tips. Iespējamās vērtības:[1] EQUAL (0, noklusējums), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Sākotnējā vērtība.
newvalue	string	(obligāts) Vērtība, uz kuru tiek kartēta sākotnējā vērtība.

Note:
Skatiet arī: Value map object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Piezīmes [1] API veselo skaitļu vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai atsaucei. Plašāku informāciju skatiet saistītajā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

4 Hosti

Pārskats

Hosti tiek eksportēti kopā ar daudziem saistītiem objektiem un objektu attiecībām.

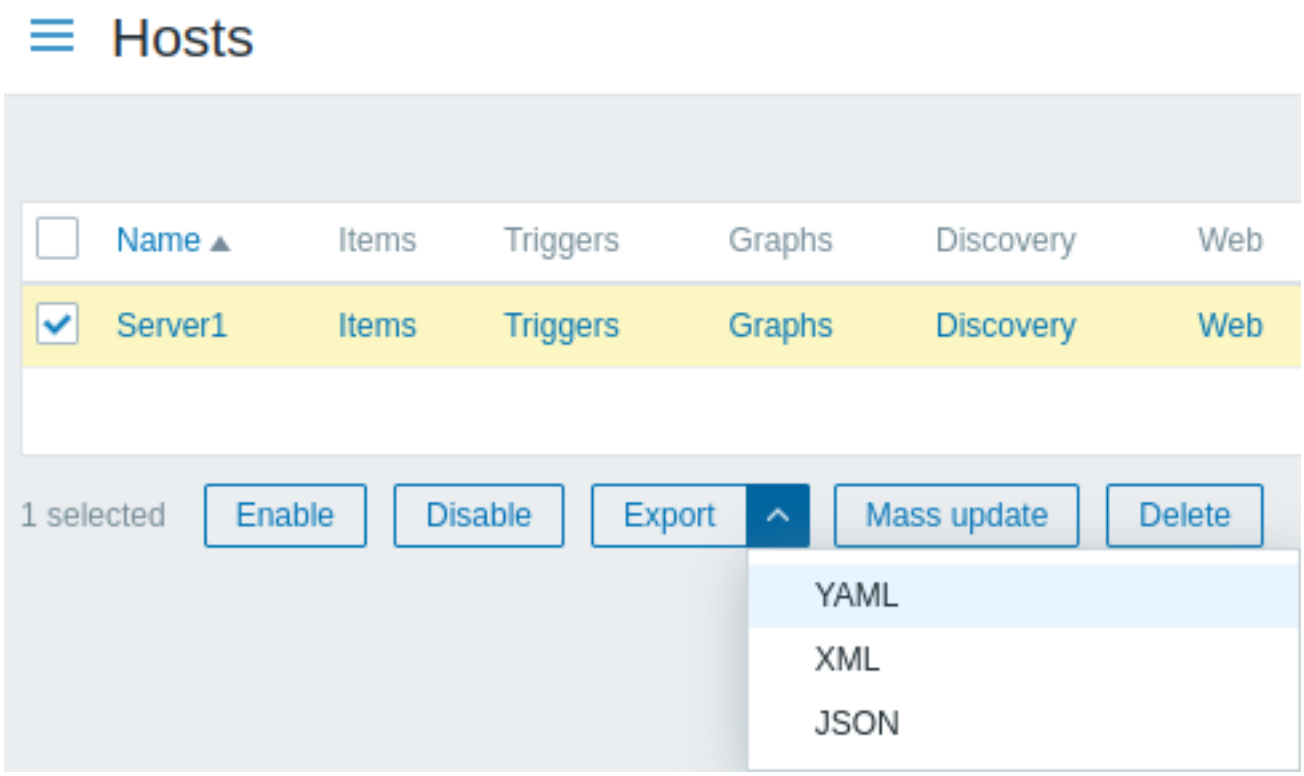
Hosta eksports satur:

- Saistītās hostu grupas
- Hosta datus
- Veidņu saites
- Hosta grupu saites
- Hosta saskarnes
- Tieši saistītos vienumus
- Tieši saistītos trigerus
- Tieši saistītos atklāšanas noteikumus ar visiem prototipiem
- Tieši saistītos tīmekļa scenārijus
- Hosta makrosus
- Hosta inventāra datus
- Vērtību kartes
- Saistītos grafikus

Eksportēšana

Lai eksportētu hostus, rīkojieties šādi:

1. Atveriet *Datu ievākšana* → *Hosti*.
2. Atzīmējiet eksportējamo hostu izvēles rūtiņas.
3. Zem saraksta noklikšķiniet uz *Eksportēt*.



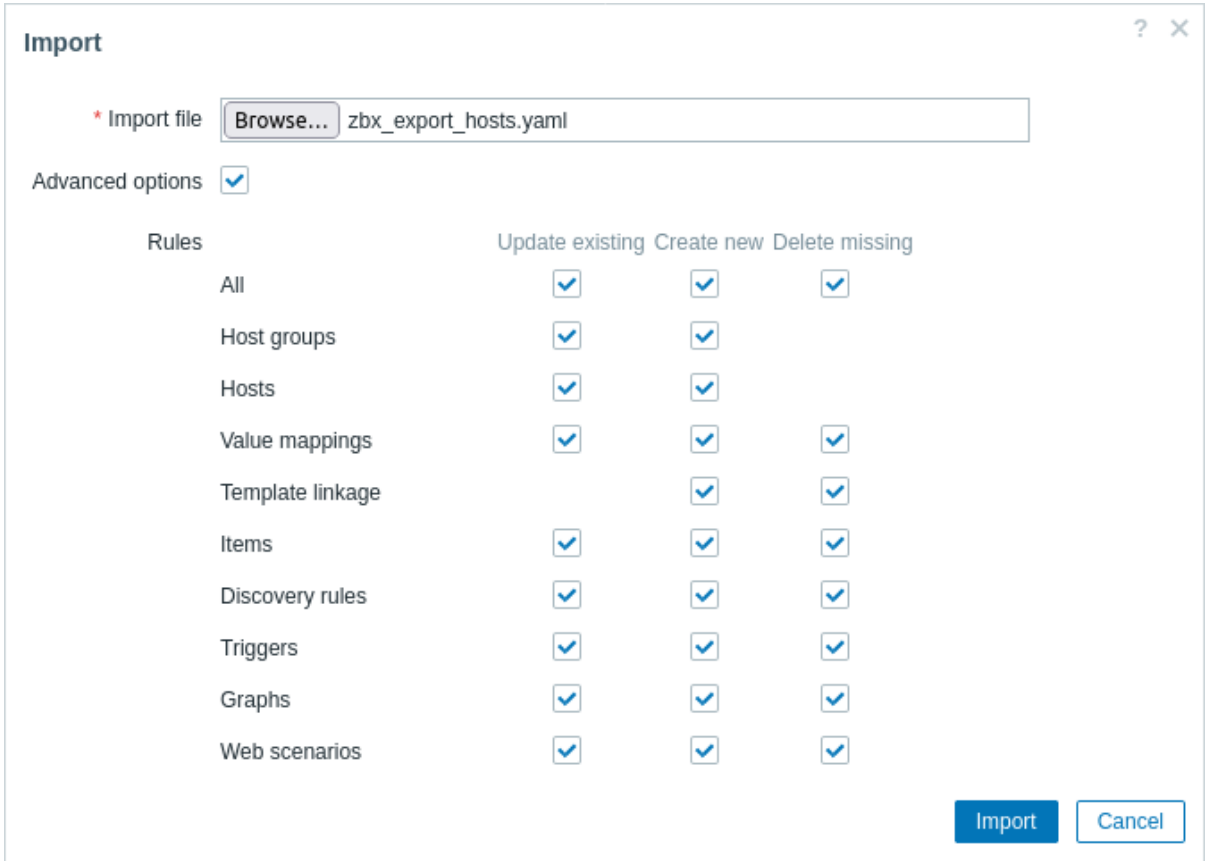
Atkarībā no izvēlētā formāta hosti tiek eksportēti uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_hosts.yaml` - YAML eksportā (noklusējuma eksporta opcija);
- `zabbix_export_hosts.xml` - XML eksportā;
- `zabbix_export_hosts.json` - JSON eksportā.

Importēšana

Lai importētu hostus, rīkojieties šādi:

1. Atveriet *Datu ievākšana* → *Hosti*.
2. Augšējā labajā stūrī noklikšķiniet uz *Importēt*.
3. Atlasiet importēšanas failu.
4. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Importēt*.



Ja atzīmēsiet izvēles rūtiņu *Papildu opcijas*, tiks parādīts detalizēts visu importējamo elementu saraksts — atzīmējiet vai noņemiet atzīmi katram importēšanas noteikumam pēc nepieciešamības.

Ja noklikšķināsiet uz izvēles rūtiņas rindā *Visi*, visi zem tās esošie elementi tiks atzīmēti/noņemta to atzīme.

Importēšanas noteikumi:

Noteikums	Apraksts
Atjaunināt esošos	Esošie elementi tiks atjaunināti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks atjaunināti.
Izveidot jaunus	Jauni elementi tiks izveidoti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks izveidoti.
Dzēst trūkstošos	Esošie elementi, kas nav importēšanas failā, tiks noņemti. Pretējā gadījumā tie netiks noņemti. Ja opcija *Dzēst trūkstošos* ir atzīmēta vienumam *Veidņu piesaiste*, pašreizējā veidņu piesaiste, kas nav importēšanas failā, tiks atsaistīta. Entītijas (vienumi, trigeri, grafiki u.c.), kas mantotas no atsaistītajām veidnēm, netiks noņemtas (ja vien opcija *Dzēst trūkstošos* nav atlasīta arī katrai entītijai atsevišķi).

Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts

Kad hosts tiek eksportēts, Zabbix ģenerē strukturētu YAML, JSON vai XML formātu. Eksports ietver hosta elementus, piemēram, hosta metadatus, vienumus, makrosus, trigerus un citus.

Katrs elements kalpo noteiktam mērķim un var saturēt ligzdotus elementus.

Tālākajās sadaļās ir aprakstīts katrs elements eksporta formātā. Piemēros tiek izmantots hosts ar veidni MySQL by Zabbix agent 2. Elementu demonstrēšanas nolūkos veidne pēc hosta izveides tiek atsaistīta. Atsaistīšana saglabā visus elementus konfigurācijā (atšķirībā no atsaistīšanas un notīrīšanas, kas tos noņem).

Daudzpunkte (`...`) norāda uz elementiem, kas īsuma labad ir izlaisti. Piezīme (`see table below`) tiek izmantota, ja elements ir sīkāk izskaidrots turpmākajās sadaļās.

```
zabbix_export:
  version: '8.0'
```

```
  host_groups:
    - uuid: 748ad4d098d447d492bb935c907f652f
      name: Databases
  hosts: (skatiet tabulu zemāk)
  graphs: (skatiet tabulu zemāk)
```

Hosta metadati

Elements		Type	Apraksts
version		string	(obligāti) Zabbix versija.
host_groups			(obligāti) Saknes elements hostu grupām.
	uuid	string	(obligāti) Unikāls identifikators šai hostu grupai.
	name	string	(obligāti) Hostu grupas nosaukums.
hosts			Saknes elements hostiem.
graphs			Saknes elements hostu grafikiem.

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      name: 'MySQL server'
      groups:
        - name: Databases
      interfaces: (skatiet tabulu zemāk)
      items: (skatiet tabulu zemāk)
      discovery_rules: (skatiet tabulu zemāk)
      tags:
        - tag: location
          value: Riga
      macros:
        (...)
        - macro: '{$MYSQL.DSN}'
          value: 192.0.2.0
        (...)
        - macro: '{$MYSQL.PASSWORD}'
          type: SECRET_TEXT
          description: 'MySQL lietotāja parole.'
        (...)
      valuemaps: (skatiet tabulu zemāk)
  (...)
```

Hosti

Element	Type	Apraksts
host	string	(obligāti) Unikāls hosta nosaukums.
name	string	Redzamais hosta nosaukums.
description	text	Hosta apraksts.
monitored_by	string	Kā hosts tiek uzraudzīts. Iespējamās vērtības:[1] SERVER (0, noklusējums), PROXY (1) vai PROXY_GROUP (2).
proxy		Saknes elements starpniekserverim.
name	string	(obligāti) Tā starpniekservera nosaukums (ja tāds ir), kas uzrauga hostu.
proxy_group		Saknes elements starpniekserveru grupai.
name	string	(obligāti) Tās starpniekserveru grupas nosaukums (ja tāda ir), kas tiek izmantota hosta uzraudzībai.
status	string	Hosta statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
ipmi_authtype	string	IPMI sesijas autentifikācijas tips. Iespējamās vērtības:[1] DEFAULT (-1, noklusējums), NONE (0), MD2 (1), MD5 (2), STRAIGHT (4), OEM (5), RMCP_PLUS (6).

Element	Type	Apraksts
ipmi_privilege	string	IPMI sesijas privilēģiju līmenis. Iespējamās vērtības:[1] CALLBACK (1), USER (2, noklusējums), OPERATOR (3), ADMIN (4), OEM (5).
ipmi_username	string	Lietotājvārds IPMI pārbaudēm.
ipmi_password	string	Parole IPMI pārbaudēm.
templates		Saknes elements piesaistītajām veidnēm.
name	string	(obligāti) Veidnes nosaukums.
groups		Saknes elements hostu grupām, kurām hosts pieder.
name	string	(obligāti) Hostu grupas nosaukums.
interfaces		Saknes elements hosta saskarnēm.
items		Saknes elements hosta vienumiem.
discovery_rules		Saknes elements hosta zema līmeņa atklāšanas noteikumiem.
http_tests		Saknes elements hosta tīmekļa scenārijiem.
tags		Saknes elements hosta tagiem.
tag	string	(obligāti) Taga nosaukums.
value	string	Taga vērtība.
macros		Saknes elements hosta makrosi.
macro	string	(obligāti) Lietotāja makrosa nosaukums.
type	string	Lietotāja makrosa tips. Iespējamās vērtības:[1] TEXT (0, noklusējums), SECRET_TEXT (1), VAULT (2).
value	string	Lietotāja makrosa vērtība.
description	string	Lietotāja makrosa apraksts.
inventory		Saknes elements hosta inventarizācijai.
<inventory_property>	string	Inventarizācijas rekvizīts. Visiem rekvizītiem ir atbilstošs elements (`type`, `name`, `os` utt.; piemēru skatiet Eksporta formātā).
inventory_mode	string	Inventarizācijas režīms. Iespējamās vērtības:[1] DISABLED (-1), MANUAL (0, noklusējums), AUTOMATIC (1).
valuemaps		Saknes elements hosta vērtību kartēm.

Note:
Skatiet arī: Host object (skatiet atbilstošo rekvizītu ar tādu pašu nosaukumu).

Hostu saskarnes

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      interfaces:
        - ip: 192.0.2.0
          interface_ref: if1
      (...)
```

Elements	Type	Apraksts
default	string	Vai šī ir primārā hosta saskarne. Ņemiet vērā, ka hostam var būt tikai viena viena tipa primārā saskarne. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
type	string	Saskarnes tips. Iespējamās vērtības:[1] ZABBIX (1, noklusējums), SNMP (2), IPMI (3), JMX (4).
useip	string	Vai izmantot IP kā saskarni savienojumam ar hostu (pretējā gadījumā tiks izmantots DNS). Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
ip	string	(obligāts IP savienojumiem) IP adrese (IPv4 vai IPv6).
dns	string	(obligāts DNS savienojumiem) DNS nosaukums.
port	string	Porta numurs.
details		Saknes elements saskarnes detalizētai informācijai.
version	string	Izmantot šo SNMP versiju. Iespējamās vērtības:[1] SNMPV1 (1), SNMP_V2C (2, noklusējums), SNMP_V3 (3).
community	string	(obligāts SNMPv1 un SNMPv2 vienumiem) SNMP kopiena.

Elements	Type	Apraksts
max_repetitions	string	Maksimālā atkārtojumu vērtība vietējiem SNMP lielapjoma pieprasījumiem (GetBulkRequest-PDUs). Atbalstīts SNMPv2 un SNMPv3 vienumiem (`discovery[]` un `walk[]` vienumi). Noklusējums: 10.
contextname	string	SNMPv3 konteksta nosaukums. Atbalstīts SNMPv3 vienumiem.
securityname	string	SNMPv3 drošības nosaukums. Atbalstīts SNMPv3 vienumiem.
securitylevel	string	SNMPv3 drošības līmenis. Atbalstīts SNMPv3 vienumiem. Iespējamās vērtības:[1] NOAUTHNOPRIV (0, noklusējums), AUTHNOPRIV (1), AUTHPRIV (2).
authprotocol	string	SNMPv3 autentifikācijas protokols. Atbalstīts SNMPv3 vienumiem. Iespējamās vērtības:[1] MD5 (0, noklusējums), SHA1 (1), SHA224 (2), SHA256 (3), SHA384 (4), SHA512 (5).
authpassphrase	string	SNMPv3 autentifikācijas parole. Atbalstīts SNMPv3 vienumiem.
privprotocol	string	SNMPv3 privātuma protokols. Atbalstīts SNMPv3 vienumiem. Iespējamās vērtības:[1] DES (0, noklusējums), AES128 (1), AES192 (2), AES256 (3), AES192C (4), AES256C (5).
privpassphrase	string	SNMPv3 privātuma parole. Atbalstīts SNMPv3 vienumiem.
bulk	string	Izmantot SNMP lielapjoma pieprasījumus. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
interface_ref	string	Saskarnes atsauces nosaukums, ko izmantot vienumos (formāts: `if<N>`).

Note:
Skatiet arī: Host interface object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hostu vienumi

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL serveris'
      (...)
      items:
        (...)
        - name: 'Binlog cache disk use'
          type: DEPENDENT
          key: mysql.binlog_cache_disk_use
          value_type: FLOAT
          description: 'Darījumu skaits, kas izmantoja pagaidu diska kešatmiņu, jo tie neietilpa parastajā
          preprocessing: (skatiet tabulu zemāk)
          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: 'Izmantoto lapu attiecība pret kopējo lapu skaitu buferu pūlā.'
        tags:
          - tag: component
            value: memory
        triggers: (skatiet tabulu zemāk)
      (...)
      - name: Uptime
        type: DEPENDENT
        key: mysql.uptime
        units: uptime
        description: 'Sekunžu skaits, cik ilgi serveris darbojas.'
        preprocessing: (skatiet tabulu zemāk)
        master_item:
          key: 'mysql.get_status_variables["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
        tags:
          - tag: component
            value: application
        triggers: (skatiet tabulu zemāk)
      (...)
```

Element	Type	Apraksts
name	string	(obligāts) Vienuma nosaukums.
type	string	Vienuma tips. Iespējamās vērtības:[1] ZABBIX_PASSIVE (0, noklusējums), 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	(obligāts SNMP_AGENT vienumiem) SNMP objekta ID.
key	string	(obligāts) Vienuma atslēga.
delay	string	Vienuma atjaunināšanas intervāls. Noklusējums: 1m. Vērtība vienmēr būs 0 TRAP vienumiem.
history	string	Laika periods (izmantojot laika sufiksu, lietotāja makrosu vai LLD makrosu), cik ilgi jāsaglabā vēstures dati. Noklusējums: 31d.
trends	string	Laika periods (izmantojot laika sufiksu, lietotāja makrosu vai LLD makrosu), cik ilgi jāsaglabā tendenču dati. Noklusējums: 365d.
status	string	Vienuma statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
value_type	string	Saņemtās vērtības tips. Iespējamās vērtības:[1] FLOAT (0), CHAR (1), LOG (2), UNSIGNED (3, noklusējums), TEXT (4), BINARY (5), JSON (6).
allowed_hosts	string	Ar komatiem atdalīts hostu IP adrešu saraksts, kuriem ir atļauts sūtīt datus šim vienumam. Atbalstīts TRAP un HTTP_AGENT vienumiem.
units	string	Saņemtās vērtības mērvienības (bps, B utt.).
params	text	Papildu parametri atkarībā no vienuma tipa (izpildāmais skripts SSH un TELNET vienumiem; SQL vaicājums ODBC vienumiem; formula CALCULATED vienumiem; skripts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem).
ipmi_sensor	string	IPMI sensors. Atbalstīts IPMI vienumiem.
authtype	string	Autentifikācijas tips. Atbalstīts SSH un HTTP_AGENT vienumiem. Iespējamās vērtības SSH vienumiem:[1] PASSWORD (0, noklusējums), PUBLIC_KEY (1). Iespējamās vērtības HTTP_AGENT vienumiem:[1] NONE (0, noklusējums), BASIC (1), NTLM (2), Kerberos (3) vai Digest (4).

Element	Type	Apraksts
username	string	(obligāts SSH un TELNET vienumiem) Lietotājvārds autentifikācijai. Atbalstīts SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Ja tiek izmantots JMX vienumiem, jānorāda arī `password` (skatiet zemāk), vai arī abi elementi jāatstāj tukši.
password	string	(obligāts SSH un TELNET vienumiem) Parole autentifikācijai. Atbalstīts SIMPLE, ODBC, JMX un HTTP_AGENT vienumiem. Ja tiek izmantots JMX vienumiem, jānorāda arī `username` (skatiet iepriekš), vai arī abi elementi jāatstāj tukši.
publickey	string	(obligāts SSH vienumiem) Publiskās atslēgas faila nosaukums.
privatekey	string	(obligāts SSH vienumiem) Privātās atslēgas faila nosaukums.
description	text	Vienuma apraksts.
inventory_link	string	Hosta inventarizācijas lauks, ko aizpilda vienums. Iespējamās vērtības:[1] NONE (0), ALIAS (4) utt. (atbalstītos laukus skatiet Host inventory).
valuemap		Saknes elements vienuma vērtību kartēm.
name	string	(obligāts) Izmantojamās vērtību kartes nosaukums šim vienumam.
logtimefmt	string	Laika formāts žurnāla ierakstos. Atbalstīts vienumiem ar LOG vērtības tipu.
preprocessing		Saknes elements vienuma vērtības priekšapstrādei.
step		Saknes elements hosta vienuma vērtības priekšapstrādes soļiem.
interface_ref	string	Atsauce uz hosta saskarni (formāts: `if<N>`).
jmx_endpoint	string	JMX galapunkts. Atbalstīts JMX vienumiem.
master_item		(obligāts DEPENDENT vienumiem) Saknes elements atkarīgā vienuma galvenajam vienumam.
key	string	(obligāts) Atkarīgā vienuma galvenā vienuma atslēga.
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Atbalstīts vienumu tipu sarakstam Timeouts.
url	string	(obligāts HTTP_AGENT vienumiem) URL virkne.
query_fields		Saknes elements vaicājuma parametriem. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība. Atbalstīts HTTP_AGENT vienumiem.
parameters		Saknes elements lietotāja definētiem parametriem. Atbalstīts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem.
name	string	(obligāts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem) Lietotāja definētā parametra nosaukums.
value	string	Lietotāja definētā parametra vērtība. Atbalstīts ITEM_TYPE_SCRIPT un ITEM_TYPE_BROWSER vienumiem.
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Atbalstīts HTTP_AGENT vienumiem.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalstīts HTTP_AGENT vienumiem.
follow_redirects	string	Sekot atbildes pāradresācijām datu aptaujas laikā. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
post_type	string	Nosūtāmo datu pamatteksta tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējums), JSON (2), XML (3).
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Atbalstīts HTTP_AGENT vienumiem.
headers		Saknes elements HTTP(S) pieprasījuma galvenēm. Atbalstīts HTTP_AGENT vienumiem.
name	string	(obligāts HTTP_AGENT vienumiem) Galvenes nosaukums.
value	string	(obligāts HTTP_AGENT vienumiem) Galvenes vērtība.
retrieve_mode	string	Kura atbildes daļa jāsaglabā. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] BODY (0, noklusējums), HEADERS (1), BOTH (2).
request_method	string	Pieprasījuma metodes tips. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] GET (0, noklusējums), POST (1), PUT (2), HEAD (3).

Element	Type	Apraksts
output_format	string	Kā apstrādāt atbildi. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] RAW (0, noklusējums), JSON (1).
allow_traps	string	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Atbalstīts HTTP_AGENT vienumiem.
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Atbalstīts HTTP_AGENT vienumiem.
ssl_key_password	string	SSL atslēgas faila parole. Atbalstīts HTTP_AGENT vienumiem.
verify_peer	string	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
verify_host	string	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Atbalstīts HTTP_AGENT vienumiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
tags		Saknes elements vienuma tagiem.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.
triggers		Saknes elements hosta vienuma trigeriem.

Note:
Skatiet arī: Item object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosta vienuma vērtības priekšapstrādes soļi

```
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
          (...)
```

Elements	Type	Apraksts
type	string	(obligāti) Vienuma vērtības priekšapstrādes soļa tips. Iespējamās vērtības:[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, aprēķināts: saņemtā vērtība - iepriekšējā vērtība), CHANGE_PER_SECOND (10, aprēķināts: (saņemtā vērtība - iepriekšējā vērtība)/(pašreizējais laiks - pēdējās pārbaudes laiks)), 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		(obligāti) Saknes elements vienuma vērtības priekšapstrādes soļa parametriem.

Elements	Type	Apraksts
parameter	string	Atsevišķs vienuma vērtības priekšapstrādes soļa parametrs.
error_handler	string	Darbības tips, kas tiek izmantots priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības:[1] ORIGINAL_ERROR (0, pēc noklusējuma), DISCARD_VALUE (1), CUSTOM_VALUE (2), CUSTOM_ERROR (3).
error_handler_params	string	Kļūdu apstrādātāja parametri.

Note:
Skatiet arī: Vienuma priekšapstrādes objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosta vienuma trigeri

```
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
        (...)
```

Elements	Type	Apraksts
uuid	string	(obligāts) Unikāls šī trigera identifikators.
expression	string	(obligāts) Trigera izteiksme.
recovery_mode	string	Pamats OK notikumu ģenerēšanai. Iespējamās vērtības:[1] EXPRESSION (0, noklusējums), RECOVERY_EXPRESSION (1), NONE (2).
recovery_expression	string	Trigera atkopšanas izteiksme.
correlation_mode	string	Korelācijas režīms (bez notikumu korelācijas vai notikumu korelācija pēc taga). Iespējamās vērtības:[1] DISABLED (0, noklusējums), TAG_VALUE (1).
correlation_tag	string	Taga nosaukums, kas tiks izmantots notikumu korelācijai.
name	string	(obligāts) Trigera nosaukums.
event_name	string	Notikuma nosaukums.
opdata	string	Operatīvie dati.
url_name	string	Etiķete URL, kas saistīts ar trigeri.
url	string	URL, kas saistīts ar trigeri.
status	string	Trigera statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).

Elements	Type	Apraksts
priority	string	Trigera nopietnības līmenis. Iespējamās vērtības:[1] NOT_CLASSIFIED (0, noklusējums), INFO (1), WARNING (2), AVERAGE (3), HIGH (4), DISASTER (5).
description	text	Trigera apraksts.
type	string	Notikumu ģenerēšanas tips (viens problēmas notikums vai vairāki problēmas notikumi). Iespējamās vērtības:[1] SINGLE (0, noklusējums), MULTIPLE (1).
manual_close	string	Problēmu notikumu manuāla aizvēršana. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
dependencies		Saknes elements atkarībām.
name	string	(obligāts) Atkarīgā trigera nosaukums.
expression	string	(obligāts) Atkarīgā trigera izteiksme.
recovery_expression	string	Atkarīgā trigera atkopšanas izteiksme.
tags		Saknes elements trigera tagiem.
tag	string	(obligāts) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Trigger object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hostu zema līmeņa atklāšanas noteikumi

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      discovery_rules:
        - name: 'Database discovery'
          key: 'mysql.db.discovery["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}"]'
          delay: 1h
          filter: (skatiet tabulu zemāk)
          description: 'Datubāzu skenēšana DBVS.'
          interface_ref: if1
          item_prototypes:
            - name: 'Datubāzes {#DATABASE} izmērs'
              key: 'mysql.db.size["{$MYSQL.DSN}","{$MYSQL.USER}","{$MYSQL.PASSWORD}","{#DATABASE}"]'
              delay: 5m
              units: B
              description: 'Datubāzes izmērs.'
              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:
Lielākā daļa hostu zema līmeņa atklāšanas noteikumu elementu ir tādi paši kā hostu vienumiem. Tālāk esošajā tabulā ir aprakstīti tie elementi, kas atšķiras no hostu vienumiem.

Elements	Type	Apraksts
type	string	Vienuma tips. Iespējamās vērtības:[1] ZABBIX_PASSIVE (0, noklusējums), 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	(obligāti) Zema līmeņa atklāšanas noteikuma atslēga.
filter		Saknes elements hostu zema līmeņa atklāšanas noteikumu filtriem.
lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makrosu), pēc kura vairs neatklātie resursi tiks dzēsti. Noklusējums: 7d.
lifetime_type	string	Scenārijs zaudēto LLD resursu dzēšanai. Iespējamās vērtības: DELETE_NEVER, DELETE_IMMEDIATELY, DELETE_AFTER.
enabled_lifetime	string	Laika periods (izmantojot sekundes, laika sufiksu vai lietotāja makrosu), pēc kura vairs neatklātie resursi tiks atspējoti.
enabled_lifetime_type	string	Scenārijs zaudēto LLD resursu atspējošanai. Iespējamās vērtības: DISABLE_NEVER, DISABLE_IMMEDIATELY, DISABLE_AFTER.
item_prototypes		Saknes elements hostu vienumu prototipu elementiem, kas ir tādi paši kā hostu vienumiem.
trigger_prototypes		Saknes elements hostu trigeru prototipu elementiem, kas ir tādi paši kā hostu vienumu trigeriem.
graph_prototypes		Saknes elements hostu grafiku prototipiem, kas ir tādi paši kā hostu grafikiem.
host_prototypes		Saknes elements hostu prototipiem, kas ir tādi paši kā hostiem.
parent_discovery_rule		Saknes elements zema līmeņa atklāšanas noteikuma prototipa vecākajam zema līmeņa atklāšanas noteikumam (vai noteikuma prototipam). Šī īpašība norāda, ka tas ir LLD noteikuma prototips, tiešs atsauktā noteikuma/noteikuma prototipa bērnelements.
key	string	(obligāti) Vecākā zema līmeņa atklāšanas noteikuma (vai noteikuma prototipa) atslēga.
master_item	string	(obligāti DEPENDENT noteikumiem) Saknes elements atkarīgā noteikuma galvenajam vienumam.
lld_macro_paths		Saknes elements zema līmeņa atklāšanas noteikuma makrosu ceļiem.
lld_macro	string	(obligāti) Zema līmeņa atklāšanas noteikuma makrosa nosaukums.
path	string	(obligāti) Selektors vērtībai, kas tiks piešķirta atbilstošajam makrosam.
preprocessing		Saknes elements zema līmeņa atklāšanas noteikuma vērtības priekšapstrādei.
step		Saknes elements zema līmeņa atklāšanas noteikuma vērtības priekšapstrādes soļu elementiem, kas ir tādi paši kā hostu vienumu vērtības priekšapstrādes soļiem, izņemot ar mazāku iespējamo vērtību skaitu. Skatiet arī: LLD rule preprocessing object.
overrides		Saknes elements zema līmeņa atklāšanas noteikuma pārrakstīšanas noteikumiem.
name	string	(obligāti) Unikāls pārrakstīšanas nosaukums.
step	string	(obligāti) Unikāls pārrakstīšanas secības numurs.
stop	string	Pārtraukt nākamo pārrakstīšanu apstrādi, ja ir atbilstība.
filter		Saknes elements zema līmeņa atklāšanas noteikuma pārrakstīšanas noteikuma filtra elementiem, kas ir tādi paši kā hostu zema līmeņa atklāšanas noteikumu filtriem.
operations		Saknes elements hostu zema līmeņa atklāšanas noteikuma pārrakstīšanas operācijām.

Note:
Skatiet arī: LLD rule object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hostu zema līmeņa atklāšanas kārtulas filtri

```
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
          (...)
```

Elements	Type	Apraksts
evaltype	string	Pārrakstīt filtra nosacījumu novērtēšanas metodi. Iespējamās vērtības:[1] AND_OR (0, noklusējums), AND (1), OR (2), FORMULA (3).
formula	string	Pielāgota aprēķina formula filtra nosacījumiem.
conditions		Saknes elements filtra nosacījumiem.
macro	string	(obligāti) Zema līmeņa atklāšanas makrosa nosaukums, kuram veikt pārbaudi.
value	string	Vērtība, ar ko salīdzināt.
operator	string	Nosacījuma operators. Iespējamās vērtības:[1] MATCHES_REGEX (8, noklusējums), NOT_MATCHES_REGEX (9).
formulaid	string	(obligāti) Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu no pielāgotās izteiksmes. Var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, bet pēc tam, tos pieprasot atkārtoti, tas tiks ģenerēts no jauna.

Note:
Skatiet arī: LLD kārtulas filtra objektu (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosts zema līmeņa atklāšanas kārtulas aizstāšanas operācijas

Elements	Type	Apraksts
operationobject	string	Objekts, kuram tiek piemērota operācija. Iespējamās vērtības:[1] ITEM_PROTOTYPE (0), TRIGGER_PROTOTYPE (1), GRAPH_PROTOTYPE (2), HOST_PROTOTYPE (3), DISCOVERY_PROTOTYPE (4).
operator	string	Aizstāšanas nosacījuma operators. Iespējamās vērtības:[1] EQUAL (1), NOT_EQUAL (2), LIKE (3), NOT_LIKE (4), REGEXP (5), NOT_REGEXP (6).
value	string	Regulārā izteiksme vai virkne aizstāšanas nosacījuma operatoram.
status	string	Objekta statuss pēc aizstāšanas operācijas.
discover	string	Vai objekts tiek pievienots atklāšanas rezultātā.
delay	string	Vienuma prototipam iestatītais atjaunināšanas intervāls pēc aizstāšanas operācijas.
history	string	Vienuma prototipam iestatītais vēstures glabāšanas periods pēc aizstāšanas operācijas.
trends	string	Vienuma prototipam iestatītais tendenču glabāšanas periods pēc aizstāšanas operācijas.
severity	string	Trigera prototipa nopietnības līmenis, kas iestatīts pēc aizstāšanas operācijas.
tags		Saknes elements birkām, kas objektam iestatītas pēc aizstāšanas operācijas.
tag	string	(obligāti) Birkas nosaukums.
value	string	Birkas vērtība.
templates		Saknes elements veidnēm, kas saistītas ar hosts prototipu pēc aizstāšanas operācijas.
name	string	(obligāti) Veidnes nosaukums.
inventory_mode	string	Hosts prototipa inventarizācijas režīms, kas iestatīts pēc aizstāšanas operācijas.

Note:
Skatiet arī: LLD rule override operation object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosta tīmekļa scenāriji

Element	Type	Apraksts
uuid	string	(obligāti) Unikāls šī tīmekļa scenārija identifikators.
name	string	(obligāti) Tīmekļa scenārija nosaukums.
delay	string	Tīmekļa scenārija izpildes biežums (izmantojot sekundes, laika sufiksu vai lietotāja makrosu). Noklusējums: 1m.
attempts	integer	Mēģinājumu skaits tīmekļa scenārija soļu izpildei. Iespējamās vērtības: 1-10 (noklusējums: 1).
agent	string	Klienta aģents. Zabbix uzdosies par izvēlēto pārlūku. Tas ir noderīgi, ja vietne dažādām pārlūkprogrammām atgriež atšķirīgu saturu. Noklusējums: Zabbix.
http_proxy	string	Starpniekserveris, kas tiks izmantots tīmekļa scenārijam, norādīts šādā formātā: `http://[username[:password]@]proxy.example.com[:port]`
variables		Saknes elements tīmekļa scenārija mainīgajiem, kurus var izmantot scenārija soļos.
name	string	(obligāti) Mainīgā nosaukums.
value	text	(obligāti) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas tiks nosūtītas, izpildot pieprasījumu. Galvenes jānorāda, izmantojot to pašu sintaksi, kāda tiktu izmantota HTTP protokolā.
name	string	(obligāti) Galvenes nosaukums.
value	text	(obligāti) Galvenes vērtība.
status	string	Tīmekļa scenārija statuss. Iespējamās vērtības:[1] ENABLED (0, noklusējums), DISABLED (1).
authentication	string	Autentifikācijas metode. Iespējamās vērtības:[1] NONE (0, noklusējums), BASIC (1), NTLM (2), Kerberos (3) vai Digest (4).
http_user	string	Lietotājvārds, kas tiek izmantots BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
http_password	string	Parole, kas tiek izmantota BASIC (HTTP), NTLM, Kerberos vai Digest autentifikācijai.
verify_peer	string	Pārbaudīt tīmekļa servera SSL sertifikātu. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
verify_host	string	Pārbaudīt, vai tīmekļa servera sertifikāta lauks *Common Name* vai lauks *Subject Alternate Name* atbilst. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
ssl_cert_file	string	SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai (jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai (jābūt PEM formātā).
ssl_key_password	string	SSL privātās atslēgas faila parole.
steps		(obligāti) Saknes elements hosta tīmekļa scenārija soļiem.
tags		Saknes elements tīmekļa scenārija tagiem.
tag	string	(obligāti) Taga nosaukums.
value	string	Taga vērtība.

Note:
Skatiet arī: Web scenario object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosta tīmekļa scenārija soļi

Elements	Tips	Apraksts
name	string	(obligāti) Tīmekļa scenārija soļa nosaukums.
url	string	(obligāti) URL uzraudzībai.
query_fields		Saknes elements vaicājuma parametriem (HTTP lauku masīvs, kas tiks pievienots URL, izpildot pieprasījumu).
name	string	(obligāti) Vaicājuma parametra nosaukums.
value	string	Vaicājuma parametra vērtība.
posts		Saknes elements HTTP POST mainīgajiem (virkne (neapstrādāti POST dati) vai HTTP lauku masīvs (veidlapas lauku dati)).
name	string	(obligāti) POST lauka nosaukums.
value	string	(obligāti) POST lauka vērtība.

Elements	Tips	Apraksts
variables		Soļa līmeņa mainīgo (makrosu) saknes elements, kas jāpiemēro pēc šī soļa. Ja mainīgā vērtībai ir prefikss 'regex:', tad tā vērtība tiek iegūta no šī soļa atgrieztajiem datiem atbilstoši regulārās izteiksmes šablonam, kas seko prefiksam 'regex:'
name	string	(obligāti) Mainīgā nosaukums.
value	text	(obligāti) Mainīgā vērtība.
headers		Saknes elements HTTP galvenēm, kas tiks nosūtītas, izpildot pieprasījumu.
name	string	(obligāti) Galvenes nosaukums.
value	text	(obligāti) Galvenes vērtība.
follow_redirects	string	Sekot HTTP pāradresācijām. Iespējamās vērtības:[1] NO (0), YES (1, pēc noklusējuma).
retrieve_mode	string	HTTP atbildes iegūšanas režīms. Iespējamās vērtības:[1] BODY (0, pēc noklusējuma), HEADERS (1), BOTH (2).
timeout	string	Soļa izpildes noildze (izmantojot sekundes, laika sufiksu vai lietotāja makrosu). Pēc noklusējuma: 15s.
required	string	Teksts, kam jābūt atbildē (tiek ignorēts, ja tukšs).
status_codes	string	Ar komatiem atdalīts pieņemamo HTTP statusa kodu saraksts (piemēram, 200-201,210-299; tiek ignorēts, ja tukšs).

Note:
Skatiet arī: Tīmekļa scenārija soļa objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hostu grafiki

```
zabbix_export:
  (...)
  graphs:
    - name: 'MySQL: Bandwidth'
      graph_items: (skatiet tabulu zemāk)
    (...)
```

Elements	Type	Apraksts
uuid	string	Unikāls grafika identifikators.
name	string	(obligāts) Grafika nosaukums.
width	integer	Grafika platums pikseļos. Izmanto priekšskatījumam un sektoru/izceltajiem grafikiem. Iespējamās vērtības:[1] 20-65535 (noklusējums: 900).
height	integer	Grafika augstums pikseļos. Izmanto priekšskatījumam un sektoru/izceltajiem grafikiem. Iespējamās vērtības:[1] 20-65535 (noklusējums: 900).
yaxismin	double	Y ass minimālā vērtība. Atbalstīts Y ass fiksētajai minimālajai vērtībai. Noklusējums: 0.
yaxismax	double	Y ass maksimālā vērtība. Atbalstīts X ass fiksētajai minimālajai vērtībai. Noklusējums: 0.
show_work_period	string	Izcelt ārpusdarba stundas. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
show_triggers	string	Rādīt vienkāršās trigeru vērtības kā līniju. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
type	string	Grafika tips. Iespējamās vērtības:[1] NORMAL (0, noklusējums), STACKED (1), PIE (2), EXPLODED (3).
show_legend	string	Rādīt grafika leģendu. Iespējamās vērtības:[1] NO (0), YES (1, noklusējums).
show_3d	string	Iespējot 3D stilu. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).

Elements	Type	Apraksts
percent_left	double	Rādīt procentiles līniju kreisajai asij. Atbalstīts NORMAL grafikiem. Noklusējums: 0.
percent_right	double	Rādīt procentiles līniju labajai asij. Atbalstīts NORMAL grafikiem. Noklusējums: 0.
ymin_type_1	string	Y ass minimālā vērtība. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] CALCULATED (0, noklusējums), FIXED (1), ITEM (2).
ymin_item_1		(obligāts, ja `ymin_type_1` ir iestatīts uz `ITEM`) Saknes elements atsevišķa vienuma detalizētai informācijai.
host	string	(obligāts) Vienuma hosts.
key	string	(obligāts) Vienuma atslēga.
ymax_type_1	string	Y ass maksimālā vērtība. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] CALCULATED (0, noklusējums), FIXED (1), ITEM (2).
ymax_item_1		(obligāts, ja `ymax_type_1` ir iestatīts uz `ITEM`) Saknes elements atsevišķa vienuma detalizētai informācijai.
host	string	(obligāts) Vienuma hosts.
key	string	(obligāts) Vienuma atslēga.
graph_items		(obligāts) Saknes elements hosta grafika vienumiem.

Note:
Skatiet arī: Graph object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hosta grafika vienumi

```
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
    (...)
```

Elements	Type	Apraksts
sortorder	integer	Zīmēšanas secība. Vispirms tiek zīmēta mazākā vērtība. Var izmantot, lai zīmētu līnijas vai apgabalus aiz (vai priekšā) citam.
drawtype	string	Grafika vienuma zīmēšanas stils. Atbalstīts NORMAL grafikiem. Iespējamās vērtības:[1] SINGLE_LINE (0, noklusējums), FILLED_REGION (1), BOLD_LINE (2), DOTTED_LINE (3), DASHED_LINE (4), GRADIENT_LINE (5).
color	string	Elementa krāsa (6 simboli, hex).
yaxisside	string	Grafika puse, kurā tiks zīmēta grafika vienuma Y skala. Atbalstīts NORMAL un STACKED grafikiem. Iespējamās vērtības:[1] LEFT (0, noklusējums), RIGHT (1).
calc_fnc	string	Zīmējamie dati, ja vienumam pastāv vairāk nekā viena vērtība. Iespējamās vērtības:[1] MIN (1), AVG (2, noklusējums), MAX (4), ALL (7; minimums, vidējais un maksimums; atbalstīts vienkāršiem grafikiem), LAST (9, atbalstīts sektoru/izceltajiem grafikiem).

Elements	Type	Apraksts
type	string	Grafika vienuma tips. Iespējamās vērtības:[1] SIMPLE (0, noklusējums), GRAPH_SUM (2; vienuma vērtība attēlo visu sektoru; atbalstīts sektoru/izceltajiem grafikiem).
item		(obligāti) Atsevišķs vienums.
host	string	(obligāti) Vienuma hosts.
key	string	(obligāti) Vienuma atslēga.

Note:
Skatiet arī: Graph item object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Hostu vērtību kartes

```
zabbix_export:
  (...)
  hosts:
    - host: 'MySQL server'
      (...)
      valuemaps:
        - name: Piemēra vērtību karte
          mappings:
            - value: '1'
              newvalue: Piemēra vērtība
  (...)
```

Elements	Type	Apraksts
uuid	string	(obligāti) Unikāls šīs vērtību kartes identifikators.
name	string	(obligāti) Vērtību kartes nosaukums.
mapping		Saknes elements kartējumiem.
type	string	Kartējuma atbilstības tips. Iespējamās vērtības:[1] EQUAL (0, pēc noklusējuma), GREATER_OR_EQUAL (2), LESS_OR_EQUAL (3), IN_RANGE (4), REGEXP (5), DEFAULT (6).
value	string	Sākotnējā vērtība.
newvalue	string	(obligāti) Vērtība, uz kuru tiek kartēta sākotnējā vērtība.

Note:
Skatiet arī: Vērtību kartes objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Piezīmes [1] API integer vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai atsaucei. Lai iegūtu vairāk informācijas, skatiet saistīto API objekta lapu tabulas ierakstā vai katras sadaļas beigās.

5 Tīkla kartes

Pārskats

Tīkla kartes eksports satur:

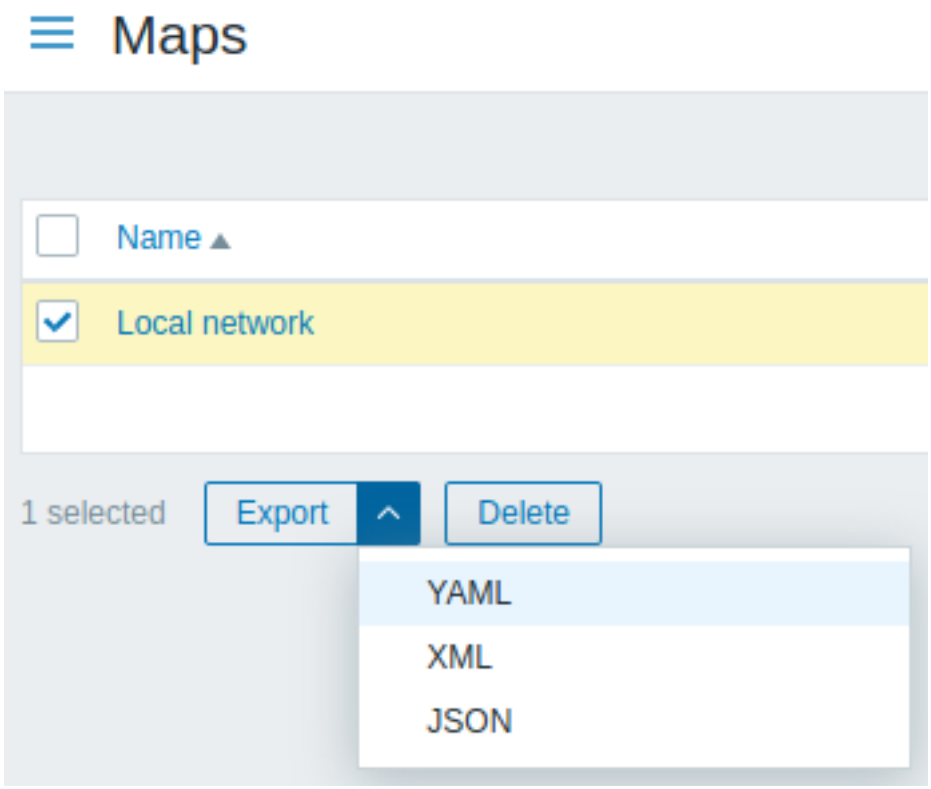
- Visus saistītos attēlus
- Kartes struktūru (visus kartes iestatījumus, visus iekļautos elementus ar to iestatījumiem, kartes saites un kartes saišu statusa indikatorus)

Warning:
Nevienas hostu grupas, hosti, trigeri, citas kartes vai citi elementi, kas var būt saistīti ar eksportēto karti, netiek eksportēti. Tādēļ, ja trūkst vismaz viena elementa, uz kuru karte atsaucas, tās importēšana neizdosies.

Eksportēšana

Lai eksportētu tīkla kartes, rīkojieties šādi:

1. Dodieties uz *Monitoring* → *Maps*.
2. Atzīmējiet eksportējamo tīkla karšu izvēles rūtiņas.
3. Zem saraksta noklikšķiniet uz *Export*.



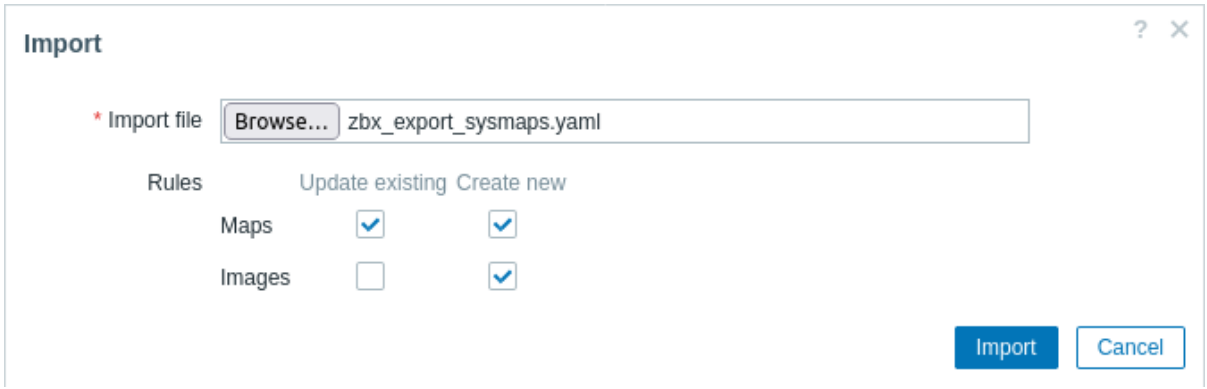
Atkarībā no izvēlētā formāta kartes tiek eksportētas uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_maps.yaml` — YAML eksportā (noklusējuma eksporta opcija);
- `zabbix_export_maps.xml` — XML eksportā;
- `zabbix_export_maps.json` — JSON eksportā.

Importēšana

Lai importētu tīkla kartes, rīkojieties šādi:

1. Atveriet *Monitoring* → *Maps*.
2. Augšējā labajā stūrī noklikšķiniet uz *Import*.
3. Atlasiet importēšanas failu.
4. Importēšanas noteikumos atzīmējiet nepieciešamās opcijas.
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Import*.



Importēšanas noteikumi:

Noteikums	Apraksts
Update existing	Esošās kartes tiks atjauninātas, izmantojot datus no importēšanas faila. Pretējā gadījumā tās netiks atjauninātas.

Noteikums	Apraksts
Create new	Jaunas kartes tiks izveidotas, izmantojot datus no importēšanas faila. Pretējā gadījumā tās netiks izveidotas.

Ja noņemsiet atzīmi abām karšu opcijām un atzīmēsiet attiecīgās opcijas attēliem, tiks importēti tikai attēli. Attēlu importēšana ir pieejama tikai *Super admin* lietotājiem.

Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Warning:
Ja aizstājat esošu attēlu, tas ietekmēs visas kartes, kurās šis attēls tiek izmantots.

Eksporta formāts

Eksportēšana uz YAML:

```yaml
zabbix_export:
  version: '8.0'
  images:
    - name: Zabbix_server_3D_(128)
      imagetype: '1'
      encodedImage: iVBOR...5CYII=
  maps:
    - name: 'Lokālais tīkls'
      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: {  }
```

Eksportētie elementi Eksportētie elementi ir izskaidroti zemāk esošajā tabulā.

Element	Type	Description
images		Saknes elements attēliem.
name	string	Unikāls attēla nosaukums.
imagetype	integer	Attēla tips. Iespējamās vērtības: 1 - Attēls; 2 - Fons.
encodedImage	string	Base64 kodēts attēls.
maps		Saknes elements kartēm.

Kartes

Element	Type	Description
name	string	Unikāls kartes nosaukums.
width	integer	Kartes platums pikseļos.
height	integer	Kartes augstums pikseļos.
label_type	integer	Kartes elementa etiķetes tips. Iespējamās vērtības: 0 - Etiķete; 1 - hosta IP adrese; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas.

Element	Type	Description
label_location	integer	Kartes elementa etiķetes noklusējuma atrašanās vieta. Iespējamās vērtības: 0 - Apakšā; 1 - Pa kreisi; 2 - Pa labi; 3 - Augšā.
highlight	integer	Iespējot ikonu izcelšanu aktīviem trigeriem un hostu statusiem. Iespējamās vērtības: 0 - Nē; 1 - Jā.
expandproblem	integer	Rādīt problēmas trigeri elementiem ar vienu problēmu. Iespējamās vērtības: 0 - Nē; 1 - Jā.
markelements	integer	Izcelt kartes elementus, kuriem nesen mainījies statuss. Iespējamās vērtības: 0 - Nē; 1 - Jā.
show_unack	integer	Problēmu attēlošana. Iespējamās vērtības: 0 - Visu problēmu skaits; 1 - Neapstiprināto problēmu skaits; 2 - Apstiprināto un neapstiprināto problēmu skaits atsevišķi.
severity_min	integer	Minimālā trigera nopietnība, ko pēc noklusējuma rādīt kartē. Iespējamās vērtības: 0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa.
show_suppressed	integer	Rādīt problēmas, kuras citādi tiktu apspiestas (nerādītas) hosta uzturēšanas dēļ. Iespējamās vērtības: 0 - Nē; 1 - Jā.
grid_size	integer	Kartes režģa šūnas izmērs pikseļos. Atbalstīts, ja `grid_show` ir iestatīts uz 0. Iespējamās vērtības: 20, 40, 50, 75 vai 100.
grid_show	integer	Rādīt režģi kartes konfigurācijā. Iespējamās vērtības: 0 - Jā; 1 - Nē.
grid_align	integer	Automātiski līdzināt ikonas kartes konfigurācijā. Iespējamās vērtības: 0 - Jā; 1 - Nē.
label_format	integer	Izmantot paplašināto etiķešu konfigurāciju. Iespējamās vērtības: 0 - Nē; 1 - Jā.
label_type_host	integer	Rādīt etiķeti kā hosta etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 1 - hosta IP adrese; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas; 5 - Pielāgota etiķete.

Element	Type	Description
label_type_hostgroup	integer	Rādīt etiķeti kā hostu grupas etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas; 5 - Pielāgota etiķete.
label_type_trigger	integer	Rādīt etiķeti kā trigera etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas; 5 - Pielāgota etiķete.
label_type_map	integer	Rādīt etiķeti kā kartes etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 2 - Elementa nosaukums; 3 - Tikai statuss; 4 - Nekas; 5 - Pielāgota etiķete.
label_type_image	integer	Rādīt etiķeti kā attēla etiķeti. Atbalstīts, ja `label_format` ir iestatīts uz 1. Iespējamās vērtības: 0 - Etiķete; 2 - Elementa nosaukums; 4 - Nekas; 5 - Pielāgota etiķete.
label_string_host	string	Pielāgota etiķete hosta elementiem. Atbalstīts, ja `label_type_host` ir iestatīts uz 5.
label_string_hostgroup	string	Pielāgota etiķete hostu grupas elementiem. Atbalstīts, ja `label_type_hostgroup` ir iestatīts uz 5.
label_string_trigger	string	Pielāgota etiķete trigera elementiem. Atbalstīts, ja `label_type_trigger` ir iestatīts uz 5.
label_string_map	string	Pielāgota etiķete kartes elementiem. Atbalstīts, ja `label_type_map` ir iestatīts uz 5.
label_string_image	string	Pielāgota etiķete attēla elementiem. Atbalstīts, ja `label_type_image` ir iestatīts uz 5.
expand_macros	integer	Izvērst makrosus etiķetēs kartes konfigurācijā. Iespējamās vērtības: 0 - Nē; 1 - Jā.
background		Saknes elements fona attēlam (ja tāds ir). Atbalstīts, ja `imagetype` ir iestatīts uz 2.
name	string	Fona attēla nosaukums.
iconmap		Saknes elements ikonu kartēšanai (ja tāda ir).
name	string	Ikonu kartēšanas nosaukums.
urls		Saknes elements URL, ko izmanto kartes vai katrs kartes elements.
name	string	Saites nosaukums.
url	string	Saites URL.
elementtype	integer	Kartes vienuma tips, kuram pieder saite. Iespējamās vērtības: 0 - Hosts; 1 - Karte; 2 - Trigeris; 3 - Hostu grupa; 4 - Attēls.
selements		Saknes elements kartes selementiem.

Element	Type	Description
shapes		Saknes elements kartes formām.
type	integer	Formas tips. Iespējamās vērtības: 0 - Taisnstūris; 1 - Elipse.
x	integer	Formas X koordinātas pikseļos.
y	integer	Formas Y koordinātas pikseļos.
width	integer	Formas platums.
height	integer	Formas augstums.
text	string	Teksts formas iekšpusē.
font	integer	Teksta fonta stils. Iespējamās vērtības: 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	Fonta izmērs pikseļos.
font_color	string	Fonta krāsa heksadecimālā kodā.
text_halign	integer	Teksta horizontālā līdzināšana. Iespējamās vērtības: 0 - Centrēts; 1 - Pa kreisi; 2 - Pa labi.
text_valign	integer	Teksta vertikālā līdzināšana. Iespējamās vērtības: 0 - Vidū; 1 - Augšā; 2 - Apakšā.
border_type	integer	Formas apmales tips. Iespējamās vērtības: 0 - Nav; 1 - Bieza līnija; 2 - Punktēta līnija; 3 - Pārtraukta līnija.
border_width	integer	Apmales platums pikseļos.
border_color	string	Apmales krāsa heksadecimālā kodā.
background_color	string	Fona (aizpildījuma) krāsa heksadecimālā kodā.
zindex	integer	Vērtība visu formu un līniju secības noteikšanai (z-index).
lines		Saknes elements kartes līnijām.
x1	integer	Līnijas 1. punkta X koordinātas pikseļos.
y1	integer	Līnijas 1. punkta Y koordinātas pikseļos.
x2	integer	Līnijas 2. punkta X koordinātas pikseļos.
y2	integer	Līnijas 2. punkta Y koordinātas pikseļos.
line_type	integer	Līnijas tips. Iespējamās vērtības: 0 - Nav; 1 - Bieza līnija; 2 - Punktēta līnija; 3 - Pārtraukta līnija.
line_width	integer	Līnijas platums pikseļos.
line_color	string	Līnijas krāsa heksadecimālā kodā.
zindex	integer	Vērtība visu formu un līniju secības noteikšanai (z-index).

Element	Type	Description
links		Saknes elements saitēm starp kartes elementiem.
drawtype	integer	Saites stils. Iespējamās vērtības: 0 - Līnija; 2 - Bieza līnija; 3 - Punktēta līnija; 4 - Pārtraukta līnija.
color	string	Saites krāsa (6 simboli, hex).
label	string	Saites etiķete.
selementid1	id	Viena savienojamā elementa ID.
selementid2	id	Otra savienojamā elementa ID.
linktriggers		Saknes elements saišu statusa indikatoriem.

Note:
Skatiet arī: Map object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Kartes selementi

Elements	Tips	Apraksts
elementtype	integer	Kartes elementa tips. Iespējamās vērtības: 0 - Hosts; 1 - Karte; 2 - Trigeris; 3 - Hostu grupa; 4 - Attēls.
elements		Saknes elements Zabbix entītijām (hosts, hostu grupa, karte utt.), kas tiek attēlotas kartē. Visām entītijām ir atbilstošais elements (`host` utt.; skatiet, piemēram, Eksporta formāts).
label	string	Ikonas etiķete.
label_location	integer	Etiķetes atrašanās vieta. Iespējamās vērtības: -1 - Izmantot kartes noklusējumu; 0 - Apakšā; 1 - Kreisajā pusē; 2 - Labajā pusē; 3 - Augšā.
x	integer	Atrašanās vieta uz X ass.
y	integer	Atrašanās vieta uz Y ass.
elementsubtype	integer	Elementa apakštips. Atbalstīts, ja `elementtype` ir iestatīts uz 3. Iespējamās vērtības: 0 - Viena hostu grupa; 1 - Visas hostu grupas.
areatype	integer	Apgabala izmērs. Atbalstīts, ja `elementtype` ir iestatīts uz 1. Iespējamās vērtības: 0 - Tāds pats kā visai kartei; 1 - Pielāgots izmērs.
width	integer	Apgabala platums. Atbalstīts, ja `areatype` ir iestatīts uz 1.
height	integer	Apgabala augstums. Atbalstīts, ja `areatype` ir iestatīts uz 1.
viewtype	integer	Apgabala izvietošanas algoritms. Atbalstīts, ja `elementsubtype` ir iestatīts uz 1. Iespējamās vērtības: 0 - Izvietot vienmērīgi apgabalā.

Elements	Tips	Apraksts
use_iconmap	integer	Izmantot ikonu kartēšanu šim elementam. Attiecas tikai tad, ja ikonu kartēšana ir aktivizēta kartes līmenī. Iespējamās vērtības: 0 - Nē; 1 - Jā.
selementid	id	Unikāls elementa ieraksta ID.
icon_off		Saknes elements attēlam, ko izmantot, kad elements ir `OK` stāvoklī.
name	string	Unikāls attēla nosaukums.
icon_on		Saknes elements attēlam, ko izmantot, kad elements ir `Problem` stāvoklī.
name	string	Unikāls attēla nosaukums.
icon_disabled		Saknes elements attēlam, ko izmantot, kad elements ir atspējots.
name	string	Unikāls attēla nosaukums.
icon_maintenance		Saknes elements attēlam, ko izmantot, kad elements ir uzturēšanā.
name	string	Unikāls attēla nosaukums.
urls		Saknes elements URL adresēm, ko izmanto kartes vai katrs kartes elements.
name	string	Saites nosaukums.
url	string	Saites URL.
evaltype	integer	Novērtēšanas tips birkām.
tags		Saknes elements problēmu birkām (hosta un hostu grupas elementiem). Ja birkas ir norādītas, kartē tiks attēlotas tikai problēmas ar šīm birkām.
tag	string	Birkas nosaukums.
value	string	Birkas vērtība.
operator	integer	Operators.

Note:
Skatiet arī: Kartes elementa objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Kartes saites statusa indikatori

Elements	Tips	Apraksts
drawtype	integer	Saites stils, kad trigeris ir stāvoklī ’Problēma’. Iespējamās vērtības: 0 - Līnija; 2 - Treknā līnija; 3 - Punktota līnija; 4 - Pārtraukta līnija.
color	string	Saites krāsa (6 simboli, hex), kad trigeris ir stāvoklī ’Problēma’.
trigger		Saknes elements trigerim, ko izmanto saites statusa norādīšanai.
description	string	Trigera nosaukums.
expression	string	Trigera izteiksme.
recovery_expression	string	Trigera atkopšanas izteiksme.

Note:
Skatiet arī: Kartes saites trigera objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

6 Mediju tipi

Pārskats

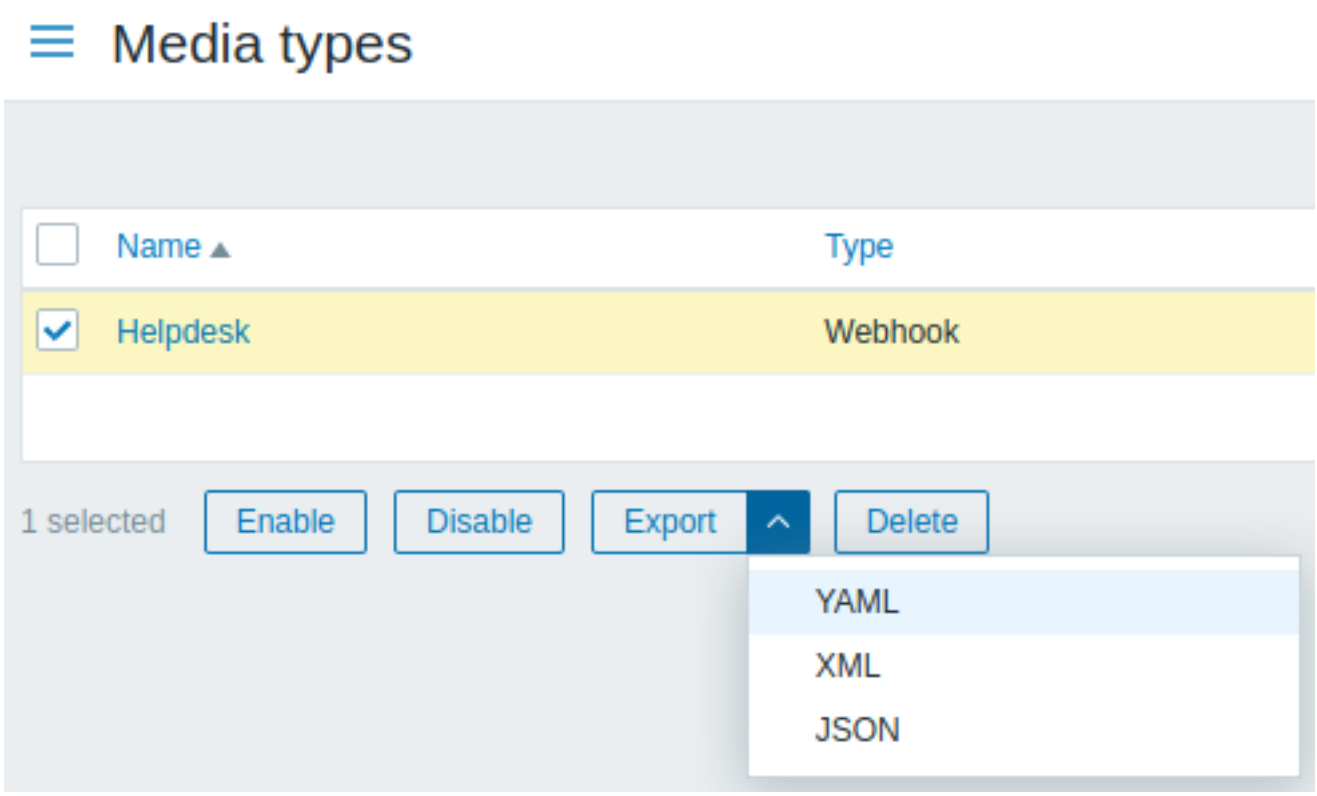
Paziņojumu nesēju tipi tiek eksportēti kopā ar visiem saistītajiem objektiem un objektu attiecībām.

Eksportēšana

Lai eksportētu paziņojumu nesēju tipus, rīkojieties šādi:

1. Atveriet *Brīdinājumi → Paziņojumu nesēju tipi*.
2. Atzīmējiet eksportējamo paziņojumu nesēju tipu izvēles rūtiņas.

3. Zem saraksta noklikšķiniet uz *Eksportēt*.



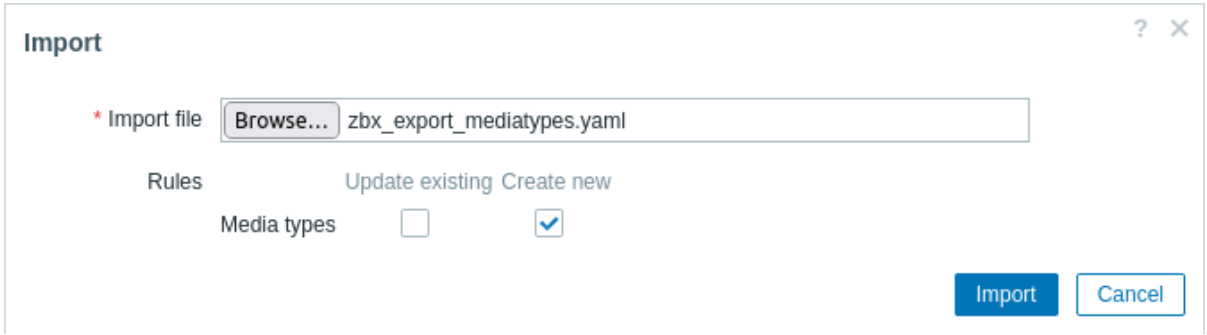
Atkarībā no izvēlētā formāta paziņojumu nesēju tipi tiek eksportēti uz lokālu failu ar noklusējuma nosaukumu:

- `zabbix_export_mediatypes.yaml` - YAML eksportam (noklusējuma eksporta opcija);
- `zabbix_export_mediatypes.xml` - XML eksportam;
- `zabbix_export_mediatypes.json` - JSON eksportam.

Importēšana

Lai importētu mediju tipus, rīkojieties šādi:

1. Atveriet *Brīdinājumi* → *Mediju tipi*.
2. Augšējā labajā stūrī noklikšķiniet uz *Importēt*.
3. Atlasiet importēšanas failu.
4. Importēšanas noteikumos atzīmējiet nepieciešamās opcijas.
5. Konfigurācijas formas apakšējā labajā stūrī noklikšķiniet uz *Importēt*.



Importēšanas noteikumi:

Noteikums	Apraksts
Atjaunināt esošos	Esošie elementi tiks atjaunināti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks atjaunināti.
Izveidot jaunus	Jauni elementi tiks izveidoti, izmantojot datus no importēšanas faila. Pretējā gadījumā tie netiks izveidoti.

Lietotāja saskarnē tiks parādīts importēšanas veiksmes vai kļūmes ziņojums.

Eksporta formāts

Eksportēt uz 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 'Incorrect "event_source" parameter given: "' + params.event_source + '".\nMust be 0
    }

    if (params.event_value !== '0' && params.event_value !== '1'
        && (params.event_source === '0' || params.event_source === '3')) {
        throw 'Incorrect "event_value" parameter given: ' + params.event_value + '\nMust be 0 or 1
    }

    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" should be -2..2';
    }

    if (params.event_source === '0' && isNaN(params.triggerid)) {
        throw 'field "triggerid" is not a number';
    }

    if (isNaN(params.eventid)) {
        throw 'field "eventid" is not a number';
    }

    if (typeof params.message !== 'string' || params.message.trim() === '') {
        throw 'field "message" cannot be empty';
    }

    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 'field "retry" should be a number with value of at least 30 if "priority" is set
```

```
            }

            if (isNaN(params.expire) || params.expire > 10800) {
                throw 'field "expire" should be a number with value of at most 10800 if "priority" is
            }

            data.retry = params.retry;
            data.expire = params.expire;
        }

        data = JSON.stringify(data);
        Zabbix.log(4, '[ Pushover Webhook ] Sending request: ' + params.endpoint + '\n' + data);

        request.addHeader('Content-Type: application/json');
        response = request.post(params.endpoint, data);

        Zabbix.log(4, '[ Pushover Webhook ] Received response with status code ' + request.getStatus()

        if (response !== null) {
            try {
                response = JSON.parse(response);
            }
            catch (error) {
                Zabbix.log(4, '[ Pushover Webhook ] Failed to parse response received from Pushover');
                response = null;
            }
        }

        if (request.getStatus() != 200 || response === null || typeof response !== 'object' || respons
            if (response !== null && typeof response === 'object' && typeof response.errors === 'objec
                    && typeof response.errors[0] === 'string') {
                throw response.errors[0];
            }
            else {
                throw 'Unknown error. Check debug log for more information.';
            }
        }

        return 'OK';
    }
    catch (error) {
        Zabbix.log(4, '[ Pushover Webhook ] Pushover notification failed: ' + error);
        throw 'Pushover notification failed: ' + error;
    }
description: |
    Lūdzu, skatiet iestatīšanas ceļvedi šeit: https://git.zabbix.com/projects/ZBX/repos/zabbix/browse/

    Iestatiet token parametru ar savu Pushover lietotnes atslēgu.
    Piešķirot Pushover mediju Zabbix lietotājam, pievienojiet lietotāja atslēgu laukā send to.
message_templates:
    - event_source: TRIGGERS
      operation_mode: PROBLEM
      subject: 'Problēma: {EVENT.NAME}'
      message: |
        Problēma sākās {EVENT.TIME} {EVENT.DATE}
        Problēmas nosaukums: {EVENT.NAME}
        Hosts: {HOST.NAME}
        Nopietnība: {EVENT.SEVERITY}
        Operatīvie dati: {EVENT.OPDATA}
        Sākotnējās problēmas ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
```

```
      operation_mode: RECOVERY
      subject: 'Atrisināts pēc {EVENT.DURATION}: {EVENT.NAME}'
      message: |
        Problēma tika atrisināta {EVENT.RECOVERY.TIME} {EVENT.RECOVERY.DATE}
        Problēmas nosaukums: {EVENT.NAME}
        Problēmas ilgums: {EVENT.DURATION}
        Hosts: {HOST.NAME}
        Nopietnība: {EVENT.SEVERITY}
        Sākotnējās problēmas ID: {EVENT.ID}
        {TRIGGER.URL}
    - event_source: TRIGGERS
      operation_mode: UPDATE
      subject: 'Atjaunināta problēma pēc {EVENT.AGE}: {EVENT.NAME}'
      message: |
        {USER.FULLNAME} {EVENT.UPDATE.ACTION} problēmu {EVENT.UPDATE.DATE} {EVENT.UPDATE.TIME}.
        {EVENT.UPDATE.MESSAGE}

        Pašreizējais problēmas stāvoklis ir {EVENT.STATUS}, vecums ir {EVENT.AGE}, apstiprināta: {EVEN
    - event_source: DISCOVERY
      operation_mode: PROBLEM
      subject: 'Atklāšana: {DISCOVERY.DEVICE.STATUS} {DISCOVERY.DEVICE.IPADDRESS}'
      message: |
        Atklāšanas kārtula: {DISCOVERY.RULE.NAME}

        Ierīces IP: {DISCOVERY.DEVICE.IPADDRESS}
        Ierīces DNS: {DISCOVERY.DEVICE.DNS}
        Ierīces statuss: {DISCOVERY.DEVICE.STATUS}
        Ierīces darbības laiks: {DISCOVERY.DEVICE.UPTIME}

        Ierīces servisa nosaukums: {DISCOVERY.SERVICE.NAME}
        Ierīces servisa ports: {DISCOVERY.SERVICE.PORT}
        Ierīces servisa statuss: {DISCOVERY.SERVICE.STATUS}
        Ierīces servisa darbības laiks: {DISCOVERY.SERVICE.UPTIME}
    - event_source: AUTOREGISTRATION
      operation_mode: PROBLEM
      subject: 'Automātiskā reģistrācija: {HOST.HOST}'
      message: |
        Hosta nosaukums: {HOST.HOST}
        Hosta IP: {HOST.IP}
        Aģenta ports: {HOST.PORT}
```

Eksportētie elementi

Eksportētie elementi ir izskaidroti zemāk esošajā tabulā.

Elements	Tips	Apraksts
name	string	(obligāti) Mediju tipa nosaukums.
type	string	(obligāti) Pārsūtīšanas veids, ko izmanto mediju tips. Iespējamās vērtības:[1] EMAIL (0), SMS (1), SCRIPT (2), WEBHOOK (4).
status	string	Vai mediju tips ir iespējots. Iespējamās vērtības:[1] ENABLED (0, pēc noklusējuma), DISABLED (1)
max_sessions	integer	Maksimālais brīdinājumu skaits, ko var apstrādāt paralēli. Iespējamās vērtības SMS gadījumā:[1] 1 (pēc noklusējuma). Iespējamās vērtības citiem mediju tipiem:[1] 0-100 (kur 0 — neierobežoti).
attempts	integer	Maksimālais mēģinājumu skaits brīdinājuma nosūtīšanai. Iespējamās vērtības:[1] 1-10 (pēc noklusējuma: 3).
attempt_interval	string	Intervāls (izmantojot sekundes vai laika sufiksu) starp atkārtotiem mēģinājumiem. Iespējamās vērtības:[1] 0-60s (pēc noklusējuma: 10s).
description	string	Mediju tipa apraksts.
message_templates		Saknes elements mediju tipa ziņojumu veidnēm.
event_source	string	(obligāti) Notikuma avots. Iespējamās vērtības:[1] TRIGGERS (0), DISCOVERY (1), AUTOREGISTRATION (2), INTERNAL (3), SERVICE (4).

Elements	Tips	Apraksts
operation_mode	string	Darbības režīms. Iespējamās vērtības:[1] PROBLEM (0), RECOVERY (1), UPDATE (2).
subject	string	Ziņojuma tēma.
message	string	Ziņojuma teksts.

Note:
Skatiet arī: Media type object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

E-pasts

Tālāk norādītie papildu elementi tiek eksportēti tikai *E-pasts* medija tipam.

Elements	Type	Apraksts
provider	string	E-pasta pakalpojumu sniedzējs.
smtp_server	string	SMTP serveris.
smtp_port	integer	SMTP servera ports, kuram izveidot savienojumu. Noklusējums: 25.
smtp_helo	string	SMTP helo.
smtp_email	string	E-pasta adrese, no kuras tiks sūtīti paziņojumi.
smtp_security	string	Izmantojamais SMTP savienojuma drošības līmenis. Iespējamās vērtības:[1] NONE (0, noklusējums), STARTTLS (1), SSL_OR_TLS (2).
smtp_verify_host	string	SMTP SSL hosta verifikācija. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
smtp_verify_peer	string	SMTP SSL partnera verifikācija. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
smtp_authentication	string	Izmantojamā SMTP autentifikācijas metode. Iespējamās vērtības:[1] NONE (0, noklusējums), PASSWORD (1), OAUTH (2).
username	string	Lietotājvārds.
password	string	Autentifikācijas parole.
redirection_url	string	Zabbix lietotāja saskarne URL, uz kuru novirzīt atpakaļ pēc OAuth autorizācijas.
client_id	string	Klienta identifikators, kas reģistrēts OAuth autorizācijas serverī.
authorization_url	string	OAuth URL ar parametriem, lai iegūtu piekļuves un atsvaidzināšanas marķierus.
token_url	string	OAuth URL, lai apmainītu autorizācijas marķieri pret piekļuves un atsvaidzināšanas marķieriem.
message_format	string	Ziņojuma formāts. Iespējamās vērtības:[1] TEXT (0), HTML (1, noklusējums).

Note:
Skatiet arī: Media type object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

SMS

Tālāk norādītie papildu elementi tiek eksportēti tikai *SMS* medija tipam.

Elements	Type	Apraksts
gsm_modem	string	(obligāti) GSM modema seriālās ierīces nosaukums.

Note:
Skatiet arī: Medija tipa objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Skripts

Tālāk norādītie papildu elementi tiek eksportēti tikai *Skripts* paziņojumu nesēja tipam.

Elements	Type	Apraksts
skripta nosaukums	string	(obligāti) Skripta nosaukums.
parametri		Saknes elements skripta parametriem.
sortorder	string	(obligāti) Skripta parametru secība, kādā tie tiek nodoti skriptam kā komandrindas argumenti.
value	string	Skripta parametra vērtība.

Note:
Skatiet arī: Paziņojumu nesēja tipa objekts (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Webhook

Tālāk norādītie papildu elementi tiek eksportēti tikai *Webhook* medija tipam.

Elements	Type	Apraksts
script	string	Skripts.
timeout	string	Javascript skripta HTTP pieprasījuma noildzes intervāls. Iespējamās vērtības:[1] 1-60s (noklusējums: 30s).
process_tags	string	Vai apstrādāt atgrieztos tagus. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
show_event_menu	string	Norāda ieraksta esamību notikuma izvēlnē, ja makross {EVENT.TAGS.*} tika veiksmīgi atrisināts laukos `event_menu_url` un `event_menu_name`. Iespējamās vērtības:[1] NO (0, noklusējums), YES (1).
event_menu_url	string	Notikuma izvēlnes ieraksta URL. Atbalsta makrosu {EVENT.TAGS.*}.
event_menu_name	string	Notikuma izvēlnes ieraksta nosaukums. Atbalsta makrosu {EVENT.TAGS.*}.
parameters		Saknes elements webhook medija tipa parametriem.
name	string	(obligāti) Webhook parametra nosaukums.
value	string	Webhook parametra vērtība.

Note:
Skatiet arī: Media type object (skatiet atbilstošo īpašību ar tādu pašu nosaukumu).

Piezīmes

[1] API integer vērtības iekavās, piemēram, ENABLED (0), ir minētas tikai atsaucei. Papildinformāciju skatiet saistītā API objekta lapā tabulas ierakstā vai katras sadaļas beigās.

7 Dashboards

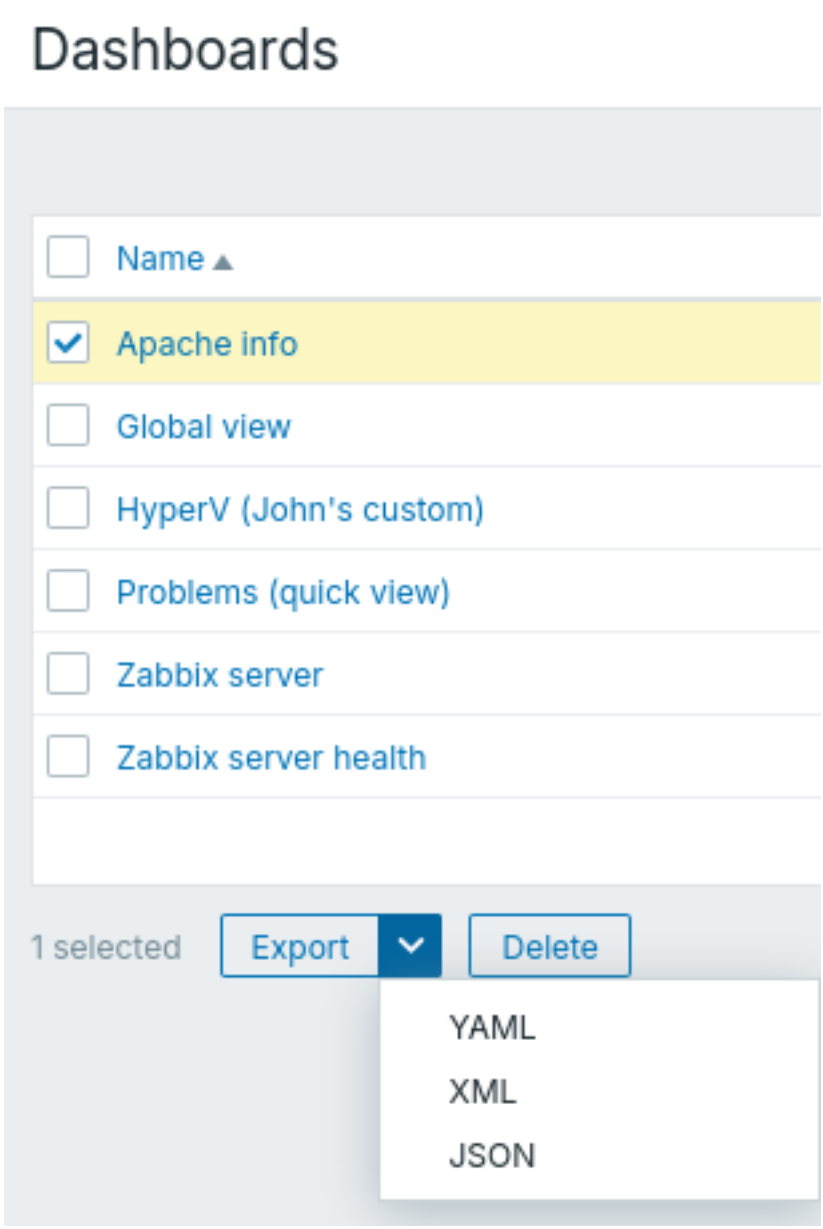
Overview

Dashboards are exported with all related objects and object relations.

Exporting

To export dashboards, do the following:

1. Go to *Dashboards* > *All dashboards*.
2. Mark the checkboxes of the dashboards to export.
3. Click *Export* below the list.



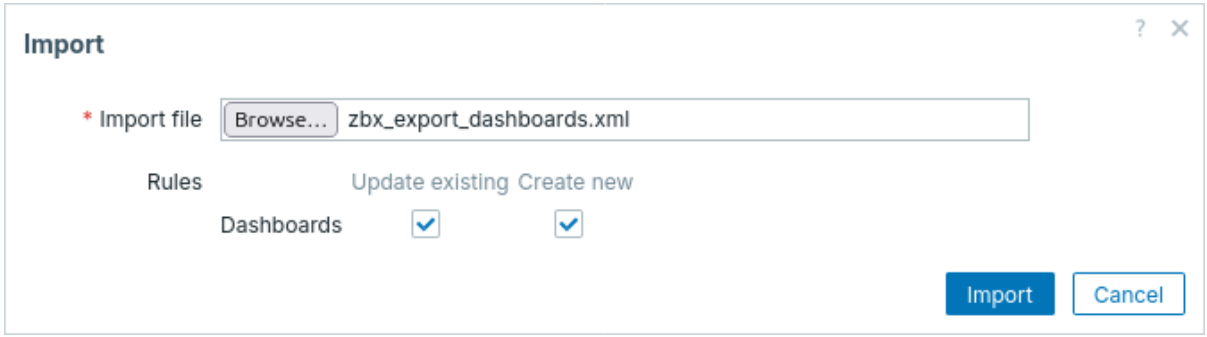
Depending on the selected format, dashboards are exported to a local file with a default name:

- `zbx_export_dashboards.yaml` - in YAML export (default option for export);
- `zbx_export_dashboards.xml` - in XML export;
- `zbx_export_dashboards.json` - in JSON export.

Importing

To import dashboards, do the following:

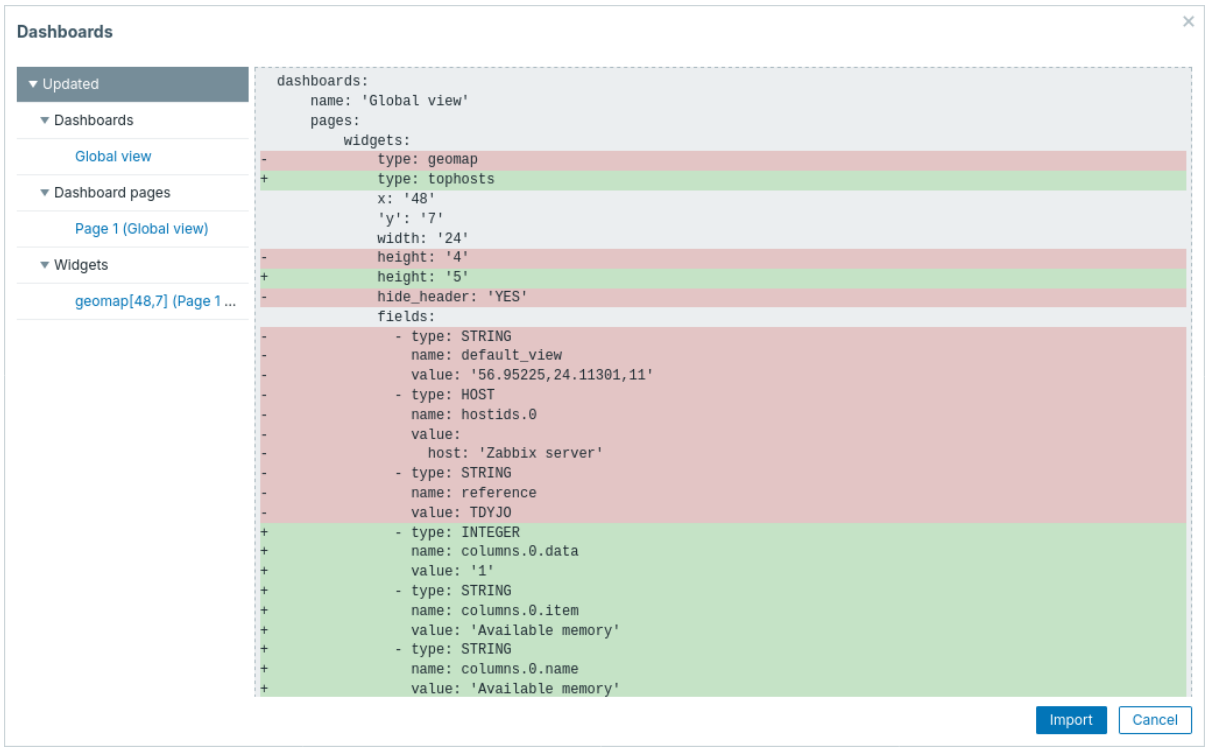
1. Go to *Dashboards* > *All dashboards*.
2. Click *Import* in the upper-right corner.
3. Select the import file.
4. Mark the required options in import rules.
5. Click *Import* in the lower-right corner of the configuration form.



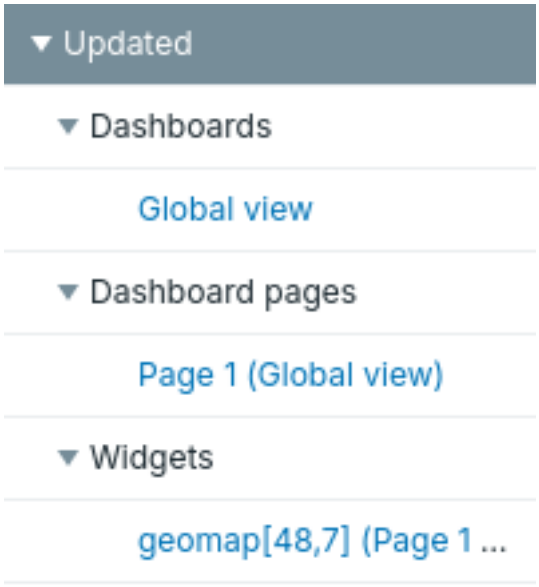
Import rules:

Rule	Description
Update existing	Existing elements will be updated using data from the import file. Otherwise, they will not be updated.
Create new	New elements will be created using data from the import file. Otherwise, they will not be created.

On the next screen, you will be able to view the content of dashboards being imported. If this is a new dashboard, all elements will be listed in green. If updating an existing dashboard, new dashboard elements will be highlighted in green; removed dashboard elements will be highlighted in red; elements that have not changed will be listed on a gray background.



The menu on the left can be used to navigate through the list of changes. The *Updated* section highlights all changes made to existing dashboard elements. The *Added* section lists new dashboard elements. The elements in each section are grouped by element type; click the gray arrow to expand or collapse the group of elements.



Review dashboard changes and then click *Import* to perform the dashboard import. A success or failure message of the import will be displayed in the frontend.

Export format

Export to YAML:

```yaml
zabbix_export:
  version: '8.0'
  dashboards:
    - name: 'Global view'
      pages:
        - widgets:
            - type: tophosts
              name: 'Top hosts by CPU utilization'
              width: '25'
              height: '4'
              fields:
                - type: INTEGER
                  name: column
                  value: '1'
                - type: INTEGER
                  name: columns.0.data
                  value: '2'
                - type: STRING
                  name: columns.0.item
                  value: 'System name'
                - type: STRING
                  name: columns.0.name
                  value: 'Host name'
                - type: STRING
                  name: columns.1.base_color
                  value: 4CAF50
                - type: INTEGER
                  name: columns.1.data
                  value: '1'
                - type: INTEGER
                  name: columns.1.display
                  value: '3'
                - type: STRING
                  name: columns.1.item
                  value: 'CPU utilization'
                - type: STRING
                  name: columns.1.max
                  value: '100'
                - type: STRING
                  name: columns.1.min
                  value: '0'
                - type: STRING
                  name: columns.1.name
                  value: Utilization
                - type: INTEGER
                  name: columns.2.data
                  value: '1'
                - type: STRING
                  name: columns.2.item
                  value: 'Load average (1m avg)'
                - type: STRING
                  name: columns.2.name
                  value: '1m avg'
                - type: INTEGER
                  name: columns.3.data
                  value: '1'
                - type: STRING
```

```
            name: columns.3.item
            value: 'Load average (5m avg)'
          - type: STRING
            name: columns.3.name
            value: '5m avg'
          - type: INTEGER
            name: columns.4.data
            value: '1'
          - type: STRING
            name: columns.4.item
            value: 'Load average (15m avg)'
          - type: STRING
            name: columns.4.name
            value: '15m avg'
          - type: INTEGER
            name: columns.5.data
            value: '1'
          - type: INTEGER
            name: columns.5.decimal_places
            value: '0'
          - type: STRING
            name: columns.5.item
            value: 'Number of processes'
          - type: STRING
            name: columns.5.name
            value: Processes
          - type: STRING
            name: columnsthresholds.1.color.0
            value: FFFF00
          - type: STRING
            name: columnsthresholds.1.color.1
            value: FF8000
          - type: STRING
            name: columnsthresholds.1.color.2
            value: FF4000
          - type: STRING
            name: columnsthresholds.1.threshold.0
            value: '50'
          - type: STRING
            name: columnsthresholds.1.threshold.1
            value: '80'
          - type: STRING
            name: columnsthresholds.1.threshold.2
            value: '90'
          - type: STRING
            name: reference
            value: NLVMO
      - type: hostavail
        'y': '4'
        width: '25'
        fields:
          - type: INTEGER
            name: interface_type.0
            value: '1'
      - type: problems
        name: 'Current problems'
        'y': '6'
        width: '48'
        height: '5'
        fields:
          - type: STRING
            name: reference
```

```
            value: BRUQE
          - type: INTEGER
            name: show
            value: '3'
          - type: INTEGER
            name: show_tags
            value: '3'
      - type: systeminfo
        x: '25'
        width: '23'
        height: '4'
      - type: problemsbysv
        name: 'Problems by severity'
        x: '25'
        'y': '4'
        width: '23'
        fields:
          - type: STRING
            name: reference
            value: LBSVB
          - type: INTEGER
            name: show_type
            value: '1'
      - type: gauge
        name: 'Zabbix server'
        x: '48'
        width: '12'
        height: '4'
        hide_header: 'YES'
        fields:
          - type: INTEGER
            name: angle
            value: '270'
          - type: INTEGER
            name: desc_size
            value: '8'
          - type: ITEM
            name: itemid.0
            value:
              host: 'Zabbix server'
              key: vm.memory.utilization
          - type: STRING
            name: max
            value: '100'
          - type: STRING
            name: min
            value: '0'
          - type: INTEGER
            name: scale_size
            value: '11'
          - type: INTEGER
            name: show.0
            value: '1'
          - type: INTEGER
            name: show.1
            value: '2'
          - type: INTEGER
            name: show.2
            value: '3'
          - type: INTEGER
            name: show.3
            value: '4'
```

```
      - type: INTEGER
        name: show.4
        value: '5'
      - type: STRING
        name: thresholds.0.color
        value: 42A5F5
      - type: STRING
        name: thresholds.0.threshold
        value: '0'
      - type: STRING
        name: thresholds.1.color
        value: 9CCC65
      - type: STRING
        name: thresholds.1.threshold
        value: '20'
      - type: STRING
        name: thresholds.2.color
        value: FFEE58
      - type: STRING
        name: thresholds.2.threshold
        value: '40'
      - type: STRING
        name: thresholds.3.color
        value: FFA726
      - type: STRING
        name: thresholds.3.threshold
        value: '60'
      - type: STRING
        name: thresholds.4.color
        value: FF7043
      - type: STRING
        name: thresholds.4.threshold
        value: '80'
      - type: INTEGER
        name: th_arc_size
        value: '9'
      - type: INTEGER
        name: th_show_arc
        value: '1'
      - type: INTEGER
        name: th_show_labels
        value: '1'
      - type: INTEGER
        name: units_bold
        value: '1'
      - type: INTEGER
        name: units_size
        value: '24'
      - type: INTEGER
        name: value_arc_size
        value: '30'
      - type: INTEGER
        name: value_bold
        value: '1'
      - type: INTEGER
        name: value_size
        value: '24'
    - type: svggraph
      x: '48'
      'y': '4'
      width: '24'
      height: '3'
```

```
      hide_header: 'YES'
      fields:
        - type: INTEGER
          name: axisx
          value: '0'
        - type: STRING
          name: ds.0.color
          value: AB47BC
        - type: INTEGER
          name: ds.0.fill
          value: '0'
        - type: STRING
          name: ds.0.hosts.0
          value: 'Zabbix server'
        - type: STRING
          name: ds.0.items.0
          value: 'Available memory'
        - type: INTEGER
          name: ds.0.transparency
          value: '2'
        - type: INTEGER
          name: ds.0.type
          value: '2'
        - type: INTEGER
          name: ds.0.width
          value: '2'
        - type: INTEGER
          name: ds.1.axisy
          value: '1'
        - type: STRING
          name: ds.1.color
          value: 42A5F5
        - type: INTEGER
          name: ds.1.fill
          value: '0'
        - type: STRING
          name: ds.1.hosts.0
          value: 'Zabbix server'
        - type: STRING
          name: ds.1.items.0
          value: 'CPU utilization'
        - type: INTEGER
          name: ds.1.transparency
          value: '0'
        - type: INTEGER
          name: ds.1.type
          value: '2'
        - type: INTEGER
          name: ds.1.width
          value: '2'
        - type: STRING
          name: reference
          value: TOYGS
    - type: geomap
      x: '48'
      'y': '7'
      width: '24'
      height: '4'
      hide_header: 'YES'
      fields:
        - type: STRING
          name: default_view
```

```
            value: '56.95225,24.11301,11'
          - type: HOST
            name: hostids.0
            value:
              host: 'Zabbix server'
          - type: STRING
            name: reference
            value: TDYJO
        - type: clock
          name: 'Local time'
          x: '60'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: clock_type
              value: '1'
            - type: INTEGER
              name: show.0
              value: '1'
            - type: INTEGER
              name: show.1
              value: '2'
            - type: INTEGER
              name: show.2
              value: '3'
            - type: INTEGER
              name: time_sec
              value: '0'
            - type: INTEGER
              name: tzone_format
              value: '1'
        - type: item
          name: 'Server performance'
          x: '60'
          'y': '2'
          width: '12'
          hide_header: 'YES'
          fields:
            - type: INTEGER
              name: aggregate_function
              value: '2'
            - type: INTEGER
              name: decimal_size
              value: '32'
            - type: STRING
              name: description
              value: 'Values per second'
            - type: INTEGER
              name: desc_h_pos
              value: '0'
            - type: INTEGER
              name: desc_v_pos
              value: '0'
            - type: ITEM
              name: itemid.0
              value:
                host: 'Zabbix server'
                key: 'zabbix[wcache,values]'
            - type: INTEGER
              name: show.0
              value: '1'
```

```
              - type: INTEGER
                name: show.1
                value: '2'
              - type: INTEGER
                name: show.2
                value: '5'
              - type: INTEGER
                name: sparkline.fill
                value: '2'
              - type: INTEGER
                name: sparkline.width
                value: '2'
              - type: STRING
                name: thresholds.0.color
                value: DCEDC8
              - type: STRING
                name: thresholds.0.threshold
                value: '20'
              - type: STRING
                name: thresholds.1.color
                value: FFF9C4
              - type: STRING
                name: thresholds.1.threshold
                value: '40'
              - type: STRING
                name: thresholds.2.color
                value: FFCCBC
              - type: STRING
                name: thresholds.2.threshold
                value: '60'
              - type: INTEGER
                name: units_bold
                value: '0'
              - type: INTEGER
                name: units_show
                value: '0'
              - type: INTEGER
                name: value_h_pos
                value: '0'
              - type: INTEGER
                name: value_size
                value: '32'
```

Exported elements Exported elements are explained in the table below.

Element	Type	Description
name	string	(required) Dashboard name.
pages		Root element for dashboard pages
widgets		Root element for dashboard pages

Note:
See also: Dashboard object (refer to the relevant property with a matching name).

Dashboard widget

Element	Type	Description
type	string	(required) Type of the dashboard widget. Possible values: actionlog - Action log; clock - Clock; discovery - Discovery status; favgraphs - Favorite graphs; favmaps - Favorite maps; gauge - Gauge; geomap - Geomap; graph - Graph (classic); graphprototype - Graph prototype; honeycomb - Honeycomb; hostavail - Host availability; hostcard - Host card; hostnavigator - Host navigator; itemcard - Item card; itemhistory - Item history; itemnavigator - Item navigator; item - Item value; map - Map; navtree - Map Navigation Tree; piechart - Pie chart; problemhosts - Problem hosts; problems - Problems; problemsbysv - Problems by severity; scatterplot - Scatter plot; slareport - SLA report; svggraph - Graph; systeminfo - System information; tophosts - Top hosts; topitems - Top items; toptriggers - Top triggers; trigover - Trigger overview; url - URL; web - Web monitoring.
x	integer	A horizontal position from the left side of the dashboard. Possible values:[1] 0-71.
y	integer	A vertical position from the top of the dashboard. Possible values:[1] 0-63.
name	string	Custom widget name.
width	integer	The widget width. Possible values:[1] 1-72.
height	integer	The widget height. Possible values:[1] 1-64.
hide_header	string	Hide widget header. Possible values:[1] NO (0, default), YES (1).
fields		Root element for the dashboard widget fields.

Note:
See also: Dashboard widget (refer to the relevant property with a matching name).

Dashboard widget field

Element	Type	Description
type	string	(required) Widget field type. Possible values:[1] INTEGER (0), STRING (1), HOST_GROUP (2), HOST (3), ITEM (4), ITEM_PROTOTYPE (5), GRAPH (6), GRAPH_PROTOTYPE (7), MAP (8), SERVICE (9), SLA (10), USER (11), ACTION (12), MEDIA_TYPE (13).

Element	Type	Description
name	string	(required) Widget field name.
value	mixed	(required) Widget field value, depending on the field type.
host	string	Item host.
key	string	Item key.

Note:
See also: Dashboard widget field (refer to the relevant property with a matching name).

Footnotes

[1] API integer values in brackets, for example, ENABLED (0), are mentioned only for reference. For more information, see the linked API object page in the table entry or at the end of each section.

13 Atklāšana

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam sadaļā Atklāšana.

1 Tīkla atklāšana

Pārskats

Zabbix piedāvā automātisku tīkla atklāšanas funkcionalitāti, kas ir efektīva un ļoti elastīga.

Ja tīkla atklāšana ir pareizi iestatīta, jūs varat:

- paātrināt Zabbix ieviešanu
- vienkāršot administrēšanu
- izmantot Zabbix strauji mainīgās vidēs bez pārmērīgas administrēšanas

Zabbix tīkla atklāšana ir balstīta uz šādu informāciju:

- IP diapazoni
- Ārējo pakalpojumu pieejamība (FTP, SSH, WEB, POP3, IMAP, TCP u.c.)
- Informācija, kas saņemta no Zabbix aģenta (tiek atbalstīts tikai nešifrēts režīms)
- Informācija, kas saņemta no SNMP aģenta

Tā NENODROŠINA:

- Tīkla topoloģijas atklāšanu

Tīkla atklāšana pamatā sastāv no divām fāzēm: atklāšana un darbības.

Atklāšana

Zabbix periodiski skenē IP diapazonus, kas definēti tīkla atklāšanas noteikumos. Pārbaudes biežumu var konfigurēt katram noteikumam atsevišķi.

Katram noteikumam ir definēts pakalpojumu pārbaužu kopums, kas jāveic norādītajam IP diapazonam.

Atklāšanas noteikumus apstrādā atklāšanas pārvaldnieks. Atklāšanas pārvaldnieks katram noteikumam izveido uzdevumu ar uzdevumu sarakstu (tīkla pārbaudes). Tīkla pārbaudes tiek veiktas paralēli ar pieejamajiem atklāšanas darbiniekiem (to skaitu var konfigurēt lietotāja saskarnē katram noteikumam atsevišķi). Tikai pārbaudes ar vienādu IP un portu tiek ieplānotas secīgi, jo dažas ierīces nepieņem paralēlus savienojumus vienā un tajā pašā portā.

Tīkla pārbaužu rindas lielums ir ierobežots līdz aptuveni 2000000 vai 4 GB atmiņas. Ja rinda kļūst pilna, atklāšanas noteikums tiks izlaists un žurnālā tiks izdrukāts brīdinājuma ziņojums. Varat izmantot iekšējo vienumu `zabbix[discovery_queue]`, lai uzraudzītu atklāšanas pārbaužu skaitu rindā.

Atklāšanas pārbaudes tiek apstrādātas neatkarīgi no citām pārbaudēm. Ja kāda no pārbaudēm neatrod pakalpojumu (vai neizdodas), pārējās pārbaudes tik un tā tiks apstrādātas.

Note:
Ja atklāšanas noteikums tiek mainīts izpildes laikā, pašreizējā atklāšanas izpilde tiks pārtraukta.

Katra pakalpojuma un hosta (IP) pārbaude, ko veic tīkla atklāšanas modulis, ģenerē atklāšanas notikumu.

Notikums	Pakalpojuma pārbaudes rezultāts
Pakalpojums atklāts	Pakalpojums ir ’up’ pēc tam, kad tas bija ’down’, vai arī tas ir atklāts pirmo reizi.
Pakalpojums darbojas	Pakalpojums ir ’up’ pēc tam, kad tas jau bija ’up’.
Pakalpojums zaudēts	Pakalpojums ir ’down’ pēc tam, kad tas bija ’up’.
Pakalpojums nedarbojas	Pakalpojums ir ’down’ pēc tam, kad tas jau bija ’down’.
Hosts atklāts	Vismaz viens hosta pakalpojums ir ’up’ pēc tam, kad visi šī hosta pakalpojumi bija ’down’, vai arī ir atklāts pakalpojums, kas pieder nereģistrētam hostam.
Hosts darbojas	Vismaz viens hosta pakalpojums ir ’up’ pēc tam, kad vismaz viens pakalpojums jau bija ’up’.
Hosts zaudēts	Visi hosta pakalpojumi ir ’down’ pēc tam, kad vismaz viens bija ’up’.
Hosts nedarbojas	Visi hosta pakalpojumi ir ’down’ pēc tam, kad tie jau bija ’down’.

Darbības

Atklāšanas notikumi var būt par pamatu atbilstošām darbībām, piemēram:

- Paziņojumu sūtīšana
- Hostu pievienošana/noņemšana
- Hostu iespējošana/atspējošana
- Hostu pievienošana grupai
- Hostu noņemšana no grupas
- Birku pievienošana hostam
- Birku noņemšana no hosta
- Veidnes piesaistīšana hostiem/veidnes atsaistīšana no hostiem
- Attālo skriptu izpilde

Šīs darbības var konfigurēt atbilstoši ierīces tipam, IP, statusam, darbības laikam/dīkstāves laikam utt. Pilnu informāciju par darbību konfigurēšanu uz tīkla atklāšanu balstītiem notikumiem skatiet lapās par darbību operācijām un nosacījumiem.

Tā kā tīkla atklāšanas darbības ir balstītas uz notikumiem, tās tiks izpildītas gan tad, kad atklātais hosts ir tiešsaistē, gan tad, kad tas ir bezsaistē. Ir ļoti ieteicams pievienot darbības nosacījumu *Atklāšanas statuss: darbojas*, lai izvairītos no tā, ka tādas darbības kā *Pievienot hostu* tiek izpildītas pie *Pakalpojums zaudēts*/*Pakalpojums nedarbojas* notikumiem. Pretējā gadījumā, ja atklātais hosts tiek noņemts manuāli, tas joprojām ģenerēs *Pakalpojums zaudēts*/*Pakalpojums nedarbojas* notikumus un tiks izveidots no jauna nākamajā atklāšanas ciklā.

Note:
Veidņu piesaistīšana atklātam hostam kolektīvi neizdosies, ja kādai no piesaistāmajām veidnēm ir unikāla entītija (piemēram, vienuma atslēga), kas sakrīt ar unikālu entītiju (piemēram, vienuma atslēgu), kura jau pastāv hostā vai citā no piesaistāmajām veidnēm.

Hosta izveide

Hosts tiek pievienots, ja ir atlasīta operācija *Pievienot hostu*. Hosts tiek pievienots arī tad, ja operācijas *Pievienot hostu* nav, bet jūs atlasāt operācijas, kuru rezultātā tiek veiktas darbības ar hostu. Šādas operācijas ir:

- iespējot hostu
- atspējot hostu
- pievienot hostu hostu grupai
- saistīt veidni ar hostu

Izveidotie hosti tiek pievienoti grupai *Atklātie hosti* (*Discovered hosts*) (pēc noklusējuma; konfigurējams sadaļā *Administrēšana* > *Vispārīgi* > *Cits*). Ja vēlaties, lai hosti tiktu pievienoti citai grupai, pievienojiet operāciju *Noņemt no hostu grupām* (norādot ”Discovered hosts”) un arī operāciju *Pievienot hostu grupām* (norādot citu hostu grupu), jo hostam ir jāpieder hostu grupai.

Atklātās ierīces IP adrese kopā ar atklāšanas avotu (Zabbix serveris, Zabbix starpniekserveris vai starpniekserveru grupa) un saskarnes tipu tiek izmantota kā kritērijs hosta atrašanai sistēmā. Ja hosts ar tādu pašu IP adresi, saskarnes tipu un atklāšanas avotu jau pastāv, šis hosts būs mērķis operāciju izpildei. Ja atklāšanas avots atšķiras, atklātā entītija tiek uzskatīta par citu hostu, un var tikt izveidots jauns hosts.

Ja atklātā hosta IP adrese tiek mainīta vai saskarne tiek dzēsta, nākamās atklāšanas laikā tiks izveidots jauns hosts.

Hostu nosaukumu piešķiršana

Pievienojot hostus, hosta nosaukums tiek noteikts pēc reversās DNS uzmeklēšanas rezultāta vai IP adreses, ja reversā uzmeklēšana neizdodas. Uzmeklēšanu veic Zabbix serveris vai Zabbix starpniekserveris atkarībā no tā, kurš veic atklāšanu. Ja uzmeklēšana neizdodas starpniekserverī, tā netiek atkārtota serverī. Ja hosts ar šādu nosaukumu jau pastāv, nākamajam hostam nosaukumam tiks pievienots **_2**, pēc tam **_3** un tā tālāk.

Ir iespējams arī ignorēt DNS/IP uzmeklēšanu un tā vietā hosta nosaukumam izmantot vienuma vērtību, piemēram:

- Varat atklāt vairākus serverus ar darbojošos Zabbix aģentu, izmantojot Zabbix aģenta vienumu atklāšanai, un automātiski piešķirt tiem atbilstošus nosaukumus, pamatojoties uz šī vienuma atgriezto virknes vērtību
- Varat atklāt vairākas SNMP tīkla ierīces, izmantojot SNMP aģenta vienumu atklāšanai, un automātiski piešķirt tām atbilstošus nosaukumus, pamatojoties uz šī vienuma atgriezto virknes vērtību

Ja hosta nosaukums ir iestatīts, izmantojot vienuma vērtību, turpmāko atklāšanas pārbaužu laikā tas netiek atjaunināts. Ja nav iespējams iestatīt hosta nosaukumu, izmantojot vienuma vērtību, tiek izmantota noklusējuma vērtība (DNS nosaukums).

Ja hosts ar atklāto IP adresi jau pastāv un atklāšanas avots (Zabbix serveris, starpniekserveris vai starpniekserveru grupa) nav mainījies, jauns hosts netiek izveidots. Ja atklāšanas avots atšķiras, atklātā entītija tiek uzskatīta par atsevišķu, un var tikt izveidots jauns hosts. Tomēr, ja atklāšanas darbība satur operācijas (veidnes piesaiste, pievienošana hostu grupai utt.), tās tiek izpildītas esošajam hostam, kas atbilst pēc IP adreses, saskarnes tipa un atklāšanas avota.

Hostu noņemšana

Hosti, ko atklāj tīkla atklāšanas kārtula, tiek automātiski noņemti no *Monitoring* > *Discovery*, ja atklātā entītija vairs neatrodas kārtulas IP diapazonā. Hosti tiek noņemti nekavējoties.

Saskarņu izveide, pievienojot hostus

Kad hosti tiek pievienoti tīkla atklāšanas rezultātā, tiem tiek izveidotas saskarnes saskaņā ar šādiem noteikumiem:

- Tiek ņemti vērā atklātie servisi — piemēram, ja SNMP pārbaude bija veiksmīga, tiks izveidota SNMP saskarne.
- Ja hosts atbildēja gan uz Zabbix aģents, gan SNMP pieprasījumiem, tiks izveidoti abu tipu saskarnes.
- Ja unikalitātes kritēriji ir Zabbix aģents vai SNMP atgrieztie dati, pirmā hostam atrastā saskarne tiks izveidota kā noklusējuma saskarne. Pārējās IP adreses tiks pievienotas kā papildu saskarnes. Darbības nosacījumi (piemēram, hosta IP) neietekmē saskarņu pievienošanu. *Ņemiet vērā*, ka tas darbosies, ja visas saskarnes tiek atklātas ar to pašu atklāšanas kārtulu. Ja cita atklāšanas kārtula atklāj citu tā paša hosta saskarni, tiks pievienots papildu hosts.
- Ja hosts atbildēja tikai uz aģents pārbaudēm, tas tiks izveidots tikai ar aģents saskarni. Ja tas vēlāk sāktu atbildēt uz SNMP, tiktu pievienotas papildu SNMP saskarnes.
- Ja sākotnēji tika izveidoti 3 atsevišķi hosti, kas tika atklāti pēc "IP" unikalitātes kritērija, un pēc tam atklāšanas kārtula tiek modificēta tā, ka hostiem A, B un C ir identisks unikalitātes kritēriju rezultāts, B un C tiek izveidoti kā papildu saskarnes hostam A, pirmajam hostam. Atsevišķie hosti B un C paliek. Sadaļā *Monitoring* > *Discovery* pievienotās saskarnes tiks parādītas kolonnā "Discovered device" ar melnu fontu un atkāpi, bet kolonnā "Monitored host" tiks parādīts tikai A — pirmais izveidotais hosts. "Uptime/Downtime" netiek mērīts IP adresēm, kas tiek uzskatītas par papildu saskarnēm.

Starpniekservera iestatījuma maiņa

Dažādu starpniekserveru atklātie hosti ne vienmēr tiek apstrādāti kā atšķirīgi hosti. Atklāšanas un unikalitātes pārbaudes ir atkarīgas no starpniekserveru grupas struktūras: kad starpniekserveris izpilda atklāšanas kārtulu un izveido hostu, šis hosts tiek pievienots starpniekservera vecākajai starpniekserveru grupai, nevis piešķirts pašam starpniekserverim. Kad Zabbix atklāšanas laikā novērtē IP unikalitāti, tas pārbauda hostus, kurus uzrauga vecākā starpniekserveru grupa. Hosti, kurus uzrauga atsevišķi starpniekserveri šajā grupā (ieskaitot starpniekserveri, kas izpildīja atklāšanu), unikalitātes pārbaudē tiek ignorēti, kas var izraisīt dublētus hostus, ja vairāki starpniekserveri uzrauga pārklājošus apakštīklus.

Lai gan šī darbība ļauj atklāšanai darboties pārklājošos IP diapazonos, ko izmanto dažādi apakštīkli, jau uzraudzīta apakštīkla piešķirtā starpniekservera maiņa ir sarežģītāka, jo starpniekservera izmaiņas ir konsekventi jāpiemēro atklātajiem hostiem un vecākās starpniekserveru grupas dalībai, lai izvairītos no dublikātiem.

Piemēram, darbības, lai aizstātu starpniekserveri atklāšanas kārtulā:

1. atspējot atklāšanas kārtulu
2. sinhronizēt starpniekservera konfigurāciju
3. aizstāt starpniekserveri atklāšanas kārtulā
4. aizstāt starpniekserveri visiem hostiem, kas atklāti ar šo kārtulu (pārliecinieties, ka tiek atjaunināti hosti vecākajā starpniekserveru grupā un visi hosti, kurus uzrauga atsevišķi starpniekserveri šajā grupā, lai izvairītos no dublikātiem)
5. iespējot atklāšanas kārtulu

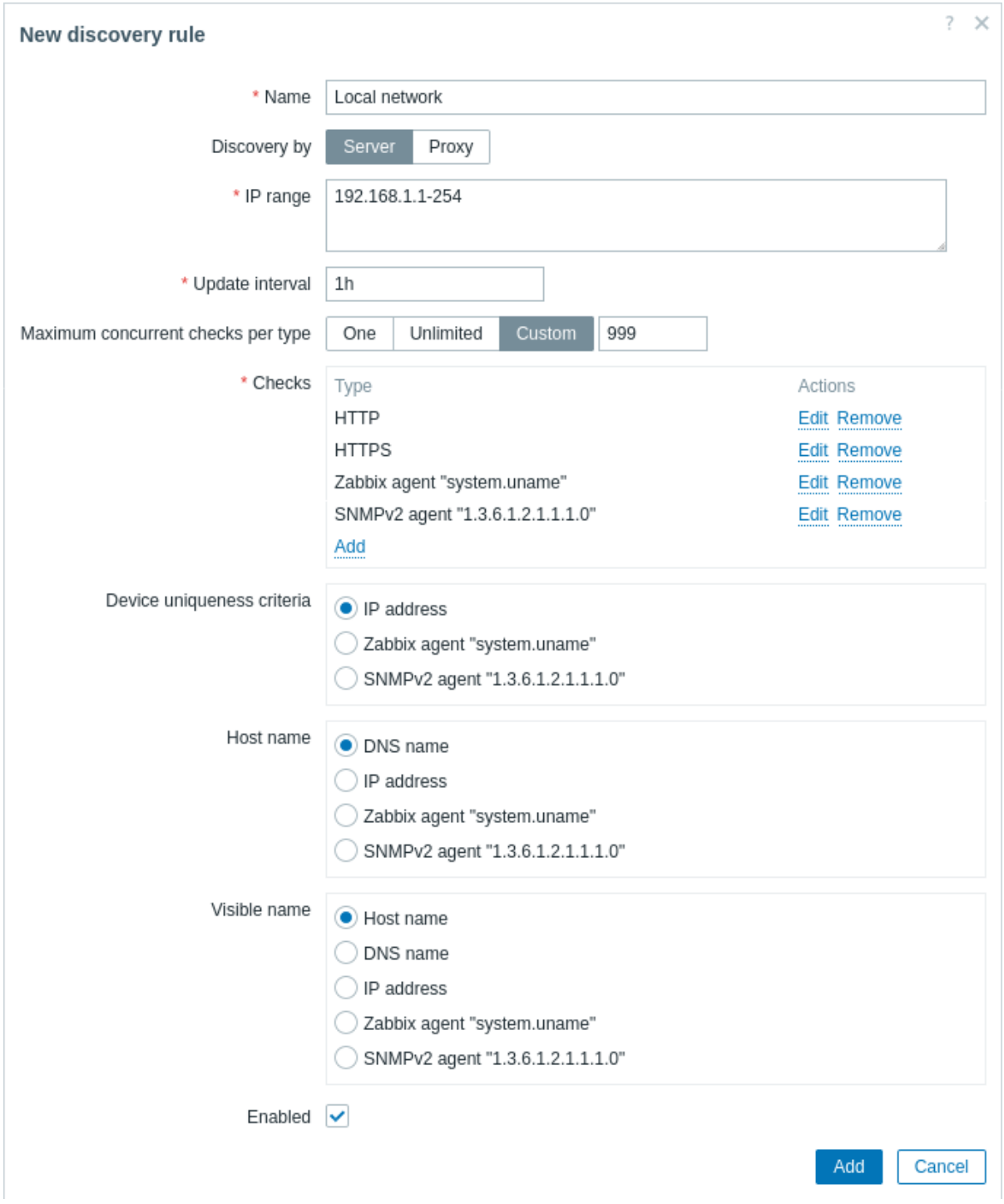
1 Tīkla atklāšanas kārtulas konfigurēšana

Pārskats

Lai konfigurētu tīkla atklāšanas kārtulu, ko Zabbix izmanto hostu un pakalpojumu atklāšanai:

- Dodieties uz *Datu ievākšana → Atklāšana*
- Noklikšķiniet uz *Izveidot atklāšanas kārtulu* (vai uz kārtulas nosaukuma, lai rediģētu esošu)
- Rediģējiet atklāšanas kārtulas atribūtus

Noteikuma atribūti



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Name	Unikāls noteikuma nosaukums. Piemēram, ”Lokālais tīkls”.
Discovery by	Atklāšana tiek veikta ar: **Server** - ar Zabbix serveri **Proxy** - ar Zabbix starpniekserveri (izvēlēts starpniekservera nosaukuma laukā)

Parametrs	Apraksts
IP range	IP adrešu diapazons atklāšanai. Tam var būt šādi formāti: Viena IP adrese: 192.168.1.33 IP adrešu diapazons: 192.168.1-10.1-255. Diapazonu ierobežo kopējais aptverto adrešu skaits (mazāk nekā 64K). IP maska: 192.168.4.0/24 atbalstītās IP maskas: /16 - /30 IPv4 adresēm /112 - /128 IPv6 adresēm Saraksts: 192.168.1.1-255, 192.168.2.1-100, 192.168.2.200, 192.168.4.0/24 Šis lauks atbalsta atstarpes, tabulāciju un vairākas rindas.
Update interval	Šis parametrs nosaka, cik bieži Zabbix izpildīs noteikumu. Intervāls tiek mērīts pēc iepriekšējās atklāšanas instances izpildes beigām, tāpēc pārklāšanās nenotiek. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Ņemiet vērā*, ka, ja tiek izmantots lietotāja makross un tā vērtība tiek mainīta (piemēram, 1w → 1h), nākamā pārbaude tiks izpildīta saskaņā ar iepriekšējo vērtību (piemēra vērtību gadījumā — tālā nākotnē).
Maximum concurrent checks per type	Iestatiet maksimālo atklāšanas pavedienu (darbinieku) skaitu katrai servisa pārbaudei, lai atklāšanas pārbaudes apstrādātu paralēli: **One** - viens pavediens **Unlimited** - neierobežots pavedienu skaits (bet ne vairāk kā parametrā StartDiscoverers) **Custom** - iestatiet pielāgotu pavedienu skaitu (0-999) Ņemiet vērā, ka visi atklāšanas noteikumi ar SNMPv3 asinhronajām servisa pārbaudēm tiek apstrādāti ar vienu darbinieku libsnmp implementācijas īpatnību dēļ, t.i., darbinieku skaita palielināšana nepalielinās atklāšanas ātrumu.
Checks	Zabbix izmantos šo pārbaužu sarakstu atklāšanai. Noklikšķiniet uz Add, lai uznirstošajā logā konfigurētu jaunu pārbaudi. Atbalstītās pārbaudes: SSH, LDAP, SMTP, FTP, HTTP, HTTPS, POP, NNTP, IMAP, TCP, Telnet, Zabbix aģents, SNMPv1 aģents, SNMPv2 aģents, SNMPv3 aģents, ICMP ping. Uz protokolu balstīta atklāšana izmanto **net.tcp.service[]** funkcionalitāti, lai pārbaudītu katru hostu, izņemot SNMP, kas vaicā SNMP OID. Zabbix aģents tiek pārbaudīts, vaicājot vienumu nešifrētā režīmā. Lūdzu, skatiet aģenta vienumi, lai iegūtu sīkāku informāciju. Parametrs 'Ports' var būt viens no šiem: Viena pieslēgvieta: 22 Pieslēgvietu diapazons: 22-45 Saraksts: 22-45,55,60-70 Sākot ar Zabbix 7.0, visas servisa pārbaudes tiek veiktas asinhroni, izņemot LDAP pārbaudes. Sākot ar Zabbix 7.0, HTTP/HTTPs pārbaude tiek veikta, izmantojot libcurl. Ja Zabbix serveris/starpniekserveris ir kompilēts bez libcurl, tad HTTP pārbaudes darbosies kā iepriekšējās versijās (t.i., kā TCP pārbaudes), bet HTTPS pārbaudes nedarbosies.
Device uniqueness criteria	Unikalitātes kritēriji var būt: **IP address** - neapstrādāt vairākas ierīces ar vienu IP adresi. Ja ierīce ar tādu pašu IP jau pastāv, tā tiks uzskatīta par jau atklātu un jauns hosts netiks pievienots. **<discovery check>** - vai nu Zabbix aģenta, vai SNMP aģenta pārbaude. Ņemiet vērā, ka atklāšanas laikā izmantotie unikalitātes kritēriji nav tas pats, kas hosta identifikācija sistēmā, izpildot darbības. Unikalitātes kritēriji atklāšanas laikā nosaka, vai divas vai vairākas atklātās ierīces ir vienādas (vai atšķirīgas), savukārt Zabbix hosta identifikācijai kritērijs ir tikai IP adrese (skatiet Hosta izveide).
Host name	Iestatiet izveidotā hosta tehnisko nosaukumu, izmantojot: **DNS name** - DNS nosaukums (noklusējums) **IP address** - IP adrese **<discovery check>** - saņemtā virknes vērtība no atklāšanas pārbaudes (piemēram, Zabbix aģents, SNMP aģenta pārbaude) Skatiet arī: Hosta nosaukuma piešķiršana.

Parametrs	Apraksts
Visible name	Iestatiet izveidotā hosta redzamo nosaukumu, izmantojot: **Host name** - tehniskais hosta nosaukums (noklusējums) **DNS name** - DNS nosaukums **IP address** - IP adrese **<discovery check>** - saņemtā virknes vērtība no atklāšanas pārbaudes (piemēram, Zabbix aģents, SNMP aģenta pārbaude) Skatiet arī: Hosta nosaukuma piešķiršana.
Enabled	Ja izvēles rūtiņa ir atzīmēta, noteikums ir aktīvs un to izpildīs Zabbix serveris. Ja tā nav atzīmēta, noteikums nav aktīvs. Tas netiks izpildīts.

Failu deskriptoru limita pārsniegšana

Liela skaita vienlaicīgu pārbaužu gadījumā ir iespējams izsmelt failu deskriptoru limitu atklāšanas pārvaldniekam.

Atklāšanai nepieciešamo failu deskriptoru skaits ir vienāds ar atklāšanas darbinieku skaitu * 1000. Pēc noklusējuma ir 5 atklāšanas darbinieki, savukārt sistēmas mīkstais limits ir aptuveni 1024.

Ja šim limitam tiek pietuvots, Zabbix samazinās noklusējuma vienlaicīgo pārbaužu skaitu katram tipam katram darbiniekam un ierakstīs brīdinājumu žurnālfailā. Tomēr, ja lietotājs ir iestatījis lielāku vērtību parametram *Maksimālais vienlaicīgo pārbaužu skaits katram tipam* nekā Zabbix aprēķinātā vērtība, Zabbix izmantos lietotāja definēto vērtību vienam darbiniekam.

Reālas dzīves scenārijs

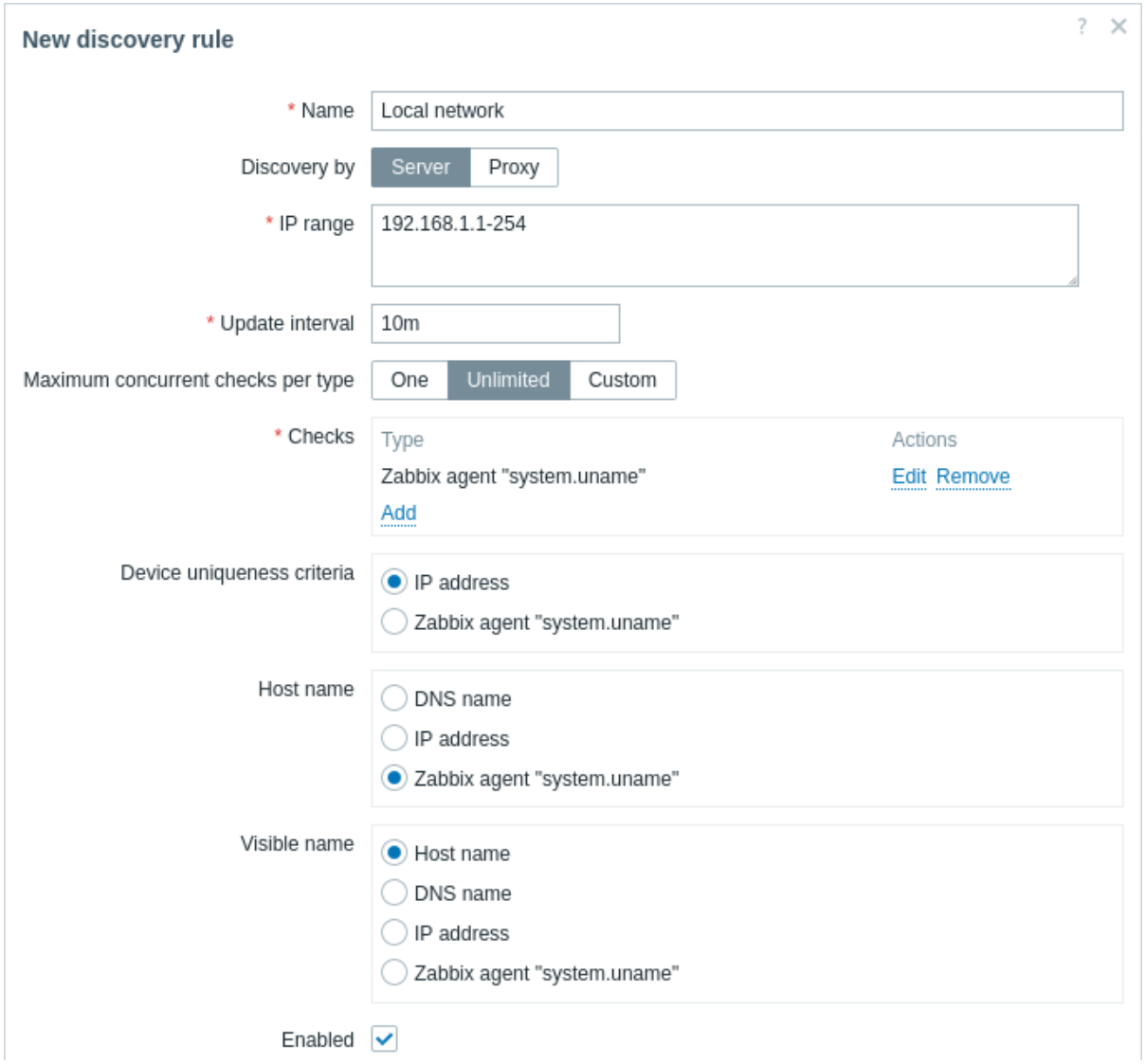
Šajā piemērā mēs vēlamies iestatīt tīkla atklāšanu lokālajam tīklam ar IP diapazonu 192.168.1.1-192.168.1.254.

Mūsu scenārijā mēs vēlamies:

- atklāt tos hostus, kuros darbojas Zabbix aģents
- palaist atklāšanu ik pēc 10 minūtēm
- pievienot hostu uzraudzībai, ja hosta darbības laiks pārsniedz 1 stundu
- noņemt hostus, ja hosta dīkstāves laiks pārsniedz 24 stundas
- pievienot Linux hostus grupai ”Linux servers”
- pievienot Windows hostus grupai ”Windows servers”
- izmantot veidni *Linux* Linux hostiem
- izmantot veidni *Windows* Windows hostiem

1. solis

Tīkla atklāšanas kārtulas definēšana mūsu IP diapazonam.



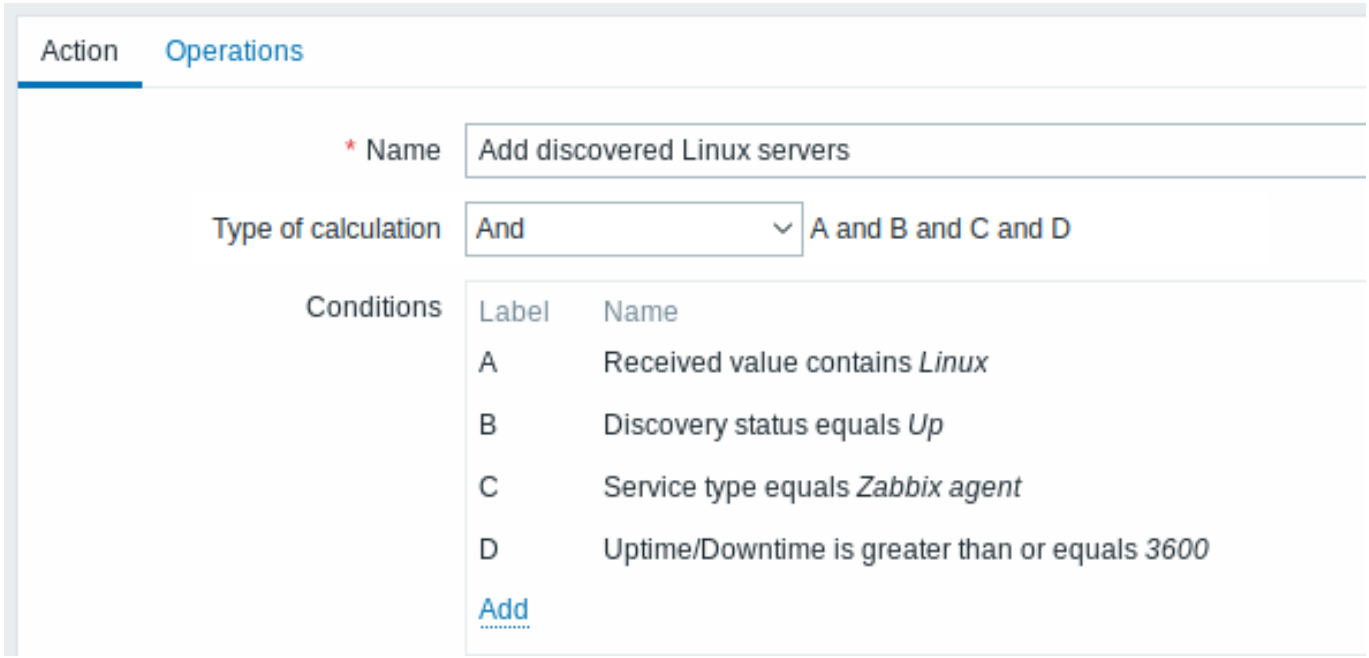
Zabbix mēģinās atklāt hostus IP diapazonā 192.168.1.1-192.168.1.254, pieslēdzoties Zabbix aģentiem un iegūstot vērtību no **system.uname** atslēgas. No aģenta saņemto vērtību var izmantot hostu nosaukšanai, kā arī dažādu darbību piemērošanai dažādām operētājsistēmām. Piemēram, saistīt Windows serverus ar veidni *Windows*, Linux serverus ar veidni *Linux*.

Kārtula tiks izpildīta ik pēc 10 minūtēm.

Kad šī kārtula būs pievienota, Zabbix automātiski sāks atklāšanu un uz atklāšanu balstītu notikumu ģenerēšanu turpmākai apstrādei.

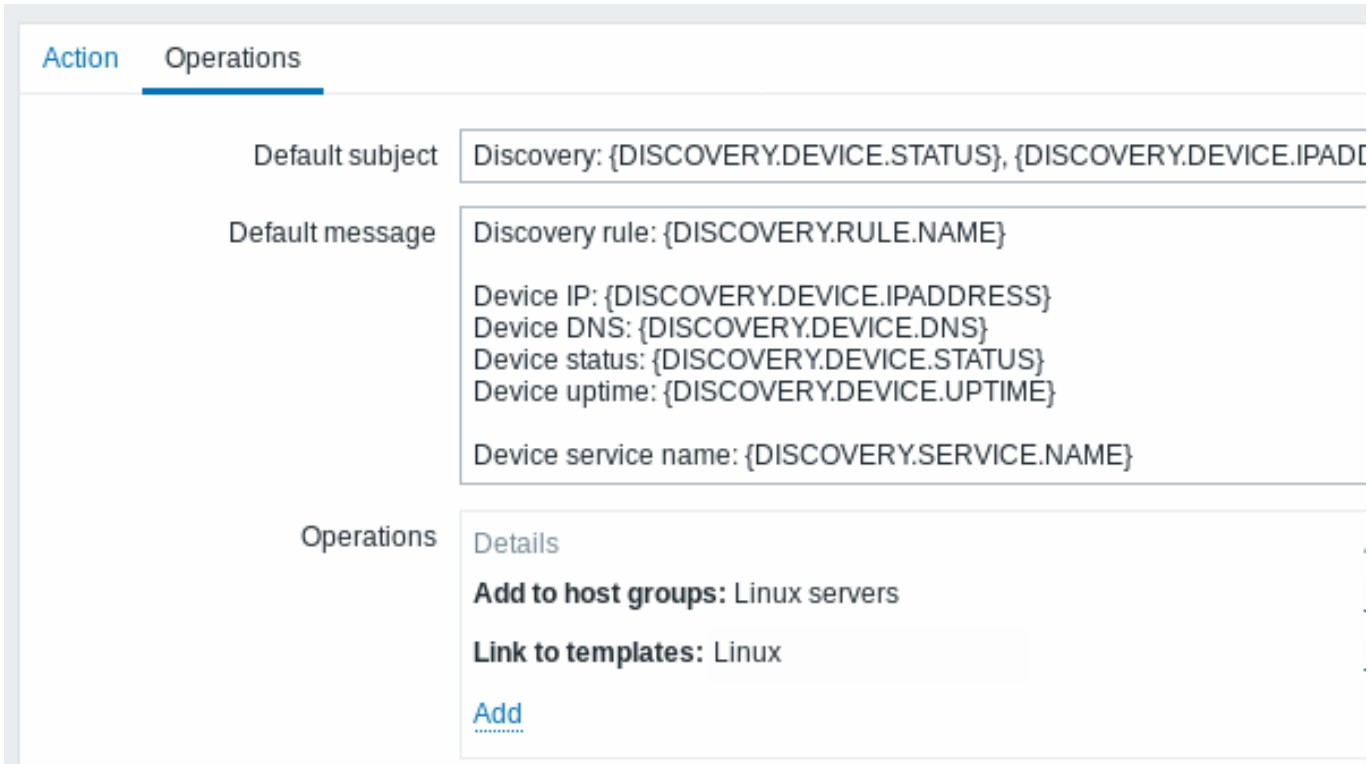
2. solis

Atklāšanas darbības definēšana, lai pievienotu atklātos Linux serverus attiecīgajai grupai/veidnei.



Darbība tiks aktivizēta, ja:

- pakalpojums ”Zabbix aģents” ir ”up”
- system.uname vērtība (Zabbix aģenta atslēga, ko izmantojām noteikuma definīcijā) satur ”Linux”
- Uptime ir 1 stunda (3600 sekundes) vai vairāk

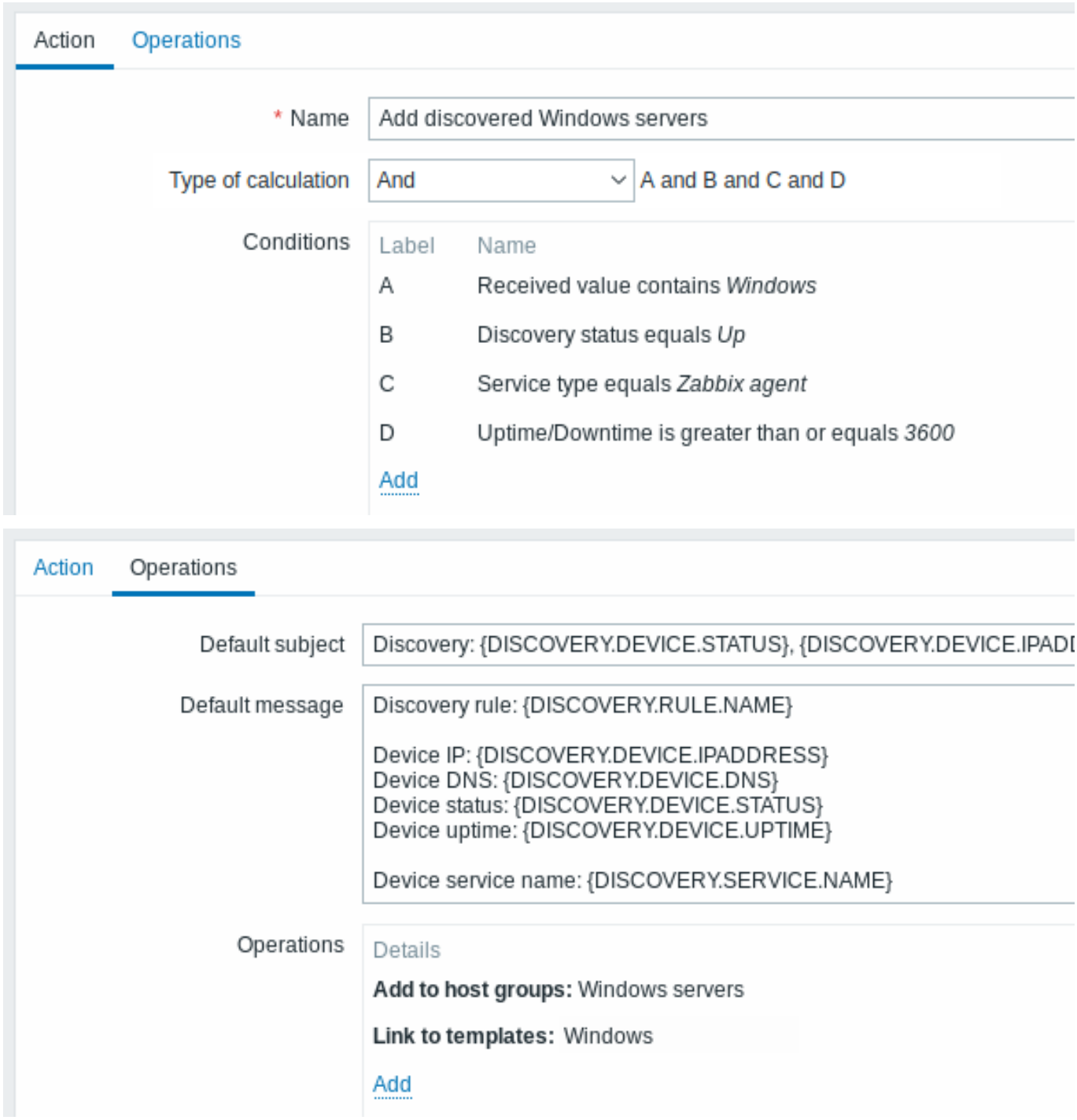


Darbība izpildīs šādas operācijas:

- pievienos atklāto hostu grupai ”Linux servers” (un arī pievienos hostu, ja tas iepriekš nebija pievienots)
- piesaistīs hostu veidnei *Linux*. Zabbix automātiski sāks uzraudzīt hostu, izmantojot vienumus un trigerus no veidnes ”Linux”.

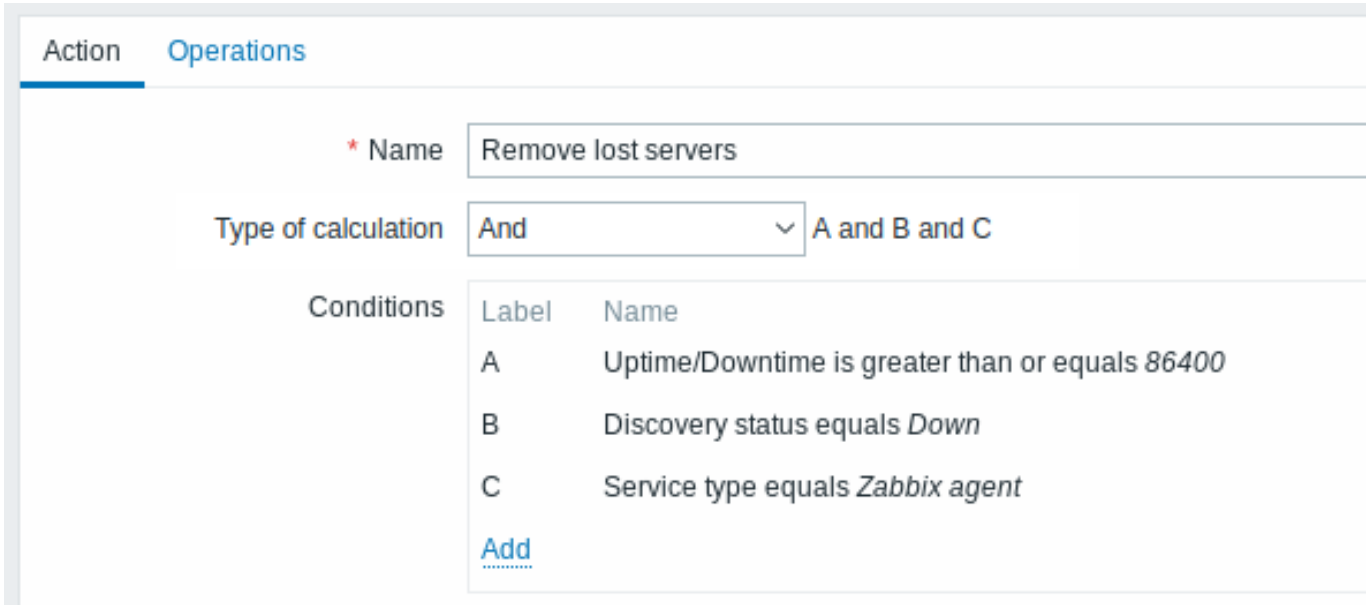
3. solis

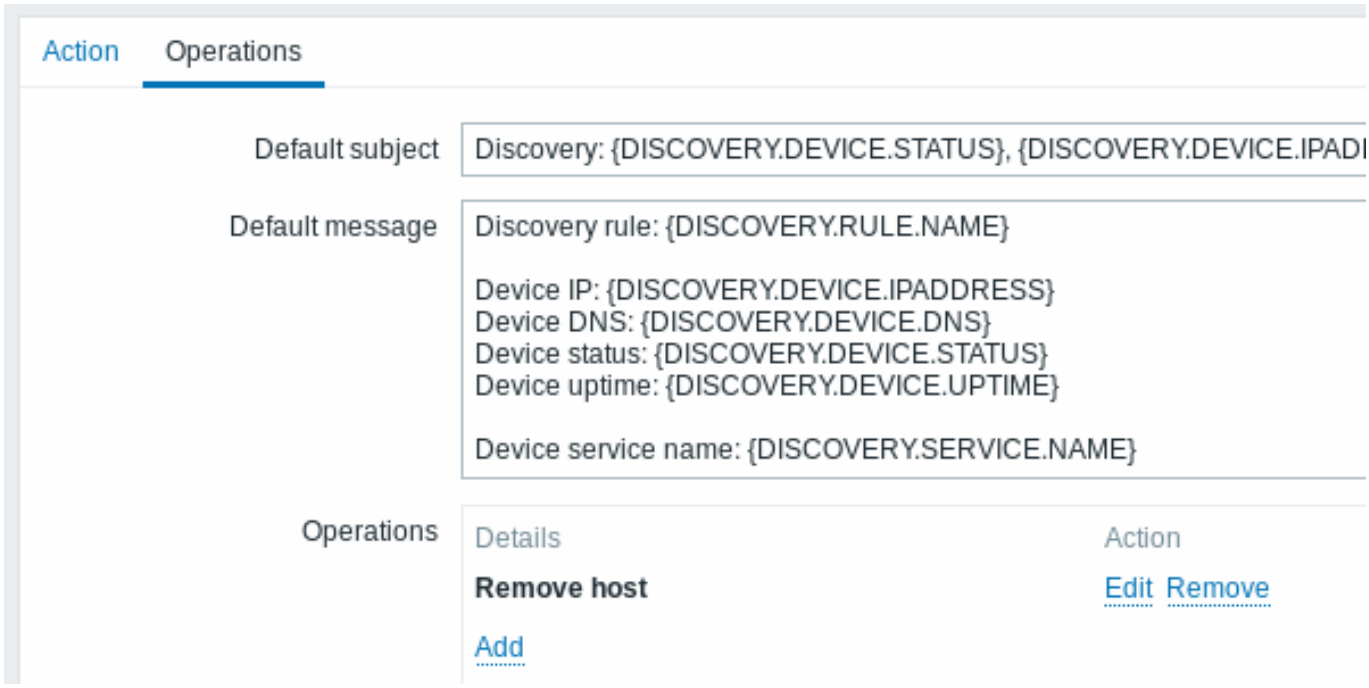
Atklāšanas darbības definēšana, lai pievienotu atklātos Windows serverus attiecīgajai grupai/veidnei.



4. solis

Atklāšanas darbības definēšana zaudētu serveru noņemšanai.





Serveris tiks noņemts, ja pakalpojums ”Zabbix aģents” būs ’down’ ilgāk par 24 stundām (86400 sekundēm).

2 Aktīvā aģenta automātiskā reģistrācija

Pārskats

Ir iespējams atļaut aktīvo Zabbix aģentu automātisko reģistrāciju, pēc kuras serveris var sākt to uzraudzību. Šādā veidā jaunus hostus var pievienot uzraudzībai, nekonfigurējot tos manuāli serverī.

Automātiskā reģistrācija var notikt, kad iepriekš nezināms aktīvs aģents pieprasa pārbaudes.

Šī funkcija var būt ļoti noderīga jaunu Cloud mezglu automātiskai uzraudzībai. Tiklīdz Cloud vidē ir jauns mezgls, Zabbix automātiski sāks hosta veiktspējas un pieejamības datu vākšanu.

Aktīvo aģentu automātiskā reģistrācija atbalsta arī pievienoto hostu uzraudzību ar pasīvajām pārbaudēm. Kad aktīvais aģents pieprasa pārbaudes, ar nosacījumu, ka tā konfigurācijas failā ir definēti konfigurācijas parametri `ListenIP` vai `ListenPort`, tie tiek nosūtīti serverim. Ja ir norādītas vairākas IP adreses, serverim tiek nosūtīta pirmā no tām.

Serveris, pievienojot jauno automātiski reģistrēto hostu, aģenta konfigurēšanai izmanto saņemto IP adresi un portu. Ja netiek saņemta IP adreses vērtība, tiek izmantota ienākošajam savienojumam izmantotā adrese. Ja netiek saņemta porta vērtība, tiek izmantots 10050.

Ir iespējams norādīt, ka hosts ir automātiski jāreģistrē ar DNS nosaukumu kā noklusējuma aģenta saskarni.

Automātiskā reģistrācija tiek palaista atkārtoti:

- ja mainās hosta metadatu informācija:
  - jo ir mainīts HostMetadata un aģents ir pārstartēts
  - jo ir mainījusies HostMetadataItem atgrieztā vērtība
- manuāli izveidotiem hostiem, kuriem trūkst metadatu
- ja hosts manuāli tiek mainīts uz uzraudzību ar citu Zabbix starpniekserveri
- ja automātiskā reģistrācija tam pašam hostam tiek saņemta no jauna Zabbix starpniekservera

Aktīvā aģenta automātiskās reģistrācijas sirdspukstu intervāls Zabbix serverim un Zabbix starpniekserverim ir 120 sekundes. Tāpēc, ja atklāts hosts tiek dzēsts, automātiskā reģistrācija tiks palaista atkārtoti pēc 120 sekundēm.

Konfigurācija

Norādiet serveri

Pārliecinieties, ka Zabbix serveris ir norādīts aģenta konfigurācijas failā - `zabbix_agentd.conf`:

```
ServerActive=10.0.0.1
```

Ja vien `zabbix_agentd.conf` failā īpaši nenorādāt *Hostname*, serveris hosta nosaukuma piešķiršanai izmantos aģenta atrašanās vietas sistēmas hosta nosaukumu. Sistēmas hosta nosaukumu Linux vidē var iegūt, izpildot komandu `hostname`.

Ja *Hostname* Zabbix aģenta konfigurācijā ir definēts kā ar komatiem atdalīts hostu saraksts, hosti tiks izveidoti visiem norādītajiem hostu nosaukumiem.

Pēc jebkādu izmaiņu veikšanas konfigurācijas failā restartējiet aģentu.

Darbība aktīvo aģentu automātiskajai reģistrācijai

Kad serveris saņem automātiskās reģistrācijas pieprasījumu no aģenta, tas izsauc darbību. Aģenta automātiskajai reģistrācijai ir jākonfigurē darbība ar notikuma avotu "Autoregistration".

> **Note:**
> Lai aktīvie aģenti varētu automātiski reģistrēties, nav nepieciešams konfigurēt tīkla atklāšanu.

Zabbix lietotāja saskarnē atveriet *Brīdinājumi → Darbības*, atlasiet *Automātiskās reģistrācijas darbības* un noklikšķiniet uz *Izveidot darbību*:

- Cilnē Darbība piešķiriet darbībai nosaukumu
- Pēc izvēles norādiet nosacījumus. Nosacījumos var izmantot apakšvirknes atbilstību vai regulārās izteiksmes atbilstību hosta nosaukumam/hosta metadatiem. Ja izmantosiet nosacījumu "Host metadata", skatiet nākamo sadaļu.
- Cilnē Darbības pievienojiet atbilstošās darbības, piemēram, - 'Pievienot hostu', 'Pievienot hostu grupai' (piemēram, *Discovered hosts*), 'Piesaistīt veidnes' utt.

> **Note:**
> Ja hosti, kas tiks automātiski reģistrēti, visticamāk atbalstīs tikai aktīvo uzraudzību (piemēram, hosti, kurus no jūsu Zabbix servera puses aizsargā ugunsmūris), iespējams, vēlēsieties izveidot īpašu veidni, piemēram, *Template_Linux-active*, ko piesaistīt.

Izveidotie hosti tiek pievienoti grupai *Discovered hosts* (pēc noklusējuma; to var konfigurēt sadaļā *Administrēšana > Vispārīgi > Citi*). Ja vēlaties, lai hosti tiktu pievienoti citai grupai, pievienojiet darbību *Noņemt no hostu grupas* (norādot "Discovered hosts") un arī darbību *Pievienot hostu grupai* (norādot citu hostu grupu), jo hostam ir jāpieder hostu grupai.

Droša automātiskā reģistrācija

Drošs automātiskās reģistrācijas veids ir iespējams, konfigurējot uz PSK balstītu autentifikāciju ar šifrētiem savienojumiem.

Šifrēšanas līmenis tiek konfigurēts globāli sadaļā *Administrēšana > Vispārīgi > Automātiskā reģistrācija*. Ir iespējams izvēlēties neizmantot šifrēšanu, TLS šifrēšanu ar PSK autentifikāciju vai abus variantus (lai daži hosti varētu reģistrēties bez šifrēšanas, bet citi — ar šifrēšanu).

Autentifikāciju ar PSK Zabbix serveris pārbauda pirms hosta pievienošanas. Ja pārbaude ir veiksmīga, hosts tiek pievienots un *Savienojumi no/uz hostu* tiek iestatīti tikai uz 'PSK', izmantojot identitāti/koplietoto atslēgu, kas ir tāda pati kā globālajā automātiskās reģistrācijas iestatījumā.

Attention:
Lai nodrošinātu automātiskās reģistrācijas drošību instalācijās, kurās tiek izmantoti starpniekserveri, ir jāiespējo šifrēšana starp Zabbix serveri un starpniekserveri.

DNS izmantošana kā noklusējuma saskarne

HostInterface un HostInterfaceItem konfigurācijas parametri ļauj norādīt pielāgotu vērtību hosta saskarnei automātiskās reģistrācijas laikā.

Konkrētāk, tie ir noderīgi, ja hosts būtu automātiski jāreģistrē ar DNS nosaukumu kā noklusējuma aģenta saskarni, nevis ar tā IP adresi. Šādā gadījumā DNS nosaukums jānorāda vai jāatgriež kā HostInterface vai HostInterfaceItem parametra vērtība. Ja kāda no šo parametru vērtībām mainās — piemēram, no IP adreses uz DNS nosaukumu vai otrādi — automātiski reģistrētā hosta noklusējuma saskarne tiks attiecīgi atjaunināta. Šis atjauninājums tiek piemērots esošajam hostam, nevis izveidojot jaunu. Lai nosūtītu jauno vērtību, aģents ir jāpārstartē, lai tas no jauna uzsāktu automātiskās reģistrācijas procesu.

Note:
Ja HostInterface vai HostInterfaceItem parametri nav konfigurēti, tā vietā tiek izmantots parametrs listen_dns. Šī vērtība tiek noteikta, veicot aģenta IP adreses reverso DNS uzmeklēšanu. Ja reversais DNS nav pareizi konfigurēts vai atgriež nederīgu nosaukumu, tas var izraisīt nepareizu vai neveiksmīgu automātisko reģistrāciju nederīgas saskarnes vērtības dēļ.

Hosta metadatu izmantošana

Kad aģents sūta automātiskās reģistrācijas pieprasījumu serverim, tas nosūta savu hosta nosaukumu. Dažos gadījumos (piemēram, Amazon mākoņa mezgliem) ar hosta nosaukumu nepietiek, lai Zabbix serveris varētu atšķirt atklātos hostus. Hosta metadatus var pēc izvēles izmantot, lai nosūtītu citu informāciju no aģenta uz serveri.

Hosta metadati tiek konfigurēti aģenta konfigurācijas failā — `zabbix_agentd.conf`. Konfigurācijas failā ir 2 veidi, kā norādīt hosta metadatus:

```
HostMetadata
HostMetadataItem
```

Skatiet šo opciju aprakstu iepriekš norādītajā saitē.

Parametrs HostMetadataItem var atgriezt līdz 65535 UTF-8 koda punktiem. Garāka vērtība tiks saīsināta.

Ņemiet vērā, ka MySQL gadījumā faktiskais maksimālais garums rakstzīmēs būs mazāks, ja atgrieztā vērtība satur vairāku baitu rakstzīmes. Piemēram, vērtība, kas satur tikai 3 baitu rakstzīmes, kopā būs ierobežota līdz 21844 rakstzīmēm, savukārt vērtība, kas satur tikai 4 baitu rakstzīmes, būs ierobežota līdz 16383 simboliem.

Attention:
Automātiskās reģistrācijas mēģinājums notiek katru reizi, kad aktīvais aģents nosūta serverim pieprasījumu atsvaidzināt aktīvās pārbaudes. Aizture starp pieprasījumiem ir norādīta aģenta parametrā RefreshActiveChecks. Pirmais pieprasījums tiek nosūtīts uzreiz pēc aģenta restartēšanas.

Piemēri

Autoreģistrācija pēc OS, izmantojot HostMetadata

Pieņemsim, ka vēlaties, lai hosti tiktu automātiski reģistrēti Zabbix serverī. Jūsu tīklā darbojas aktīvi Zabbix aģenti (skatiet iepriekš esošo sadaļu "Configuration"). Jūsu tīklā ir Windows hosti un Linux hosti, un Zabbix lietotāja saskarnē jums ir pieejamas veidnes "Linux by Zabbix agent" un "Windows by Zabbix agent". Tāpēc hosta reģistrācijas laikā vēlaties, lai reģistrējamajam hostam tiktu piemērota atbilstošā Linux/Windows veidne. Pēc noklusējuma automātiskās reģistrācijas laikā uz serveri tiek nosūtīts tikai hosta nosaukums, un ar to var nepietikt. Lai nodrošinātu, ka hostam tiek piemērota pareizā veidne, jums jāizmanto hosta metadati.

Lietotāja saskarnes konfigurācija

Pirmā lieta, kas jādara, ir konfigurēt lietotāja saskarni. Izveidojiet 2 darbības. Pirmā darbība:

- Nosaukums: Linux hosta automātiskā reģistrācija
- Nosacījumi: Hosta metadati satur *Linux*
- Darbības: Saistīt veidnes: Linux by Zabbix aģents

Note:
Šajā gadījumā varat izlaist darbību "Pievienot hostu". Lai saistītu veidni ar hostu, vispirms ir jāpievieno hosts, tāpēc serveris to izdarīs automātiski.

Otrā darbība:

- Nosaukums: Windows hosta automātiskā reģistrācija
- Nosacījumi: Hosta metadati satur *Windows*
- Darbības: Saistīt veidnes: Windows by Zabbix aģents

Aģenta konfigurācija

Tagad jums jākonfigurē aģenti. Pievienojiet nākamo rindu aģenta konfigurācijas failiem:

```
HostMetadataItem=system.uname
```

Tādējādi jūs nodrošināsiet, ka hosta metadati saturēs "Linux" vai "Windows" atkarībā no hosta, uz kura darbojas aģents. Hosta metadatu piemērs šajā gadījumā:

```
Linux: Linux server3 3.2.0-4-686-pae #1 SMP Debian 3.2.41-2 i686 GNU/Linux
Windows: Windows WIN-0PXGGSTYNHO 6.0.6001 Windows Server 2008 Service Pack 1 Intel IA-32
```

Neaizmirstiet restartēt aģentu pēc jebkādu izmaiņu veikšanas konfigurācijas failā.

HostMetadata izmantošana, lai kontrolētu automātisko reģistrāciju un veidnes

1. solis — Aizsargājiet reģistrāciju ar HostMetadata

Izmantojot hosta metadatus, var nodrošināt pamata aizsardzību pret nevēlamu hostu reģistrēšanos.

Lietotāja saskarnes konfigurācija

Izveidojiet darbību lietotāja saskarnē, izmantojot grūti uzminamu slepeno kodu, lai liegtu nevēlamiem hostiem reģistrēties:

- Nosaukums: Linux automātiskās reģistrācijas darbība
- Nosacījumi:
  - Aprēķina veids: AND
  - Nosacījums (A): Hosta metadati satur //Linux//
  - Nosacījums (B): Hosta metadati satur //21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae//
- Darbības:
  - Sūtīt ziņojumu lietotājiem: Admin, izmantojot visus datu nesējus
  - Pievienot hostu grupām: Linux serveri
  - Piesaistīt veidnes: Linux by Zabbix aģents

Lūdzu, ņemiet vērā, ka šī metode viena pati nenodrošina spēcīgu aizsardzību, jo dati tiek pārraidīti atklātā tekstā. Lai izmaiņas stātos spēkā nekavējoties, ir nepieciešama konfigurācijas kešatmiņas pārlāde.

Aģenta konfigurācija

Pievienojiet aģenta konfigurācijas failam šādu rindu:

```
HostMetadata=Linux    21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

kur "Linux" ir platforma, bet pārējā virknes daļa ir grūti uzminams slepenais teksts.

Neaizmirstiet restartēt aģentu pēc jebkādu izmaiņu veikšanas konfigurācijas failā.

2. solis — pievienojiet veidni reģistrētajam hostam

Jau reģistrētam hostam ir iespējams pievienot papildu veidnes. Šajā gadījumā veidne MySQL by Zabbix agent tiks piesaistīta tikai tiem hostiem, kuru HostMetadata satur marķieri `MySQL`.

Lietotāja saskarnes konfigurācija

Atjauniniet darbību lietotāja saskarnē:

- Nosaukums: Linux autoreģistrācijas darbība
- Nosacījumi:
  - Aprēķina veids: AND
  - Nosacījums (A): Hosta metadati satur Linux
  - Nosacījums (B): Hosta metadati satur 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
  - Nosacījums (C): Hosta metadati satur MySQL
- Operācijas:
  - Sūtīt ziņojumu lietotājiem: Admin, izmantojot visus datu nesējus
  - Pievienot hostu grupām: Linux serveri
  - Piesaistīt veidnes: Linux by Zabbix aģents
  - Piesaistīt veidnes: MySQL by Zabbix Agent

Aģenta konfigurācija

Atjauniniet nākamo rindu aģenta konfigurācijas failā:

```
HostMetadata=MySQL on Linux 21df83bf21bf0be663090bb8d4128558ab9b95fba66a6dbf834f8b91ae5e08ae
```

Neaizmirstiet restartēt aģentu pēc jebkādu izmaiņu veikšanas konfigurācijas failā.

3 Zema līmeņa atklāšana

Pārskats Zema līmeņa atklāšana (LLD) nodrošina iespēju automātiski izveidot vienumus, trigerus un grafikus dažādām entītijām hostā. Piemēram, Zabbix var automātiski sākt uzraudzīt failu sistēmas vai tīkla saskarnes jūsu datorā, bez nepieciešamības manuāli izveidot vienumus katrai failu sistēmai vai tīkla saskarnei. LLD var arī izveidot hostus, piemēram, lai aizpildītu virtuālās mašīnas, kas atklātas hipervizorā, kā arī ligzdotās atklāšanas kārtulas, nodrošinot daudzlīmeņu atklāšanu. Papildus tam ir iespējams konfigurēt Zabbix tā, lai tas automātiski noņemtu nevajadzīgās entītijas, pamatojoties uz faktiskiem periodiski veiktās atklāšanas rezultātiem.

Lietotājs var definēt savus atklāšanas tipus, ja vien tie ievēro noteiktu JSON protokolu.

Atklāšanas procesa vispārējā arhitektūra ir šāda.

Vispirms lietotājs izveido atklāšanas kārtulu sadaļā *Datu ievākšana* > *Veidnes*, kolonnā *Atklāšana*. Atklāšanas kārtula sastāv no (1) vienuma, kas atklāj nepieciešamās entītijas (piemēram, failu sistēmas vai tīkla saskarnes), un (2) vienumu, trigeru un grafiku prototipiem, kas jāizveido, pamatojoties uz šī vienuma vērtību.

Vienums, kas atklāj nepieciešamās entītijas, ir līdzīgs parastam vienumam, kas redzams citur: serveris pieprasa Zabbix aģentam (vai jebkuram citam tipam, kas iestatīts šim vienumam) šī vienuma vērtību, un aģents atbild ar teksta vērtību. Atšķirība ir tāda, ka vērtībai, ko atgriež aģents, jāietver atklāto entītiju saraksts JSON formātā. Lai gan šī formāta detaļas ir svarīgas tikai pielāgotu atklāšanas pārbaužu izstrādātājiem, ir svarīgi zināt, ka atgrieztā vērtība satur makro → vērtība pāru sarakstu. Piemēram, vienums "net.if.discovery" var atgriezt divus pārus: "{#IFNAME}" → "lo" un "{#IFNAME}" → "eth0".

Šie makro tiek izmantoti nosaukumos, atslēgās un citos prototipu laukos, kur tie pēc tam tiek aizstāti ar saņemtajām vērtībām, lai katrai atklātajai entītijai izveidotu reālus vienumus, trigerus, grafikus vai pat hostus. Skatiet pilnu opciju sarakstu LLD makro izmantošanai.

Kad serveris saņem atklāšanas vienuma vērtību, tas aplūko makro → vērtība pārus un katram pārim ģenerē reālus vienumus, trigerus un grafikus, pamatojoties uz to prototipiem. Iepriekš minētajā piemērā ar "net.if.discovery" serveris ģenerētu vienu vienumu, trigeru un grafiku kopu cilpas saskarnei "lo" un citu kopu saskarnei "eth0".

Ņemiet vērā, ka kopš **Zabbix 4.2** ir mainīts JSON formāts, ko atgriež zema līmeņa atklāšanas kārtulas. Vairs netiek sagaidīts, ka JSON saturēs objektu "data". Zema līmeņa atklāšana tagad pieņem parastu JSON, kas satur masīvu, lai atbalstītu jaunas iespējas, piemēram, vienuma vērtības priekšapstrādi un pielāgotus ceļus uz zema līmeņa atklāšanas makro vērtībām JSON dokumentā.

Iebūvētās atklāšanas atslēgas ir atjauninātas tā, lai tās atgrieztu LLD rindu masīvu JSON dokumenta saknē. Zabbix automātiski izgūs makro un vērtību, ja masīva lauks kā atslēgu izmanto sintaksi {#MACRO}. Jebkuras jaunas iebūvētās atklāšanas pārbaudes izmantos jauno sintaksi bez "data" elementiem. Apstrādājot zema līmeņa atklāšanas vērtību, vispirms tiek noteikta sakne (masīvs pie `$.` vai `$.data`).

Lai gan elements "data" ir noņemts no visiem ar atklāšanu saistītajiem iebūvētajiem vienumiem, atpakaļsaderības nolūkos Zabbix joprojām pieņems JSON pierakstu ar elementu "data", lai gan tā izmantošana nav ieteicama. Ja JSON satur objektu tikai ar vienu masīva elementu "data", tad tas automātiski izgūs šī elementa saturu, izmantojot JSONPath `$.data`. Zema līmeņa atklāšana tagad pieņem arī neobligātus lietotāja definētus LLD makro ar pielāgotu ceļu, kas norādīts JSONPath sintaksē.

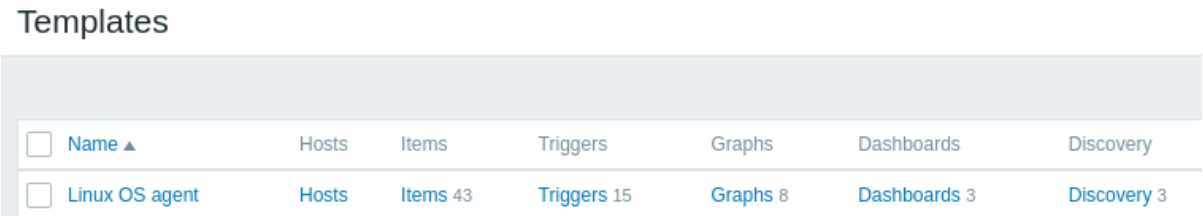
Warning:
Iepriekš minēto izmaiņu rezultātā jaunāki aģenti vairs nevarēs darboties ar vecāku Zabbix serveri.

Skatiet arī: Atklātās entītijas

Zema līmeņa atklāšanas konfigurēšana Mēs ilustrēsim zema līmeņa atklāšanu, izmantojot failu sistēmas atklāšanas piemēru.

Lai konfigurētu atklāšanu, rīkojieties šādi:

- Dodieties uz: *Datu ievākšana* > *Veidnes* vai *Hosts*.
- Noklikšķiniet uz *Atklāšana* atbilstošās veidnes/hosta rindā.



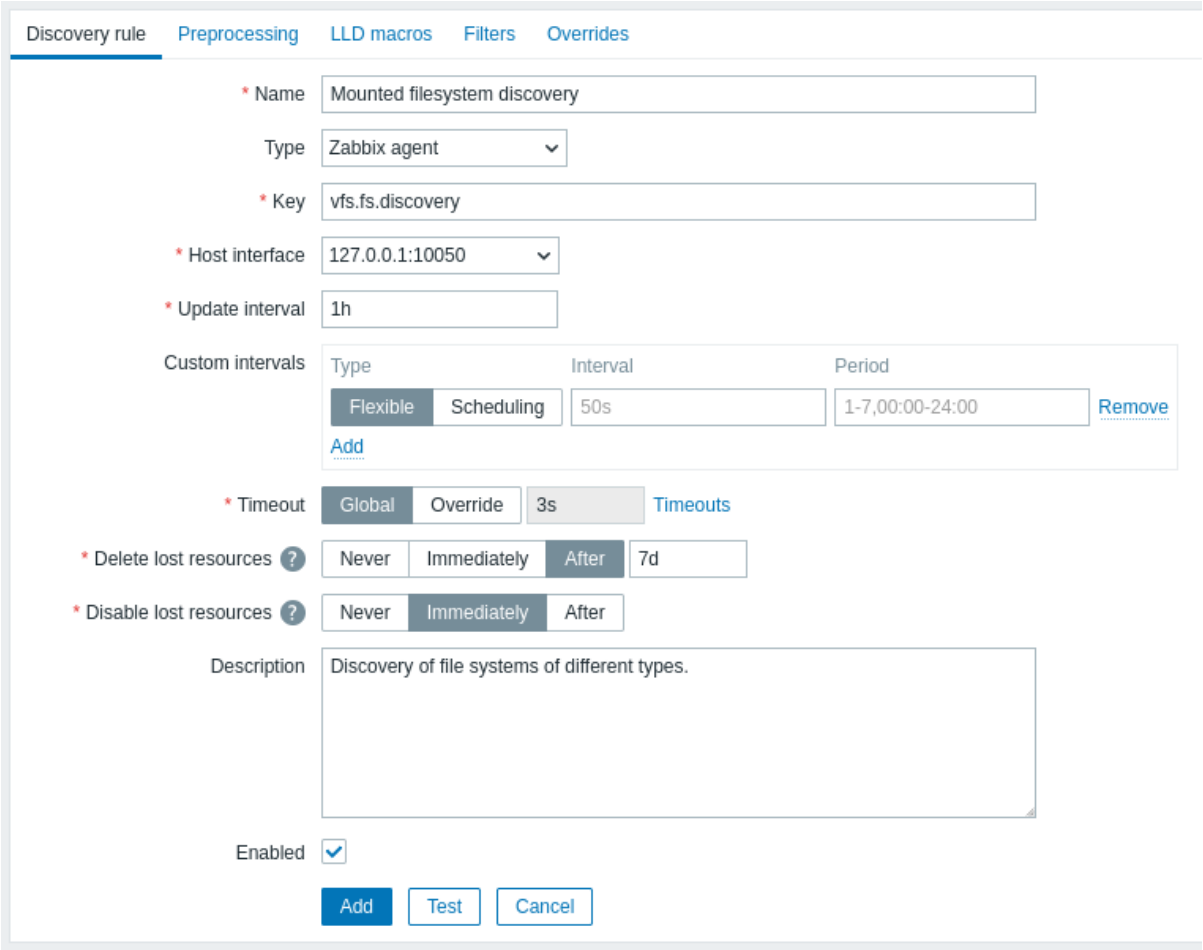
- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot atklāšanas kārtulu*.
- Aizpildiet atklāšanas kārtulas veidlapu ar nepieciešamo informāciju.

Atklāšanas kārtula

Atklāšanas kārtulas forma satur piecas cilnes, kas no kreisās uz labo pusi attēlo datu plūsmu atklāšanas laikā:

- *Atklāšanas kārtula* — nosaka, galvenokārt, iebūvēto vienumu vai pielāgotu skriptu atklāšanas datu iegūšanai.
- *Priekšapstrāde* — piemēro noteiktu priekšapstrādi atklātajiem datiem.
- *LLD makrosi* — ļauj izvilkt noteiktas makrosu vērtības izmantošanai atklātajos vienumos, trigeros utt.
- *Filtri* — ļauj filtrēt atklātās vērtības.
- *Aizstāšanas* — ļauj modificēt vienumus, trigerus, grafikus vai hostu prototipus, piemērojot tos konkrētiem atklātajiem objektiem.

Cilne **Atklāšanas kārtula** satur atklāšanai izmantojamo vienuma atslēgu (kā arī dažus vispārīgus atklāšanas kārtulas atribūtus):



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parameter	Description
Name	Atklāšanas kārtulas nosaukums.
Type	Pārbaudes tips, ko izmantot atklāšanai. Šajā piemērā mēs izmantojam *Zabbix aģents* vienuma tipu. Atklāšanas kārtula var būt arī atkarīgais vienums, kas ir atkarīgs no parasta vienuma. Tā nevar būt atkarīga no citas atklāšanas kārtulas. Atkarīgajam vienumam atlasiet attiecīgo tipu (*Atkarīgais vienums*) un laukā 'Master item' norādiet galveno vienumu. Galvenajam vienumam ir jāeksistē.

Parameter	Description
Key	Ievadiet atklāšanas vienuma atslēgu (līdz 2048 rakstzīmēm). Piemēram, varat izmantot iebūvēto vienuma atslēgu ”vfs.fs.discovery”, lai atgrieztu JSON virkni ar datorā esošo failu sistēmu sarakstu, to tipiem un montēšanas opcijām. Ņemiet vērā, ka cita failu sistēmu atklāšanas iespēja ir izmantot atklāšanas rezultātus ar aģenta atslēgu ”vfs.fs.get” (skatiet piemēru).
Update interval	Šis lauks nosaka, cik bieži Zabbix veic atklāšanu. Sākumā, kad tikai iestatāt failu sistēmu atklāšanu, iespējams, vēlēsieties iestatīt nelielu intervālu, taču, kad būsiet pārliecinājies, ka tā darbojas, varat to iestatīt uz 30 minūtēm vai vairāk, jo failu sistēmas parasti nemainās ļoti bieži. Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Piezīme*: atjaunināšanas intervālu var iestatīt uz ’0’ tikai tad, ja pastāv pielāgoti intervāli ar vērtību, kas nav nulle. Ja iestatīta vērtība ’0’ un pastāv pielāgots intervāls (elastīgs vai plānots) ar vērtību, kas nav nulle, vienums tiks aptaujāts pielāgotā intervāla darbības laikā. Jaunas atklāšanas kārtulas tiks pārbaudītas 60 sekunžu laikā pēc to izveides, ja vien tām nav iestatīts Scheduling vai Flexible update interval un *Update interval* nav iestatīts uz 0. *Ņemiet vērā*, ka esošai atklāšanas kārtulai atklāšanu var veikt nekavējoties, nospiežot pogu *Execute now*.
Custom intervals	Varat izveidot pielāgotas kārtulas vienuma pārbaudei: **Flexible** — izveidot izņēmumu *Update interval* vērtībai (intervāls ar citu biežumu) **Scheduling** — izveidot pielāgotu aptaujas grafiku. Detalizētu informāciju skatiet sadaļā Pielāgotie intervāli.
Timeout	Iestatiet atklāšanas pārbaudes noildzi. Atlasiet noildzes opciju: **Global** — tiek izmantota starpniekservera/globālā noildze (parādīta pelēkajā laukā *Timeout*); **Override** — tiek izmantota pielāgota noildze (iestatīta laukā *Timeout*; atļautais diapazons: 1 - 600s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makrosi. Noklikšķinot uz saites *Timeouts*, varat konfigurēt starpniekservera noildzes vai globālās noildzes (ja starpniekserveris netiek izmantots). Ņemiet vērā, ka saite *Timeouts* ir redzama tikai lietotājiem ar tipu *Super admin* un atļaujām uz lietotāja saskarne sadaļām *Administration* > *General* vai *Administration* > *Proxies*.
Delete lost resources	Norādiet, cik drīz atklātā entītija tiks dzēsta, tiklīdz tās atklāšanas statuss kļūs par ”Not discovered anymore”: *Never* — tā netiks dzēsta; *Immediately* — tā tiks dzēsta nekavējoties; *After* — tā tiks dzēsta pēc norādītā laika perioda. Vērtībai jābūt lielākai par *Disable lost resources* vērtību. Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makrosi. *Piezīme*: nav ieteicams izmantot ”Immediately”, jo pat nepareiza filtra rediģēšana var novest pie tā, ka entītija tiek dzēsta kopā ar visiem vēsturiskajiem datiem. Ņemiet vērā, ka manuāli atspējoti resursi netiks dzēsti ar zema līmeņa atklāšanu.
Disable lost resources	Norādiet, cik drīz atklātā entītija tiks atspējota, tiklīdz tās atklāšanas statuss kļūs par ”Not discovered anymore”: *Never* — tā netiks atspējota; *Immediately* — tā tiks atspējota nekavējoties; *After* — tā tiks atspējota pēc norādītā laika perioda. Vērtībai jābūt lielākai par atklāšanas kārtulas atjaunināšanas intervālu. Ņemiet vērā, ka automātiski atspējoti resursi tiks atkal iespējoti, ja zema līmeņa atklāšana tos atklās no jauna. Manuāli atspējoti resursi netiks atkal iespējoti, ja tie tiks atklāti no jauna. Šis lauks netiek parādīts, ja *Delete lost resources* ir iestatīts uz ”Immediately”. Tiek atbalstīti laika sufiksi, piemēram, 2h, 1d. Tiek atbalstīti lietotāja makrosi.
Description	Ievadiet aprakstu.
Enabled	Ja atzīmēts, kārtula tiks apstrādāta.

Note:
Atklāšanas kārtulas vēsture netiek saglabāta.

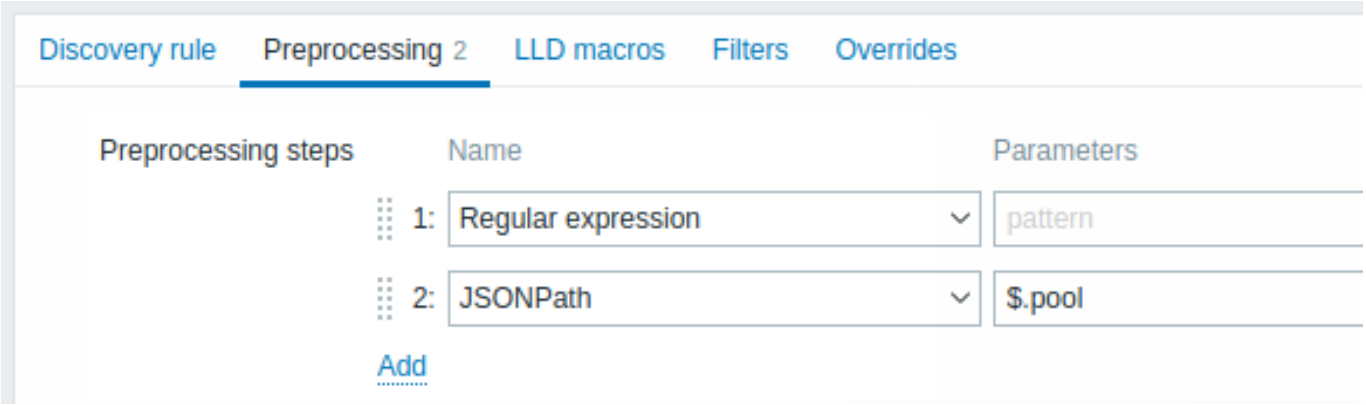
Priekšapstrāde

Cilne **Priekšapstrāde** ļauj definēt pārveidošanas noteikumus, ko piemērot atklāšanas rezultātam. Šajā solī ir iespējama viena vai vairākas pārveidošanas. Pārveidošanas tiek izpildītas tādā secībā, kādā tās ir definētas. Visa priekšapstrāde tiek veikta Zabbix

serverī.

Skatiet arī:

- Priekšapstrādes informācija
- Priekšapstrādes testēšana



Type		
	Pārveidošana	Apraksts
Text		
	Regulārā izteiksme	Salīdzināt saņemto vērtību ar <pattern> regulāro izteiksmi un aizstāt vērtību ar izvilkto <output>. Regulārā izteiksme atbalsta ne vairāk kā 10 uztvertu grupu izvilkšanu, izmantojot \N secību. Parametri: **pattern** - regulārā izteiksme **output** - izvades formatēšanas veidne. \N (kur N=1...9) atsoļa secība tiek aizstāta ar N-to atrasto grupu. \0 atsoļa secība tiek aizstāta ar atrasto tekstu. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
	Aizstāt	Atrast meklējamo virkni un aizstāt to ar citu (vai neko). Tiks aizstātas visas meklējamās virknes parādīšanās reizes. Parametri: **search string** - virkne, ko atrast un aizstāt, reģistrjutīga (obligāti) **replacement** - virkne, ar ko aizstāt meklējamo virkni. Aizstāšanas virkne var būt arī tukša, tādējādi ļaujot dzēst meklējamo virkni, kad tā ir atrasta. Ir iespējams izmantot atsoļa secības, lai meklētu vai aizstātu rindas pārtraukumus, kursora atgriešanu, tabulācijas un atstarpes ”\n \r \t \s”; slīpsvītru var atsoļot kā ”\\” un atsoļa secības var atsoļot kā ”\\n”. Rindas pārtraukumu, kursora atgriešanas un tabulāciju atsoļošana zema līmeņa atklāšanas laikā tiek veikta automātiski.
Structured data		
	JSONPath	Izvilkt vērtību vai fragmentu no JSON datiem, izmantojot JSONPath funkcionalitāti. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

	XML XPath	Izvilkt vērtību vai fragmentu no XML datiem, izmantojot XPath funkcionalitāti.
		Lai šī opcija darbotos, Zabbix serverim jābūt kompilētam ar libxml atbalstu.
		Piemēri:
		`number(/document/item/value)` izvilks 10 no `<document><item><value>10</value></item></document>`
		`number(/document/item/@attribute)` izvilks 10 no `<document><item attribute="10"></item></document>`
		`/document/item` izvilks `<item><value>10</value></item>` no `<document><item><value>10</value></item></document>`
		Ņemiet vērā, ka nosaukumvietas netiek atbalstītas.
		Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.

	CSV uz JSON	Pārveidot CSV faila datus JSON formātā. Papildinformāciju skatiet: CSV uz JSON priekšapstrāde.

Type	
XML uz JSON	Pārveidot XML formāta datus uz JSON. Papildinformāciju skatiet: Serializācijas noteikumi. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
SNMP	
SNMP walk vērtība	Izvilkt vērtību pēc norādītā OID/MIB nosaukuma un piemērot formatēšanas opcijas: **Unchanged** - atgriezt Hex-STRING kā neatsoļotu heksadecimālu virkni (*ņemiet vērā*, ka attēlošanas norādes joprojām tiek piemērotas); **UTF-8 from Hex-STRING** - pārveidot Hex-STRING par UTF-8 virkni; **MAC from Hex-STRING** - pārveidot Hex-STRING par MAC adreses virkni (kurā `' '` tiks aizstāts ar `':'`); **Integer from BITS** - pārveidot pirmos 8 baitus no bitu virknes, kas izteikta kā heksadecimālu rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu bezparaksta veselu skaitli. Bitu virknēs, kas ir garākas par 8 baitiem, nākamie baiti tiks ignorēti. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
SNMP walk uz JSON	Pārveidot SNMP vērtības uz JSON. Norādiet lauka nosaukumu JSON un atbilstošo SNMP OID ceļu. Lauku vērtības tiks aizpildītas ar vērtībām no norādītā SNMP OID ceļa. Šo priekšapstrādes soli var izmantot SNMP OID atklāšanai. Ir pieejamas līdzīgas vērtību formatēšanas opcijas kā solī *SNMP walk vērtība*. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
SNMP get vērtība	Piemērot formatēšanas opcijas SNMP get vērtībai: **UTF-8 from Hex-STRING** - pārveidot Hex-STRING par UTF-8 virkni; **MAC from Hex-STRING** - pārveidot Hex-STRING par MAC adreses virkni (kurā `' '` tiks aizstāts ar `':'`); **Integer from BITS** - pārveidot pirmos 8 baitus no bitu virknes, kas izteikta kā heksadecimālu rakstzīmju secība (piemēram, ”1A 2B 3C 4D”), par 64 bitu bezparaksta veselu skaitli. Bitu virknēs, kas ir garākas par 8 baitiem, nākamie baiti tiks ignorēti. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Custom scripts	
JavaScript	Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam. Ņemiet vērā, ka pieejamais JavaScript garums ir atkarīgs no izmantotās datubāzes. Papildinformāciju skatiet: Javascript priekšapstrāde
Validation	
Neatbilst regulārajai izteiksmei	Norādiet regulāro izteiksmi, kurai vērtība nedrīkst atbilst. Piemēram, `Error:(.*?)\.` Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Pārbaudīt kļūdu JSON	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas JSONPath. Apturēt apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpināt apstrādi ar vērtību, kāda tā bija pirms šī priekšapstrādes soļa. Ņemiet vērā, ka šīs ārējo servisu kļūdas lietotājam tiek ziņotas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes soli. Piemēram, `$.errors`. Ja tiek saņemts JSON, piemēram, `{"errors":"e1"}`, nākamais priekšapstrādes solis netiks izpildīts. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Pārbaudīt kļūdu XML	Pārbaudīt lietojumprogrammas līmeņa kļūdas ziņojumu, kas atrodas XPath. Apturēt apstrādi, ja pārbaude ir veiksmīga un ziņojums nav tukšs; pretējā gadījumā turpināt apstrādi ar vērtību, kāda tā bija pirms šī priekšapstrādes soļa. Ņemiet vērā, ka šīs ārējo servisu kļūdas lietotājam tiek ziņotas tādas, kādas tās ir, nepievienojot informāciju par priekšapstrādes soli. Nederīga XML parsēšanas kļūmes gadījumā kļūda netiks ziņota. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Atbilst regulārajai izteiksmei	Norādiet regulāro izteiksmi, kurai vērtībai jāatbilst. Ja atzīmējat izvēles rūtiņu *Custom on fail*, ir iespējams norādīt pielāgotas kļūdu apstrādes opcijas: vai nu atmest vērtību, iestatīt norādītu vērtību vai iestatīt norādītu kļūdas ziņojumu.
Throttling	

Type	
Atmest nemainīgu ar sirdspukstu	Atmest vērtību, ja tā nav mainījusies definētajā laika periodā (sekundēs). Sekunžu norādīšanai tiek atbalstītas pozitīvas veselas skaitļu vērtības (minimums - 1 sekunde). Šajā laukā var izmantot laika sufiksus (piemēram, 30s, 1m, 2h, 1d). Šajā laukā var izmantot lietotāja makrosus un zema līmeņa atklāšanas makrosus. Atklāšanas vienumam var norādīt tikai vienu ierobežošanas opciju. Piemēram, `1m`. Ja identisks teksts tiek nodots šim noteikumam divreiz 60 sekunžu laikā, tas tiks atmests. *Piezīme*: vienumu prototipu maiņa neatiestata ierobežošanu. Ierobežošana tiek atiestatīta tikai tad, kad tiek mainīti priekšapstrādes soļi.
Prometheus	
Prometheus uz JSON	Pārveidot nepieciešamās Prometheus metrikas uz JSON. Sīkāku informāciju skatiet Prometheus pārbaudes.

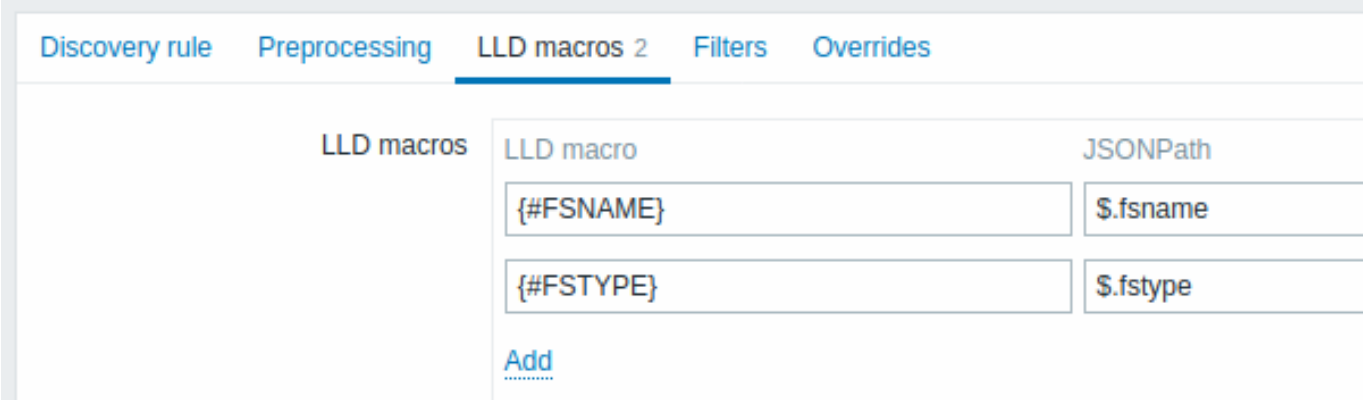
Ņemiet vērā, ka, ja atklāšanas kārtula hostam ir piemērota, izmantojot veidni, tad šīs cilnes saturs ir tikai lasāms.

Pielāgotas makrokomandas

Cilne **LLD makrokomandas** ļauj norādīt pielāgotas zema līmeņa atklāšanas makrokomandas.

Pielāgotas makrokomandas ir noderīgas gadījumos, kad atgrieztajā JSON jau nav definētas nepieciešamās makrokomandas. Tātad, piemēram:

- Vietējā `vfs.fs.discovery` atslēga failu sistēmu atklāšanai atgriež JSON ar dažām iepriekš definētām LLD makrokomandām, piemēram, `{#FSNAME}`, `{#FSTYPE}`. Šīs makrokomandas var izmantot tieši vienumu un trigeru prototipos (skatiet turpmākās šīs lapas sadaļas); pielāgotu makrokomandu definēšana nav nepieciešama;
- Arī aģenta vienums `vfs.fs.get` atgriež JSON ar failu sistēmu datiem, taču bez iepriekš definētām LLD makrokomandām. Šajā gadījumā makrokomandas varat definēt pats un sasaistīt tās ar vērtībām JSON, izmantojot JSONPath:

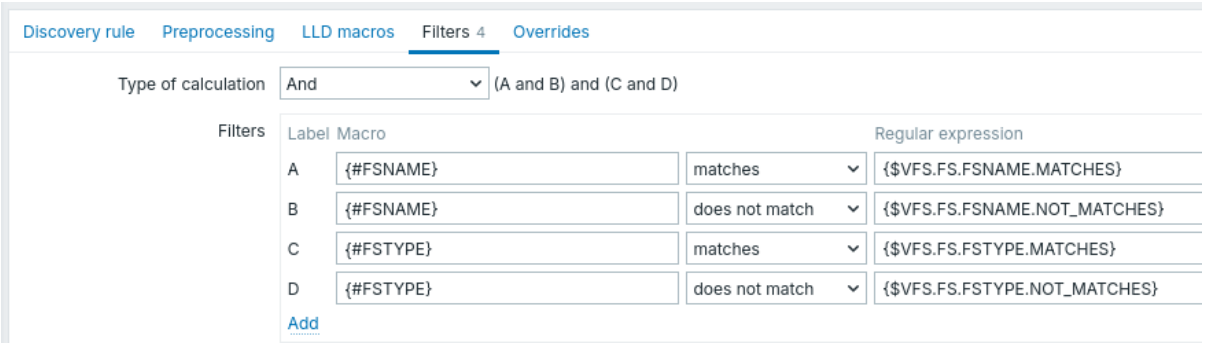


Izgūtās vērtības var izmantot atklātajos vienumos, trigeros utt. Ņemiet vērā, ka vērtības tiks izgūtas no atklāšanas rezultāta un visām līdz šim veiktajām priekšapstrādes darbībām.

Parameter	Description
LLD macro	Zema līmeņa atklāšanas makrokomandas nosaukums, izmantojot šādu sintaksi: `{#MACRO}`.
JSONPath	Ceļš, kas tiek izmantots LLD makrokomandas vērtības izgūšanai no LLD rindas, izmantojot JSONPath sintaksi. Vērtības, kas izgūtas no atgrieztā JSON, tiek izmantotas, lai aizstātu LLD makrokomandas vienumu, trigeru utt. prototipu laukos. JSONPath var norādīt, izmantojot punkta notāciju vai iekavu notāciju. Iekavu notācija jāizmanto īpašo rakstzīmju un unikoda gadījumā, piemēram, `$['unicode + special chars #1']['unicode + special chars #2']`. Piemēram, `$.foo` izgūs "bar" un "baz" no šī JSON: `[{"foo":"bar"}, {"foo":"baz"}]` Ņemiet vērā, ka `$.foo` izgūs "bar" un "baz" arī no šī JSON: `{"data":[{"foo":"bar"}, {"foo":"baz"}]}`, jo viens "data" objekts tiek apstrādāts automātiski (lai nodrošinātu atpakaļsaderību ar zema līmeņa atklāšanas ieviešanu Zabbix versijās pirms 4.2).

Filtrs

Filtru var izmantot, lai ģenerētu reālus vienumus, trigerus un grafikus tikai tām entītijām, kas atbilst kritērijiem. Cilnē **Filtri** ir ietvertas atklāšanas kārtulas filtru definīcijas, kas ļauj filtrēt atklāšanas vērtības:



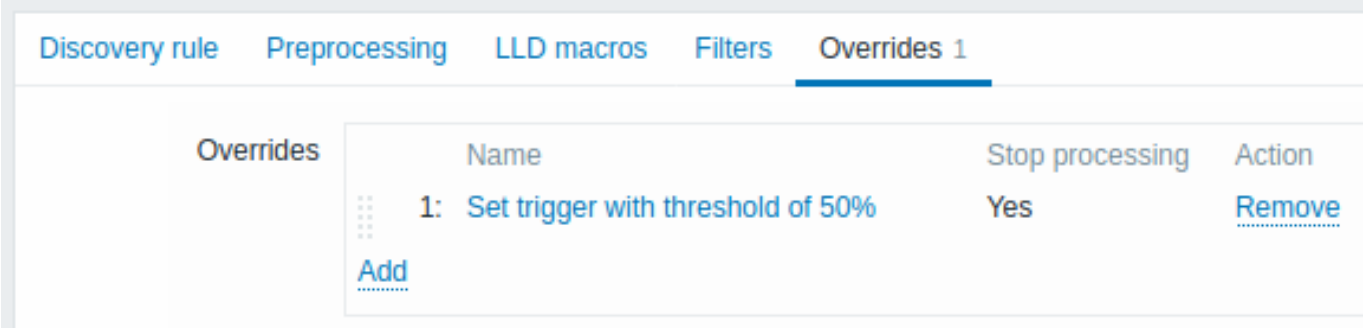
Parametrs	Apraksts
Aprēķina veids	Ir pieejamas šādas filtru aprēķināšanas opcijas: **And** - jāizpildās visiem filtriem; **Or** - pietiek, ja izpildās viens filtrs; **And/Or** - izmanto *And* ar dažādiem makrosu nosaukumiem un *Or* ar vienādiem makrosu nosaukumiem; **Custom expression** - piedāvā iespēju definēt pielāgotu filtru aprēķinu. Formulai jāietver visi sarakstā esošie filtri. Ierobežots līdz 255 simboliem.
Filtri	Ir pieejami šādi filtra nosacījumu operatori: *matches*, *does not match*, *exists*, *does not exist*. Operatoriem *matches* un *does not match* ir nepieciešama Perl Compatible Regular Expression (PCRE). Piemēram, ja jūs interesē tikai C:, D: un E: failu sistēmas, laukā "Macro" varat ievadīt {#FSNAME}, bet laukā "Regular expression" - regulāro izteiksmi "^C|^D|^E". Filtrēšana ir iespējama arī pēc failu sistēmu tipiem, izmantojot makrosu {#FSTYPE} (piemēram, "^ext|^reiserfs"), un pēc disku tipiem (atbalstīts tikai Windows aģents), izmantojot makrosu {#FSDRIVETYPE} (piemēram, "fixed"). Laukā "Regular expression" varat ievadīt regulāro izteiksmi vai atsauci uz globālu regular expression. Lai pārbaudītu regulāro izteiksmi, varat izmantot "grep -E", piemēram: `for f in ext2 nfs reiserfs smbfs; do echo $f | grep -E '^ext|^reiserfs' || echo "SKIP: $f"; done` Operatori *exists* un *does not exist* ļauj filtrēt entītijas, pamatojoties uz norādītā LLD makrosa esamību vai neesamību atbildē. Ņemiet vērā, ka, ja atbildē trūkst makrosa no filtra, atrastā entītija tiks ignorēta, ja vien šim makrosam nav norādīts nosacījums "does not exist". Tiks parādīts brīdinājums, ja makrosa neesamība ietekmē izteiksmes rezultātu. Piemēram, ja {#B} trūkst šādā izteiksmē: {#A} matches 1 and {#B} matches 2 - tiks parādīts brīdinājums {#A} matches 1 or {#B} matches 2 - brīdinājuma nebūs

Warning:
Kļūda vai drukas kļūda regulārajā izteiksmē, kas izmantota LLD kārtulā (piemēram, nekorekta regulārā izteiksme "File systems for discovery"), var izraisīt tūkstošiem konfigurācijas elementu, vēsturisko vērtību un notikumu dzēšanu daudziem hostiem.

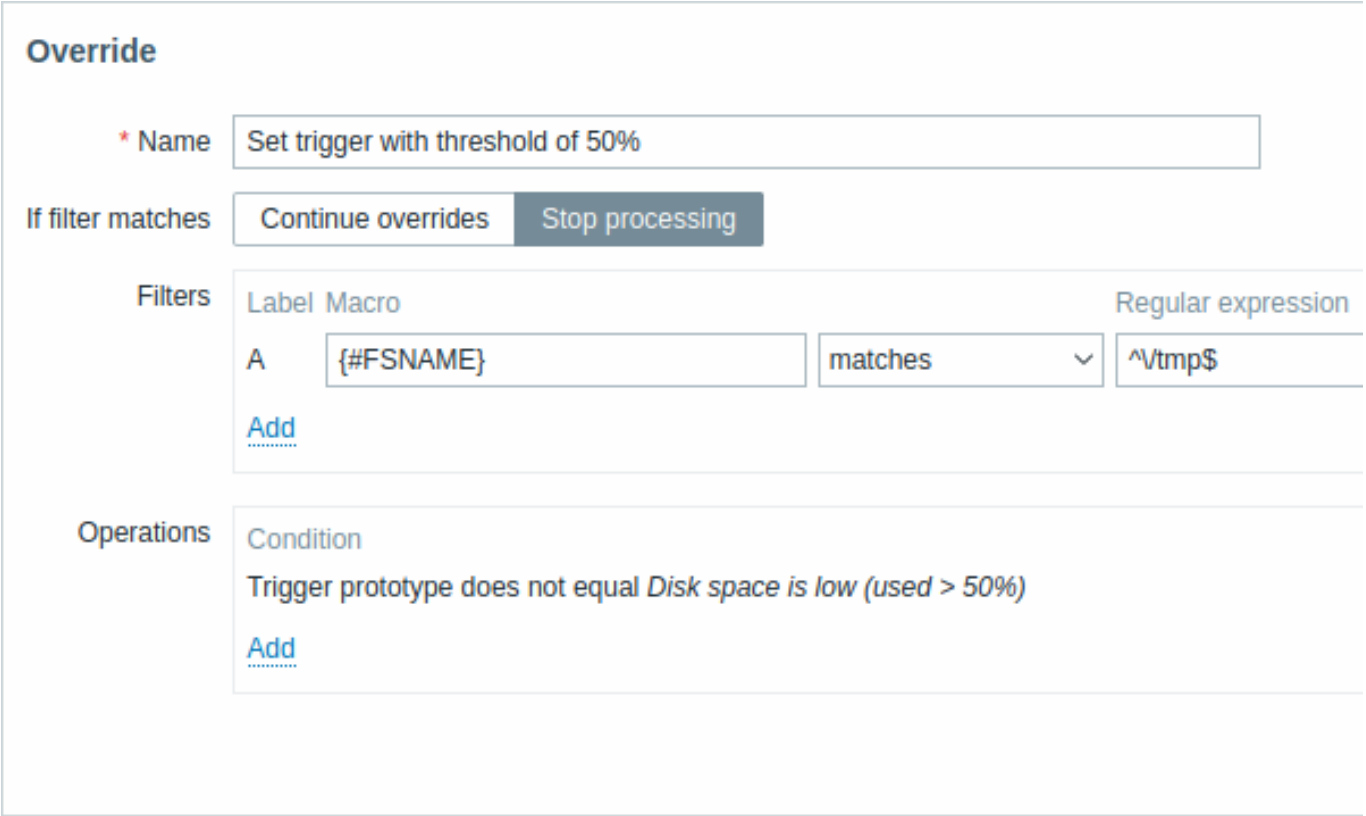
Attention:
Zabbix datubāze MySQL vidē jāizveido ar reģistrjutīgu salīdzināšanu, lai failu sistēmu nosaukumi, kas atšķiras tikai ar burtu reģistru, tiktu atklāti pareizi.

Pārrakstīšana

Cilne **Pārrakstīšana** ļauj iestatīt noteikumus, lai mainītu vienumu, trigeru, grafiku, hostu un atklāšanas prototipu sarakstu vai to atribūtus atklātajiem objektiem, kas atbilst norādītajiem kritērijiem.



Pārrakstīšanas (ja tādas ir) tiek parādītas pārkārtojamā vilkšanas un nomešanas sarakstā un izpildītas tādā secībā, kādā tās ir definētas. Lai konfigurētu jaunas pārrakstīšanas informāciju, noklikšķiniet uz Add blokā *Overrides*. Lai rediģētu esošu pārrakstīšanu, noklikšķiniet uz pārrakstīšanas nosaukuma. Tiks atvērts uznirstošais logs, kurā var rediģēt pārrakstīšanas noteikuma informāciju.

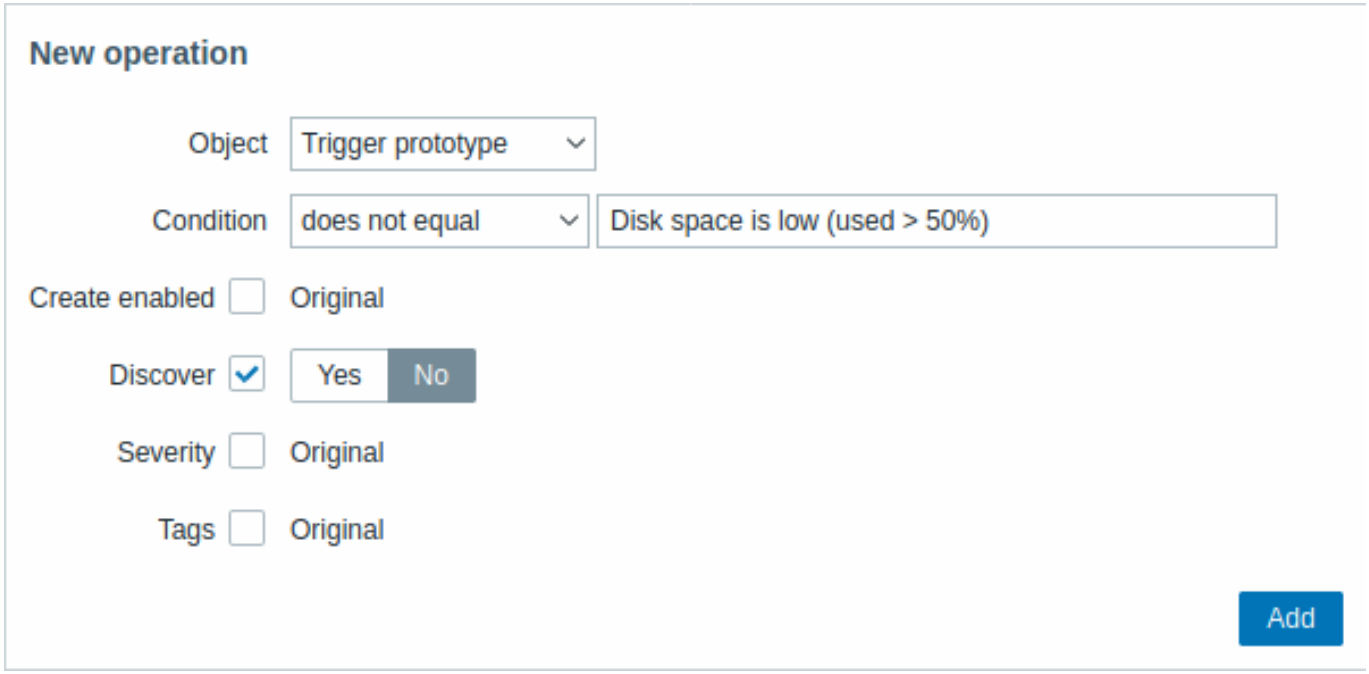


Visi obligātie parametri ir atzīmēti ar sarkanām zvaigznītēm.

Parameter	Description
Name	Unikāls (katram LLD noteikumam) pārrakstīšanas nosaukums.
If filter matches	Nosaka, vai nākamās pārrakstīšanas ir jāapstrādā, kad filtra nosacījumi ir izpildīti: **Continue overrides** - tiks apstrādātas nākamās pārrakstīšanas. **Stop processing** - tiks izpildītas iepriekšējo (ja tādas ir) un šīs pārrakstīšanas darbības, bet nākamās pārrakstīšanas atbilstošajām LLD rindām tiks ignorētas.
Filters	Nosaka, kurām atklātajām entītijām pārrakstīšana ir jāpiemēro. Pārrakstīšanas filtri tiek apstrādāti pēc atklāšanas noteikuma filtriem, un tiem ir tāda pati funkcionalitāte.
Operations	Pārrakstīšanas darbības tiek parādītas ar šādu informāciju: **Condition** - objekta tips un nosacījums, kam jābūt izpildītam objekta nosaukumam; piemēram: Trigera prototips nav vienāds ar *Disk space is low (used > 50%)*. **Actions** - tiek parādītas saites darbības rediģēšanai un noņemšanai.

Darbības konfigurēšana

Lai konfigurētu jaunas darbības informāciju, noklikšķiniet uz Add blokā Operations. Lai rediģētu esošu darbību, noklikšķiniet uz Edit blakus darbībai. Tiks atvērts uznirstošais logs, kurā var rediģēt darbības informāciju.



Parameter	Description
Object	Ir pieejami pieci objektu tipi: Vienuma prototips Trigera prototips Grafika prototips Hosta prototips Atklāšanas prototips
Condition	Ļauj filtrēt entītijas, kurām darbība ir jāpiemēro.
Operator	Atbalstītie operatori: **equals** - piemērot šim prototipam **does not equal** - piemērot visiem prototipiem, izņemot šo **contains** - piemērot, ja prototipa nosaukums satur šo virkni **does not contain** - piemērot, ja prototipa nosaukums nesatur šo virkni **matches** - piemērot, ja prototipa nosaukums atbilst regulārajai izteiksmei **does not match** - piemērot, ja prototipa nosaukums neatbilst regulārajai izteiksmei
Pattern	Regulārā izteiksme vai meklējamā virkne.
Object: *Item prototype*	
Create enabled	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos vienuma prototipa iestatījumus: *Yes* - vienums tiks pievienots iespējotā stāvoklī. *No* - vienums tiks pievienots atklātajai entītijai, bet atspējotā stāvoklī.
Discover	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos vienuma prototipa iestatījumus: *Yes* - vienums tiks pievienots. *No* - vienums netiks pievienots.
Update interval	Kad izvēles rūtiņa ir atzīmēta, parādīsies divas opcijas, kas ļaus iestatīt citu intervālu vienumam: *Delay* - vienuma atjaunināšanas intervāls. Tiek atbalstīti lietotāja makrosi un laika sufiksi (piemēram, 30s, 1m, 2h, 1d). Jāiestata uz 0, ja tiek izmantots *Custom interval*. *Custom interval* - noklikšķiniet uz Add, lai norādītu elastīgus/plānotus intervālus. Detalizētu informāciju skatiet sadaļā Pielāgoti intervāli.

Parameter		Description
	History	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus vienumam iestatīt citu vēstures glabāšanas periodu: *Do not store* - ja atlasīts, vēsture netiks glabāta. *Store up to* - ja atlasīts, labajā pusē parādīsies ievades lauks glabāšanas perioda norādīšanai. Tiek atbalstīti lietotāja makrosi un LLD makrosi.
	Trends	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus vienumam iestatīt citu tendenču glabāšanas periodu: *Do not store* - ja atlasīts, tendences netiks glabātas. *Store up to* - ja atlasīts, labajā pusē parādīsies ievades lauks glabāšanas perioda norādīšanai. Tiek atbalstīti lietotāja makrosi un LLD makrosi.
	Tags	Kad izvēles rūtiņa ir atzīmēta, parādīsies jauns bloks, kas ļaus norādīt birkas-vērtības pārus. Šīs birkas tiks pievienotas vienuma prototipā norādītajām birkām, pat ja birku nosaukumi sakrīt.
Object: *Trigger prototype*		
	Create enabled	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos trigera prototipa iestatījumus: *Yes* - trigeris tiks pievienots iespējotā stāvoklī. *No* - trigeris tiks pievienots atklātajai entītijai, bet atspējotā stāvoklī.
	Discover	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos trigera prototipa iestatījumus: *Yes* - trigeris tiks pievienots. *No* - trigeris netiks pievienots.
	Severity	Kad izvēles rūtiņa ir atzīmēta, parādīsies trigera būtiskuma pogas, kas ļaus mainīt trigera būtiskumu.
	Tags	Kad izvēles rūtiņa ir atzīmēta, parādīsies jauns bloks, kas ļaus norādīt birkas-vērtības pārus. Šīs birkas tiks pievienotas trigera prototipā norādītajām birkām, pat ja birku nosaukumi sakrīt.
Object: *Graph prototype*		
	Discover	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos grafika prototipa iestatījumus: *Yes* - grafiks tiks pievienots. *No* - grafiks netiks pievienots.
Object: *Host prototype*		
	Create enabled	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos hosta prototipa iestatījumus: *Yes* - hosts tiks izveidots iespējotā stāvoklī. *No* - hosts tiks izveidots atspējotā stāvoklī.
	Discover	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus pārrakstīt sākotnējos hosta prototipa iestatījumus: *Yes* - hosts tiks atklāts. *No* - hosts netiks atklāts.
	Link templates	Kad izvēles rūtiņa ir atzīmēta, parādīsies ievades lauks veidņu norādīšanai. Sāciet rakstīt veidnes nosaukumu vai noklikšķiniet uz *Select* blakus laukam un atlasiet veidnes no saraksta uznirstošajā logā. Veidnes no šīs pārrakstīšanas tiek pievienotas visām veidnēm, kas jau ir piesaistītas hosta prototipam.

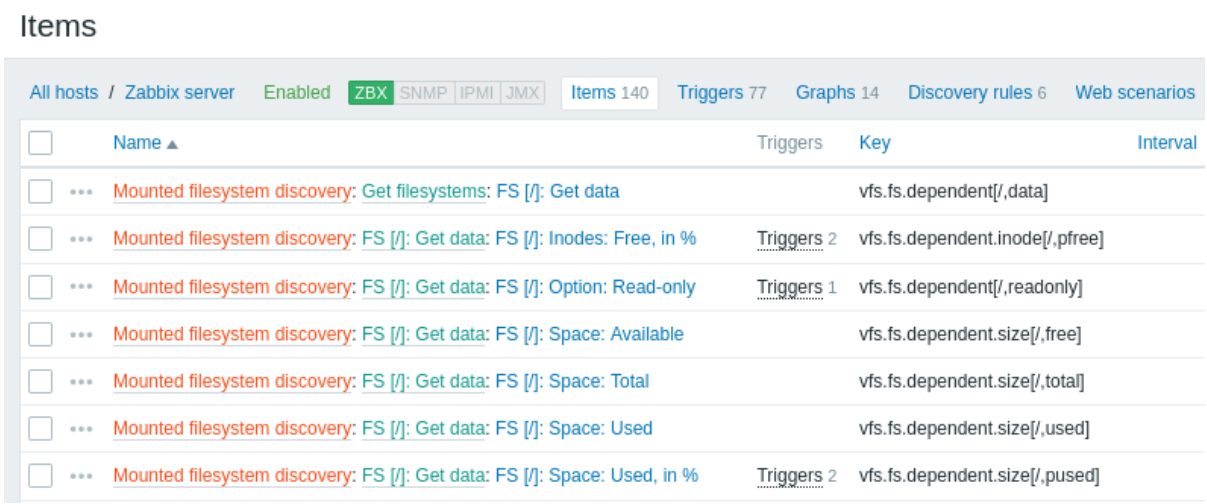
Parameter	Description
Tags	Kad izvēles rūtiņa ir atzīmēta, parādīsies jauns bloks, kas ļaus norādīt birkas-vērtības pārus. Šīs birkas tiks pievienotas hosta prototipā norādītajām birkām, pat ja birku nosaukumi sakrīt.
Host inventory	Kad izvēles rūtiņa ir atzīmēta, parādīsies pogas, kas ļaus atlasīt citu hosta prototipa inventāra režīmu: *Disabled* - neaizpildīt hosta inventāru *Manual* - norādīt informāciju manuāli *Automated* - automātiski aizpildīt hosta inventāra datus, pamatojoties uz savāktajiem mērījumiem.

Veidlapas pogas

Pogas veidlapas apakšdaļā ļauj veikt vairākas darbības.

Add	Pievienot atklāšanas kārtulu. Šī poga ir pieejama tikai jaunām atklāšanas kārtulām.
Update	Atjaunināt atklāšanas kārtulas rekvizītus. Šī poga ir pieejama tikai esošām atklāšanas kārtulām.
Clone	Izveidot citu atklāšanas kārtulu, pamatojoties uz pašreizējās atklāšanas kārtulas rekvizītiem.
Execute now	Nekavējoties veikt atklāšanu, pamatojoties uz atklāšanas kārtulu. Atklāšanas kārtulai jau ir jāeksistē. Skatiet vairāk informācijas. *Piezīme:* veicot atklāšanu nekavējoties, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ rezultāts neatspoguļos pavisam nesenās izmaiņas atklāšanas kārtulas konfigurācijā.
Test	Pārbaudīt atklāšanas kārtulas konfigurāciju. Izmantojiet šo pogu, lai pārbaudītu konfigurācijas iestatījumus (piemēram, savienojamību un parametru pareizību), pastāvīgi nepiemērojot nekādas izmaiņas.
Delete	Dzēst atklāšanas kārtulu.
Cancel	Atcelt atklāšanas kārtulas rekvizītu rediģēšanu.

Atklātās entītijas Tālāk redzamie ekrānuzņēmumi ilustrē, kā hosta konfigurācijā izskatās atklātie vienumi, trigeri un grafiki. Atklātajām entītijām priekšā ir oranža saite uz atklāšanas kārtulu, no kuras tās ir iegūtas.



Ņemiet vērā, ka atklātās entītijas netiks izveidotas, ja jau pastāv entītijas ar tiem pašiem unikalitātes kritērijiem, piemēram, vienums ar to pašu atslēgu vai grafiks ar to pašu nosaukumu. Šādā gadījumā lietotāja saskarnē tiek parādīts kļūdas ziņojums, ka zema līmeņa atklāšanas kārtula nevarēja izveidot noteiktas entītijas. Tomēr pati atklāšanas kārtula nekļūs par neatbalstītu tikai tāpēc, ka kādu entītiju nevarēja izveidot un tā bija jāizlaiž. Atklāšanas kārtula turpinās izveidot/atjaunināt citas entītijas.

Ja atklātā entītija (hosts, failu sistēma, saskarne utt.) vairs netiek atklāta (vai vairs neatbilst filtram), entītijas, kas tika izveidotas, pamatojoties uz to, var tikt automātiski atspējotas un vēlāk dzēstas.

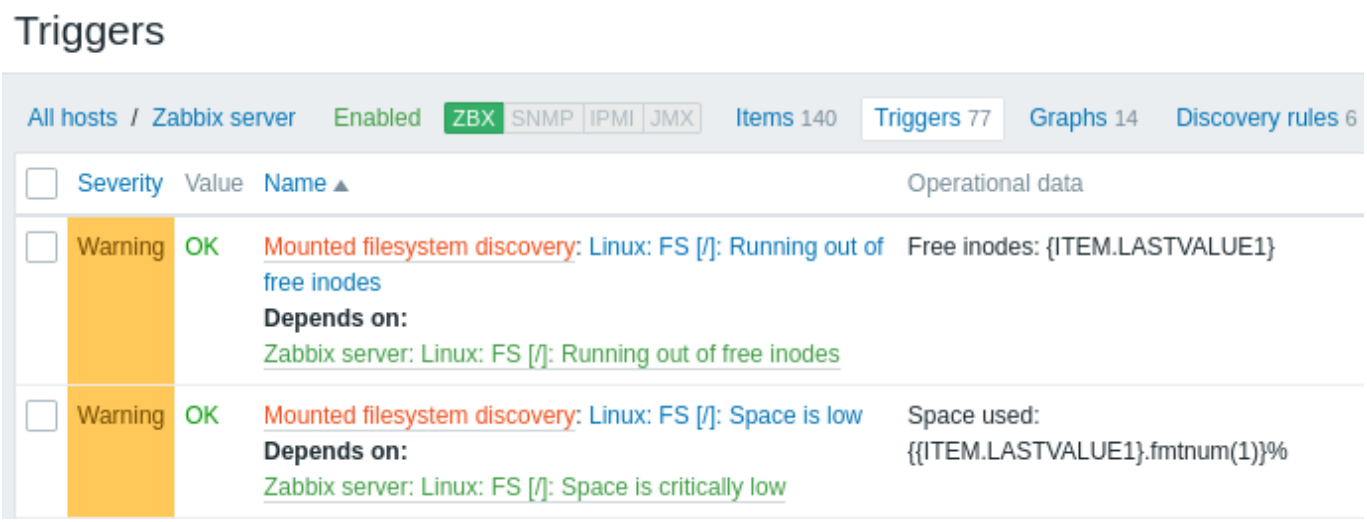
Zaudētie resursi var tikt automātiski atspējoti atkarībā no parametra *Disable lost resources* vērtības. Tas attiecas uz zaudētajiem hostiem, vienumiem un trigeriem.

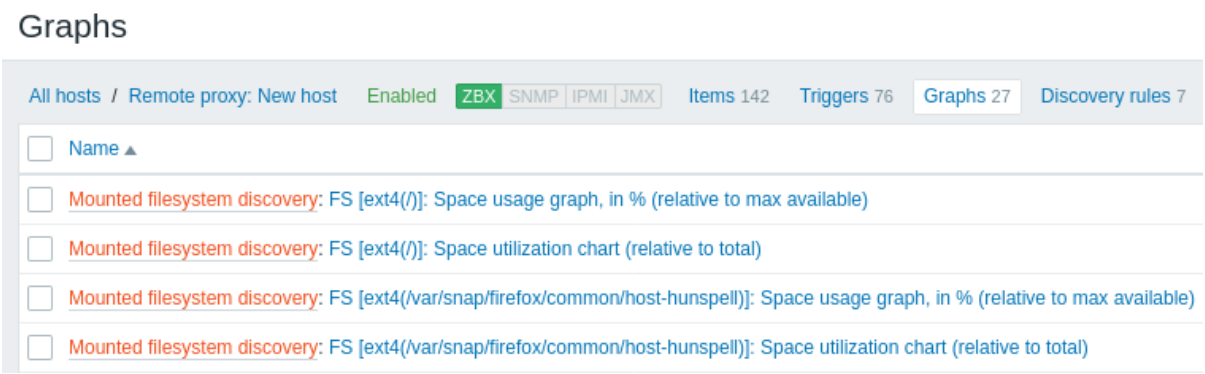
Zaudētie resursi var tikt automātiski dzēsti atkarībā no parametra *Delete lost resources* vērtības. Tas attiecas uz zaudētajiem hostiem, hostu grupām, vienumiem, trigeriem un grafikiem.

Kad atklātās entītijas kļūst par "Vairs netiek atklātas", entītiju sarakstā tiek parādīts dzīves ilguma indikators. Pārvietojiet peles kursoru virs tā, un tiks parādīts ziņojums ar detalizētu informāciju par statusu.

Ja entītijas tika atzīmētas dzēšanai, bet netika dzēstas paredzētajā laikā (atspējota atklāšanas kārtula vai vienuma hosts), tās tiks dzēstas nākamajā reizē, kad tiks apstrādāta atklāšanas kārtula.

Entītijas, kas satur citas entītijas un ir atzīmētas dzēšanai, netiks atjauninātas, ja izmaiņas tiks veiktas atklāšanas kārtulas līmenī. Piemēram, uz LLD balstīti trigeri netiks atjaunināti, ja tie satur vienumus, kas ir atzīmēti dzēšanai.





Citi atklāšanas veidi Plašāka informācija un instrukcijas par citiem gatavajiem atklāšanas veidiem ir pieejamas šādās sadaļās:

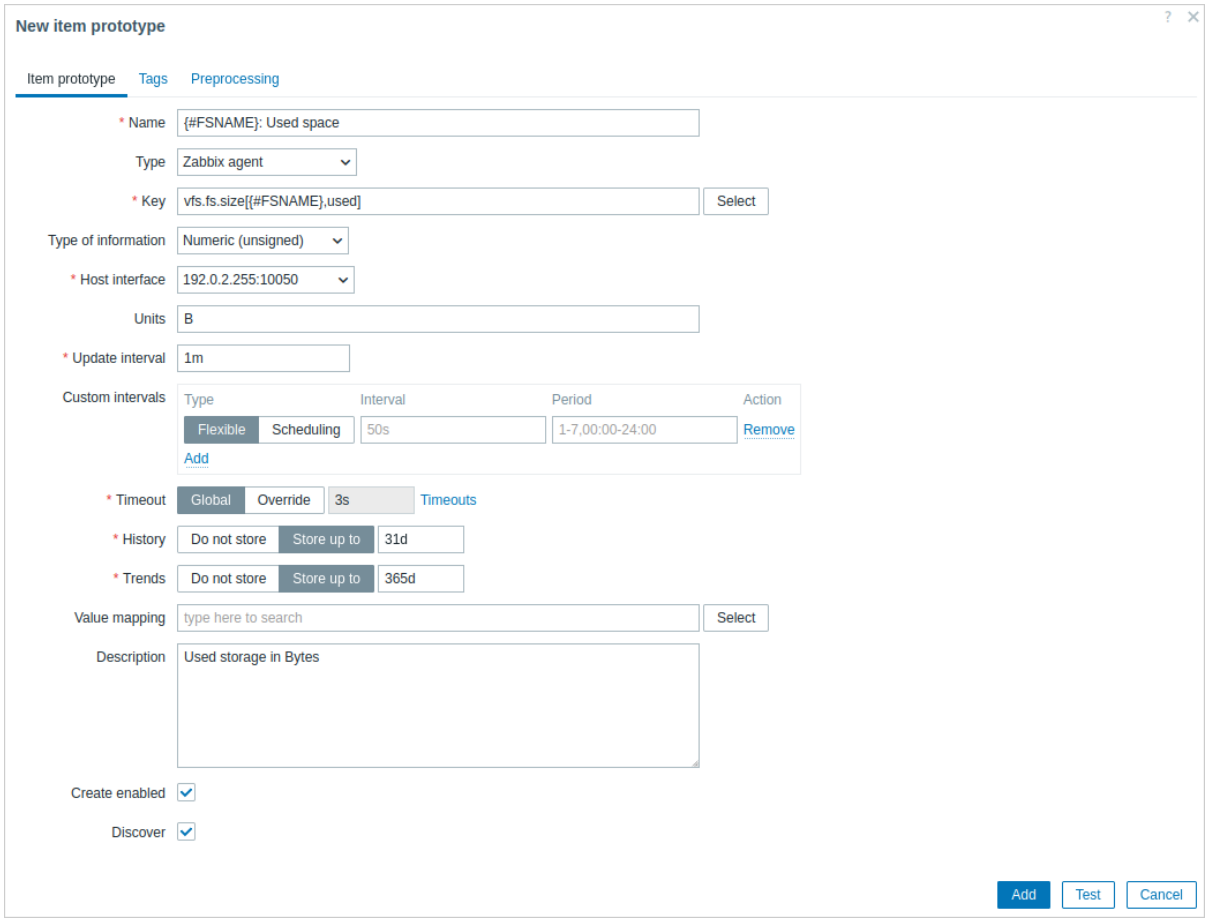
- tīkla saskarņu atklāšana
- CPU un CPU kodolu atklāšana
- SNMP OID atklāšana
- JMX objektu atklāšana;
- atklāšana, izmantojot ODBC SQL vaicājumus
- Windows pakalpojumu atklāšana
- hostu saskarņu atklāšana Zabbix vidē

Plašāku informāciju par JSON formātu atklāšanas vienumiem un piemēru, kā ieviest savu failu sistēmas atklājēju kā Perl skriptu, skatiet sadaļā pielāgotu LLD noteikumu izveide.

1 Vienumu prototipi

Kad kārta ir izveidota, atveriet šīs kārtas vienumus un nospiediet ”Izveidot vienuma prototipu”, lai izveidotu vienuma prototipu.

Ņemiet vērā, kā makross {#FSNAME} tiek izmantots vietā, kur ir nepieciešams failu sistēmas nosaukums. Zema līmeņa atklāšanas makrosa izmantošana vienuma atslēgā ir obligāta, lai nodrošinātu, ka atklāšana tiek apstrādāta pareizi. Kad atklāšanas kārta tiek apstrādāta, šis makross tiks aizstāts ar atklāto failu sistēmu.



Note:
Ja vienuma atslēgas parametri netiek izmantoti, ievietojiet LLD makrosu vienuma atslēgas iekavās `[...]` (piemēram, `v_[{#MACRO}]`).

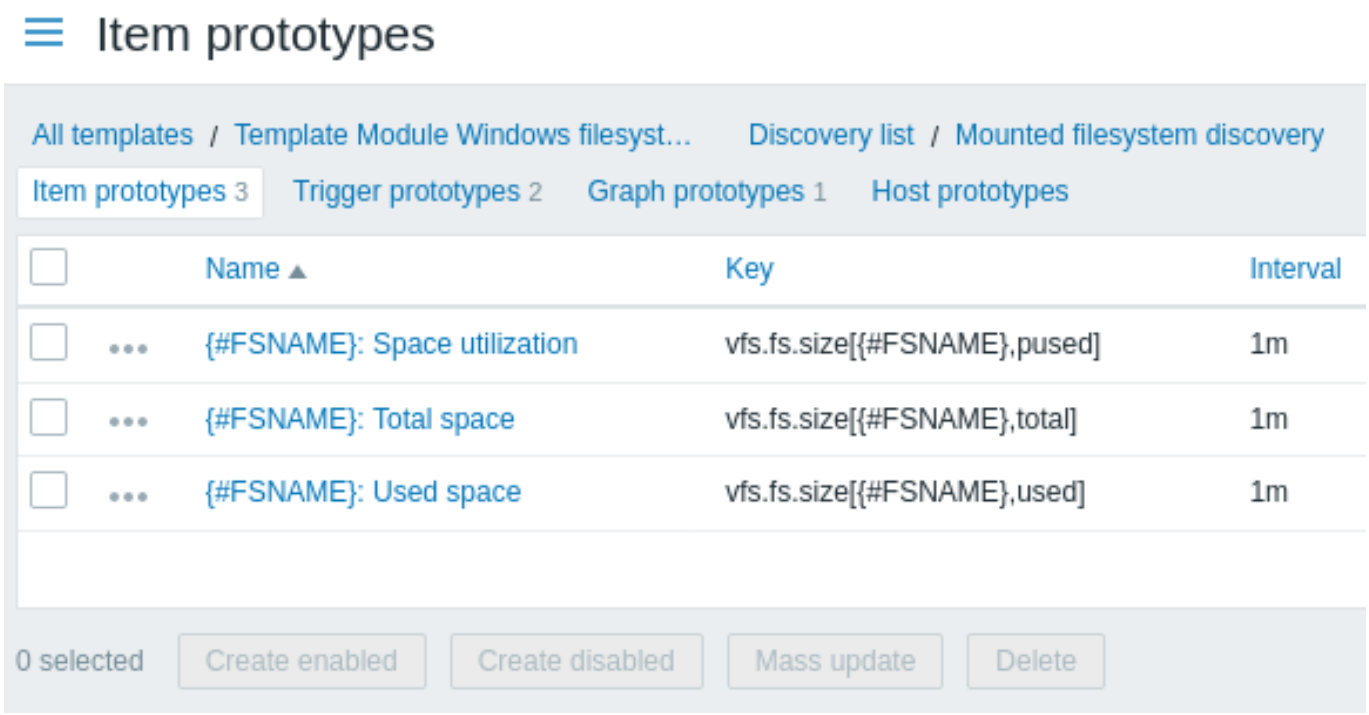
Zema līmeņa atklāšanas makrosi un lietotāja makrosi tiek atbalstīti vienuma prototipa konfigurācijā un vienuma vērtības priekšapstrādes parametros. Ņemiet vērā, ka, izmantojot tos atjaunināšanas intervālos, vienam makrosam ir jāaizpilda viss lauks. Vairāki makrosi vienā laukā vai makrosi kopā ar tekstu netiek atbalstīti.

Note:
Kontekstam atbilstoša zema līmeņa atklāšanas makrosu ekrānēšana tiek veikta drošai izmantošanai regulāro izteiksmju un XPath priekšapstrādes parametros.

Atribūti, kas ir specifiski vienumu prototipiem:

Parameter	Description
Create enabled	Ja atzīmēts, vienums tiks pievienots iespējotā stāvoklī. Ja nav atzīmēts, vienums tiks pievienots atklātajai entītijai, bet atspējotā stāvoklī.
Discover	Ja atzīmēts (noklusējums), vienums tiks pievienots atklātajai entītijai. Ja nav atzīmēts, vienums netiks pievienots atklātajai entītijai, ja vien šis iestatījums nav pārrakstīts atklāšanas kārtā.

Mēs varam izveidot vairākus vienumu prototipus katram failu sistēmas rādītājam, kas mūs interesē:



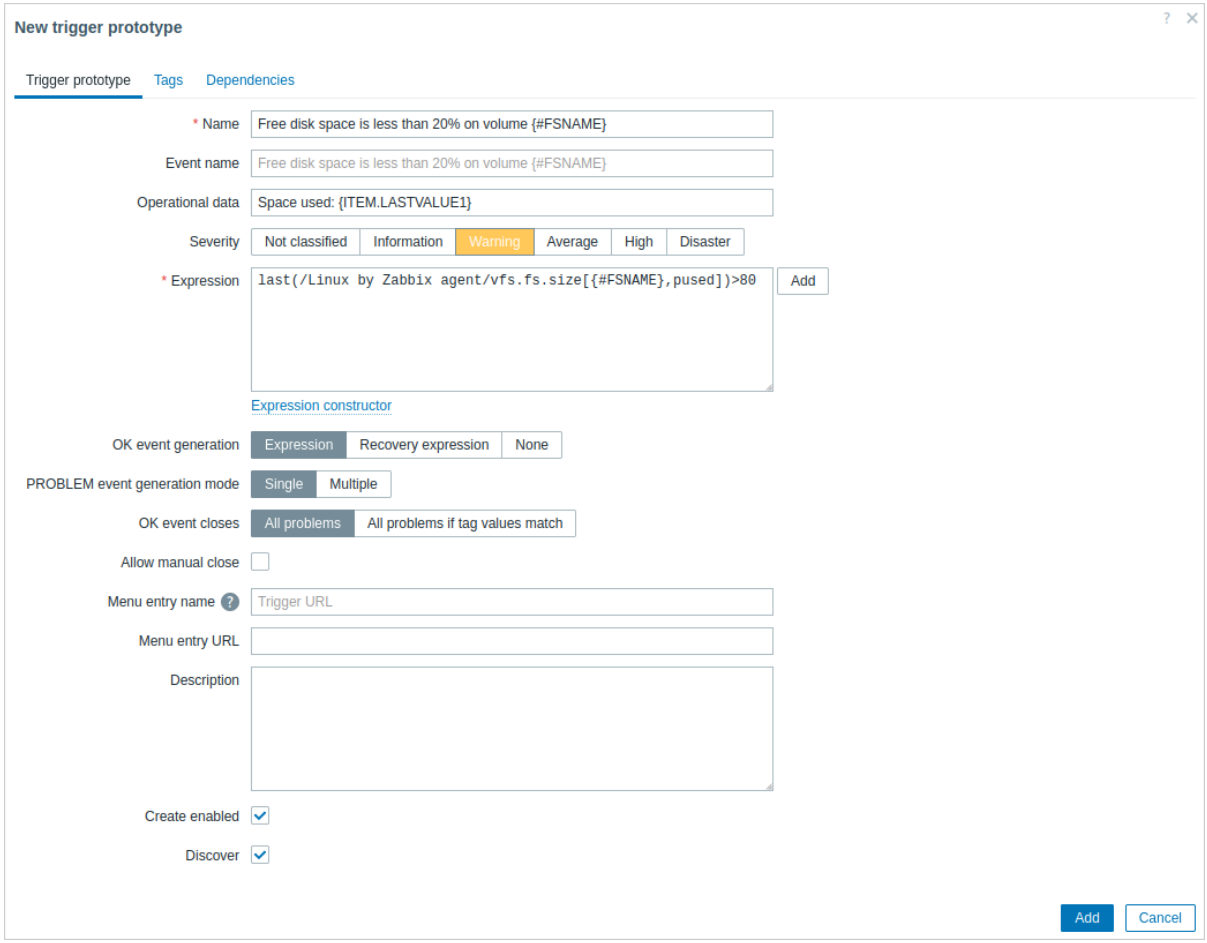
Noklikšķiniet uz trīs punktu ikonas, lai atvērtu izvēlni konkrētajam vienuma prototipam ar šādām opcijām:

- *Izveidot trigera prototipu* - izveidot trigera prototipu, pamatojoties uz šo vienuma prototipu
- *Trigeru prototipi* - noklikšķiniet, lai skatītu sarakstu ar saitēm uz jau konfigurētiem šī vienuma prototipa trigeru prototipiem
- *Izveidot atkarīgo vienumu* - izveidot atkarīgo vienumu šim vienuma prototipam

Opcija *Masveida atjaunināšana* ir pieejama, ja vēlaties vienlaikus atjaunināt vairāku vienumu prototipu īpašības.

2 Trigeru prototipi

Mēs veidojam trigeru prototipus līdzīgi kā vienumu prototipus:

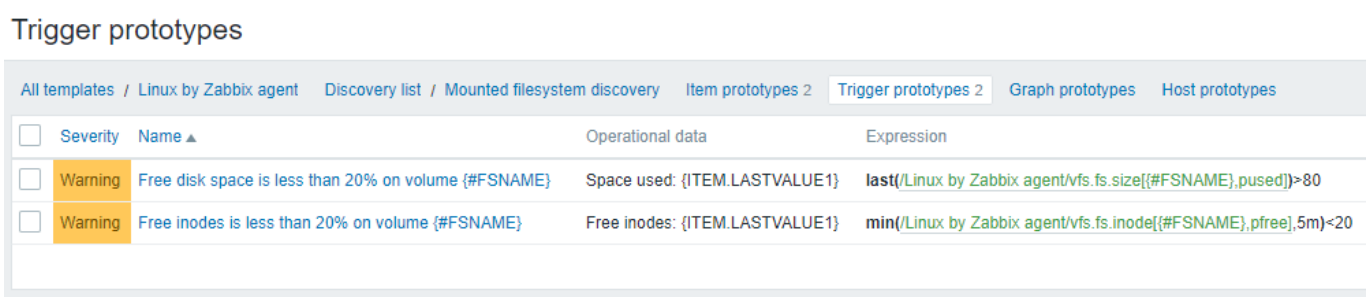


Atribūti, kas ir specifiski trigeru prototipiem:

Parametrs	Apraksts
Create enabled	Ja atzīmēts, trigeris tiks pievienots iespējotā stāvoklī. Ja nav atzīmēts, trigeris tiks pievienots atklātajai entītijai, bet atspējotā stāvoklī.
Discover	Ja atzīmēts (noklusējums), trigeris tiks pievienots atklātajai entītijai. Ja nav atzīmēts, trigeris netiks pievienots atklātajai entītijai, ja vien šis iestatījums nav pārrakstīts atklāšanas kārtulā.

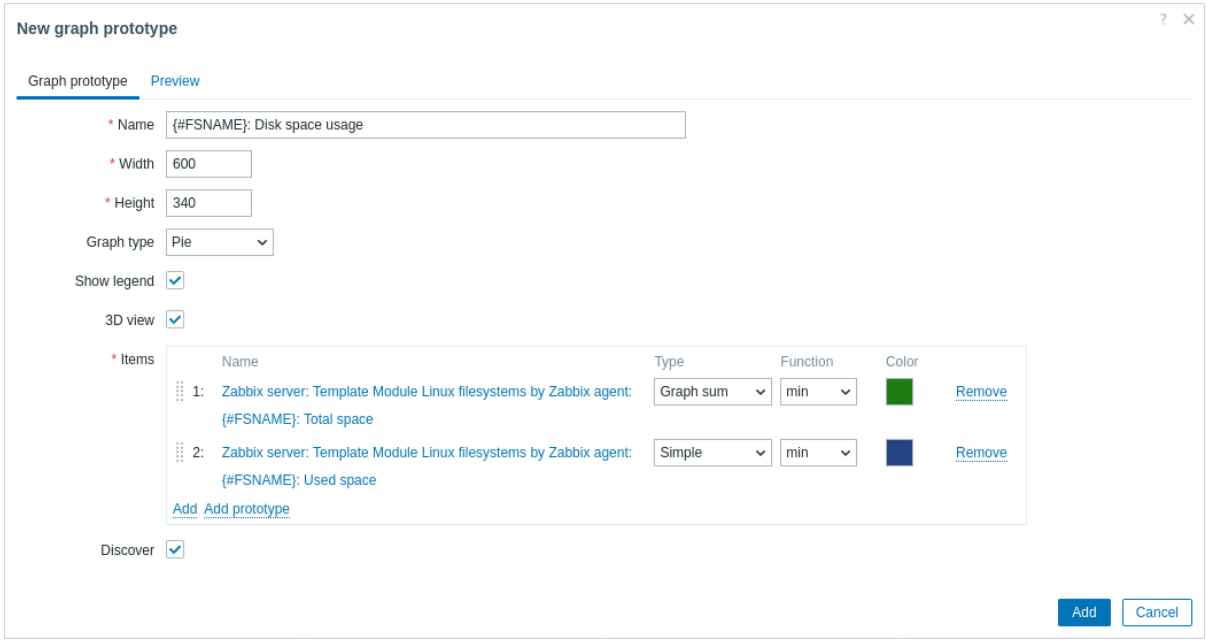
Kad no prototipiem tiek izveidoti reāli trigeri, var rasties nepieciešamība elastīgi noteikt, kāda konstante (mūsu piemērā '20') tiek izmantota salīdzināšanai izteiksmē. Skatiet, kā lietotāja makrosi ar kontekstu var būt noderīgi, lai nodrošinātu šādu elastību.

Jūs varat definēt atkarības starp trigeru prototipiem. Lai to izdarītu, atveriet cilni *Dependencies*. Trigeru prototips var būt atkarīgs no cita trigeru prototipa no tās pašas zema līmeņa atklāšanas (LLD) kārtulas vai no parasta trigera. Trigeru prototips nevar būt atkarīgs no trigeru prototipa no citas LLD kārtulas vai no trigera, kas izveidots no trigera prototipa. Hosta trigera prototips nevar būt atkarīgs no trigera no veidnes.



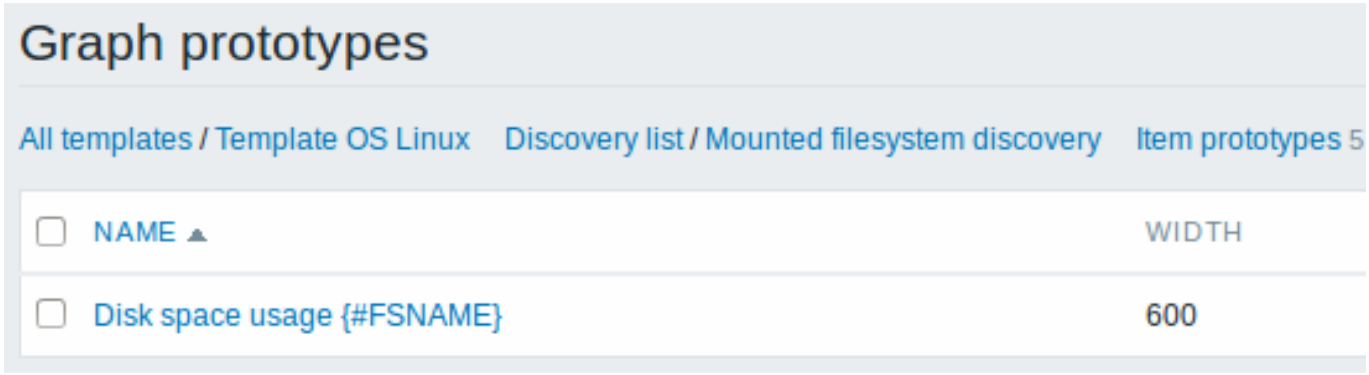
3 Grafiku prototipi

Mēs varam izveidot arī grafiku prototipus:

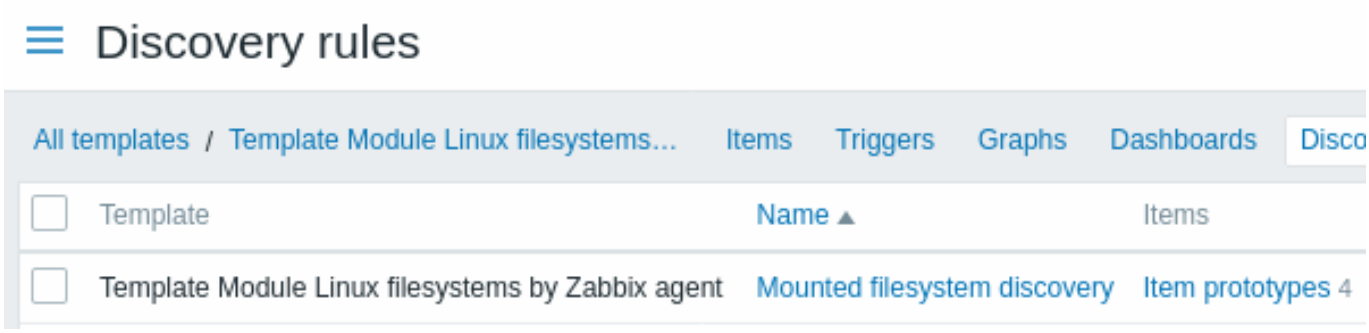


Atribūti, kas ir specifiski grafiku prototipiem:

Parametrs	Apraksts
Atklāt	Ja atzīmēts (noklusējums), grafiks tiks pievienots atklātajai entītijai. Ja nav atzīmēts, grafiks netiks pievienots atklātajai entītijai, ja vien šis iestatījums netiek pārrakstīts atklāšanas kārtulā.



Visbeidzot, mēs esam izveidojuši atklāšanas kārtulu, kas izskatās, kā parādīts zemāk. Tai ir pieci vienumu prototipi, divi trigeru prototipi un viens grafika prototips.



4 Hostu prototipi

Hostu prototipi ir rasējumi hostu izveidei, izmantojot zemā līmeņa atklāšanas noteikumus. Pirms tie tiek atklāti kā hosti, šiem prototipiem nevar būt vienumi un trigeri, izņemot tos, kas ir piesaistīti no veidnēm.

Konfigurācija

Hostu prototipi tiek konfigurēti sadaļā zema līmeņa atklāšanas noteikumi.

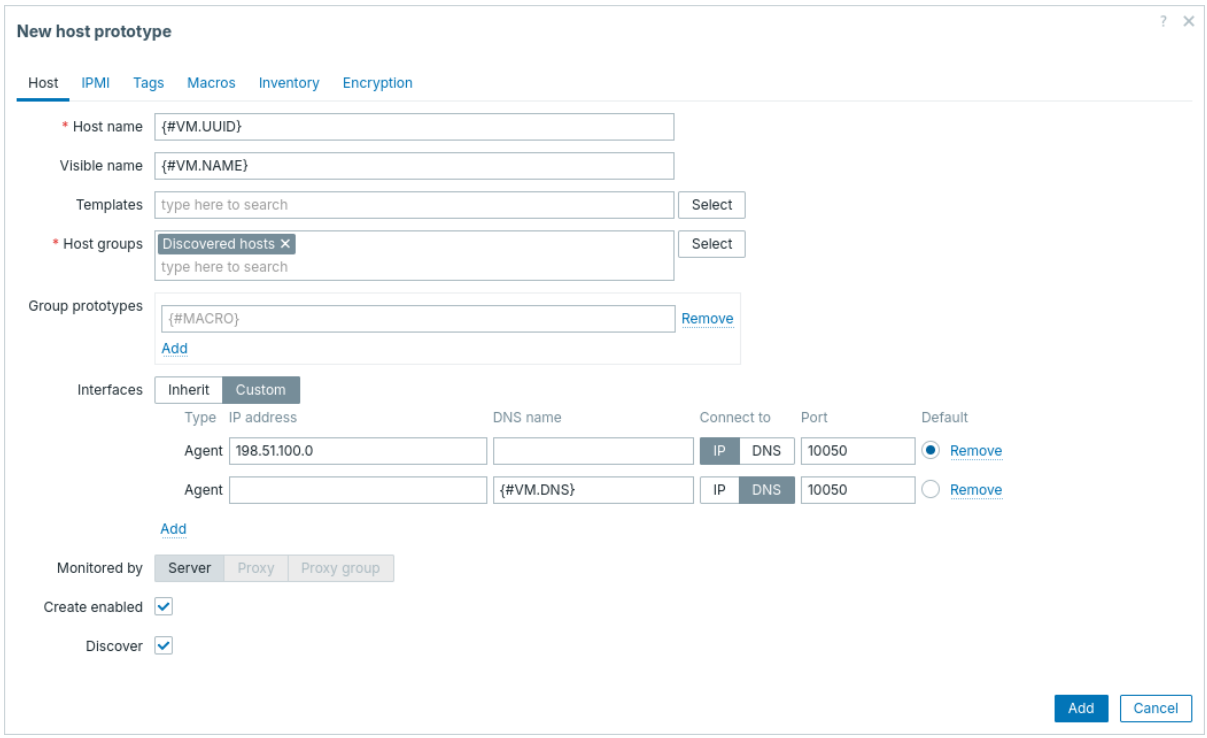
Lai izveidotu hosta prototipu:

1. Atveriet *Datu ievākšana → Hosti*.

2. Nepieciešamajam hostam noklikšķiniet uz *Atklāšana*, lai pārietu uz šim hostam konfigurēto zema līmeņa atklāšanas noteikumu sarakstu.

3. Nepieciešamajam atklāšanas noteikumam noklikšķiniet uz *Hostu prototipi*.

4. Augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot hosta prototipu* un aizpildiet konfigurācijas formu.



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Parametrs	Apraksts
Hosta nosaukums	Šim parametram jāsatur vismaz viens zema līmeņa atklāšanas makross, lai nodrošinātu unikālus hostu nosaukumus izveidotajiem hostiem.
Redzamais nosaukums	Tiek atbalstīti zema līmeņa atklāšanas makrosi.
Grupu prototipi	Ļauj norādīt hostu grupu prototipus, izmantojot zema līmeņa atklāšanas makrosus. Pamatojoties uz norādītajiem grupu prototipiem, tiks atklātas, izveidotas un ar izveidotajiem hostiem sasaistītas hostu grupas; atklātās grupas, kuras jau ir izveidotas ar citiem zema līmeņa atklāšanas noteikumiem, arī tiks sasaistītas ar izveidotajiem hostiem. Tomēr atklātās hostu grupas, kas atbilst manuāli izveidotām hostu grupām, netiks sasaistītas ar izveidotajiem hostiem.
Saskarnes	Norādiet, vai atklātie hosti pārmanto IP adresi no hosta, kuram pieder atklāšanas noteikums (noklusējums), vai saņem pielāgotas saskarnes. Tiek atbalstīti zema līmeņa atklāšanas makrosi un lietotāja makrosi.
Uzrauga	Šis lauks tiek automātiski sinhronizēts ar atbilstošo lauku hosta konfigurācijā.
Izveidot iespējotu	Iestatiet atklāto hostu statusu; ja izvēles rūtiņa nav atzīmēta, hosti tiks izveidoti kā atspējoti.
Atklāt	Norādiet, vai hosti tiks izveidoti no hosta prototipa; ja izvēles rūtiņa nav atzīmēta, hosti no hosta prototipa netiks izveidoti (ja vien šis iestatījums nav pārrakstīts zema līmeņa atklāšanas noteikumā).

Note:
zema līmeņa atklāšanas makrosi tiek atbalstīti arī birku vērtībām un hosta prototipa lietotāja makrosu vērtībām.
Vērtību kartes hostu prototipiem netiek atbalstītas.

Hosta prototipa konfigurēšanas piemēru skatiet sadaļā *Virtuālo mašīnu uzraudzība*.

Hostu saskarnes

Lai pievienotu pielāgotas saskarnes, pārslēdziet *Interfaces* atlasītāju no "Inherit" uz "Custom". Noklikšķiniet uz Add un atlasiet saskarnes tipu — Zabbix aģents, SNMP, JMX, IPMI.

Note:
Ja ir atlasīts *Custom*, bet nav iestatīta neviena saskarne, hosti tiks izveidoti bez saskarnēm.
 Ja ir atlasīts *Inherit* un hosta prototips pieder veidnei, visi atklātie hosti mantos hosta saskarni no hosta, ar kuru veidne ir saistīta.

Ja ir norādītas vairākas pielāgotas saskarnes, primāro saskarni var iestatīt kolonnā *Default*.

Piemēru, kā konfigurēt pielāgotu hosta saskarni, skatiet *VMware monitoringa iestatīšanas piemērā*.

Warning:
Hosts tiks izveidots tikai tad, ja hosta saskarne saturēs pareizus datus.

Atklātie hosti

Hostu sarakstā atklātajiem hostiem priekšā ir tās atklāšanas kārtulas nosaukums, kas tos izveidoja.

Atklātie hosti manto lielāko daļu parametru no hostu prototipiem kā *tikai lasāmus*. Konfigurēt var tikai šādus atklātā hosta parametrus:

- *Veidnes* - piesaistīt jaunas veidnes vai atsaistīt manuāli pievienotās. Veidnes, kas mantotas no hosta prototipa, nevar atsaistīt.
- *Apraksts* - pievienot vai rediģēt hosta aprakstu.
- *Statuss* - iespējot vai atspējot hostu.
- *Birkas* - pievienot jaunas birkas vai noņemt manuāli pievienotās. Birkas, kas mantotas no hosta prototipa, nevar noņemt. Ņemiet vērā, ka birkas nevar dublēties (birkas ar vienādu nosaukumu un vērtību). Ja hosta prototipam tiek pievienota jauna birka un tā sakrīt ar manuāli pievienotu birku atklātajā hostā, atklāšanas laikā manuāli pievienotā birka tiks aizstāta ar mantoto birku.
- *Makrosi* - pievienot jaunus hosta makrosus vai noņemt manuāli pievienotos, kā arī tos, kas mantoti no hosta prototipa; mainīt makrosu vērtības un tipus.

Note:
Atklātie hosti manto arī lietotāja makrosus no hosta, kurā ir konfigurēta atklāšanas kārtula. Šos mantotos makrosus var noņemt, kā arī var mainīt to vērtības un tipus.

Atklātie hosti var atklāt arī citus hostus. Piemēram, ja hosta prototipam (ko izmanto hipervizoru atklāšanai) ir piesaistīta veidne, kurā ir zema līmeņa atklāšanas kārtula ar savu hosta prototipu (ko izmanto virtuālo mašīnu atklāšanai), Zabbix atklās gan hipervizorus, gan to virtuālās mašīnas. Lai papildus atklātu konteinerus šajās VM, varat izveidot zema līmeņa atklāšanas kārtulas vai piesaistīt jaunu veidni ar hostu prototipiem atklātajām VM, vai arī iepriekš konfigurēt VM hosta prototipu, piesaistot veidni, kurā pašā ir iekļauti hostu prototipi.

Atklātos hostus var dzēst manuāli. Tomēr ņemiet vērā, ka tie tiks atklāti atkārtoti, ja tiem būs iespējota atklāšana.

Hosti, kas vairs netiek atklāti, var tikt:

- automātiski atspējoti (pamatojoties uz atklāšanas kārtulas *Disable lost resources* vērtību)
- automātiski dzēsti (pamatojoties uz atklāšanas kārtulas *Delete lost resources* vērtību).

5 Atklāšanas prototipi

Pārskats

Atklāšanas prototipi ir **ligzdoti zema līmeņa atklāšanas** noteikumi "vecākā" atklāšanas noteikuma ietvaros, kas ļauj izveidot daudzlīmeņu objektu atklāšanu ar tiem piederošajiem vienumiem, trigeriem utt. Piemēram, jūs varat vēlēties atklāt visas datubāzes instances datubāzes serverī, pēc tam atklāt tabulvietas katrai instancei un pēc tam atklāt tabulas katrai tabulvietai.

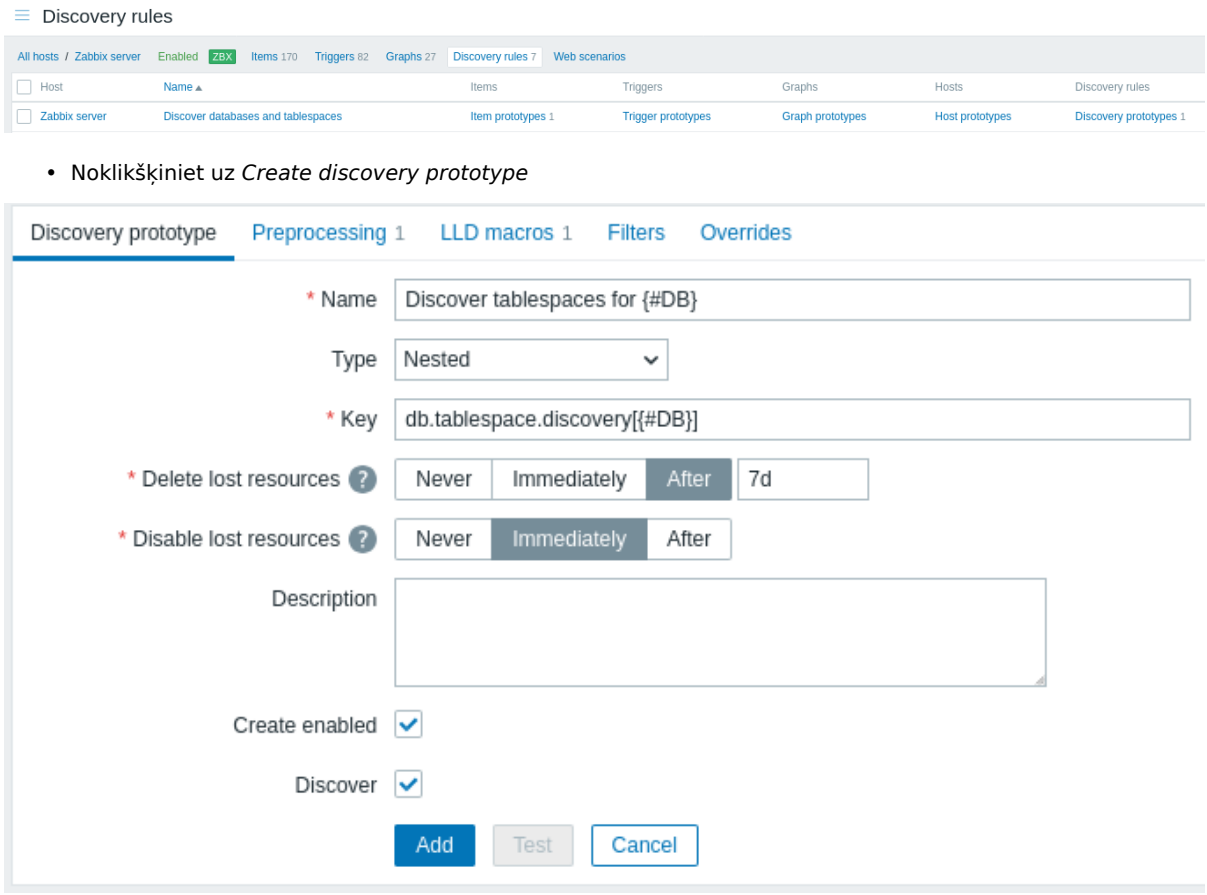
Atklāšanas prototipiem ir savi vienumu, trigeru, grafiku, hostu un atklāšanas prototipi. Ligzdots atklāšanas prototips izmantos to pašu JSON vērtību kā vecākais noteikums, ja norādīsiet tipu *Ligzdots*.

Atklāšanas prototipu ligzdošanas līmeņu skaits nav ierobežots.

Konfigurācija

Lai izveidotu atklāšanas prototipu:

- Noklikšķiniet uz *Discovery prototypes* esoša atklāšanas noteikuma rindā



- Noklikšķiniet uz *Create discovery prototype*

Šīs formas konfigurācijas lauki ir kopīgi ar parasto zema līmeņa atklāšanu.

Ja atvērtajā atklāšanas prototipa formā kā *Type* izvēlaties ”Nested”, tad atklāšanas noteikumi (no atklāšanas prototipa) tiek ģenerēti, pamatojoties uz JSON objektu no tās pašas JSON vērtības kā vecākatklāšanas noteikums. Piemēram, ja sākotnējais JSON ir [<objekts A>, <objekts B>] un ir viens ligzdota atklāšanas noteikuma prototips, tad tiks ģenerēti divi atklāšanas noteikumi, attiecīgi balstoties uz objekta A un objekta B datiem.

Šajā gadījumā atklāšanas prototips tiek aktivizēts vienlaikus ar vecāknoteikumu. Tādējādi ligzdotais noteikums var izmantot priekšapstrādi, lai strādātu ar citu to pašu datu ”šķēlumu”, ko vecāknoteikums jau ir ieguvis.

LLD makrosi no vecākā LLD noteikuma ir pieejami ligzdotajiem atklāšanas noteikumiem.

Ligzdotie LLD noteikumi atklātajos hostos

Ligzdotu zema līmeņa atklāšanas noteikumu var izmantot hosta veidnē, kas piešķirta hosta prototipam. Ja atklātajā hostā pastāv *Ligzdots* atklāšanas noteikums, tad JSON objekts, kas izmantots hosta atklāšanai, tiek nosūtīts arī visiem šī hosta ligzdotā tipa LLD noteikumiem. Papildinformāciju skatiet piemērā.

LLD makrosi no atklāšanas noteikuma, kas izveidoja hostu, ir pieejami ligzdotajiem atklāšanas noteikumiem.

Piemērs

Apskatīsim iespējamu atklāšanas prototipu pielietojumu, pamatojoties uz šādu daudzlīmeņu JSON piemēru.

```
[
  {
    "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" }
    ]
  }
]
```

Gadījums 1

Datubāzes instanču atklāšana datubāzes serverī, pēc tam katras instances tabulvietu atklāšana.

1. Jums ir vismaz viens hosts, kas saistīts ar datubāzes servera atklāšanu.
2. Izveidojiet šim hostam LLD kārtulu ar nosaukumu *Atklāt datubāzes un tabulvietas*.
3. Šajā kārtulā atveriet cilni *LLD makrosi* un pievienojiet makrosu `{#DB}=$.database`.
4. Pievienojiet šai kārtulai vienuma prototipu ar nosaukumu *Aktīvie savienojumi ar {#DB}* (Tips: Aģents, Atslēga: `db.connections[{#DB}]`).
5. Tiek atklāti ar katru datubāzi saistītie vienumi:

```
Active connections to db1, Active connections to db2, Active connections to db3.
```

6. Izveidojiet šai kārtulai atklāšanas prototipu ar nosaukumu *Atklāt tabulvietas priekš {#DB}* (Tips: Ligzdots, Atslēga: `db.tablespace.discovery[{#DB}]`).
7. Atveriet šī atklāšanas prototipa cilni *Preprocessing* un pievienojiet soli `JSONPath=$.tablespaces`.
8. Atveriet šī atklāšanas prototipa cilni *LLD makrosi* un pievienojiet makrosu `{#TSNAME}=$.name`.
9. Izveidojiet šim atklāšanas prototipam vienuma prototipu ar nosaukumu *Tabulvietas {#TSNAME} izmērs priekš {#DB}* (Tips: Aģents, Atslēga: `db.ts.size[{#DB}, {#TSNAME}]`).
10. Tiek atklāti ar katras datubāzes katru tabulvietu saistītie vienumi:

```
Size of tablespace ts1 for db1, Size of tablespace ts2 for db1, Size of tablespace ts3 for db1,
Size of tablespace ts1 for db2, Size of tablespace ts2 for db2, Size of tablespace ts3 for db2,
Size of tablespace ts1 for db3, Size of tablespace ts2 for db3, Size of tablespace ts3 for db3.
```

ar atslēgām `db.ts.size[db1,ts1]`, `db.ts.size[db1,ts2]`, ... `db.ts.size[db3,ts3]`.

Gadījums 2

Datubāzes instanču atklāšana datubāzes serverī, attēlojot tās kā atklātos hostus, un pēc tam tabulvietu atklāšana katrai instancei.

1. Jums ir vismaz viens hosts (saknes hosts), kas saistīts ar datubāzes servera atklāšanu.
2. Izveidojiet veidni, lai atklātu tabulvietas katrai datubāzei.
3. Izveidojiet šajā veidnē vienumu ar nosaukumu *Aktīvie savienojumi ar {#DB}* (Tips: Aģents, Atslēga: `db.connections[{#DB}]`).
4. Izveidojiet šai veidnei LLD kārtulu ar nosaukumu *Atklāt tabulvietas* (Tips: Ligzdots).
5. Pārslēdzieties uz šīs kārtulas cilni *Preprocessing* un pievienojiet soli `JSONPath=$.tablespaces`.
6. Pārslēdzieties uz šīs kārtulas cilni *LLD Macros*, pievienojiet makrosu `{#TSNAME}=$.name`.
7. Izveidojiet šai kārtulai vienuma prototipu ar nosaukumu *Tabulvietas {#TSNAME} izmērs datubāzei {#DB}* (Tips: Aģents, Atslēga: `db.ts.size[{#DB}, {#TSNAME}]`).
8. Atgriežoties pie saknes hosta, izveidojiet šim hostam LLD kārtulu ar nosaukumu *Atklāt datubāzes un tabulvietas*.
9. Pārslēdzieties uz šīs kārtulas cilni *LLD Macros*, pievienojiet makrosu `{#DB}=$.database`.

10. Pievienojiet šai kārtulai hosta prototipu ar nosaukumu *Hosts datubāzei {#DB}*.
11. Pārslēdzieties uz šī hosta prototipa cilni *Macros*, pievienojiet makrosu `{$DB}={#DB}` (vienuma nosaukumam un atslēgai no 3. soļa).
12. Piesaistiet 2. solī izveidoto veidni šim hosta prototipam.
13. Atklātie hosti satur atklātos vienumus, kas saistīti ar katru datubāzi un tās tabulvietām:

Hosts	Vienumi
Hosts datubāzei db1	Aktīvie savienojumi ar db1
	Tabulvietas ts1 izmērs datubāzei db1
	Tabulvietas ts2 izmērs datubāzei db1
	Tabulvietas ts3 izmērs datubāzei db1
Hosts datubāzei db2	Aktīvie savienojumi ar db2
	Tabulvietas ts1 izmērs datubāzei db2
	Tabulvietas ts2 izmērs datubāzei db2
	Tabulvietas ts3 izmērs datubāzei db2
Hosts datubāzei db3	Aktīvie savienojumi ar db3
	Tabulvietas ts1 izmērs datubāzei db3
	Tabulvietas ts2 izmērs datubāzei db3
	Tabulvietas ts3 izmērs datubāzei db3

6 Piezīmes par zema līmeņa atklāšanu

LLD makrosu izmantošana lietotāja makrosa kontekstos

LLD makrosus var izmantot lietotāja makrosa kontekstā, piemēram, trigeru prototipos.

Vairāki LLD noteikumi vienam un tam pašam vienumam

Ir iespējams definēt vairākus zema līmeņa atklāšanas noteikumus ar vienu un to pašu atklāšanas vienumu.

Lai to izdarītu, jums jādefinē aģenta Alias parameter, kas ļauj izmantot mainītas atklāšanas vienuma atslēgas dažādos atklāšanas noteikumos, piemēram, `vfs.fs.discovery[foo]`, `vfs.fs.discovery[bar]` utt.

Datu ierobežojumi atgrieztajām vērtībām

Nav ierobežojuma zema līmeņa atklāšanas kārtulas JSON datiem, ja tie tiek saņemti tieši Zabbix serverī. Tas ir tāpēc, ka atgrieztās vērtības tiek apstrādātas, tās nesaglabājot datubāzē.

Nav arī ierobežojuma pielāgotām zema līmeņa atklāšanas kārtulām. Tomēr, ja pielāgotas zema līmeņa atklāšanas kārtulas dati tiek iegūti, izmantojot lietotāja parametru, tiek piemērots lietotāja parametra atgrieztās vērtības ierobežojums.

Ja datiem ir jāiziet caur Zabbix starpniekserveri, tam šie dati ir jāsaglabā datubāzē. Šādā gadījumā tiek piemēroti datubāzes ierobežojumi.

7 Atklāšanas noteikumi

Lūdzu, izmantojiet sānjoslu, lai skatītu atklāšanas noteikumu konfigurācijas piemērus dažādiem gadījumiem.

1 Pievienoto failu sistēmu atklāšana

Pārskats

Ir iespējams atklāt piemontētās failu sistēmas un to īpašības:

- montēšanas punkta nosaukums
- failu sistēmas tips
- failu sistēmas izmērs
- inode statistika
- montēšanas opcijas

Lai to izdarītu, var izmantot šādu kombināciju:

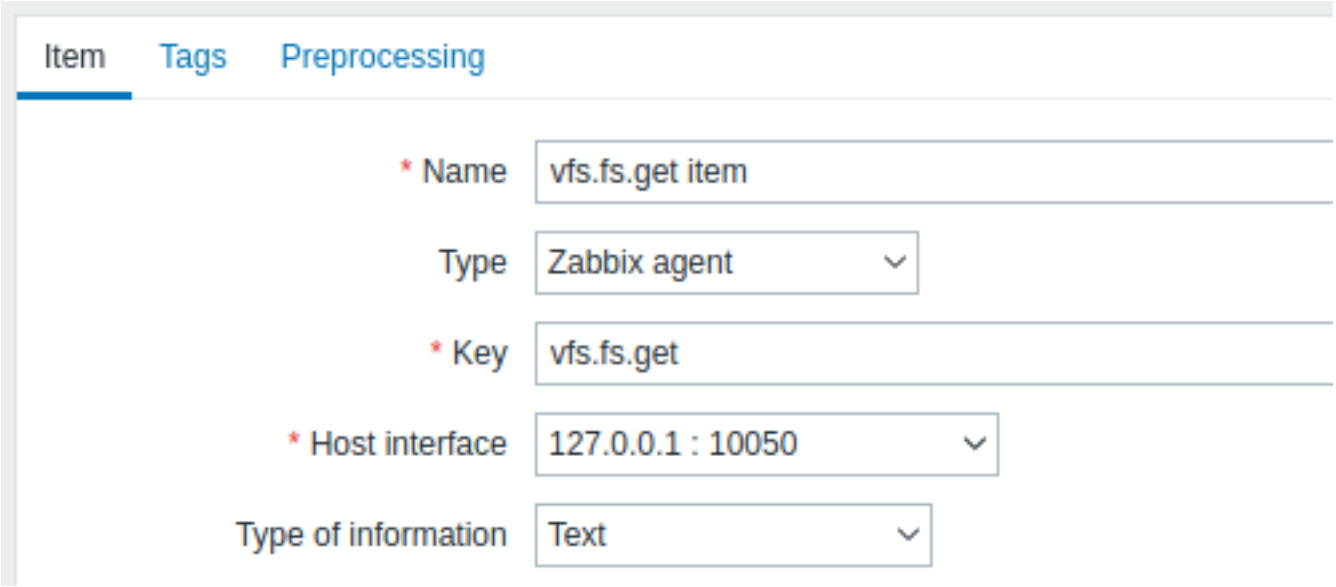
- aģenta vienumu `vfs.fs.get` kā galveno vienumu
- atkarīgo zema līmeņa atklāšanas kārtulu un vienumu prototipus

Konfigurācija

Galvenais vienums

Izveidojiet Zabbix aģenta vienumu, izmantojot šādu atslēgu:

`vfs.fs.get`



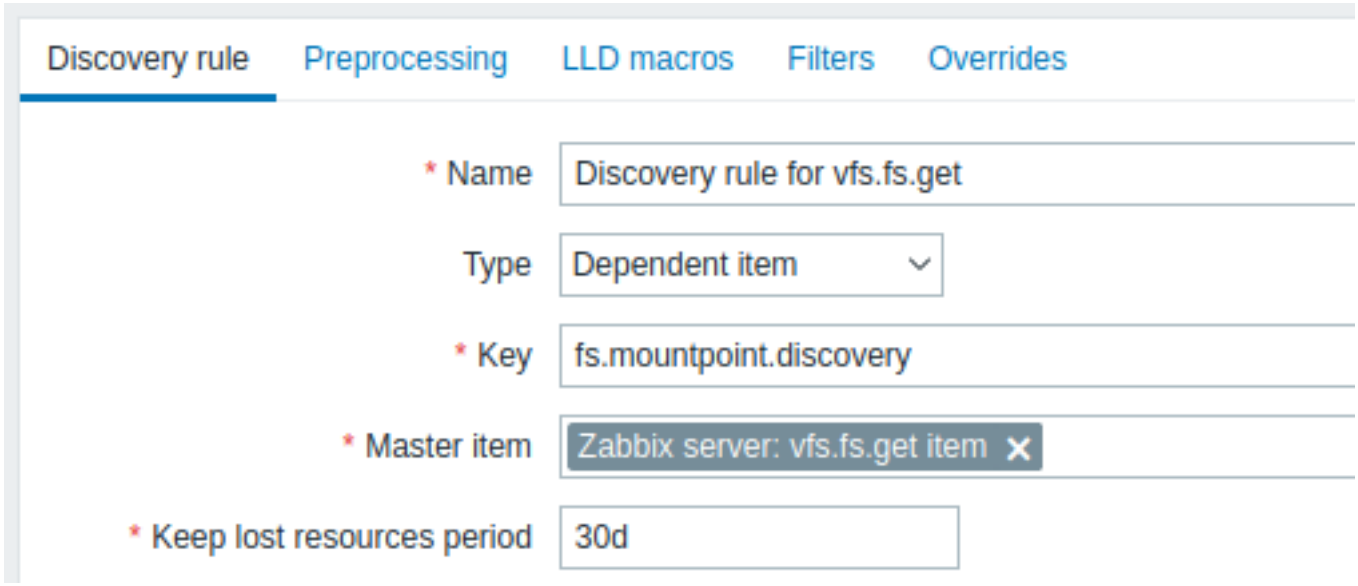
Iestatiet informācijas tipu uz "Text", lai varētu apstrādāt iespējami lielus JSON datus.

Šī vienuma atgrieztie dati piemontētai failu sistēmai saturēs aptuveni šādu informāciju:

```
[
  {
    "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"
  }
]
```

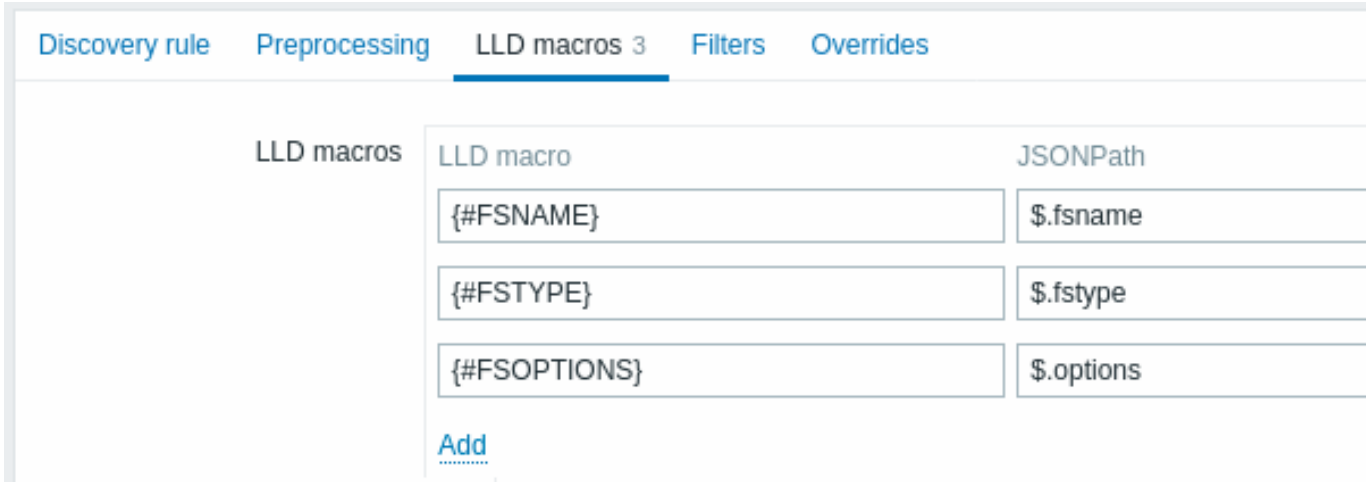
Atkarīgais LLD noteikums

Izveidojiet zema līmeņa atklāšanas noteikumu ar tipu "Atkarīgais vienums":

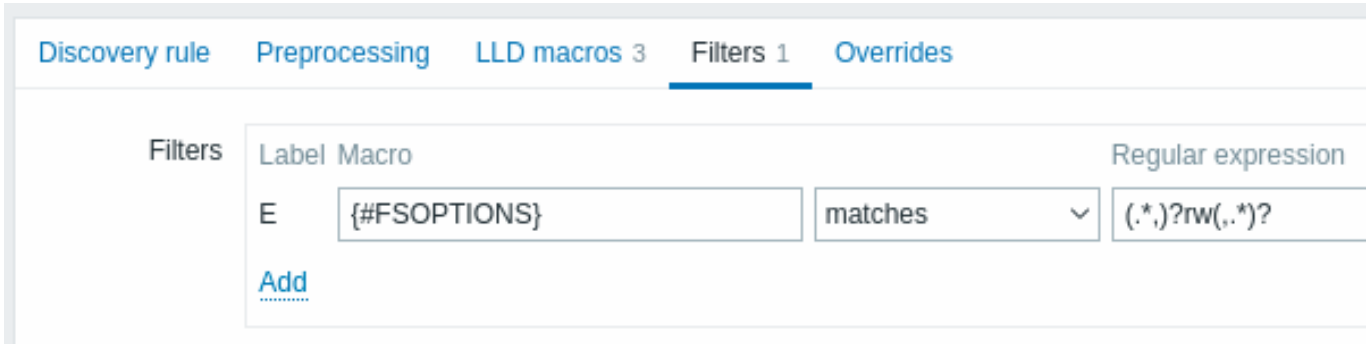


Kā galveno vienumu atlasiet mūsu izveidoto `vfs.fs.get` vienumu.

Cilnē ”LLD makrosi” definējiet pielāgotus makrosus ar atbilstošo JSONPath:

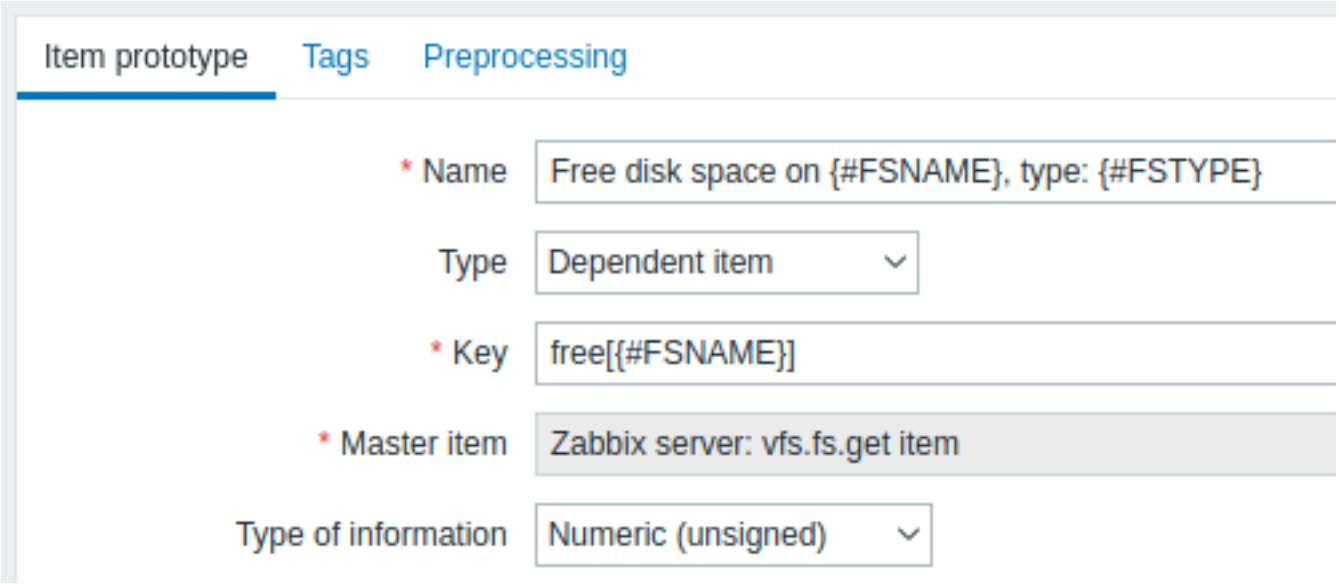


Cilnē ”Filtri” varat pievienot regulāro izteiksmi, kas filtrē tikai **lasīšanas-rakstīšanas** datņu sistēmas:



Atkarīgā vienuma prototips

Izveidojiet šajā LLD kārtulā vienuma prototipu ar tipu ”Dependent item”. Kā galveno vienumu šim prototipam atlasiet iepriekš izveidoto `vfs.fs.get` vienumu.



Ņemiet vērā pielāgoto makrosu izmantošanu vienuma prototipa nosaukumā un atslēgā:

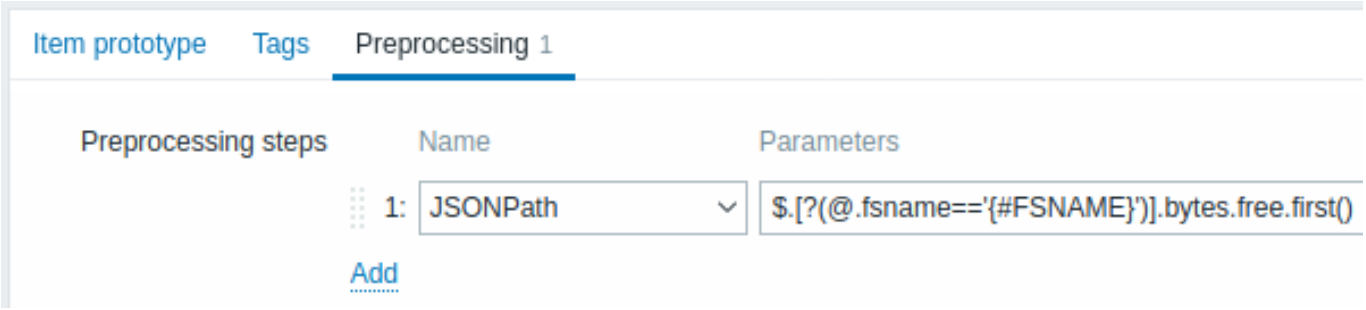
- *Name*: Brīvā diska vieta uz {#FSNAME}, tips: {#FSTYPE}
- *Key*: Free[{#FSNAME}]

Kā informācijas tipu izmantojiet:

- *Numeric (unsigned)* tādiem rādītājiem kā 'free', 'total', 'used'
- *Numeric (float)* tādiem rādītājiem kā 'pfree', 'pused' (procenti)

Vienuma prototipa cilnē "Preprocessing" atlasiet JSONPath un izmantojiet šādu JSONPath izteiksmi kā parametru:

```
$.[?(@.fsname=='{#FSNAME}')].bytes.free.first()
```



Kad atklāšana sāksies, tiks izveidots viens vienums katram montēšanas punktam. Šis vienums atgriezīs brīvo baitu skaitu norādītajam montēšanas punktam.

2 Tīkla saskarņu atklāšana

Līdzīgi kā tiek atklātas failu sistēmas, ir iespējams atklāt arī tīkla saskarnes.

Vienuma atslēga

Vienuma atslēga, ko izmantot atklāšanas noteikumā, ir

```
net.if.discovery
```

Atbalstītie makrosi

Jūs varat izmantot makrosu {#IFNAME} atklāšanas kārtulas filtrā un vienumu, trigeru un grafiku prototipos.

Vienumu prototipu piemēri, kurus jūs varētu vēlēties izveidot, pamatojoties uz "net.if.discovery":

- "net.if.in[{#IFNAME},bytes]",
- "net.if.out[{#IFNAME},bytes]".

Ņemiet vērā, ka Windows vidē tiek atgriezts arī {#IFGUID}.

3 CPU un CPU kodolu atklāšana

Līdzīgi kā tiek atklātas failu sistēmas, ir iespējams atklāt arī CPU un CPU kodolus.

Vienuma atslēga

Vienuma atslēga, ko izmantot atklāšanas kārtulā, ir

`system.cpu.discovery`

Atbalstītie makrosi

Šī atklāšanas atslēga atgriež divus makrosus — {#CPU.NUMBER} un {#CPU.STATUS}, kas attiecīgi identificē CPU kārtas numuru un statusu. Tā uzskaita procesorus no 0 līdz N - 1, kur N ir kopējais procesoru skaits.

```
[
    {
        "{#CPU.NUMBER}": 0,
        "{#CPU.STATUS}": "online"
    },
    {
        "{#CPU.NUMBER}": 1,
        "{#CPU.STATUS}": "offline"
    },
    {
        "{#CPU.NUMBER}": 2,
        "{#CPU.STATUS}": "unknown" /* "unknown" ir tikai Windows sistēmās */
    },
    {
        "{#CPU.NUMBER}": 3,
        "{#CPU.STATUS}": "online"
    }
]
```

Ņemiet vērā, ka nav iespējams skaidri nošķirt faktiskos fiziskos procesorus, kodolus un hiperpavedienus. {#CPU.STATUS} Linux, UNIX un BSD sistēmās atgriež procesora statusu, kas var būt vai nu "online", vai "offline". Windows sistēmās šis pats makross var attēlot arī trešo vērtību — "unknown" —, kas norāda, ka procesors ir noteikts, bet informācija par to vēl nav savākta.

CPU atklāšana balstās uz aģenta kolektora procesu, lai saglabātu saskaņotību ar kolektora sniegtajiem datiem un taupītu resursus datu iegūšanas laikā. Tas nozīmē, ka šī vienuma atslēga nedarbojas ar aģenta binārā faila komandrindas karodziņu testēšanai (-t), kas atgriezīs statusu NOT_SUPPORTED un tam pievienotu ziņojumu, ka kolektora process nav palaists.

Vienumu prototipi, ko var izveidot, pamatojoties uz CPU atklāšanu, ietver, piemēram:

- `system.cpu.util[{#CPU.NUMBER},<type>,<mode>]`
- `system.hw.cpu[{#CPU.NUMBER},<info>]`

Detalizētu vienumu atslēgu aprakstu skatiet sadaļā Zabbix aģenta vienumu atslēgas.

Piezīmes par atklātajiem veiktspējas skaitītāju vienumiem NUMA sistēmās

Daži Windows veiktspējas skaitītāji var nebūt pieejami dažiem loģiskajiem procesoriem NUMA sistēmās.

Piemēram, vienumi, kas atklāti ar vienuma prototipu, izmantojot tālāk norādīto atslēgu, var darboties tikai pirmajam NUMA mezglam. Vienumi citiem NUMA mezgliem var būt neatbalstītā stāvoklī.

`perf_counter[\Processor({#CPU.NUMBER})\% Processor Time,60]`

Tāpat atklāšanas kārtula ar atslēgu "system.cpu.discovery" nenodrošina nevienu zema līmeņa atklāšanas makro vērtību, ko aizvietot šādos veiktspējas skaitītājos kā:

`perf_counter["\Processor Information(<NUMA node index>,<CPU index in NUMA node>)\% Processor Time",60]`

Savukārt system.cpu.util vienumi izmanto atšķirīgus veiktspējas skaitītājus, lai uzraudzītu CPU noslodzes procentu atkarībā no procesoru grupu skaita un procesoru skaita. Pastāv arī reta kļūda, kas saistīta ar procesoru grupām Windows vidē.

4 SNMP OID atklāšana

Pārskats

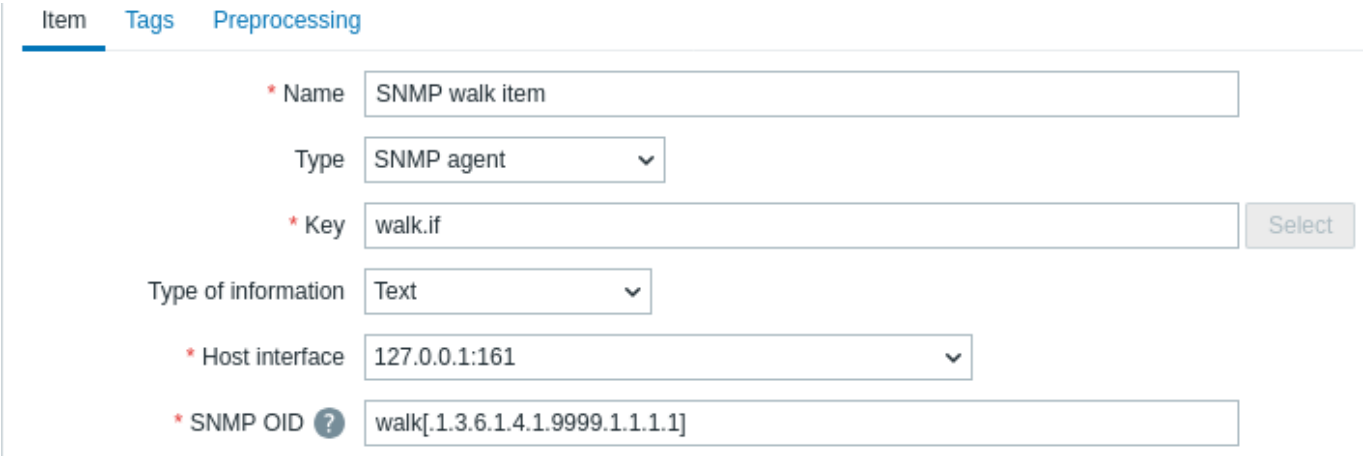
Šajā sadaļā mēs veiksim zemā līmeņa atklāšanu SNMP ierīcē.

Šī SNMP OID atklāšanas metode tiek atbalstīta kopš Zabbix serveris/starpniekserveris 6.4.

Konfigurācijas piemērs

1. Izveidojiet SNMP aģenta vienumu ar atslēgu, piemēram:

```
walk[.1.3.6.1.4.1.9999.1.1.1.1]
```



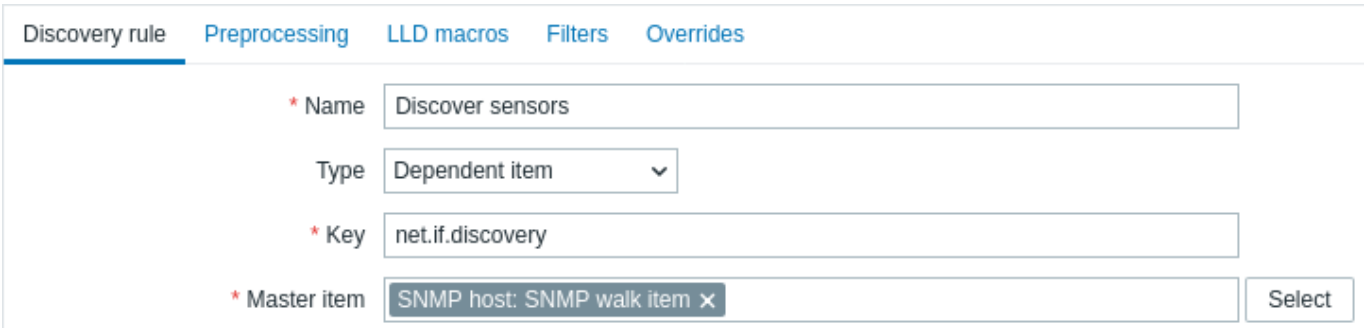
Šis vienums izpilda vienu SNMP tabulas izgājienu un atgriež visus tabulas ierakstus vienā pieprasījumā formātā, kas atbilst `snmpwalk` utilītas izvadei ar formatēšanas opcijām `-Oe -Ot -On`.

Tas atgriezīs šādu vairākrindu teksta vērtību:

```
.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. Izveidojiet atklāšanas kārtulu:

- Laukā *Name* ievadiet aprakstošu atklāšanas kārtulas nosaukumu (piemēram, ”Atklāt sensorus”).
- Laukā *Type* atlasiet ”Dependent item”.
- Laukā *Key* ievadiet aprakstošu atslēgu (piemēram, ”net.if.discovery”).
- Laukā *Master item* atlasiet ”SNMP walk item”.



3. Cilnē *Preprocessing* pievienojiet priekšapstrādes soli, nolaižamajā sarakstā *Name* izvēloties ”SNMP walk to JSON” ar 3 parametriem:

- *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”.

Pēc priekšapstrādes atklāšanas kārtula atgriež JSON masīvu ar makrosu kopām.

Piemēram:

```
[
    {
        "{#SNMPINDEX}": "1",
        "{#SENSORNAME}": "Temperature Sensor",
        "{#SENSORTYPE}": "temp",
        "{#SENSORVALUE}": "100"
```

```
    },
    {
        "{#SNMPINDEX}": "2",
        "{#SENSORNAME}": "Humidity Sensor",
        "{#SENSORTYPE}": "humidity",
        "{#SENSORVALUE}": "200"
    }
]
```

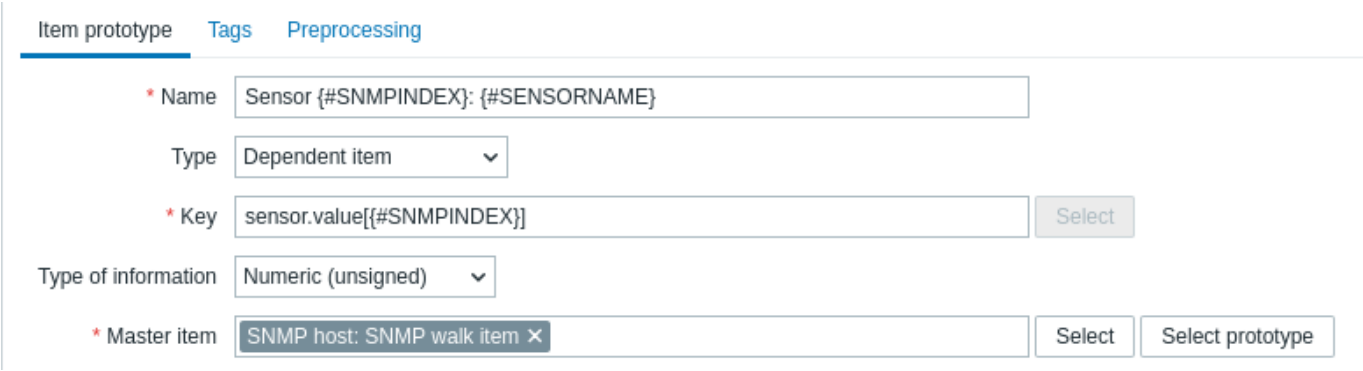
Katrs objekts attēlo vienu atklātu sensoru un nodrošina tādus makrosus kā {#SNMPINDEX}, {#SENSORNAME}, {#SENSORTYPE} un {#SENSORVALUE}.

Tie ir grupēti pēc SNMP indeksa, kas ir skaitliskais sufikss katra OID beigās (piemēram, .1, .2) — šis indekss unikāli identificē katru rindu SNMP tabulā un tiek automātiski iegūts kā {#SNMPINDEX}.

4. Zem atklāšanas kārtulas izveidojiet vienu vai vairākus vienumu prototipus (ar atklāšanas kārtulu kā galveno vienumu).

Piemēram, sensora vērtības atkarīgais vienums:

- Laukā *Name* ievadiet "Sensor {#SNMPINDEX}: {#SENSORNAME}".
- Laukā *Type* atlasiet "Dependent item".
- Laukā *Key* ievadiet "sensor.value[{#SNMPINDEX}]".
- Laukā *Master item* atlasiet "SNMP walk item".



Cilnē *Preprocessing* pievienojiet priekšapstrādes soli ar nosaukumu "SNMP walk value", laukā *Parameter* norādot OID ".1.3.6.1.4.1.9999.1.1.1.1.3.{#SNMPINDEX}". *Format*: "Unchanged".

Tiks atklāti šādi vienumi:

Nosaukums	Atslēga	OID, no kura tiek iegūta vērtība	Vienuma vērtība
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

Kad atklāšanas kārtula tiek izpildīta, tiek izveidoti tādi vienumi kā `sensor.value[1]`, `sensor.value[2]`.

Katrs atkarīgais vienums iegūst savu vērtību no galvenā vienuma SNMP walk rezultāta, izmantojot priekšapstrādi, neveicot atsevišķus SNMP pieprasījumus.

5. Atsaucieties uz atkarīgo vienumu prototipiem trigeru prototipos, izmantojot tos pašus makrosus no atklāšanas kārtulas. Piemērs:

```
{Template_Sensor:sensor.value[{#SNMPINDEX}].last()} > 75
```

Tas izveido trigeri katram atklātajam sensoram (piemēram, sensor.value[1], sensor.value[2]) un nostrādā, ja jaunākā vērtība (temperatūra vai mitrums) pārsniedz 75.

6. Iekļaujiet atkarīgos vienumus katrai atklātajai entītijai. Grafika vienuma atslēgas piemērs:

```
sensor.value[{#SNMPINDEX}]
```

Katram {#SNMPINDEX} tiek izveidots viens grafiks, kurā laika gaitā tiek attēlota temperatūra un mitrums.

Šī konfigurācija katrā aptaujas ciklā izpilda tikai vienu SNMP walk pieprasījumu neatkarīgi no atklāto vienumu skaita. Visi atkarīgie vienumi iegūst savas vērtības no galvenā SNMP walk rezultāta, izmantojot priekšapstrādi, tādējādi būtiski samazinot SNMP datplūsmu un slodzi.

Dinamiskie indeksi ar walk[]

Dinamiskie indeksi (piemēram, saskarņu indeksi) var mainīties, kad tiek pārkonfigurēta aparatūra. Lai pielāgotos šādai darbībai, tiek izveidots galvenais SNMP walk atklāšanas kārtulas vienums ar atslēgu, piemēram:

```
walk[1.3.6.1.2.1.2.2.1.10]
```

Pēc SNMP walk uz JSON priekšapstrādes rezultāts var izskatīties šādi:

```
[
    {
        "{#SNMPINDEX}": "2",
        "{#VALUE}": "123456"
    },
    {
        "{#SNMPINDEX}": "3",
        "{#VALUE}": "654321"
    }
]
```

Atkarīgā vienuma prototips izmanto makrosu {#SNMPINDEX}, lai izveidotu atslēgu:

```
net.if.in[{#SNMPINDEX}]
```

Šī prototipa priekšapstrāde ietver "SNMP walk value" nosaukumu ar OID "1.3.6.1.2.1.2.2.1.10.{#SNMPINDEX}" laukā *Parameter*. *Format*: "Unchanged".

Izpildes laikā tiek izveidoti faktiskie vienumi, piemēram, net.if.in[2] un net.if.in[3]. Ja konkrētas saskarnes indekss mainās (piemēram, ja indekss 2 SNMP tabulā tiek aizstāts ar 5), tad nākamajā atklāšanas kārtulas izpildes reizē:

- Vecais atkarīgais vienums net.if.in[2] tiek atzīmēts kā "lost" vai noņemts, un šim vienumam vairs netiek ievākti jauni dati.
- Tiek izveidots jauns atkarīgais vienums net.if.in[5], sākot ar tukšu vēsturi.
- Vēsturiskie dati no net.if.in[2] netiek automātiski pārvietoti uz net.if.in[5].

Trigera prototipa piemērs:

```
{Template_Interface:net.if.in[{#SNMPINDEX}].last()} > 1000000000
```

Grafika prototipa piemērs ietver vienumus:

```
net.if.in[{#SNMPINDEX}]
net.if.out[{#SNMPINDEX}]
```

Šī konfigurācija nodrošina uzticamu tabulu ar dinamiskiem indeksiem uzraudzību, vienlaikus samazinot SNMP datplūsmu — katram aptaujas ciklam ir nepieciešams tikai viens SNMP walk, bet atkarīgo vienumu prototipi iegūst nepieciešamās vērtības.

Atklātās entītijas

Kad serveris darbojas, tas izveidos reālus atkarīgos vienumus, trigerus un grafikus, pamatojoties uz vērtībām, ko atgriež SNMP atklāšanas kārtula.

5 SNMP OID atklāšana (mantotā)

Pārskats

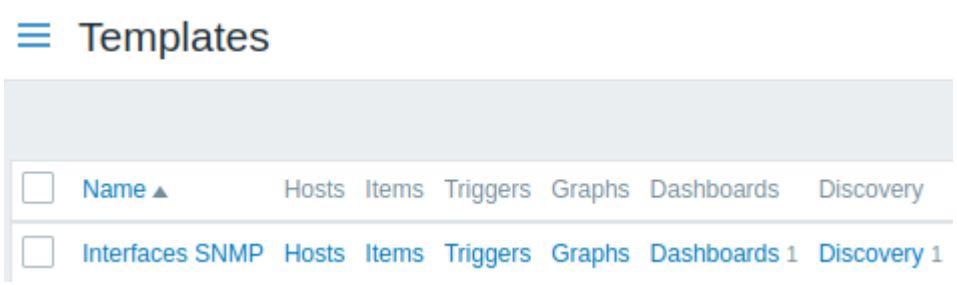
Šajā sadaļā mēs veiksim slēdža SNMP atklāšanu.

Vienuma atslēga

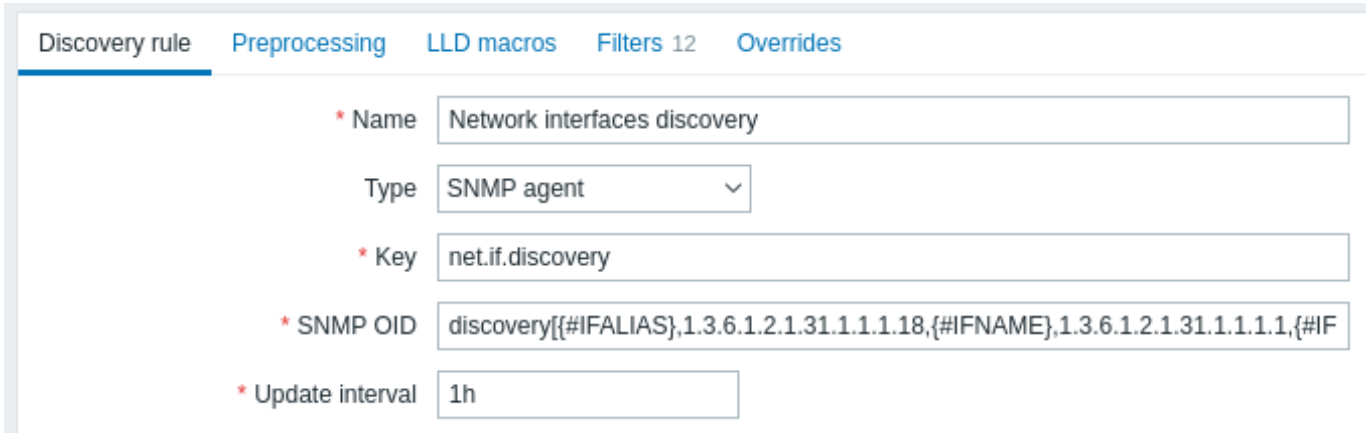
Atšķirībā no failu sistēmas un tīkla saskarnes atklāšanas, vienumam nav obligāti jābūt ar "snmp.discovery" atslēgu — pietiek ar SNMP aģenta vienuma tipu.

Lai konfigurētu atklāšanas kārtulu, rīkojieties šādi:

- Dodieties uz: *Datu ievākšana* → *Veidnes*
- Atbilstošās veidnes rindā noklikšķiniet uz *Atklāšana*



- Ekrāna augšējā labajā stūrī noklikšķiniet uz *Izveidot atklāšanas kārtulu*
- Aizpildiet atklāšanas kārtulas formu ar nepieciešamo informāciju, kā parādīts zemāk esošajā ekrānuzņēmumā



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Atklājamie OID tiek definēti laukā SNMP OID šādā formātā: `discovery[{#MACRO1}, oid1, {#MACRO2}, oid2, …,]`

kur *{#MACRO1}*, *{#MACRO2}* ... ir derīgi lld makrosu nosaukumi, bet *oid1*, *oid2*... ir OID, kas spēj ģenerēt jēgpilnas vērtības šiem makrosiem. Iebūvētais makross *{#SNMPINDEX}*, kas satur atklātā OID indeksu, tiek piemērots atklātajām entītijām. Atklātās entītijas tiek grupētas pēc makrosa *{#SNMPINDEX}* vērtības.

Šeit ir piemērs, kur slēdzim tiek izpildītas vairākas snmpwalk komandas:

```
$ 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
```

Un SNMP OID ir iestatīts šādi: `discovery[{#IFDESCR}, ifDescr, {#IFPHYSADDRESS}, ifPhysAddress]`

Tagad šī kārtula atklās entītijas ar makrosiem {#IFDESCR}, kas iestatīti uz **WAN**, **LAN1** un **LAN2**, makrosiem {#IFPHYSADDRESS}, kas iestatīti uz **8:0:27:90:7a:75**, **8:0:27:90:7a:76** un **8:0:27:2b:af:9e**, un makrosiem {#SNMPINDEX}, kas iestatīti uz atklāto OID indeksiem **1**, **2** un **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"
    }
]
```

Ja entītijai nav norādītā OID, tad atbilstošais makross šai entītijai tiks izlaists. Piemēram, ja mums ir šādi dati:

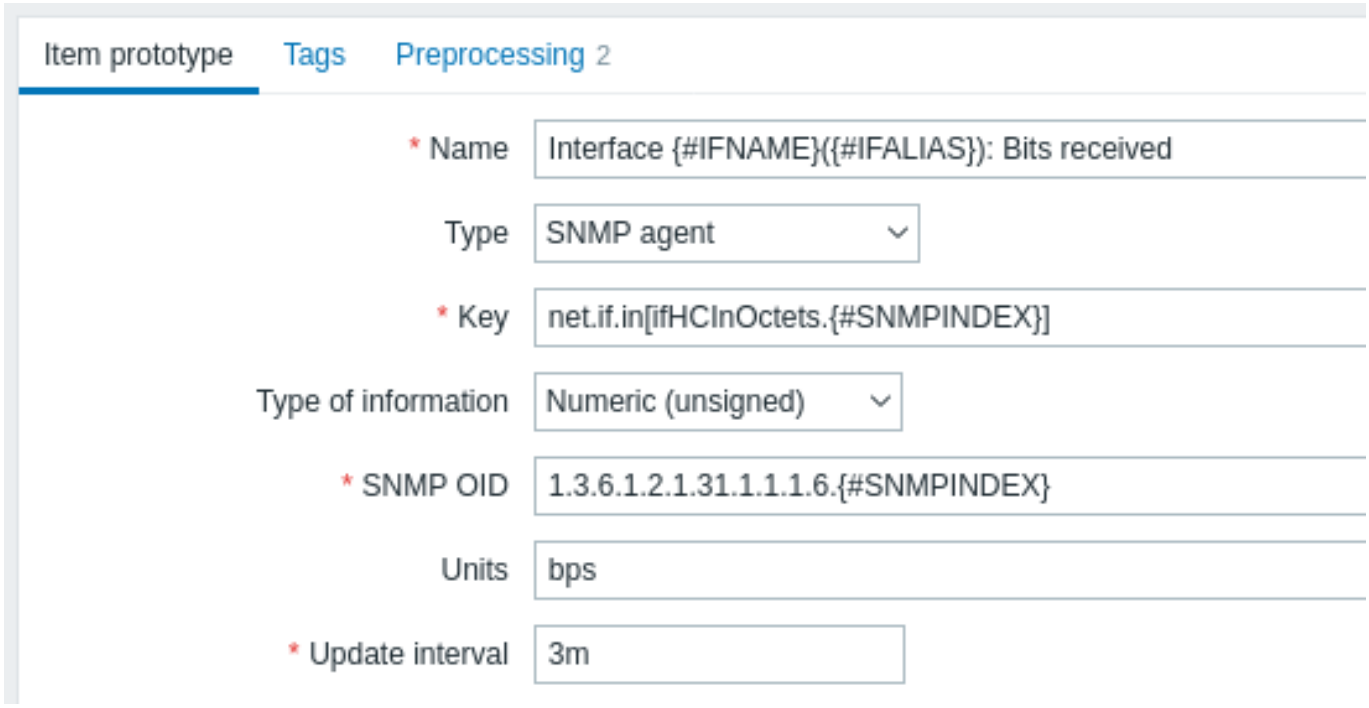
```
ifDescr.1 "Interface #1"
ifDescr.2 "Interface #2"
ifDescr.4 "Interface #4"

ifAlias.1 "eth0"
ifAlias.2 "eth1"
ifAlias.3 "eth2"
ifAlias.5 "eth4"
```

Tad šajā gadījumā SNMP atklāšana `discovery[{#IFDESCR}, ifDescr, {#IFALIAS}, ifAlias]` atgriezīs šādu struktūru:

```
[
    {
        "{#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"
    }
]
```

Vienumu prototipi

Šis ekrānuzņēmums ilustrē, kā mēs varam izmantot šos makrosus vienumu prototipos:



Jūs varat izveidot tik daudz vienumu prototipu, cik nepieciešams:

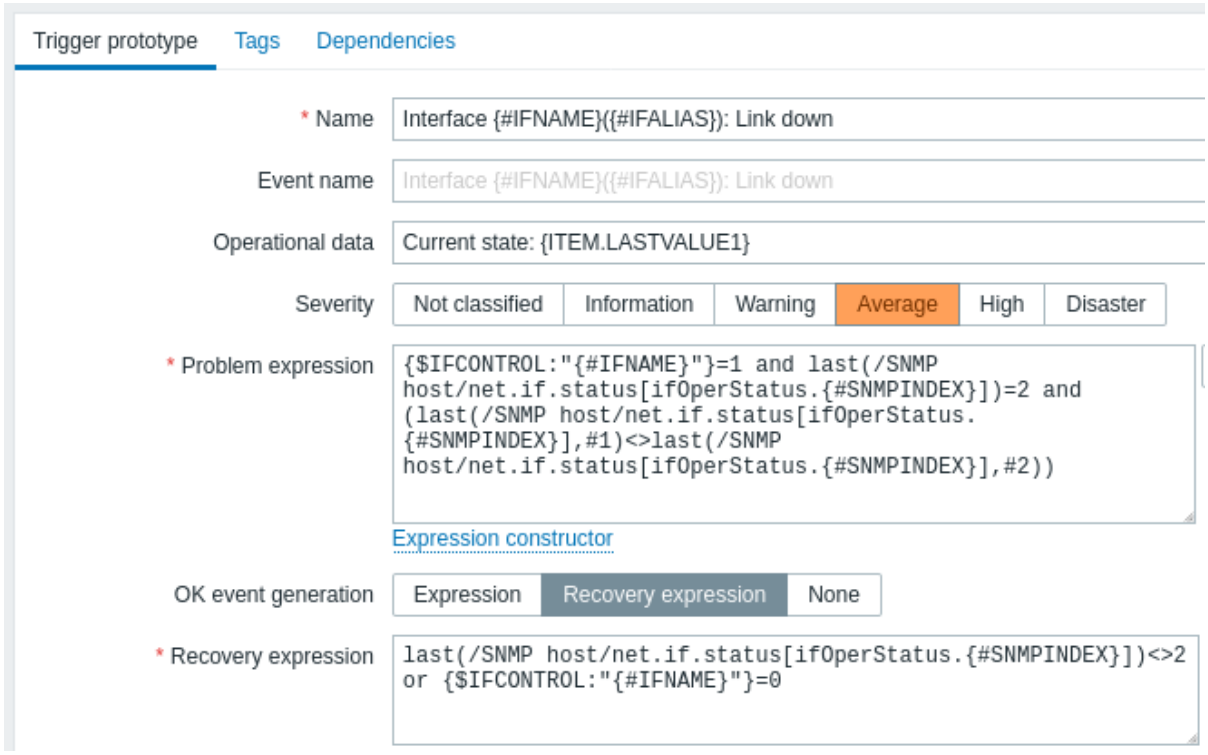
Item prototypes

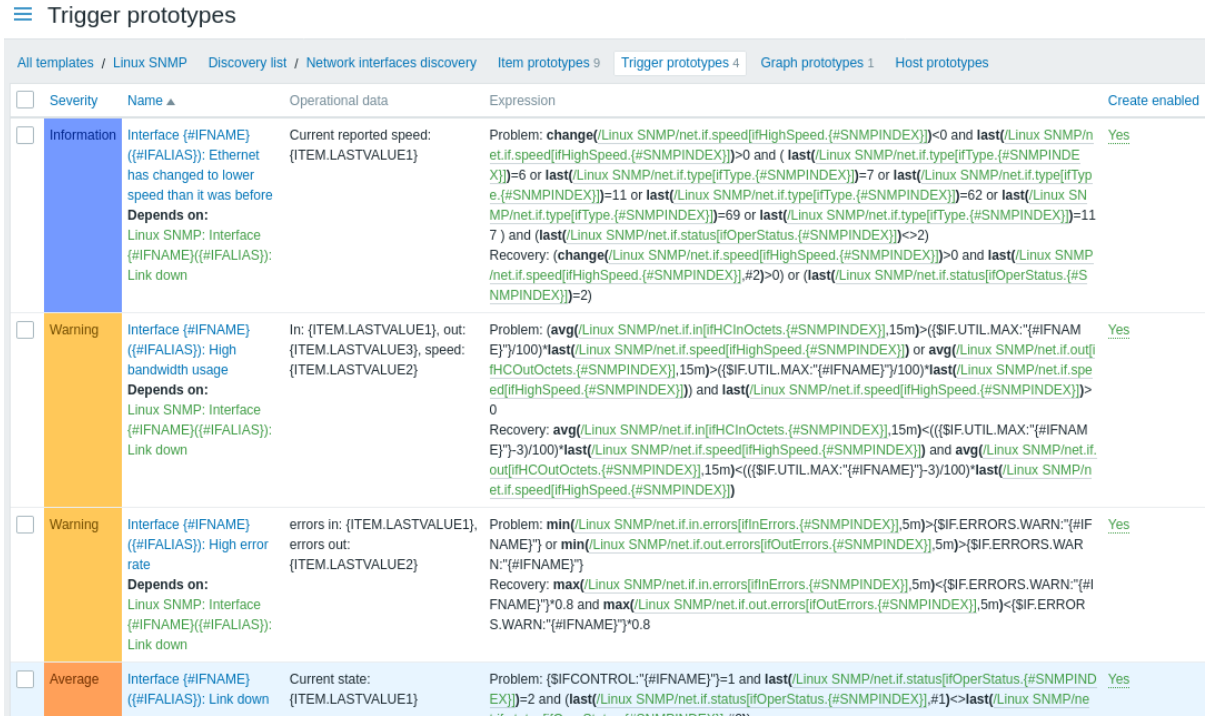
All templates / Linux SNMP Discovery list / Network interfaces discovery Item prototypes 9 Trigger prototypes 4 Graph prototypes 1 Host prototypes

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

Trigeru prototipi

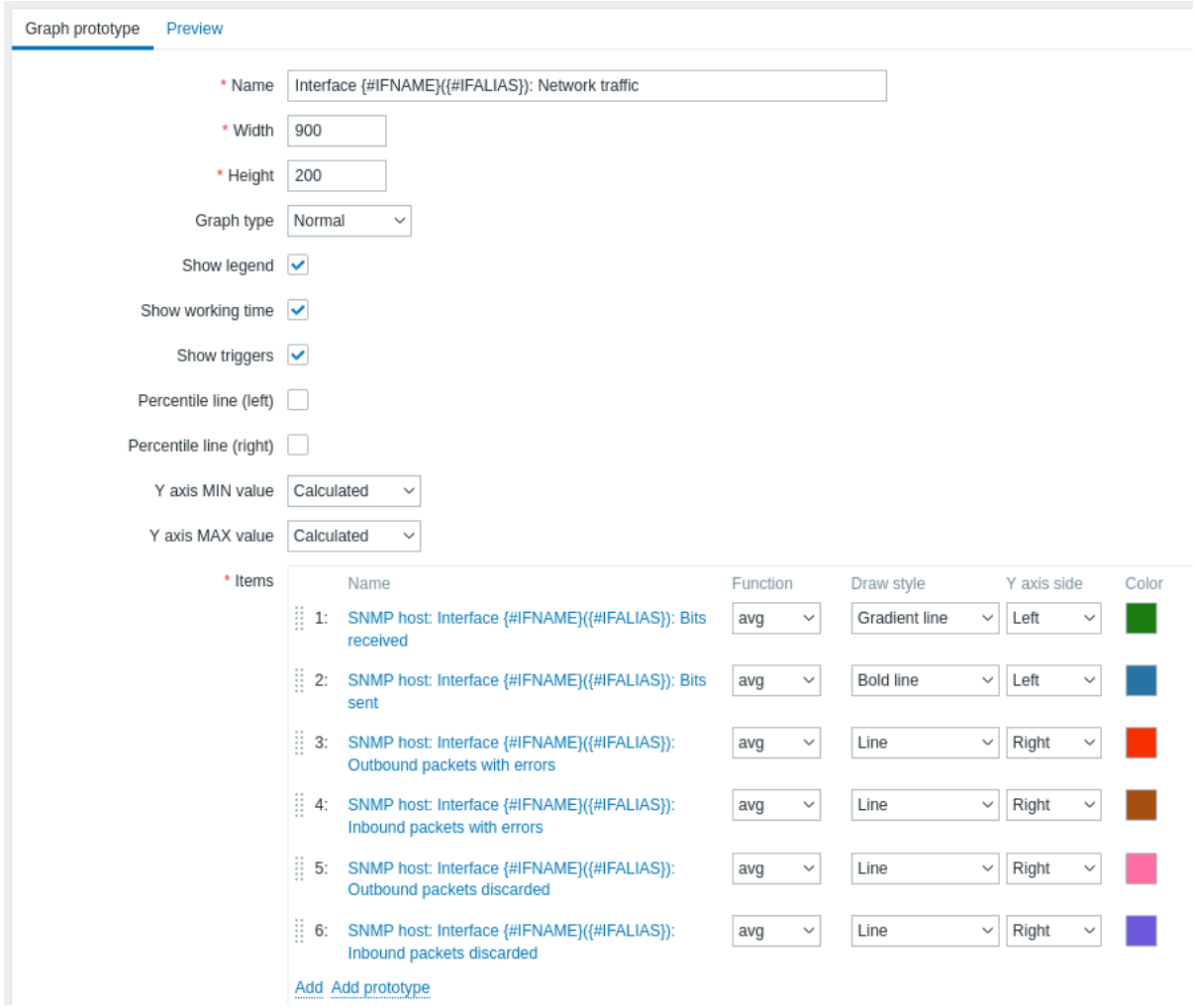
Tālāk redzamais ekrānuzņēmums ilustrē, kā mēs varam izmantot šos makrosus trigeru prototipos:

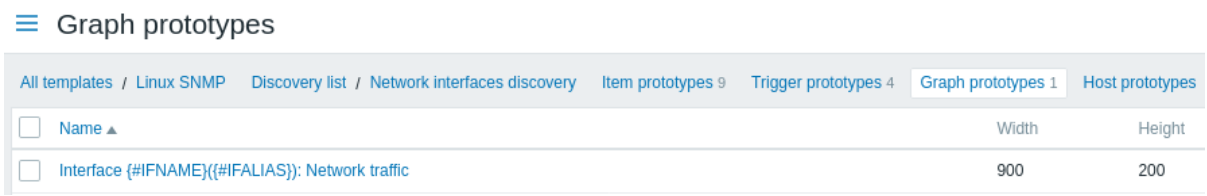




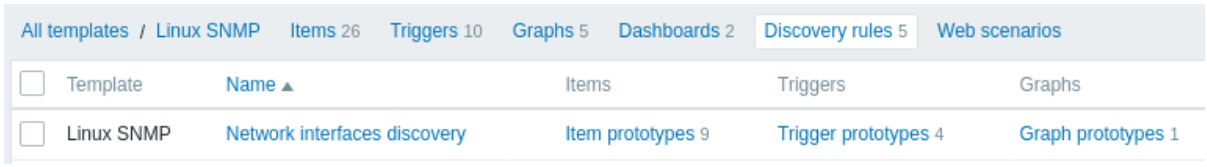
Grafiku prototipi

Šis ekrānuzņēmums ilustrē, kā mēs varam izmantot šos makrosus grafiku prototipos:





Mūsu atklāšanas kārtulas kopsavilkums:



Atklātās entītijas

Kad serveris darbojas, tas izveidos reālus vienumus, trigerus un grafikus, pamatojoties uz vērtībām, ko atgriež SNMP atklāšanas kārtula. Hosta konfigurācijā tiem ir prefikss ar oranžu saiti uz atklāšanas kārtulu, no kuras tie ir izveidoti.

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 JMX objektu atklāšana

Pārskats

Ir iespējams atklāt visus JMX MBean vai MBean atribūtus, vai arī norādīt šablonu šo objektu atklāšanai.

Atklāšanas kārtulas konfigurēšanai ir obligāti jāsaprot atšķirība starp MBean un MBean atribūtiem. MBean ir objekts, kas var attēlot ierīci, lietotni vai jebkuru resursu, kuru nepieciešams pārvaldīt.

Piemēram, ir MBean, kas attēlo tīmekļa serveri. Tā atribūti ir savienojumu skaits, pavedienu skaits, pieprasījuma noildze, HTTP failu kešatmiņa, atmiņas lietojums utt. Izsakot šo domu cilvēkam saprotamā valodā, mēs varam definēt kafijas automātu kā MBean, kam ir šādi uzraugāmie atribūti: ūdens daudzums uz tasi, vidējais ūdens patēriņš noteiktā laika periodā, kafijas pupiņu skaits, kas nepieciešams vienai tasītei, kafijas pupiņu un ūdens papildināšanas laiks utt.

Vienuma atslēga

Atklāšanas kārtulas konfigurācijā laukā *Type* atlasiet **JMX agent**.

JMX objektu atklāšanai tiek atbalstītas divas vienumu atslēgas — jmx.discovery[] un jmx.get[]:

Vienuma atslēga	Atgrieztā vērtība	Parametri	Komentārs
jmx.discovery[<atklāšanas režīms>,<objekta nosaukums>,<unikāls īss apraksts>]	Šis vienums atgriež JSON masīvu ar LLD makro, kas apraksta MBean objektus vai to atribūtus.	**atklāšanas režīms** — viens no šiem: *attributes* (iegūt JMX MBean atribūtus, pēc noklusējuma) vai *beans* (iegūt JMX MBeans) **objekta nosaukums** — objekta nosaukuma raksts (skatiet dokumentāciju), kas identificē iegūstamo MBean nosaukumus (pēc noklusējuma tukšs, iegūstot visus reģistrētos beanus) **unikāls īss apraksts** — unikāls apraksts, kas hostā ļauj izmantot vairākus JMX vienumus ar vienādu atklāšanas režīmu un objekta nosaukumu (neobligāts)	Piemēri: → jmx.discovery — iegūt visus JMX MBean atribūtus → jmx.discovery[beans] — iegūt visus JMX MBeans → jmx.discovery[attributes,"*:type=GarbageCollector,name — iegūt visus atkritumu savācēja atribūtus → jmx.discovery[beans,"*:type=GarbageCollector,name=* — iegūt visus atkritumu savācējus Pastāv daži ierobežojumi attiecībā uz to, kādas MBean īpašības šis vienums var atgriezt, jo makro nosaukumu ģenerēšanā tiek atbalstīts ierobežots rakstzīmju kopums (atbalstītās rakstzīmes var aprakstīt ar šādu regulāro izteiksmi: `A-Z0-9_\.`). Tāpēc, piemēram, lai atklātu MBean īpašības ar vārdu, kas satur defisi, vai rakstzīmes ārpus ASCII, ir jāizmanto `jmx.get[]`.
jmx.get[<atklāšanas režīms>,<objekta nosaukums>,<unikāls īss apraksts>]			

Vienuma atslēga			
	Šis vienums atgriež JSON masīvu ar MBean objektiem vai to atribūtiem. Salīdzinājumā ar `jmx.discovery[]` tas nedefinē LLD makro.	**atklāšanas režīms** — viens no šiem: *attributes* (iegūt JMX MBean atribūtus, pēc noklusējuma) vai *beans* (iegūt JMX MBeans) **objekta nosaukums** — objekta nosaukuma raksts (skatiet dokumentāciju), kas identificē iegūstamo MBean nosaukumus (pēc noklusējuma tukšs, iegūstot visus reģistrētos beanus) **unikāls īss apraksts** — unikāls apraksts, kas hostā ļauj izmantot vairākus JMX vienumus ar vienādu atklāšanas režīmu un objekta nosaukumu (neobligāts)	Izmantojot šo vienumu, ir jādefinē pielāgoti zema līmeņa atklāšanas makro, kas norāda uz vērtībām, kuras no atgrieztā JSON tiek iegūtas, izmantojot JSONPath.

Attention:
Ja netiek nodoti parametri, tiek pieprasīti visi JMX MBean atribūti. Parametru nenorādīšana JMX atklāšanai vai mēģinājums saņemt visus atribūtus plašam diapazonam, piemēram, `*:type=*,name=*`, var radīt iespējamās veiktspējas problēmas.

jmx.discovery izmantošana

Šis vienums atgriež JSON objektu ar zema līmeņa atklāšanas makrosiem, kas apraksta MBean objektus vai atribūtus. Piemēram, MBean atribūtu atklāšanā (pārformatēts skaidrības labad):

```
[
    {
        "{#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"
    }
]
```

MBeans atklāšanā (pārformatēts skaidrības labad):

```
[
    {
        "{#JMXDOMAIN}":"java.lang",
        "{#JMXTYPE}":"GarbageCollector",
        "{#JMXOBJ}":"java.lang:type=GarbageCollector,name=PS Scavenge",
        "{#JMXNAME}":"PS Scavenge"
    }
]
```

Atbalstītie makrosi

Tālāk norādītie makrosi tiek atbalstīti izmantošanai atklāšanas kārtulas filtrā un vienumu, trigeru un grafiku prototipos:

Macro	Description
MBean atribūtu atklāšana	
{#JMXVALUE}	Atribūta vērtība.
{#JMXTYPE}	Atribūta tips.
{#JMXOBJ}	Objekta nosaukums.
{#JMXDESC}	Objekta nosaukums, ieskaitot atribūta nosaukumu.
{#JMXATTR}	Atribūta nosaukums.
MBeans atklāšana	
{#JMXDOMAIN}	MBean domēns. (*Zabbix rezervēts nosaukums*)
{#JMXOBJ}	Objekta nosaukums. (*Zabbix rezervēts nosaukums*)
{#JMX<key property>}	MBean īpašības (piemēram, {#JMXTYPE}, {#JMXNAME}) (skatiet Ierobežojumi zemāk).

Ierobežojumi

Pastāv daži ierobežojumi, kas saistīti ar algoritmu LLD makro nosaukumu izveidei no MBean īpašību nosaukumiem:

- atribūtu nosaukumi tiek pārveidoti uz lielajiem burtiem
- atribūtu nosaukumi tiek ignorēti (LLD makro netiek ģenerēti), ja tie sastāv no LLD makro nosaukumiem neatbalstītām rakstzīmēm. Atbalstītās rakstzīmes var aprakstīt ar šādu regulāro izteiksmi: `A-Z0-9_\.`
- ja atribūts tiek saukts par "obj" vai "domain", tas tiks ignorēts, jo pārklājas ar rezervēto Zabbix īpašību {#JMXOBJ} un {#JMXDOMAIN} vērtībām

Lūdzu, apskatiet šo jmx.discovery piemēru (ar "beans" režīmu). MBean ir definētas šādas īpašības (dažas no tām tiks ignorētas; skatiet zemāk):

```
name=test
  =Type
attributes []=1,2,3
Name=NameOfTheTest
domAin=some
```

JMX atklāšanas rezultātā tiks ģenerēti šādi LLD makro:

- {#JMXDOMAIN} - Zabbix iekšējs, apraksta MBean domēnu
- {#JMXOBJ} - Zabbix iekšējs, apraksta MBean objektu
- {#JMXNAME} - izveidots no īpašības "name"

Ignorētās īpašības ir:

- тип : tās nosaukums satur neatbalstītas rakstzīmes (ne-ASCII)
- attributes[] : tās nosaukums satur neatbalstītas rakstzīmes (kvadrātiekavas netiek atbalstītas)
- Name : tas jau ir definēts (name=test)
- domAin : tas ir Zabbix rezervēts nosaukums

Piemēri

Apskatīsim vēl divus praktiskus LLD kārtulas izveides piemērus, izmantojot MBean. Lai labāk saprastu atšķirību starp LLD kārtulu, kas apkopo MBeans, un LLD kārtulu, kas apkopo MBean atribūtus, lūdzu, aplūkojiet šo tabulu:

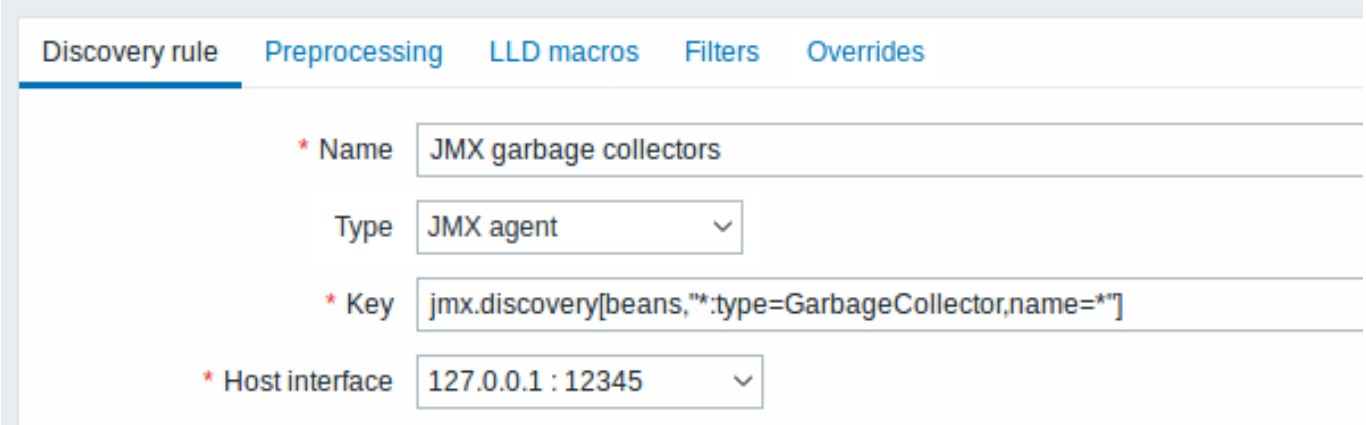
MBean1	**MBean2**	**MBean3**
MBean1Attribute1	MBean2Attribute1	MBean3Attribute1
MBean1Attribute2	MBean2Attribute2	MBean3Attribute2
MBean1Attribute3	MBean2Attribute3	MBean3Attribute3

Piemērs 1: MBean atklāšana

Šis noteikums atgriezīs 3 objektus: kolonnas augšējā rinda: MBean1, MBean2, MBean3.

Plašāku informāciju par objektiem skatiet atbalstīto makrosu tabulā, sadaļā *MBean atklāšana*.

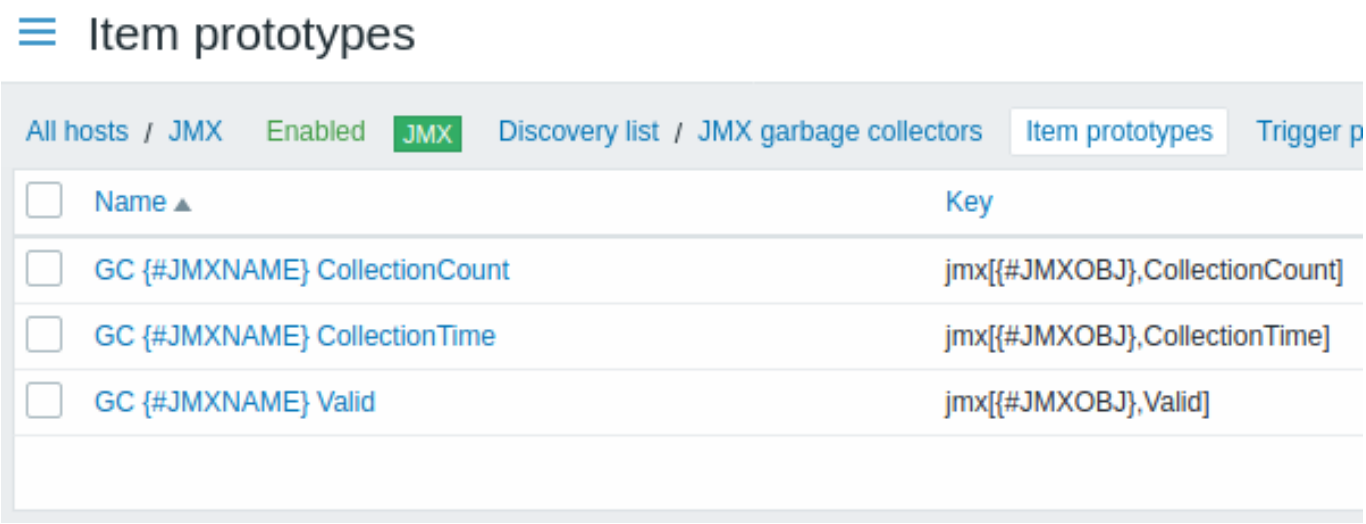
Atklāšanas noteikuma konfigurācija MBean apkopošanai (bez atribūtiem) izskatās šādi:



Šeit izmantotā atslēga:

```
jmx.discovery[beans,"*:type=GarbageCollector,name=*"]
```

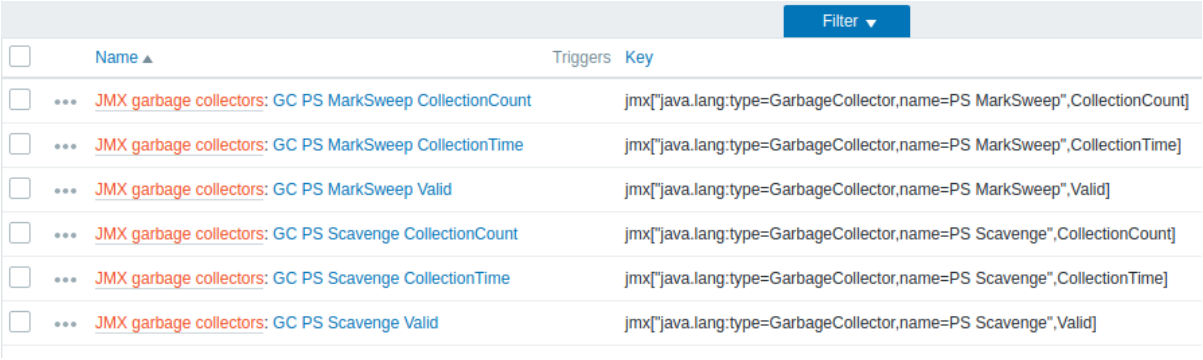
Tiks atklāti visi atkritumu savācēji bez atribūtiem. Tā kā atkritumu savācējiem ir viens un tas pats atribūtu kopums, mēs varam izmantot vēlamos atribūtus vienumu prototipos šādā veidā:



Šeit izmantotās atslēgas:

```
jmx[{#JMXOBJ},CollectionCount]
jmx[{#JMXOBJ},CollectionTime]
jmx[{#JMXOBJ},Valid]
```

LLD atklāšanas noteikuma rezultāts būs aptuveni šāds (vienumi tiek atklāti diviem atkritumu savācējiem):

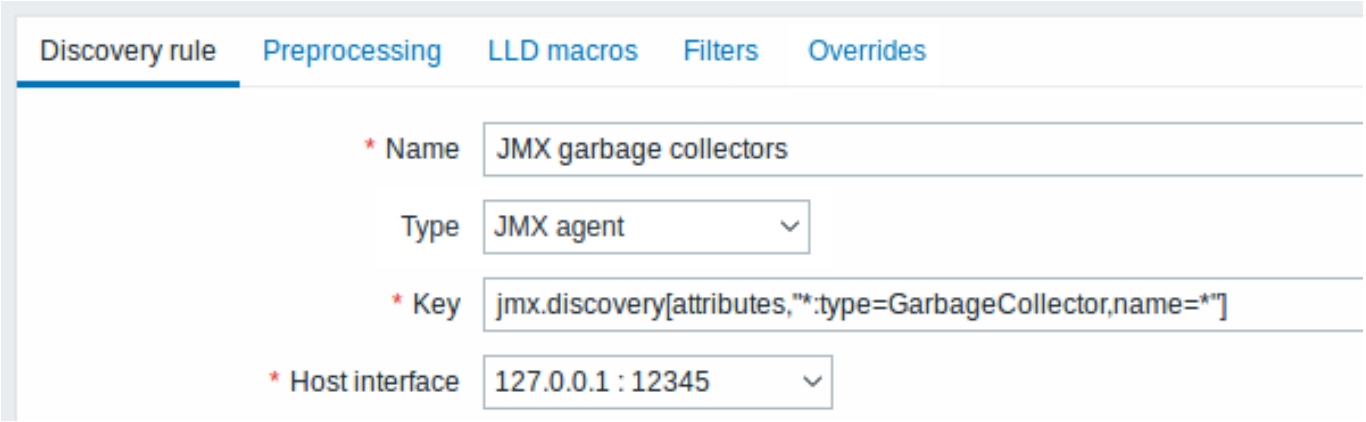


Piemērs 2: MBean atribūtu atklāšana

Šis noteikums atgriezīs 9 objektus ar šādiem laukiem: MBean1Attribute1, MBean2Attribute1, MBean3Attribute1,MBean1Attribute2,MBean2Attr MBean3Attribute2, MBean1Attribute3, MBean2Attribute3, MBean3Attribute3.

Plašāku informāciju par objektiem skatiet atbalstīto makrosu tabulā, sadaļā *MBean atribūtu atklāšana*.

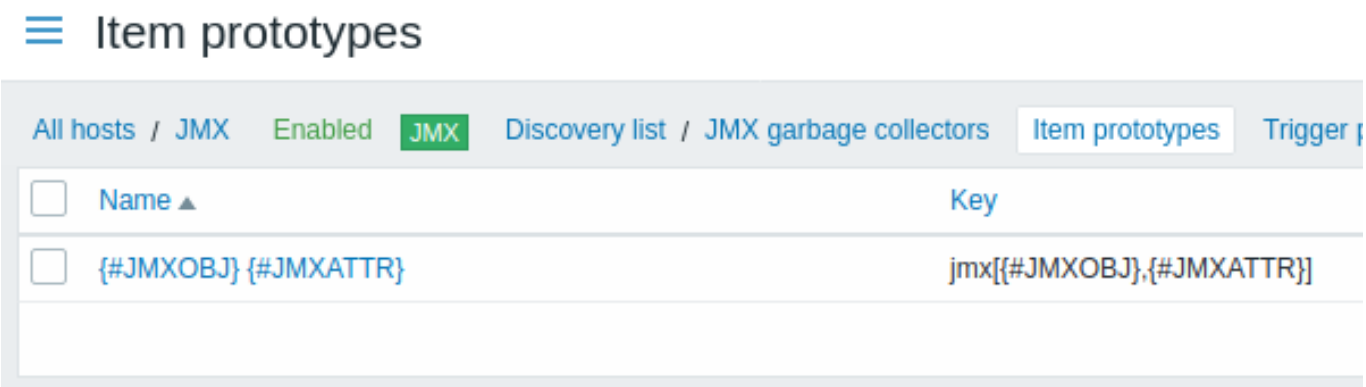
Atklāšanas noteikuma konfigurācija MBean atribūtu vākšanai izskatās šādi:



Šeit izmantotā atslēga:

```
jmx.discovery[attributes,"*:type=GarbageCollector,name=*"]
```

Tiks atklāti visi atkritumu savācēji ar viena vienuma atribūtu.



Šajā konkrētajā gadījumā no prototipa tiks izveidots vienums katram MBean atribūtam. Šīs konfigurācijas galvenais trūkums ir tas, ka trigeru izveide no trigeru prototipiem nav iespējama, jo visiem atribūtiem ir tikai viens vienuma prototips. Tāpēc šo iestatījumu var izmantot datu vākšanai, taču tas nav ieteicams automātiskai uzraudzībai.

`jmx.get` izmantošana

`jmx.get[]` ir līdzīgs `jmx.discovery[]` vienumam, taču tas nepārvērš Java objektu īpašības par zema līmeņa atklāšanas makro nosaukumiem un tādēļ var atgriezt vērtības bez ierobežojumiem, kas ir saistīti ar LLD makro nosaukumu ģenerēšanu, piemēram, defisēm vai rakstzīmēm, kas nav ASCII.

Izmantojot `jmx.get[]` atklāšanai, zema līmeņa atklāšanas makro var definēt atsevišķi atklāšanas kārtulas konfigurācijas pielāgotajā LLD makro cilnē, izmantojot JSONPath, lai norādītu uz nepieciešamajām vērtībām.

MBeans atklāšana

Atklāšanas vienums: `jmx.get[beans,"com.example:type=*,*"]`

Atbilde:

```
[
    {
        "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"
        }
    }
]
```

MBean atribūtu atklāšana

Atklāšanas vienums: `jmx.get[attributes,"com.example:type=*,*"]`

Atbilde:

```
[
    {
        "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 IPMI sensoru atklāšana

Pārskats

Ir iespējams automātiski atklāt IPMI sensorus.

Lai to izdarītu, var izmantot šādu kombināciju:

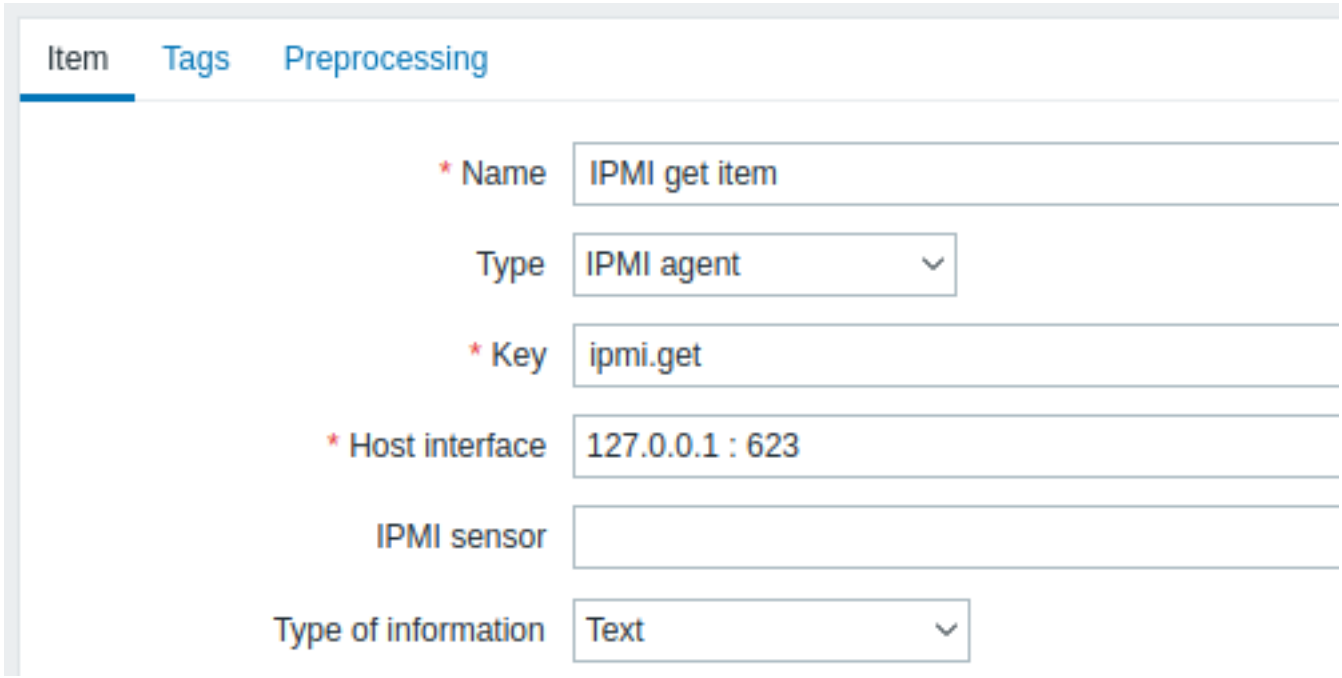
- `ipmi.get` IPMI vienums kā galveno vienumu
- atkarīgo zema līmeņa atklāšanas kārtulu un vienumu prototipus

Konfigurācija

Galvenais vienums

Izveidojiet IPMI vienumu, izmantojot šādu atslēgu:

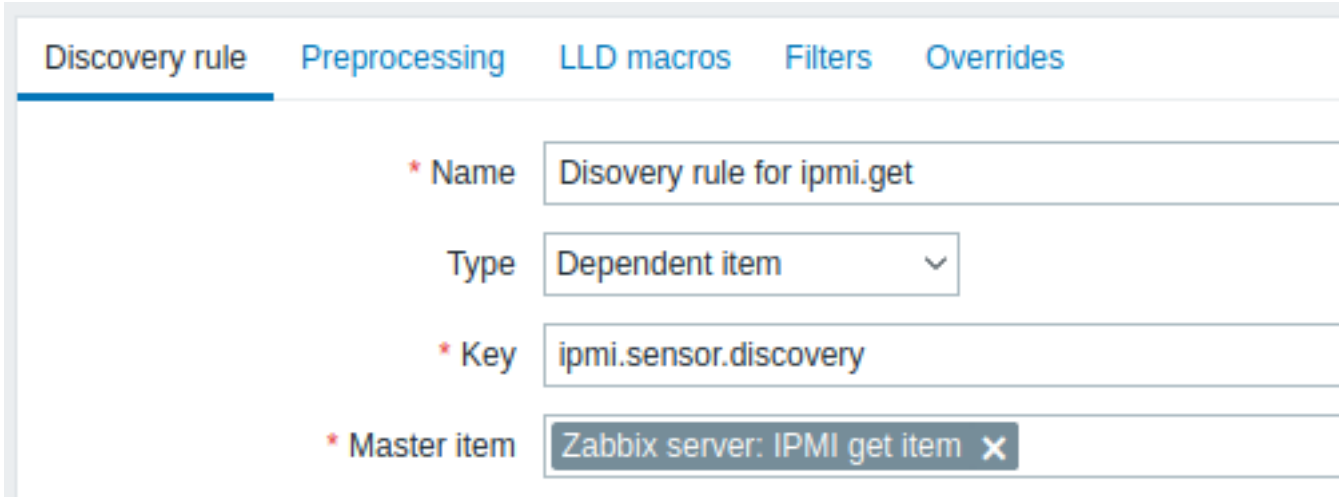
`ipmi.get`



Iestatiet informācijas tipu uz ”Teksts”, lai apstrādātu iespējami lielus JSON datus.

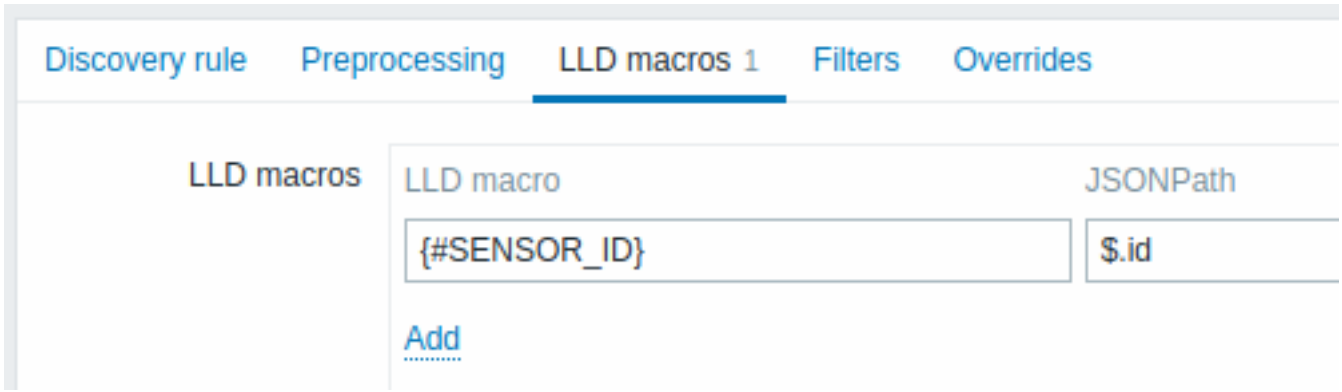
Atkarīgais LLD noteikums

Izveidojiet zema līmeņa atklāšanas noteikumu ar tipu ”Atkarīgais vienums”:



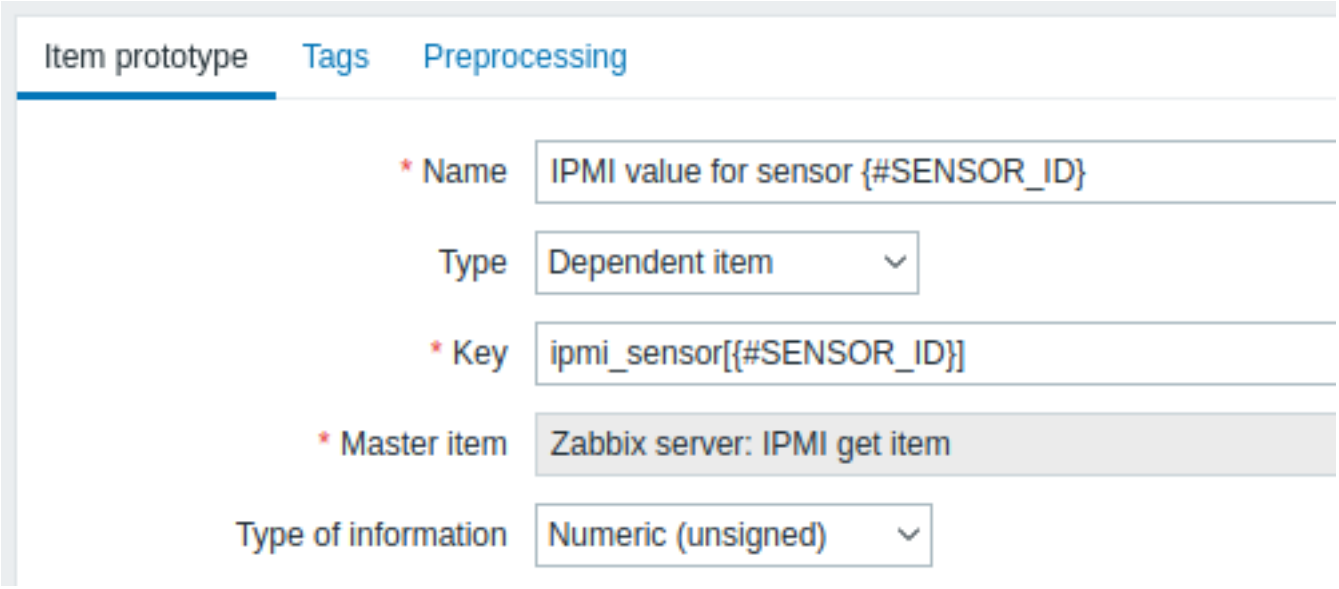
Kā galveno vienumu atlasiet mūsu izveidoto `ipmi.get` vienumu.

Cilnē ”LLD makrosi” definējiet pielāgotu makrosu ar atbilstošo JSONPath:



Atkarīgā vienuma prototips

Izveidojiet šajā LLD kārtulā vienuma prototipu ar tipu ”Dependent item”. Kā galveno vienumu šim prototipam atlasiet iepriekš izveidoto `ipmi.get` vienumu.



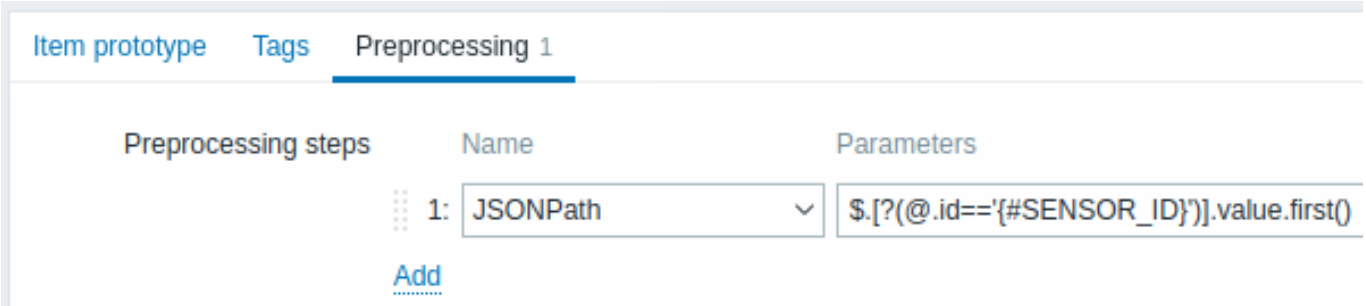
Ņemiet vērā makrosa {#SENSOR_ID} izmantošanu vienuma prototipa nosaukumā un atslēgā:

- *Name*: IPMI vērtība sensoram {#SENSOR_ID}
- *Key*: ipmi_sensor[{#SENSOR_ID}]

Kā informācijas tipu norādiet *Numeric (unsigned)*.

Vienuma prototipa cilnē "Preprocessing" atlasiet JSONPath un izmantojiet šādu JSONPath izteiksmi kā parametru:

```
$.[?(@.id=='{#SENSOR_ID}')].value.first()
```



Kad atklāšana sāksies, tiks izveidots viens vienums katram IPMI sensoram. Šis vienums atgriezīs dotā sensora veselo skaitlisko vērtību.

8 systemd pakalpojumu atklāšana

Pārskats

Ir iespējams ar Zabbix palīdzību atklāt systemd vienības (pēc noklusējuma — pakalpojumus).

Vienuma atslēga

Vienums, ko izmantot atklāšanas kārtulā, ir

```
systemd.unit.discovery
```

Attention:
Šī vienuma atslēga tiek atbalstīta tikai Zabbix aģents 2.

Šis vienums atgriež JSON ar informāciju par systemd vienībām, piemēram:

```
[{
    "{#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}": "Žurnāla ligzda",
    "{#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"
}]
```

Atspējotu systemd vienību atklāšana

Ir iespējams arī atklāt **atspējotas** systemd vienības. Šajā gadījumā iegūtajā JSON tiek atgriezti trīs makrosi:

- {#UNIT.PATH}
- {#UNIT.ACTIVESTATE}
- {#UNIT.UNITFILESTATE}.

Attention:
Lai no prototipiem tiktu izveidoti vienumi un trigeri atspējotām systemd vienībām, pārliecinieties, ka pielāgojat (vai noņemat) aizliedzošos LLD filtrus priekš {#UNIT.ACTIVESTATE} un {#UNIT.UNITFILESTATE}.

Atbalstītie makrosi

Tālāk norādītie makrosi tiek atbalstīti izmantošanai atklāšanas kārtulas filtrā un vienumu, trigeru un grafiku prototipos:

Macro	Description
{#UNIT.NAME}	Primārās vienības nosaukums.
{#UNIT.DESCRIPTION}	Cilvēkam lasāms apraksts.
{#UNIT.LOADSTATE}	Ielādes stāvoklis (t.i., vai vienības fails ir veiksmīgi ielādēts).
{#UNIT.ACTIVESTATE}	Aktīvais stāvoklis (t.i., vai vienība pašlaik ir palaista vai ne).
{#UNIT.SUBSTATE}	Apakšstāvoklis (smalkāk detalizēta aktīvā stāvokļa versija, kas ir specifiska vienības tipam, atšķirībā no aktīvā stāvokļa).
{#UNIT.FOLLOWED}	Vienība, kuras stāvoklim šī vienība seko, ja tāda ir; pretējā gadījumā tukša virkne.
{#UNIT.PATH}	Vienības objekta ceļš.
{#UNIT.JOBID}	Skaitlisks darba ID, ja darba vienībai ir rindā ievietots darbs; pretējā gadījumā 0.
{#UNIT.JOBTYPE}	Darba tips.
{#UNIT.JOBPATH}	Darba objekta ceļš.
{#UNIT.UNITFILESTATE}	Vienības faila instalēšanas stāvoklis.
{#UNIT.SERVICETYPE}	Servisa vienības tips (piemēram, `simple`, `forking`, `oneshot`, `idle` u.c.). Šis makross tiek atgriezts tikai tad, ja vienība ir serviss.

Vienumu prototipi

Vienumu prototipi, kurus var izveidot, pamatojoties uz systemd servisu atklāšanu, ietver, piemēram:

- Vienuma nosaukums: `{#UNIT.DESCRIPTION} active state info`; vienuma atslēga: `systemd.unit.info["{#UNIT.NAME}"]`
- Vienuma nosaukums: `{#UNIT.DESCRIPTION} load state info`; vienuma atslēga: `systemd.unit.info["{#UNIT.NAME}",Lo`

9 Windows servisu atklāšana

Pārskats

Līdzīgi kā tiek atklātas failu sistēmas, ir iespējams atklāt arī Windows pakalpojumus.

Vienuma atslēga

Vienums, ko izmantot atklāšanas kārtulā, ir

`service.discovery`

Atbalstītie makrosi

Tālāk norādītie makrosi tiek atbalstīti izmantošanai atklāšanas kārtulas filtrā un vienumu, trigeru un grafiku prototipos:

Makross	Apraksts
{#SERVICE.NAME}	Servisa nosaukums.
{#SERVICE.DISPLAYNAME}	Attēlotais servisa nosaukums.
{#SERVICE.DESCRIPTION}	Servisa apraksts.
{#SERVICE.STATE}	Servisa stāvokļa skaitliskā vērtība. Sīkāku informāciju skatiet service.info vienumā.
{#SERVICE.STATENAME}	Servisa stāvokļa nosaukums. Sīkāku informāciju skatiet service.info vienumā.
{#SERVICE.PATH}	Servisa ceļš.
{#SERVICE.USER}	Servisa lietotājs.
{#SERVICE.STARTUP}	Servisa palaišanas tipa skaitliskā vērtība. Sīkāku informāciju skatiet service.info vienumā.
{#SERVICE.STARTUPNAME}	Servisa palaišanas tipa nosaukums. Sīkāku informāciju skatiet service.info vienumā.
{#SERVICE.STARTUPTRIGGER}	Skaitliska vērtība, kas norāda, vai servisa palaišanas tipam ir: 0 - nav palaišanas trigeru 1 - ir palaišanas trigeri Tas ir noderīgi, lai atklātu tādus servisa palaišanas tipus kā *Automatic (trigger start)*, *Automatic delayed (trigger start)* un *Manual (trigger start)*.

Pamatojoties uz Windows servisu atklāšanu, jūs varat izveidot vienuma prototipu, piemēram,

`service.info[{#SERVICE.NAME},<param>]`

kur `param` pieņem šādas vērtības: *state*, *displayname*, *path*, *user*, *startup* vai *description*.

Piemēram, lai iegūtu servisa attēlojamo nosaukumu, varat izmantot vienumu "service.info[{#SERVICE.NAME},displayname]". Ja `param` vērtība nav norādīta ("service.info[{#SERVICE.NAME}]"), tiek izmantots noklusējuma parametrs *state*.

10 Windows veiktspējas skaitītāju instanču atklāšana

Pārskats

Ir iespējams atklāt Windows veiktspējas skaitītāju objektu instances. Tas ir noderīgi vairāku instanču veiktspējas skaitītājiem.

Vienuma atslēga

Lai konfigurētu atklāšanas kārtulu, izmantojiet šādu vienumu:

- `perf_instance.discovery[object]`

Ņemiet vērā, ka objekta nosaukums var būt lokalizēts. Piemēram:

```
perf_instance.discovery[Processor] # Objekta nosaukums ir angļu valodā.
perf_instance.discovery[Processador] # Objekta nosaukums ir portugāļu valodā.
```

Alternatīvi, lai nodrošinātu, ka objekta nosaukums tiek norādīts angļu valodā neatkarīgi no OS lokalizācijas, izmantojiet šādu vienumu:

- `perf_instance_en.discovery[object]`

Piemēram:

```
perf_instance_en.discovery[Processor]
perf_instance_en.discovery[Memory]
```

Atbalstītie makrosi

Atklāšanas process atgriezīs visas norādītā objekta instances makrosā {#INSTANCE}:

```
[
    {"{#INSTANCE}":"0"},
    {"{#INSTANCE}":"1"},
    {"{#INSTANCE}":"_Total"}
]
```

Šo makrosu var izmantot `perf_counter[]` un `perf_counter_en[]` vienumu prototipos.

Piemēram, ja vienuma atslēga, kas izmantota atklāšanas kārtulā, ir `perf_instance.discovery[Processor]`, varat izveidot šādu vienuma prototipu:

```
perf_counter["\Processor({#INSTANCE})\% Processor Time"]
```

Piezīme:

- Ja norādītais objekts netiek atrasts vai neatbalsta mainīgas instances, atklāšanas vienums kļūs par NOTSUPPORTED.
- Ja norādītais objekts atbalsta mainīgas instances, bet pašlaik tam nav nevienas instances, tiks atgriezts tukšs JSON masīvs.
- Dublētas instances tiks izlaistas.

11 Atklāšana, izmantojot WMI vaicājumus

Pārskats

WMI ir jaudīga saskarne sistēmā Windows, ko var izmantot, lai iegūtu dažādu informāciju par Windows komponentiem, pakalpojumiem, stāvokli un instalēto programmatūru.

To var izmantot fizisko disku atklāšanai un to veiktspējas datu vākšanai, tīkla saskarņu atklāšanai, Hyper-V viesu atklāšanai, Windows pakalpojumu uzraudzībai un daudz kam citam Windows OS.

Šāda veida zema līmeņa atklāšana tiek veikta, izmantojot WQL vaicājumus, kuru rezultāti tiek automātiski pārveidoti par JSON objektu, kas ir piemērots zema līmeņa atklāšanai.

Vienuma atslēga

Vienums, ko izmantot atklāšanas kārtulā, ir

`wmi.getall[<namespace>,<query>]`

Šis vienums pārveido vaicājuma rezultātu JSON masīvā. Piemēram:

`select * from Win32_DiskDrive where Name like '%PHYSICALDRIVE%'`

var atgriezt kaut ko līdzīgu šim:

```
[
    {
        "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"
    }
]
```


Zema līmeņa atklāšanas makrosi

Lai gan atgrieztajā JSON netiek izveidoti nekādi zema līmeņa atklāšanas makrosi, lietotājs var definēt šos makrosus kā papildu soli, izmantojot pielāgota LLD makrosa funkcionalitāti ar JSONPath, kas norāda uz atklātajām vērtībām atgrieztajā JSON.

Pēc tam šos makrosus var izmantot, lai izveidotu vienuma, trigera u.c. prototipus.

12 Atklāšana, izmantojot ODBC SQL vaicājumus

Pārskats

Šis zema līmeņa atklāšanas veids tiek veikts, izmantojot SQL vaicājumus, kuru rezultāti tiek automātiski pārveidoti par JSON objektu, kas ir piemērots zema līmeņa atklāšanai.

Vienuma atslēga

SQL vaicājumi tiek izpildīti, izmantojot ”Database monitor” vienuma tipu. Tādēļ lielākā daļa norādījumu lapā ODBC monitoring ir piemērojami, lai iegūtu strādājošu ”Database monitor” atklāšanas kārtulu.

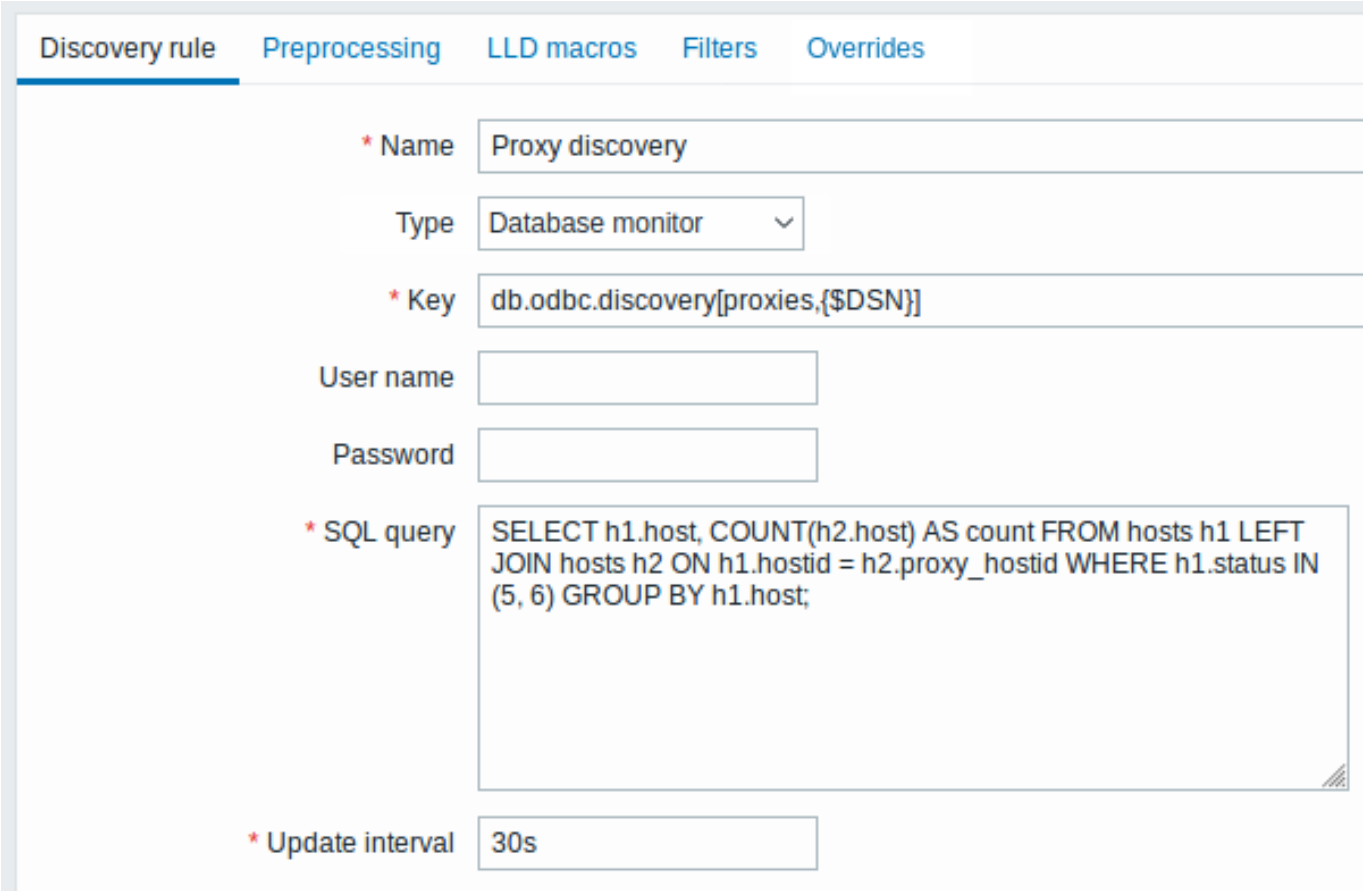
”Database monitor” atklāšanas kārtulās var izmantot divas vienumu atslēgas:

- **db.odbc.discovery**[<unique short description>,<dsn>,<connection string>] - šis vienums pārveido SQL vaicājuma rezultātu JSON masīvā, pārvēršot kolonnu nosaukumus no vaicājuma rezultāta par zema līmeņa atklāšanas makrosu nosaukumiem, kas ir sapāroti ar atklāto lauku vērtībām. Šos makrosus var izmantot, veidojot vienumu, trigeru u.c. prototipus. Skatiet arī: Using db.odbc.discovery.
- **db.odbc.get**[<unique short description>,<dsn>,<connection string>] - šis vienums pārveido SQL vaicājuma rezultātu JSON masīvā, saglabājot sākotnējos kolonnu nosaukumus no vaicājuma rezultāta kā lauka nosaukumu JSON formātā, sapārotu ar atklātajām vērtībām. Salīdzinot ar `db.odbc.discovery[]`, šis vienums neatgrieztajā JSON neveido zema līmeņa atklāšanas makrosus, tādēļ nav nepieciešams pārbaudīt, vai kolonnu nosaukumi var būt derīgi makrosu nosaukumi. Zema līmeņa atklāšanas makrosus var definēt kā papildu soli pēc nepieciešamības, izmantojot custom LLD macro funkcionalitāti ar JSONPath, kas norāda uz atklātajām vērtībām atgrieztajā JSON. Skatiet arī: Using db.odbc.get.

db.odbc.discovery izmantošana

Tālāk redzamais piemērs parāda, kā SQL vaicājums tiek pārveidots par JSON, izmantojot Zabbix starpniekserveru zema līmeņa atklāšanu, pamatojoties uz ODBC vaicājumu Zabbix datubāzē. Tas ir noderīgi, lai automātiski izveidotu ”zabbix[proxy,<name>,lastaccess]” iekšējos vienumus, ar kuriem uzraudzīt, kuri starpniekserveri ir aktīvi.

Sāciet ar atklāšanas kārtulas konfigurāciju:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Šeit tiek izmantots šāds tiešs vaicājums Zabbix datubāzē, lai atlasītu visus Zabbix starpniekserverus kopā ar to uzraudzīto hostu skaitu. Hostu skaitu var izmantot, piemēram, lai atfiltrētu tukšus starpniekserverus:

```
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)
```

"db.odbc.discovery[,{$DSN}]" vienuma iekšējās darbības rezultātā šī vaicājuma rezultāts tiek automātiski pārveidots šādā JSON formātā:

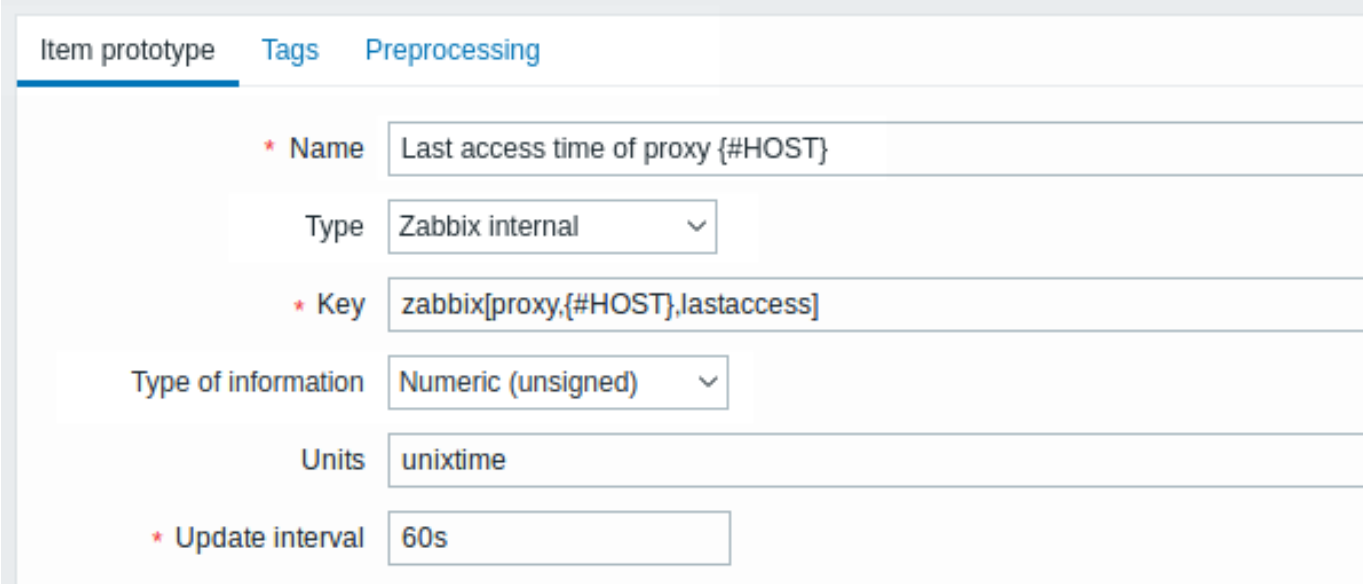
```
[
    {
        "{#HOST}": "Japan 1",
        "{#COUNT}": "5"
    },
    {
        "{#HOST}": "Japan 2",
        "{#COUNT}": "12"
    },
    {
        "{#HOST}": "Latvia",
        "{#COUNT}": "3"
    }
]
```

Var redzēt, ka kolonnu nosaukumi kļūst par makrosu nosaukumiem, bet atlasītās rindas kļūst par šo makrosu vērtībām.

Note:
Ja nav uzreiz skaidrs, kā kolonnas nosaukums tiks pārveidots par makrosa nosaukumu, ieteicams izmantot kolonnu aizstājvārdus, kā iepriekš minētajā piemērā "COUNT(h2.host) AS count".
Ja kolonnas nosaukumu nevar pārveidot par derīgu makrosa nosaukumu, atklāšanas kārtula kļūst neatbalstīta, un kļūdas ziņojumā tiek norādīts problemātiskās kolonnas numurs. Ja nepieciešama papildu palīdzība, iegūtie kolonnu nosaukumi ir pieejami zem DebugLevel=4 Zabbix serveris žurnālfailā:

```
$ 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 I
 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
```

Tagad, kad saprotam, kā SQL vaicājums tiek pārveidots par JSON objektu, varam izmantot {#HOST} makrosu vienumu prototipos:



Kad atklāšana ir veikta, katram starpniekserverim tiks izveidots vienums:

db.odbc.get izmantošana

Izmantojot db.odbc.get[,{$DSN}] un šādu SQL piemēru:

```
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)
```

tiks atgriezts šāds JSON:

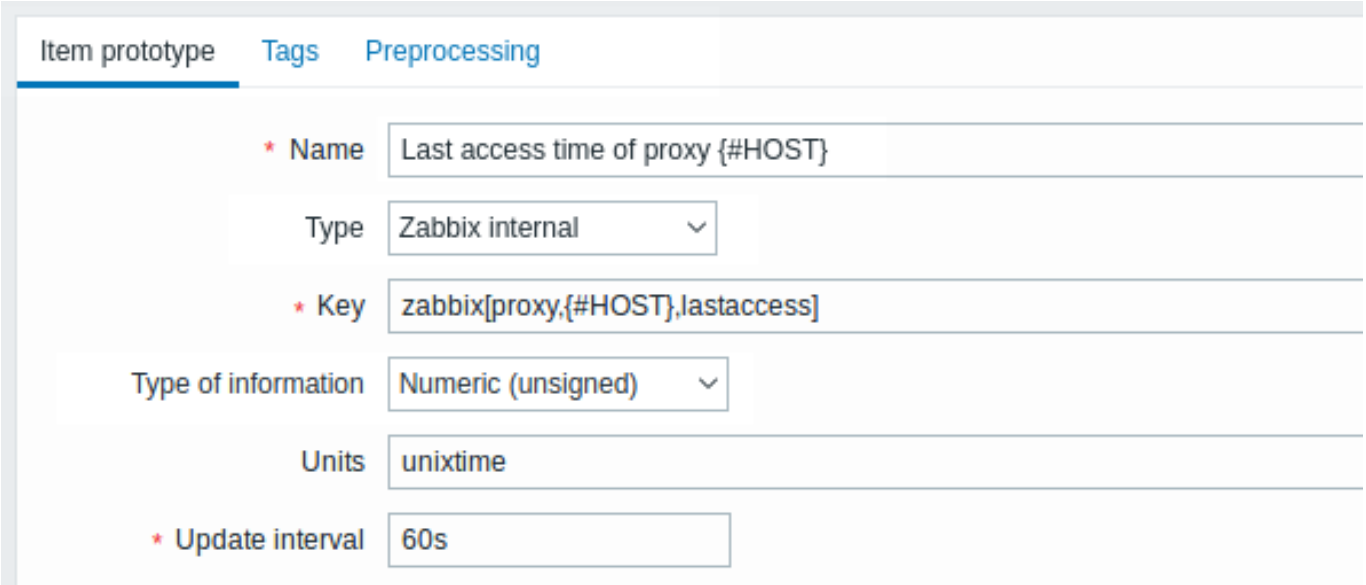
```
[
    {
```

```
        "host": "Japan 1",
        "count": "5"
    },
    {
        "host": "Japan 2",
        "count": "12"
    },
    {
        "host": "Latvia",
        "count": "3"
    }
]
```

Kā redzat, tur nav zema līmeņa atklāšanas makrosu. Tomēr pielāgotus zema līmeņa atklāšanas makrosus var izveidot atklāšanas kārtulas cilnē LLD macros, izmantojot JSONPath, piemēram:

```
{#HOST} → $.host
```

Tagad šo {#HOST} makrosu var izmantot vienumu prototipos:



13 Atklāšana, izmantojot Prometheus datus

Pārskats

Datus, kas sniegti Prometheus rindu formātā, var izmantot zema līmeņa atklāšanai.

Skatiet Prometheus checks, lai uzzinātu vairāk par to, kā Zabbix ir ieviesta Prometheus datu vaicāšana.

Konfigurācija

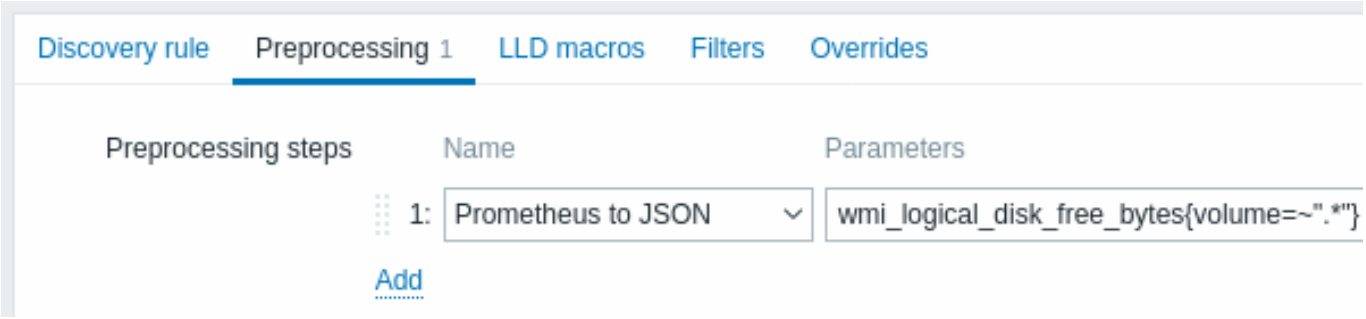
Zema līmeņa atklāšanas kārtula jāizveido kā atkarīgais vienums HTTP galvenajam vienumam, kas apkopo Prometheus datus.

Prometheus uz JSON

Atklāšanas kārtulā atveriet cilni **Preprocessing** un atlasiet priekšapstrādes opciju *Prometheus to JSON*. Atklāšanai ir nepieciešami dati JSON formātā, un priekšapstrādes opcija *Prometheus to JSON* atgriezīs tieši to ar šādiem atribūtiem:

- metrikas nosaukums
- metrikas vērtība
- help (ja ir)
- type (ja ir)
- labels (ja ir)
- neapstrādāta rinda

Piemēram, vaicājot `wmi_logical_disk_free_bytes`:



no šīm Prometheus rindām:

```
# 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
```

tiks atgriezts:

```
[
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "C:"
         },
        "value": "3.5180249088e+11",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"C:\"} 3.5180249088e+11"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "D:"
         },
        "value": "2.627731456e+09",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"D:\"} 2.627731456e+09"
    },
    {
        "name": "wmi_logical_disk_free_bytes",
        "help": "Free space in bytes (LogicalDisk.PercentFreeSpace)",
        "type": "gauge",
        "labels": {
            "volume": "HarddiskVolume4"
         },
        "value": "4.59276288e+08",
        "line_raw": "wmi_logical_disk_free_bytes{volume=\"HarddiskVolume4\"} 4.59276288e+08"
    }
]
```

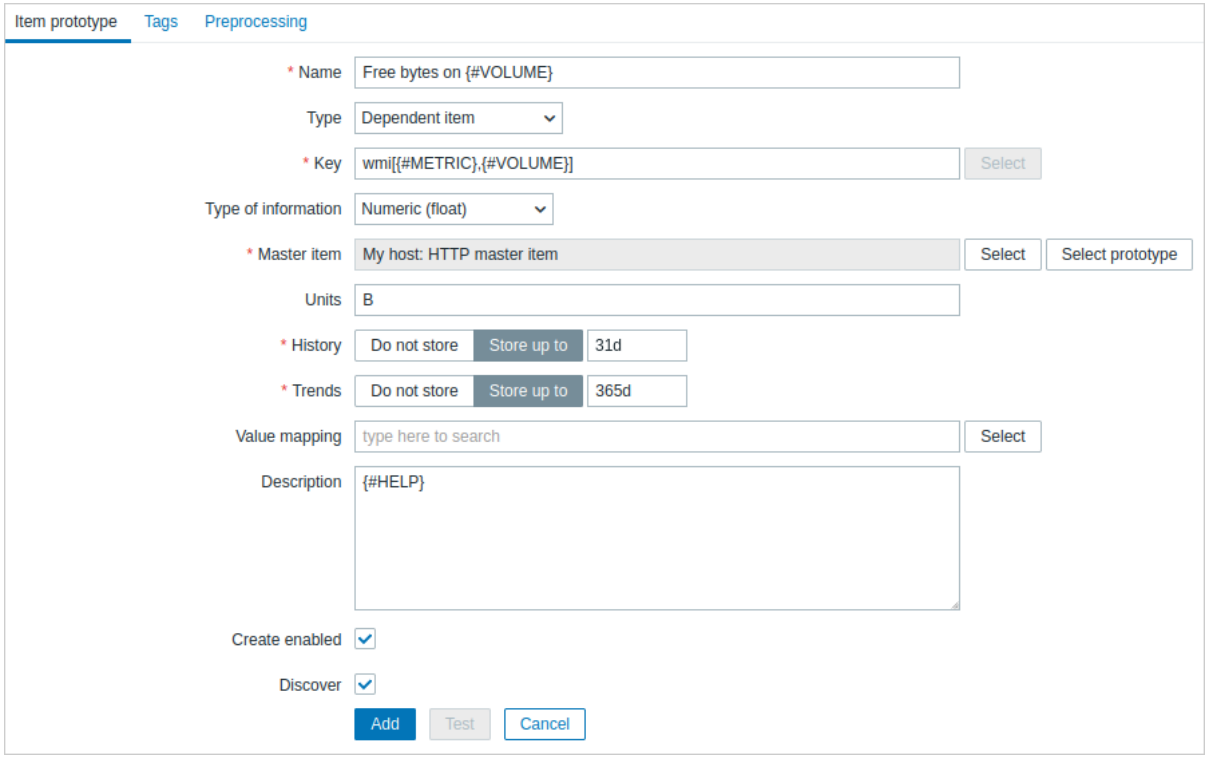
LLD makrosu kartēšana

Pēc tam jums jādodas uz LLD makrosu cilni un jāizveido šādas kartēšanas:

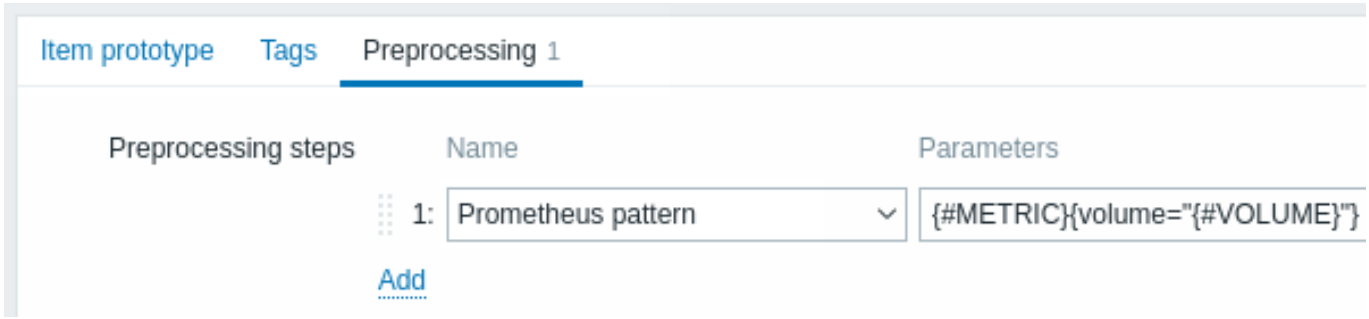
```
{#VOLUME}=$.labels['volume']
{#METRIC}=$['name']
{#HELP}=$['help']
```

Vienuma prototips

Iespējams, vēlēsieties izveidot vienuma prototipu šādi:



ar priekšapstrādes opcijām:



14 Bloku ierīču atklāšana

Līdzīgi kā tiek atklātas failu sistēmas, ir iespējams atklāt arī bloku ierīces un to tipu.

Vienuma atslēga

Vienuma atslēga, ko izmantot atklāšanas kārtulā, ir

`vfs.dev.discovery`

Šis vienums tiek atbalstīts tikai Linux platformās.

Jūs varat izveidot atklāšanas kārtulas, izmantojot šo atklāšanas vienumu un:

- filtru: **{#DEVNAME} atbilst** `sd[\D]$` - lai atklātu ierīces ar nosaukumu "sd0", "sd1", "sd2", ...
- filtru: **{#DEVTYPE} atbilst** `disk` **AND {#DEVNAME} neatbilst** `^loop.*` - lai atklātu diska tipa ierīces, kuru nosaukums nesākas ar "loop"

Atbalstītie makrosi

Šī atklāšanas atslēga atgriež divus makrosus — {#DEVNAME} un {#DEVTYPE}, kas attiecīgi identificē bloka ierīces nosaukumu un tipu, piemēram:

```
[
   {
      "{#DEVNAME}":"loop1",
      "{#DEVTYPE}":"disk"
   },
   {
      "{#DEVNAME}":"dm-0",
```

```
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda",
        "{#DEVTYPE}":"disk"
    },
    {
        "{#DEVNAME}":"sda1",
        "{#DEVTYPE}":"partition"
    }
]
```

Bloka ierīču atklāšana ļauj izmantot `vfs.dev.read[]` un `vfs.dev.write[]` vienumus, lai izveidotu vienumu prototipus, izmantojot makrosu {#DEVNAME}, piemēram:

- "vfs.dev.read[{#DEVNAME},sps]"
- "vfs.dev.write[{#DEVNAME},sps]"

{#DEVTYPE} ir paredzēts ierīču filtrēšanai.

15 Hostu saskarņu atklāšana Zabbix

Pārskats

Ir iespējams atklāt visas saskarnes, kas ir konfigurētas Zabbix lietotāja saskarnē hostam.

Vienuma atslēga

Vienums, ko izmantot atklāšanas kārtulā, ir Zabbix iekšējais vienums:

`zabbix[host,discovery,interfaces]`

Šis vienums atgriež JSON ar interfeisu aprakstu, tostarp:

- IP adrese/DNS hosta nosaukums (atkarībā no hosta iestatījuma "Connect to")
- Porta numurs
- Interfeisa tips (Zabbix aģents, SNMP, JMX, IPMI)
- Vai tas ir noklusējuma interfeiss vai nē
- Vai ir iespējota lielapjoma pieprasījumu funkcija — tikai SNMP interfeisiem.

Piemēram:

`[{"{#IF.CONN}":"192.168.3.1","{#IF.IP}":"192.168.3.1","{#IF.DNS}":"","{#IF.PORT}":"10050","{#IF.TYPE}":"AG`

Ja ir vairāki interfeisi, to ieraksti JSON tiek sakārtoti pēc:

- Interfeisa tipa,
- Noklusējuma — noklusējuma interfeiss tiek novietots pirms interfeisiem, kas nav noklusējuma,
- Interfeisa ID (augošā secībā).

Atbalstītie makrosi

Tālāk norādītie makrosi tiek atbalstīti izmantošanai atklāšanas kārtulas filtrā un vienumu, trigeru un grafiku prototipos:

Macro	Description
{#IF.CONN}	Saskarnes IP adrese vai DNS hosta nosaukums.
{#IF.IP}	Saskarnes IP adrese.
{#IF.DNS}	Saskarnes DNS hosta nosaukums.
{#IF.PORT}	Saskarnes porta numurs.
{#IF.TYPE}	Saskarnes tips ("AGENT", "SNMP", "JMX" vai "IPMI").
{#IF.DEFAULT}	Saskarnes noklusējuma statuss: 0 - nav noklusējuma saskarne 1 - noklusējuma saskarne
{#IF.SNMP.BULK}	Saskarnes SNMP lielapjoma apstrādes statuss: 0 - atspējots 1 - iespējots Šis makross tiek atgriezts tikai tad, ja saskarnes tips ir "SNMP".

8 Pielāgoti LLD noteikumi

Pārskats

Ir iespējams arī izveidot pilnībā pielāgotu LLD kārtulu, atklājot jebkāda veida entītijas — piemēram, datubāzes datubāzu serverī.

Lai to izdarītu, jāizveido pielāgots vienums, kas atgriež JSON virkni, norādot atrastos objektus un pēc izvēles — dažas to īpašības. Makrosu skaits katrai entītijai nav ierobežots — kamēr iebūvētās atklāšanas kārtulas atgriež vienu vai divus makrosus (piemēram, divus datņu sistēmu atklāšanai), ir iespējams atgriezt vairāk.

Piemērs

Nepieciešamo JSON virknes formātu vislabāk ilustrēt ar piemēru. Pieņemsim, ka jūs izmantojat vecu Zabbix 1.8 aģentu (tādu, kas neatbalsta vfs.fs.discovery atslēgu), bet jums joprojām ir nepieciešams atklāt failu sistēmas. Šeit ir vienkāršs Perl skripts Linux videi, kas atklāj piemontētās failu sistēmas un izvada tās JSON formātā, iekļaujot gan failu sistēmas nosaukumu, gan tipu. Viens no veidiem, kā to izmantot, būtu kā UserParameter ar atslēgu "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:
Atļautie simboli LLD makrosu nosaukumiem ir **0-9** , **A-Z** , **_** , **.** Nosaukumos mazie burti netiek atbalstīti.

Tālāk ir parādīts tā izvades piemērs (pārformatēts skaidrības labad). JSON pielāgotajām atklāšanas pārbaudēm ir jāatbilst tam pašam formātam.

```
[
    { "{#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"  }
]
```

Iepriekšējā piemērā ir nepieciešams, lai atslēgas atbilstu prototipos izmantotajiem LLD makrosu nosaukumiem; alternatīva ir izvilkt LLD makrosu vērtības, izmantojot JSONPath {#FSNAME} → $.fsname un {#FSTYPE} → $.fstype, tādējādi padarot iespējamu šādu skriptu:

```
####!/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";
```

Tālāk ir parādīts tā izvades piemērs (pārformatēts skaidrības labad).

```
[
    { "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"  }
]
```

Pēc tam atklāšanas kārtulas *Filter* laukā jūs varat norādīt "{#FSTYPE}" kā makrosu un "rootfs|ext3" kā regulāro izteiksmi.

Note:
Jums nav obligāti jāizmanto tādi makrosu nosaukumi kā FSNAME/FSTYPE ar pielāgotām LLD kārtulām; jūs varat brīvi izmantot jebkurus nosaukumus pēc savas izvēles. Ja tiek izmantots JSONPath, tad LLD rinda būs masīva elements, kas var būt objekts, bet tas var būt arī cits masīvs vai vērtība.

Ņemiet vērā, ka, izmantojot user parameter, atgrieztās vērtības lielums ir ierobežots līdz 16 MB. Plašāku informāciju skatiet sadaļā datu ierobežojumi LLD atgrieztajām vērtībām.

14 Izkliedētā uzraudzība

Pārskats Zabbix nodrošina efektīvu un uzticamu veidu, kā uzraudzīt izkliedētu IT infrastruktūru, izmantojot Zabbix starpniekserverus.

Starpniekserverus var izmantot, lai lokāli vāktu datus centralizēta Zabbix servera vārdā un pēc tam nosūtītu šos datus serverim.

Starpniekservera iespējas

Izvēloties izmantot vai neizmantot starpniekserveri, jāņem vērā vairāki apsvērumi.

	Starpniekserveris
Viegls	**Jā**
GUI	Nē
Darbojas neatkarīgi	**Jā**
Vienkārša uzturēšana	**Jā**
Automātiska DB izveide	**Jā**[1]

	Starpniekserveris
Lokālā administrēšana	Nē
Piemērots iegultajai aparatūrai	**Jā**
Vienvirziena TCP savienojumi	**Jā**
Centralizēta konfigurācija	**Jā**
Ģenerē paziņojumus	Nē

[1] Automātiskās DB izveides funkcija darbojas tikai ar SQLite. Citām atbalstītajām datubāzēm nepieciešama manuāla iestatīšana.

Attention:
Zabbix starpniekserveris neņem vērā uzturēšanas periodus; sīkāku informāciju skatiet sadaļā Rindu aprēķināšana uzturēšanas laikā.

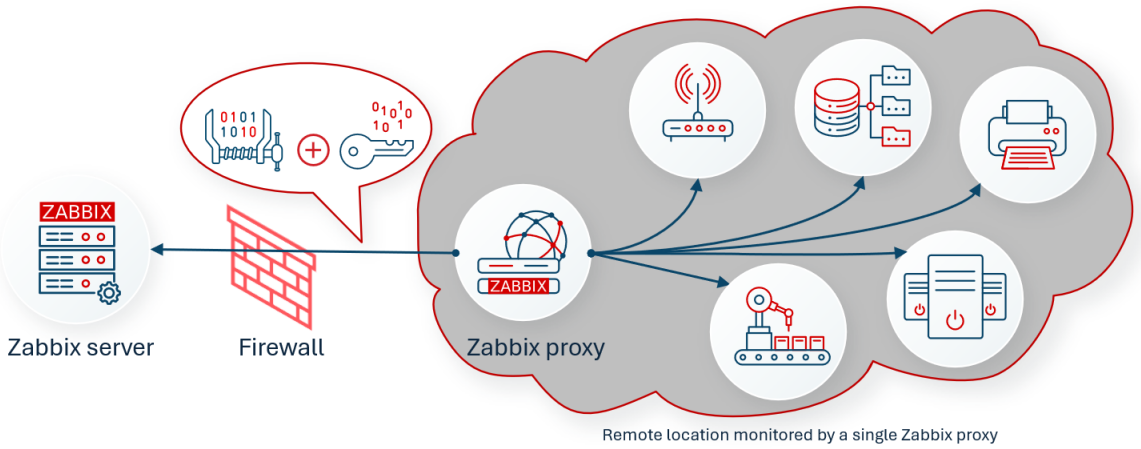
1 Starpniekserveri

Pārskats Zabbix starpniekserveris var vākt veiktspējas un pieejamības datus Zabbix servera vārdā. Tādējādi starpniekserveris var uzņemties daļu no datu vākšanas slodzes un atslogot Zabbix serveri.

Turklāt starpniekservera izmantošana ir vienkāršākais veids, kā ieviest centralizētu un izkliedētu uzraudzību, kad visi aģenti un starpniekserveri ziņo vienam Zabbix serverim un visi dati tiek vākti centralizēti.

Zabbix starpniekserveri var izmantot, lai:

- Uzraudzītu attālas atrašanās vietas
- Uzraudzītu atrašanās vietas ar nestabiliem sakariem
- Atslogotu Zabbix serveri, uzraugot tūkstošiem ierīču
- Vienkāršotu izkliedētas uzraudzības uzturēšanu



Remote location monitored by a single Zabbix proxy

Starpniekserverim ir nepieciešams tikai viens TCP savienojums ar Zabbix serveri. Tādējādi ir vieglāk apiet ugunsmūri, jo jums jākonfigurē tikai viens ugunsmūra noteikums.

Attention:
Zabbix starpniekserverim jāizmanto atsevišķa datubāze. Norādot uz Zabbix servera datubāzi, konfigurācija tiks bojāta.

Visi starpniekservera savāktie dati pirms nosūtīšanas uz serveri tiek glabāti lokāli. Tādējādi netiek zaudēti dati īslaicīgu saziņas problēmu ar serveri dēļ. Parametri *ProxyLocalBuffer* un *ProxyOfflineBuffer* starpniekservera konfigurācijas failā nosaka, cik ilgi dati tiek glabāti lokāli.

Attention:
Var gadīties, ka starpniekserverim, kas saņem jaunākās konfigurācijas izmaiņas tieši no Zabbix servera datubāzes, ir aktuālāka konfigurācija nekā Zabbix serverim, kura konfigurācija var netikt atjaunināta tik ātri CacheUpdateFrequency vērtības dēļ. Rezultātā starpniekserveris var sākt vākt datus un nosūtīt tos uz Zabbix serveri, kas šos datus ignorē.

Zabbix starpniekserveris ir datu vācējs. Tas neaprēķina trigerus, neapstrādā notikumus un nesūta brīdinājumus. Lai iegūtu pārskatu par starpniekservera funkcionalitāti, skatiet šo tabulu:

Funkcija		Atbalsta starpniekserveris
Vienumi		
	Zabbix aģenta pārbaudes	**Jā**
	Zabbix aģenta pārbaudes (aktīvās)	**Jā** [1]
	Vienkāršās pārbaudes	**Jā**
	Trapper vienumi	**Jā**
	SNMP pārbaudes	**Jā**
	SNMP trapi	**Jā**
	IPMI pārbaudes	**Jā**
	JMX pārbaudes	**Jā**
	Žurnālfailu uzraudzība	**Jā**
	Iekšējās pārbaudes	**Jā**
	SSH pārbaudes	**Jā**
	Telnet pārbaudes	**Jā**
	Ārējās pārbaudes	**Jā**
	Atkarīgie vienumi	**Jā**
	Skriptu vienumi	**Jā**
	Pārlūka vienumi	**Jā**
Iebūvētā tīmekļa uzraudzība		**Jā**
Vienumu vērtību priekšapstrāde		**Jā**
Tīkla atklāšana		**Jā**
Aktīvā aģenta automātiskā reģistrācija		**Jā**
Zema līmeņa atklāšana		**Jā** [2]
Attālās komandas		**Jā**
Trigeru aprēķināšana		*Nē*
Notikumu apstrāde		*Nē*
Notikumu korelācija		*Nē*
Brīdinājumu sūtīšana		*Nē*

Note:
[1] Lai nodrošinātu, ka aģents aktīvajām pārbaudēm vaicā starpniekserverim (nevis serverim), starpniekserverim jābūt norādītam parametrā **ServerActive** aģenta konfigurācijas failā.
 [2] LLD gadījumā Zabbix starpniekserveris tikai savāc un priekšapstrādā datus, pēc tam nosūtot tos uz Zabbix serveri turpmākai apstrādei.

Aizsardzība pret pārslodzi

Ja Zabbix serveris kādu laiku nav darbojies un starpniekserveri ir savākuši daudz datu, tad pēc servera palaišanas tas var tikt pārslogots (vēstures kešatmiņas izmantojums kādu laiku saglabājas 95–100% līmenī). Šāda pārslodze var izraisīt veiktspējas pasliktināšanos, kad pārbaudes tiek apstrādātas lēnāk, nekā tām vajadzētu. Lai izvairītos no problēmām, kas rodas vēstures kešatmiņas pārslodzes dēļ, tika ieviesta aizsardzība pret šo scenāriju.

Kad Zabbix servera vēstures kešatmiņa ir pilna, rakstīšanas piekļuve vēstures kešatmiņai tiek ierobežota, apturot servera datu vākšanas procesus. Visbiežāk vēstures kešatmiņas pārslodze rodas pēc servera dīkstāves, kad starpniekserveri augšupielādē savāktos datus. Lai no tā izvairītos, tika pievienota starpniekserveru ierobežošana (pašlaik to nevar atspējot).

Kad vēstures kešatmiņas izmantojums sasniedz 80%, Zabbix serveris pāriet ierobežošanas režīmā. Ierobežošanas režīmā serveris pieņem starpniekserveru datus tikai tad, ja vēstures kešatmiņas izmantojums ir zem 60%, rotējot pieņemtos starpniekserverus. Kad vēstures kešatmiņas izmantojums nokrītas zem 20%, serveris atgriežas normālā režīmā.

Papildus tam normālajā režīmā Zabbix serveris ierobežo atsevišķus starpniekserverus, kas sūta ļoti lielas paketes (10 000+ ierakstu), ja vēstures kešatmiņas izmantojums pārsniedz 60%. Šis lēmums tiek pieņemts brīdī, kad serveris izvērtē starpniekservera augšupielādi, tāpēc tas ne vienmēr uzreiz atspoguļojas vēstures kešatmiņas izmantojuma grafikos (iekšējais vienums `zabbix[wcache,history,pused]` un tā atjaunināšanas intervāls var nefiksēt īslaicīgus pīķus).

Šis ierobežošanas režīms turpināsies, līdz kešatmiņas izmantojums atkal sasniegs 80%, nokritīs līdz 20% vai arī ierobežošanas saraksts būs tukšs. Pirmajā gadījumā serveris atkal pārtrauks pieņemt starpniekserveru datus. Pārējos divos gadījumos serveris sāks darboties normāli, pieņemot datus no visiem starpniekserveriem.

Iepriekš minēto informāciju var attēlot šādā tabulā:

Vēstures rakstīšanas kešatmiņas izmantojums	Zabbix servera režīms	Zabbix servera darbība
Sasniedz 80%	Gaida	Pārtrauc pieņemt starpniekserveru datus, bet uztur *ierobežošanas sarakstu* (prioritāru starpniekserveru sarakstu, ar kuriem sazināties vēlāk).
Sasniedz 60%	Normāls, bet gatavs ierobežošanai	Var atteikt ļoti lielas starpniekserveru augšupielādes (vairāk nekā 10 tūkst. ierakstu), lemjot par datu pieņemšanu; turpina pieņemt citus starpniekserveru datus.
Nokrītas līdz 20%	Normāls	Atmet ierobežošanas sarakstu un sāk normāli pieņemt starpniekserveru datus.

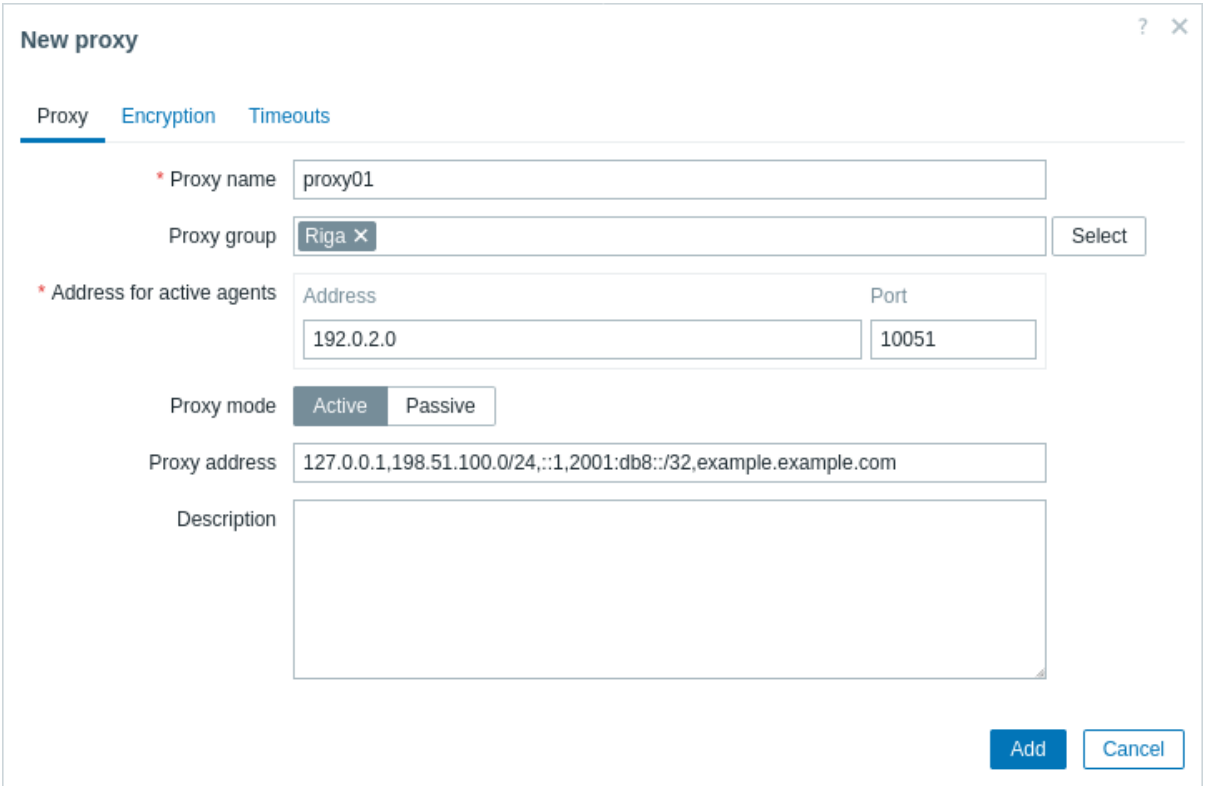
Jūs varat izmantot iekšējo vienumu zabbix[wcache,history,pused], lai sasaistītu šo Zabbix servera darbību ar metriku.

Konfigurācija Kad esat instalējis un konfigurējis starpniekserveri, ir pienācis laiks to konfigurēt Zabbix lietotāja saskarnē.

Starpniekserveru pievienošana

Lai konfigurētu starpniekserveri Zabbix lietotāja saskarnē:

- Dodieties uz: *Administration* > *Proxies*
- Noklikšķiniet uz *Create proxy*



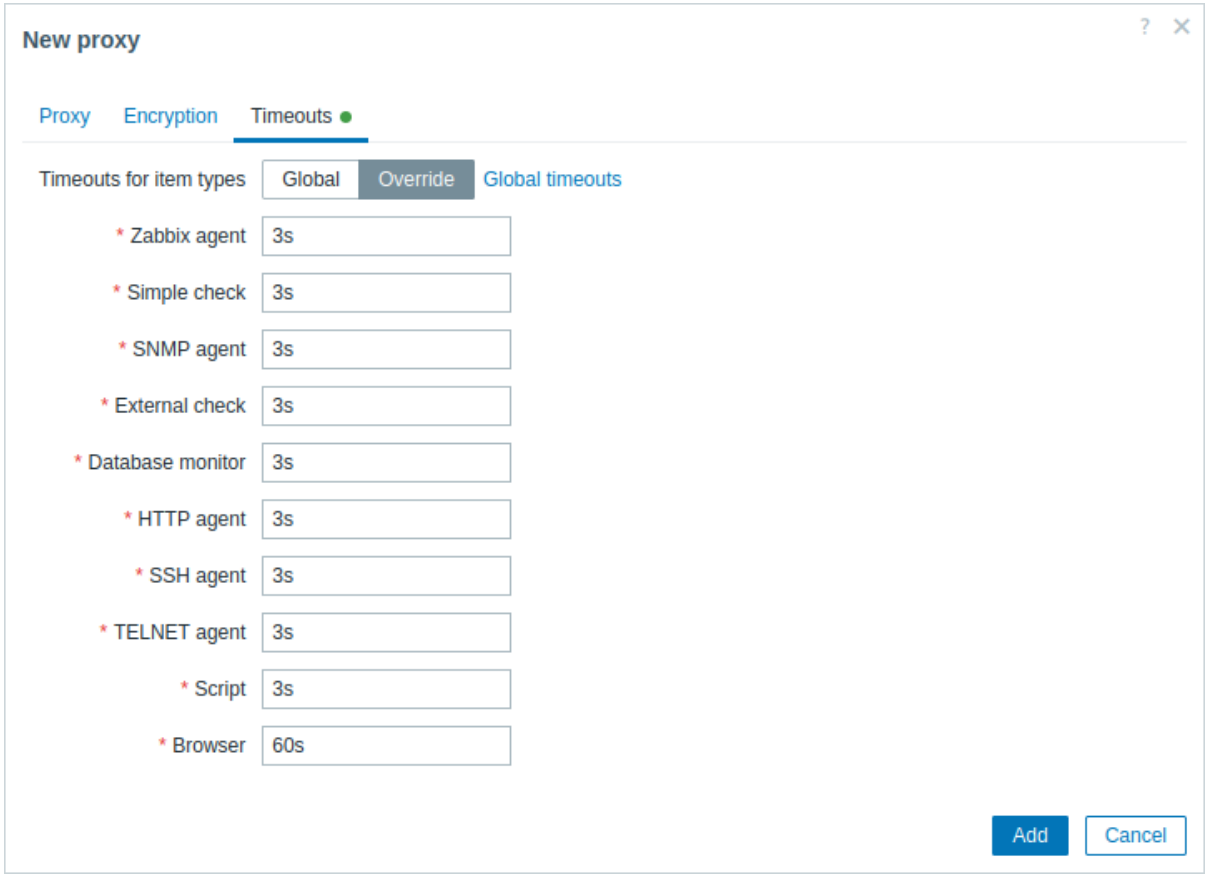
Parametrs	Apraksts
Proxy name	Ievadiet starpniekservera nosaukumu. Tam jāsakrīt ar nosaukumu *Hostname* parametrā starpniekservera konfigurācijas failā.
Proxy group	Atlasiet vienu starpniekserveru grupu starpniekservera slodzes līdzsvarošanai/augstai pieejamībai.
Address for active agents	Ievadiet adresi, pie kuras jāpieslēdzas uzraudzītajiem aktīvajiem aģentiem vai sūtītājiem. Tiek atbalstīts **tikai** Zabbix 7.0 aģentiem vai jaunākām versijām. Šī adrese tiek izmantota, lai pieslēgtos gan aktīvajiem, gan pasīvajiem starpniekserveriem. Šis lauks ir pieejams tikai tad, ja laukā *Proxy group* ir atlasīta starpniekserveru grupa.
Address	IP adrese/DNS nosaukums, pie kura pieslēgties.
Port	TCP porta numurs (pēc noklusējuma 10051), pie kura pieslēgties. Tiek atbalstīti lietotāja makrosi.

Parametrs	Apraksts
Proxy mode	Atlasiet starpniekservera režīmu. **Active** - starpniekserveris pieslēgsies Zabbix serverim un pieprasīs konfigurācijas datus **Passive** - Zabbix serveris pieslēdzas starpniekserverim *Ņemiet vērā*, ka bez šifrētas saziņas (sensitīvi) starpniekservera konfigurācijas dati var kļūt pieejami pusēm, kurām ir piekļuve Zabbix servera trapper portam, izmantojot aktīvu starpniekserveri. Tas ir iespējams, jo jebkurš var izlikties par aktīvu starpniekserveri un pieprasīt konfigurācijas datus, ja autentifikācija nenotiek vai starpniekservera adreses nav ierobežotas laukā *Proxy address*.
Proxy address	Ja norādīts, tad aktīvā starpniekservera pieprasījumi tiek pieņemti tikai no šī ar komatiem atdalītā IP adrešu saraksta, pēc izvēles CIDR pierakstā, vai aktīvā Zabbix starpniekservera DNS nosaukumiem. Šis lauks ir pieejams tikai tad, ja laukā *Proxy mode* ir atlasīts aktīvs starpniekserveris. Makrosi netiek atbalstīti.
Interface	Ievadiet saskarnes informāciju pasīvam starpniekserverim. Šis lauks ir pieejams tikai tad, ja laukā Proxy mode ir atlasīts pasīvs starpniekserveris.
Address	Pasīvā starpniekservera IP adrese/DNS nosaukums.
Port	Pasīvā starpniekservera TCP porta numurs (pēc noklusējuma 10051). Tiek atbalstīti lietotāja makrosi.
Description	Ievadiet starpniekservera aprakstu.

Cilne **Encryption** ļauj pieprasīt šifrētus savienojumus ar starpniekserveri.

Parametrs	Apraksts
Connections to proxy	Kā serveris pieslēdzas pasīvajam starpniekserverim: bez šifrēšanas (pēc noklusējuma), izmantojot PSK (iepriekš koplietotu atslēgu) vai sertifikātu.
Connections from proxy	Atlasiet, kāda veida savienojumi ir atļauti no aktīvā starpniekservera. Vienlaikus var atlasīt vairākus savienojumu veidus (noderīgi testēšanai un pārslēgšanai uz citu savienojuma veidu). Pēc noklusējuma ir ”No encryption”.
Issuer	Atļautais sertifikāta izdevējs. Sertifikāts vispirms tiek pārbaudīts ar CA (sertifikācijas iestādi). Ja tas ir derīgs un CA to ir parakstījusi, tad lauku *Issuer* var izmantot, lai vēl vairāk ierobežotu atļauto CA. Šis lauks nav obligāts un ir paredzēts lietošanai, ja jūsu Zabbix instalācijā tiek izmantoti vairāku CA sertifikāti.
Subject	Atļautais sertifikāta subjekts. Sertifikāts vispirms tiek pārbaudīts ar CA. Ja tas ir derīgs un CA to ir parakstījusi, tad lauku *Subject* var izmantot, lai atļautu tikai vienu *Subject* virknes vērtību. Ja šis lauks ir tukšs, tad tiek pieņemts jebkurš derīgs sertifikāts, ko parakstījusi konfigurētā CA.
PSK identity	Iepriekš koplietotās atslēgas identitātes virkne. Neievietojiet sensitīvu informāciju PSK identitātē, jo tā tiek pārraidīta tīklā nešifrētā veidā, lai informētu saņēmēju, kuru PSK izmantot.
PSK	Iepriekš koplietotā atslēga (hex-virkne). Maksimālais garums: 512 hex-cipari (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, 64 hex-cipari (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952

Cilne **Timeouts** ļauj pārrakstīt globālos taimautus vienumu tipiem, kas to atbalsta.



Parametrs	Apraksts
Timeouts for item types	Iestatiet vienuma taimautu (atkarībā no tā tipa): **Global** - izmantot globālo taimautu (parādīts pelēkotajā laukā *Timeout* katram vienuma tipam); **Override** - iestatīt pielāgotu taimautu (laukā *Timeout* katram vienuma tipam). Atļautais diapazons: 1 - 600s (pēc noklusējuma: mantots no globālajiem taimautiem). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makrosi. Noklikšķinot uz saites *Global timeouts*, varat konfigurēt globālos taimautus. Ņemiet vērā, ka saite *Global timeouts* ir redzama tikai *Super admin* tipa lietotājiem ar atļaujām uz lietotāja saskarnes sadaļu *Administration* > *General*. Ņemiet vērā, ka, lai gan starpniekservera līmeņa taimauti pārraksta globālos, tos pārrakstīs individuālie vienuma taimauti, ja tie ir konfigurēti.

Note:
Ja starpniekservera galvenā versija nesakrīt ar servera galveno versiju, blakus *Timeouts for item types* tiks parādīta ikona ar uznirstošo ziņojumu "Timeouts disabled because the proxy and server versions do not match". Šādos gadījumos starpniekserveris izmantos parametru `Timeout` no starpniekservera konfigurācijas faila.

Esoša starpniekservera rediģēšanas formai ir šādas papildu pogas:

- *Refresh configuration* - atsvaidzināt starpniekservera konfigurāciju
- *Clone* - izveidot jaunu starpniekserveri, pamatojoties uz esošā starpniekservera īpašībām
- *Delete* - dzēst starpniekserveri

Hosta konfigurācija

Jūs varat norādīt, ka atsevišķs hosts ir jāuzrauga starpniekserverim vai starpniekserveru grupai hosta konfigurācijas formā, izmantojot lauku *Monitored by*.

Hostu masveida atjaunināšana ir vēl viens veids, kā norādīt, ka hosti ir jāuzrauga starpniekserverim vai starpniekserveru grupai.

1 Monitoringa konfigurācijas sinhronizācija

Pārskats

Šajā lapā ir sniegta detalizēta informācija par starpniekservera uzraudzības konfigurācijas atjaunināšanu, t. i., kā serverī veiktās izmaiņas uzraudzības konfigurācijā tiek sinhronizētas ar starpniekserveri.

Inkrementāls atjauninājums

Starpniekservera konfigurācijas atjauninājums ir inkrementāls. Konfigurācijas sinhronizācijas laikā tiek atjauninātas tikai modificētās entītijas (tādējādi, ja neviena entītija nav modificēta, nekas netiks nosūtīts). Šī pieeja ļauj ietaupīt resursus un iestatīt mazāku intervālu (gandrīz tūlītēju) starpniekservera konfigurācijas atjauninājumam.

Starpniekservera konfigurācijas izmaiņas tiek izsekotas, izmantojot revīziju numurus. Konfigurācijas datos, kas tiek nosūtīti uz starpniekserveri, tiek iekļautas tikai tās entītijas, kuru revīzijas ir lielākas par starpniekservera konfigurācijas revīziju.

Konfigurācijas sinhronizācijai paredzētās entītijas ir šādas:

Entītija	Detalizēta informācija
autoregistration tls data	Visi automātiskās reģistrācijas TLS dati.
expressions	Visas izteiksmes (regulārās izteiksmes, izteiksmju tabulas).
global configuration	Globālā konfigurācija, kas definēta tabulā 'config'
host	Visas hosta īpašības, saskarnes, inventārs, vienumi, vienumu priekšapstrāde, vienumu parametri, hosta tīmekļa scenāriji.
host macros	Visi makrosi, kas definēti hostā, un visi ar to saistītie veidņu ID.
proxy discovery rule	Atklāšanas kārtulas un pārbaudes, kas piešķirtas starpniekserverim.

Tas nozīmē:

- Ja **hostā** tiek mainīts kāds **vienums**, tiks sinhronizēta visa šī hosta konfigurācija.
- Ja tiek mainīta **regulārā izteiksme**, tiks sinhronizētas visas regulārās izteiksmes.

Izņēmums ir hosta makrosi, kas tiek nosūtīti arī tad, ja hostā ir mainīts jebkas.

Komanda `-R config_cache_reload` starpniekserverī arī iniciēs inkrementālu atjauninājumu.

Ņemiet vērā, ka pilna konfigurācijas sinhronizācija starpniekserverī notiks palaišanas/pārstartēšanas laikā, HA pārslēgšanās gadījumā, ja ir mainījies sesijas marķieris vai ja starpniekserverī konfigurācijas atjaunināšana neizdevās, piemēram, ja konfigurācijas datu saņemšanas laikā tika pārtraukts savienojums.

Konfigurācijas parametri

Parametrs **ProxyConfigFrequency** nosaka, cik bieži starpniekservera konfigurācija tiek sinhronizēta ar serveri (pēc noklusējuma ik pēc 10 sekundēm).

Ņemiet vērā, ka ProxyConfigFrequency ir:

- servera parametrs pasīvajiem starpniekserveriem
- starpniekservera parametrs aktīvajiem starpniekserveriem

Aktīvajos starpniekserveros ProxyConfigFrequency ir jauns parametrs kopš Zabbix 6.4, un tas jāizmanto tagad novecojušā ConfigFrequency vietā.

> **Attention:**
> Ja tiek izmantoti gan ProxyConfigFrequency, gan ConfigFrequency, starpniekserveris reģistrēs kļūdu žurnālā un pārtrauks darbību.

Rindu aprēķināšana apkopes laikā

> **Attention:**
> Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet Rindu aprēķināšana apkopes laikā.

2 Starpniekservera slodzes balansēšana un augsta pieejamība

Pārskats

Zabbix starpniekserverus var organizēt starpniekserveru grupās, lai nodrošinātu starpniekserveru slodzes līdzsvarošanu un augstu pieejamību.

Starpniekserveru slodzes līdzsvarošana un augsta pieejamība ir automātiska hostu pārdale starp starpniekserveriem vienas starpniekserveru grupas ietvaros:

- Ja starpniekserveris kļūst bezsaistē, tā hosti tiks pārvietoti uz citiem starpniekserveriem, tādējādi saglabājot augstu starpniekserveru pieejamību.
- Ja starpniekserverim ir ievērojami lielāks/mazāks hostu skaits nekā citiem starpniekserveriem, tā hosti tiks pārvietoti uz citiem starpniekserveriem, lai līdzsvarotu starpniekserveru slodzi.

Hostu pārdale darbojas tikai starp tiem grupas starpniekserveriem, kas atbilst šādiem nosacījumiem:

- Starpniekserveros darbojas Zabbix 7.0 vai jaunāka versija.
- Starpniekservera versija atbilst Zabbix serveris versijai. Ja tiek izmantots Zabbix aģents (pasīvais), starpniekservera versijai jāatbilst aģenta versijai. Aktīvajiem aģentiem nepieciešama tikai Zabbix 7.0 vai jaunāka versija.
- Starpniekserveru grupai ir tiešsaistes stāvoklis.
- Hosti ir konfigurēti tā, lai tos uzraudzītu starpniekserveru grupa, nevis atsevišķi starpniekserveri.

Note:
Starpniekserveru grupas veselības stāvokli var uzraudzīt ar iekšējām pārbaudēm no jebkura hosta, kas piešķirts starpniekserveru grupai. Tomēr, lai uzraudzītu viena grupas starpniekservera veselības stāvokli, piešķiriet hostu šim starpniekserverim; pretējā gadījumā rezultāti var būt nekonsekventi.

Hostu pārdale

Starpniekserveru slodzes līdzsvarošanu un augstu pieejamību pārvalda Zabbix serveris, izmantojot starpniekserveru grupas pārvaldnieku, kas nepārtraukti uzrauga visu katras starpniekserveru grupas starpniekserveru stāvokli un to hostu sadalījumu.

Starpniekserveru augsta pieejamība grupā tiek nodrošināta ar starpniekserveru pārslēgšanos atteices gadījumā: kad starpniekserveris kļūst bezsaistē, tā hosti tiek nekavējoties pārdalīti citiem starpniekserveriem. Notiek arī starpniekserveru slodzes līdzsvarošana, jo hosti tiek atkārtoti piešķirti starpniekserveriem ar vismazāko piešķirto hostu skaitu.

Papildus tam starpniekserveru slodzes līdzsvarošana tiek aktivizēta, ja starpniekservera hostu skaits atšķiras no grupas vidējā rādītāja vismaz par 10 hostiem un 2 reizes (hostu pārpalikums vai hostu trūkums). Ja nelīdzsvarotība saglabājas pēc nogaidīšanas perioda (10 x pārslēgšanās aizture), starpniekserveru grupa tiek ievietota hostu pārdales rindā.

Starpniekserveru grupas pārvaldnieks pārdala hostus, izmantojot šādu loģiku:

1. Aprēķina vidējo hostu skaitu uz vienu starpniekserveri.
2. Starpniekserveriem ar hostu pārpalikumu — pārvieto liekos hostus uz nepiešķirto starpniekserveru kopu.
3. Starpniekserveriem ar hostu trūkumu — aprēķina, cik hostu nepieciešams, lai sasniegtu līdzsvaru.
4. No starpniekserveriem ar vislielāko hostu skaitu noņem nepieciešamo hostu skaitu.
5. Pārvieto nepiešķirtos hostus uz starpniekserveriem ar vismazāko hostu skaitu.

Hostu pārdales piemēri:

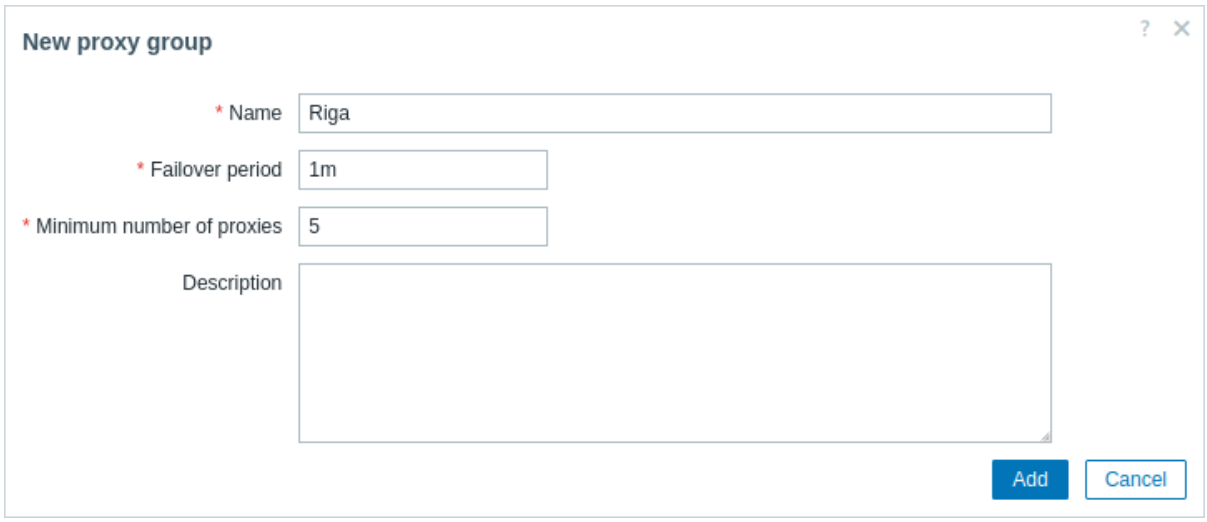
Hosti uz starpniekservera	Grupas vidējais rādītājs	Hostu atkārtota piešķiršana
100	50	Jā
60	50	Nē
40	50	Nē
25	50	Jā
15	5	Jā
10	5	Nē

Attention:
Ja starpniekserveru grupa uzrauga mazāk nekā 10 hostus, tas var izraisīt nevienmērīgu hostu sadalījumu starp grupas starpniekserveriem.

Starpniekserveru grupas konfigurēšana

Lai konfigurētu starpniekserveru grupu Zabbix lietotāja saskarnē:

1. Atveriet *Administration* > *Proxy groups*
2. Noklikšķiniet uz *Create proxy group*



Parametrs	Apraksts
Name	Starpniekserveru grupas nosaukums.
Failover period	Periods sekundēs, kura laikā starpniekserverim starpniekserveru grupā ir jāsazinās ar Zabbix serveri, lai tas tiktu uzskatīts par tiešsaistē esošu (noklusējums: 1m; diapazons: 10s–15m). Ja starpniekserveris šajā periodā nesazinās, tā stāvoklis tiek mainīts uz *Offline*, un tā hosti tiek nekavējoties pārdalīti citiem starpniekserveriem. Starpniekserveru slodzes līdzsvarošana sākas pēc 10 x šī perioda. Atbalsta laika sufiksus (piemēram, 30s, 1m) un lietotāja makrosus.
Minimum number of proxies	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa būtu tiešsaistē (noklusējums: 1; diapazons: 1–1000). Atbalsta lietotāja makrosus. Šai vērtībai jābūt mazākai par kopējo starpniekserveru skaitu grupā. Piemēram, 10 starpniekserveru grupā, iestatot minimālo skaitu uz 10, grupa pāries bezsaistes režīmā, ja atteiks jebkurš starpniekserveris. Ņemiet vērā, ka tiešsaistē esošie starpniekserveri bezsaistes grupā turpina darboties normāli, taču slodzes līdzsvarošana/augsta pieejamība nenotiks.
Description	Starpniekserveru grupas apraksts.
Proxies	Rediģējot grupu ar starpniekserveriem, tiek parādīts saraksts ar ne vairāk kā pieciem starpniekserveriem (kā saites vai vienkāršs teksts atkarībā no lietotāja piekļuves tiesībām starpniekserveriem).

Starpniekserveru slodzes līdzsvarošanas konfigurēšana

Lai izmantotu starpniekserveru slodzes līdzsvarošanu, jums Zabbix lietotāja saskarnē jākonfigurē starpniekserveru grupa (skatiet iepriekš) un jāpārliecinās, ka hostus uzrauga starpniekserveru grupa, nevis atsevišķi starpniekserveri (varat izmantot hostu masveida atjaunināšanu, lai pārvietotu hostus no starpniekserveriem uz starpniekserveru grupu).

Ja izmantojat Zabbix aģentu, konfigurējiet to arī šādi:

- Pasīvajām pārbaudēm parametrā Server norādiet visus starpniekserveru grupas starpniekserverus.
- Aktīvajām pārbaudēm ieteicams parametrā ServerActive norādīt visus starpniekserveru grupas starpniekserverus vai Zabbix serveri. Ņemiet vērā, ka ar starpniekserveru grupām aktīvajā režīmā darbosies tikai Zabbix aģents 7.0 (vai jaunāka versija).

Ja parametrā `ServerActive` ir iekļauts tikai viens starpniekserveris no starpniekserveru grupas (vai Zabbix serveris), aģents joprojām varēs izveidot savienojumu ar pareizo starpniekserveri. Kad aģenta serviss tiek palaists un izveido savienojumu ar norādīto starpniekserveri, aģents saņems un kešatmiņā saglabās pilnu starpniekserveru IP adrešu sarakstu un to pašreizējo slodzi grupā. Pēc tam aktīvās pārbaudes tiks novirzītas uz hostam atbilstošo tiešsaistē esošo starpniekserveri, pamatojoties uz pašreizējo starpniekservera un hosta piesaisti starpniekserveru grupā.

Warning:
Ja Zabbix aģenta parametrā `ServerActive` ir norādīts tikai viens starpniekserveris, tas var izraisīt uzraudzības datu zudumu, ja aģents tiek palaists vai pārstartēts laikā, kad norādītais starpniekserveris nav tiešsaistē.

Note:
Izmantojot Zabbix sender, datu pieprasījumi arī tiek novirzīti uz hostam atbilstošo tiešsaistē esošo starpniekserveri, pamatojoties uz pašreizējo starpniekservera un hosta piesaisti starpniekserveru grupā. Tomēr, ja sūtāt vairāku hostu vērtības no ievades faila, izmantojiet −g option, lai novērstu datu nosūtīšanu nepareizajam starpniekserverim.

Zabbix aģentam arī jāspēj caur ugunsmūri izveidot savienojumu ar visiem starpniekserveriem starpniekserveru grupā. Pretējā gadījumā aktīvās pārbaudes novirzīšanas vai pārslēgšanās uz rezerves mezglu laikā var uzkārties vai neizdoties. Piemēram:

- Aktīvo pārbaužu laikā starpniekserveris var novirzīt aģentu uz citu starpniekserveri. Ja šo starpniekserveri bloķē ugunsmūris, saziņa uzkārties, gaidot atbildi.
- Stabilās augstas pieejamības vidēs bez nesenas pārbalansēšanas aģenti var nekad nesazināties ar rezerves starpniekserveriem. Ja ugunsmūra noteikumi ir mainīti un nav pārbaudīti, pārslēgšanās uz rezerves mezglu var neizdoties.

Starpniekserveru slodzes līdzsvarošanas testēšana

Lai testētu starpniekserveru slodzes līdzsvarošanu:

1. Konfigurējiet starpniekserveru grupu.
2. Pārliecinieties, ka starpniekserveru grupai ir tiešsaistes stāvoklis.
3. Pārliecinieties, ka hostus uzrauga starpniekserveru grupa, nevis atsevišķi starpniekserveri (varat izmantot hostu masveida atjaunināšanu, lai pārvietotu hostus no starpniekserveriem uz starpniekserveru grupu).
4. Uzgaidiet dažas sekundes, līdz tiek atjaunināta konfigurācija un hosti tiek sadalīti starp starpniekserveriem starpniekserveru grupā. Novērojiet izmaiņas, atsvaidzinot hostu sarakstu sadaļā *Administrēšana* > *Starpniekserveri*.

Svarīgas piezīmes

- SNMP trapi netiek atbalstīti starpniekserveriem starpniekserveru grupā.
- Pārbaudēm, kas ir atkarīgas no ārējās konfigurācijas (piemēram, skriptiem ārējām pārbaudēm vai ODBC konfigurācijai datubāzes pārbaudēm), jābūt vienādai konfigurācijai visos starpniekserveros starpniekserveru grupā.
- Datubāzes pārbaudēm ir nepieciešamas paplašinātas atļaujas datubāzes objektam/serverim.
- VMware hosti, kurus uzrauga starpniekserveru grupa, tiks nejauši sadalīti starp grupas starpniekserveriem. Tas izraisa to, ka katrs starpniekserveris kešo visus VMware datus, radot papildu slodzi vCenter.
- Hosti, kas izveidoti, pamatojoties uz automātiskās reģistrācijas datiem no starpniekservera starpniekserveru grupā, tiek iestatīti tā, lai tos uzraudzītu šī starpniekserveru grupa. Tomēr hosti, kas izveidoti, pamatojoties uz tīkla atklāšanas datiem no starpniekservera starpniekserveru grupā, tiek iestatīti tā, lai tos uzraudzītu šis starpniekserveris.

15 Šifrēšana

Pārskats Zabbix atbalsta šifrētu saziņu starp Zabbix komponentiem, izmantojot Transport Layer Security (TLS) protokolu v.1.2 un 1.3 (atkarībā no kriptogrāfijas bibliotēkas). Tiek atbalstīta šifrēšana, kas balstīta uz sertifikātiem un iepriekš koplietotām atslēgām.

Šifrēšanu var konfigurēt savienojumiem:

- Starp Zabbix serveri, Zabbix starpniekserveri, Zabbix aģentu, Zabbix tīmekļa pakalpojumu, zabbix_sender un zabbix_get utilītām
- Uz Zabbix datubāzi no Zabbix lietotāja saskarnes un servera/starpniekservera
- Starp Zabbix lietotāja saskarni un Zabbix serveri

Šifrēšana ir izvēles iespēja, un to var konfigurēt atsevišķiem komponentiem:

- Dažus starpniekserverus un aģentus var konfigurēt tā, lai tie izmantotu uz sertifikātiem balstītu šifrēšanu ar serveri, savukārt citi var izmantot uz iepriekš koplietotām atslēgām balstītu šifrēšanu, bet vēl citi var turpināt izmantot nešifrētu saziņu (kā iepriekš).
- Serveris (starpniekserveris) var izmantot atšķirīgas šifrēšanas konfigurācijas dažādiem hostiem.

Zabbix dēmonu programmas izmanto vienu klausīšanās portu šifrētiem un nešifrētiem ienākošajiem savienojumiem. Šifrēšanas pievienošanai nav nepieciešams atvērt jaunus portus ugunsmūros.

Ierobežojumi

- Privātās atslēgas tiek glabātas vienkāršā tekstā failos, kurus Zabbix komponenti var nolasīt palaišanas laikā.
- Iepriekš koplietotās atslēgas tiek ievadītas Zabbix lietotāja saskarne un glabātas Zabbix datubāzē vienkāršā tekstā.
- Iebūvētā šifrēšana neaizsargā saziņu starp tīmekļa serveri, kurā darbojas Zabbix lietotāja saskarne, un lietotāja tīmekļa pārlūku.
- Pašlaik katrs šifrētais savienojums tiek atvērts ar pilnu TLS rokasspiedienu; sesiju kešošana un biļetes netiek izmantotas.

- Šifrēšanas pievienošana palielina laiku, kas nepieciešams vienumu pārbaudēm un darbībām, atkarībā no tīkla latentuma:
  - Piemēram, ja pakešu aizture ir 100 ms, tad TCP savienojuma atvēršana un nešifrēta pieprasījuma nosūtīšana aizņem apmēram 200 ms. Izmantojot šifrēšanu, TLS savienojuma izveidei tiek pievienotas apmēram 1000 ms.
  - Var būt nepieciešams palielināt noildzes vērtības, pretējā gadījumā daži vienumi un darbības, kas izpilda attālinātus skriptus uz aģentiem, var darboties ar nešifrētiem savienojumiem, bet neizdoties noildzes dēļ ar šifrētiem.
- Šifrēšanu neatbalsta tīkla atklāšana. Zabbix aģenta pārbaudes, ko veic tīkla atklāšana, būs nešifrētas, un, ja Zabbix aģents ir konfigurēts noraidīt nešifrētus savienojumus, šādas pārbaudes nebūs veiksmīgas.

Zabbix kompilēšana ar šifrēšanas atbalstu Lai atbalstītu šifrēšanu, Zabbix ir jākompilē un jāsasaista ar vienu no atbalstītajām kriptogrāfijas bibliotēkām:

- GnuTLS - начиная ar versiju 3.1.18
- OpenSSL - versijas 1.0.1, 1.0.2, 1.1.0, 1.1.1, 3.0.x
- LibreSSL - testēts ar versijām 2.7.4, 2.8.2:
  - LibreSSL 2.6.x netiek atbalstīts
  - LibreSSL tiek atbalstīts kā ar OpenSSL saderīgs aizstājējs; jaunās `tls_*()` LibreSSL specifiskās API funkcijas netiek izmantotas. Zabbix komponenti, kas kompilēti ar LibreSSL, nevarēs izmantot PSK, varēs izmantot tikai sertifikātus.

Note:
Plašāku informāciju par SSL iestatīšanu Zabbix lietotāja saskarnei var atrast šajā labākajā praksē.

Bibliotēka tiek izvēlēta, norādot attiecīgo opciju skriptam "configure":

- `--with-gnutls[=DIR]`
- `--with-openssl[=DIR]` (izmanto arī LibreSSL)

Piemēram, lai konfigurētu pirmkodu serverim un aģentam ar *OpenSSL*, var izmantot kaut ko līdzīgu šim:

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

Dažādi Zabbix komponenti var tikt kompilēti ar dažādām kriptogrāfijas bibliotēkām (piemēram, serveris ar *OpenSSL*, aģents ar *GnuTLS*).

Attention:
Ja plānojat izmantot iepriekš koplietotas atslēgas (PSK), apsveriet iespēju izmantot *GnuTLS* vai *OpenSSL 1.1.0* (vai jaunākas) bibliotēkas tajos Zabbix komponentos, kas izmanto PSK. *GnuTLS* un *OpenSSL 1.1.0* bibliotēkas atbalsta PSK šifru kopas ar Perfect Forward Secrecy. Arī vecākas *OpenSSL* bibliotēkas versijas (1.0.1, 1.0.2c) atbalsta PSK, taču pieejamās PSK šifru kopas nenodrošina Perfect Forward Secrecy.

Savienojumu šifrēšanas pārvaldība Savienojumi Zabbix var izmantot:

- bez šifrēšanas (noklusējums)
- uz RSA sertifikātiem balstītu šifrēšanu
- uz PSK balstītu šifrēšanu

Ir divi svarīgi parametri, ko izmanto, lai norādītu šifrēšanu starp Zabbix komponentiem:

- TLSConnect - norāda, kādu šifrēšanu izmantot izejošajiem savienojumiem (bez šifrēšanas, PSK vai sertifikāts)
- TLSAccept - norāda, kādi savienojumu tipi ir atļauti ienākošajiem savienojumiem (bez šifrēšanas, PSK vai sertifikāts). Var norādīt vienu vai vairākas vērtības.

`TLSConnect` tiek izmantots Zabbix starpniekserveris konfigurācijas failos (aktīvajā režīmā norāda tikai savienojumus ar serveris) un Zabbix aģents konfigurācijas failos (aktīvajām pārbaudēm). Zabbix lietotāja saskarne parametrs, kas atbilst TLSConnect, ir lauks *Connections to host* cilnē *Data collection → Hosts → <some host> → Encryption* un lauks *Connections to proxy* cilnē *Administration → Proxies → <some proxy> → Encryption*. Ja konfigurētais savienojuma šifrēšanas tips neizdodas, citi šifrēšanas tipi netiks mēģināti.

`TLSAccept` tiek izmantots Zabbix starpniekserveris konfigurācijas failos (pasīvajā režīmā norāda tikai savienojumus no serveris) un Zabbix aģents konfigurācijas failos (pasīvajām pārbaudēm). Zabbix lietotāja saskarne parametrs, kas atbilst TLSAccept, ir lauks *Connections from host* cilnē *Data collection → Hosts → <some host> → Encryption* un lauks *Connections from proxy* cilnē *Administration → Proxies → <some proxy> → Encryption*.

Parasti ienākošajiem savienojumiem konfigurē tikai vienu šifrēšanas tipu. Taču, iespējams, vēlēsieties pārslēgt šifrēšanas tipu, piemēram, no nešifrēta uz uz sertifikātiem balstītu, ar minimālu dīkstāvi un iespēju atgriezties pie iepriekšējās konfigurācijas. Lai to panāktu:

- Iestatiet `TLSAccept=unencrypted,cert` aģents konfigurācijas failā un restartējiet Zabbix aģents

- Pārbaudiet savienojumu ar aģents, izmantojot zabbix_get ar sertifikātu. Ja tas darbojas, varat pārkonfigurēt šifrēšanu šim aģents Zabbix lietotāja saskarne cilnē *Data collection* → *Hosts* → *<some host>* → *Encryption*, iestatot *Connections to host* uz "Certificate".
- Kad serveris konfigurācijas kešatmiņa tiks atjaunināta (un starpniekserveris konfigurācija tiks atjaunināta, ja hosts tiek uzraudzīts ar starpniekserveris), savienojumi ar šo aģents tiks šifrēti
- Ja viss darbojas, kā paredzēts, varat iestatīt `TLSAccept=cert` aģents konfigurācijas failā un restartēt Zabbix aģents. Tagad aģents pieņems tikai šifrētus, uz sertifikātiem balstītus savienojumus. Nešifrēti un uz PSK balstīti savienojumi tiks noraidīti.

Līdzīgā veidā tas darbojas serveris un starpniekserveris. Ja Zabbix lietotāja saskarne hosts konfigurācijā lauks *Connections from host* ir iestatīts uz "Certificate", tad no aģents tiks pieņemti tikai uz sertifikātiem balstīti šifrēti savienojumi (aktīvās pārbaudes) un zabbix_sender (trapper vienumi).

Visticamāk, jūs konfigurēsiet ienākošos un izejošos savienojumus tā, lai tie izmantotu vienu un to pašu šifrēšanas tipu vai vispār neizmantotu šifrēšanu. Taču tehniski ir iespējams to konfigurēt asimetriski, piemēram, uz sertifikātiem balstītu šifrēšanu ienākošajiem savienojumiem un uz PSK balstītu šifrēšanu izejošajiem savienojumiem.

Katra hosts šifrēšanas konfigurācija tiek parādīta Zabbix lietotāja saskarne sadaļā *Data collection* → *Hosts* kolonnā *Agent encryption*. Piemēram:

Piemērs	Savienojumi ar hostu	Atļautie savienojumi no hosta	Noraidītie savienojumi no hosta
NONE	Nešifrēti	Nešifrēti	Šifrēti, uz sertifikātiem balstīti un uz PSK balstīti šifrēti
CERT NONE PSK CERT	Šifrēti, uz sertifikātiem balstīti	Šifrēti, uz sertifikātiem balstīti	Nešifrēti un uz PSK balstīti šifrēti
PSK NONE PSK CERT	Šifrēti, uz PSK balstīti	Šifrēti, uz PSK balstīti	Nešifrēti un uz sertifikātiem balstīti šifrēti
PSK NONE PSK CERT	Šifrēti, uz PSK balstīti	Nešifrēti un uz PSK balstīti šifrēti	Uz sertifikātiem balstīti šifrēti
CERT NONE PSK CERT	Šifrēti, uz sertifikātiem balstīti	Nešifrēti, uz PSK balstīti vai uz sertifikātiem balstīti šifrēti	-

Attention:
Savienojumi pēc noklusējuma nav šifrēti. Šifrēšana jākonfigurē katram hostam un starpniekserverim atsevišķi.

zabbix_get un zabbix_sender ar šifrēšanu Skatiet zabbix_get un zabbix_sender rokasgrāmatas lapas, lai uzzinātu, kā tās izmantot ar šifrēšanu.

Šifru kopas Pēc noklusējuma šifru kopas tiek konfigurētas iekšēji Zabbix palaišanas laikā.

Tiek atbalstītas arī lietotāja konfigurētas šifru kopas GnuTLS un OpenSSL bibliotēkām. Lietotāji var konfigurēt šifru kopas atbilstoši savām drošības politikām. Šīs iespējas izmantošana nav obligāta (iebūvētās noklusējuma šifru kopas joprojām darbojas).

Kriptogrāfijas bibliotēkām, kas kompilētas ar noklusējuma iestatījumiem, Zabbix iebūvētie noteikumi parasti nodrošina šādas šifru kopas (secībā no augstākas uz zemāku prioritāti):

Bibliotēka	Sertifikātu šifru kopas	PSK šifru kopas
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

Bibliotēka	Sertifikātu šifru kopas	PSK šifru kopas
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

Lietotāja konfigurēti šifru komplekti Iebūvētos šifru komplektu atlases kritērijus var pārrakstīt ar lietotāja konfigurētiem šifru komplektiem.

> **Attention:**
> Lietotāja konfigurēti šifru komplekti ir funkcija, kas paredzēta pieredzējušiem lietotājiem, kuri izprot TLS šifru komplektus, to drošību un kļūdu sekas, kā arī pārzina TLS problēmu novēršanu.

Iebūvētos šifru komplektu atlases kritērijus var pārrakstīt, izmantojot šādus parametrus:

Pārrakstīšanas tvērums	Parametrs	Vērtība	Apraksts
Šifru komplektu atlase sertifikātiem	TLSCipherCert13	Derīgas OpenSSL 1.1.1 cipher strings TLS 1.3 protokolam (to vērtības tiek nodotas OpenSSL funkcijai `SSL_CTX_set_ciphersuites()`).	Uz sertifikātiem balstīti šifru komplektu atlases kritēriji TLS 1.3 Tikai OpenSSL 1.1.1 vai jaunāka versija.
	TLSCipherCert	Derīgas OpenSSL cipher strings TLS 1.2 vai derīgas GnuTLS priority strings. To vērtības tiek nodotas attiecīgi funkcijām `SSL_CTX_set_cipher_list()` vai `gnutls_priority_init()`.	Uz sertifikātiem balstīti šifru komplektu atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Šifru komplektu atlase PSK	TLSCipherPSK13	Derīgas OpenSSL 1.1.1 cipher strings TLS 1.3 protokolam (to vērtības tiek nodotas OpenSSL funkcijai `SSL_CTX_set_ciphersuites()`).	Uz PSK balstīti šifru komplektu atlases kritēriji TLS 1.3 Tikai OpenSSL 1.1.1 vai jaunāka versija.
	TLSCipherPSK	Derīgas OpenSSL cipher strings TLS 1.2 vai derīgas GnuTLS priority strings. To vērtības tiek nodotas attiecīgi funkcijām `SSL_CTX_set_cipher_list()` vai `gnutls_priority_init()`.	Uz PSK balstīti šifru komplektu atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)
Apvienots šifru komplektu saraksts sertifikātam un PSK	TLSCipherAll13	Derīgas OpenSSL 1.1.1 cipher strings TLS 1.3 protokolam (to vērtības tiek nodotas OpenSSL funkcijai `SSL_CTX_set_ciphersuites()`).	Šifru komplektu atlases kritēriji TLS 1.3 Tikai OpenSSL 1.1.1 vai jaunāka versija.

Pārrakstīšanas tvērums	Parametrs	Vērtība	Apraksts
	TLSCipherAll	Derīgas OpenSSL cipher strings TLS 1.2 vai derīgas GnuTLS priority strings. To vērtības tiek nodotas attiecīgi funkcijām `SSL_CTX_set_cipher_list()` vai `gnutls_priority_init()`.	Šifru komplektu atlases kritēriji TLS 1.2/1.3 (GnuTLS), TLS 1.2 (OpenSSL)

Lai pārrakstītu šifru komplektu atlasi zabbix_get un zabbix_sender utilītās, izmantojiet komandrindas parametrus:

- `--tls-cipher13`
- `--tls-cipher`

Jaunie parametri nav obligāti. Ja parametrs nav norādīts, tiek izmantota iekšējā noklusējuma vērtība. Ja parametrs ir definēts, tas nedrīkst būt tukšs.

Ja TLSCipher* vērtības iestatīšana kriptogrāfijas bibliotēkā neizdodas, serveris, starpniekserveris vai aģents netiks palaists un tiks reģistrēta kļūda.

Ir svarīgi saprast, kad katrs parametrs ir piemērojams.

Izejošie savienojumi

Vienkāršākais gadījums ir izejošie savienojumi:

- Izejošajiem savienojumiem ar sertifikātu izmantojiet TLSCipherCert13 vai TLSCipherCert
- Izejošajiem savienojumiem ar PSK izmantojiet TLSCipherPSK13 vai TLSCipherPSK
- zabbix_get un zabbix_sender utilītu gadījumā var izmantot komandrindas parametrus `--tls-cipher13` vai `--tls-cipher` (šifrēšana ir nepārprotami norādīta ar parametru `--tls-connect`)

Ienākošie savienojumi

Ar ienākošajiem savienojumiem tas ir nedaudz sarežģītāk, jo noteikumi ir atkarīgi no komponentiem un konfigurācijas.

Zabbix **aģentam**:

Aģenta savienojuma iestatīšana	Šifru konfigurācija
TLSConnect=cert	TLSCipherCert, TLSCipherCert13
TLSConnect=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert	TLSCipherCert, TLSCipherCert13
TLSAccept=psk	TLSCipherPSK, TLSCipherPSK13
TLSAccept=cert,psk	TLSCipherAll, TLSCipherAll13

Zabbix **serverim** un **starpniekserverim**:

Savienojuma iestatīšana	Šifru konfigurācija
Izejošie savienojumi, izmantojot PSK	TLSCipherPSK, TLSCipherPSK13
Ienākošie savienojumi, izmantojot sertifikātus	TLSCipherAll, TLSCipherAll13
Ienākošie savienojumi, izmantojot PSK, ja serverim nav sertifikāta	TLSCipherPSK, TLSCipherPSK13
Ienākošie savienojumi, izmantojot PSK, ja serverim ir sertifikāts	TLSCipherAll, TLSCipherAll13

Abās iepriekš redzamajās tabulās var saskatīt noteiktu likumsakarību:

- TLSCipherAll un TLSCipherAll13 var norādīt tikai tad, ja tiek izmantots apvienots uz sertifikātiem **un** PSK balstītu šifru komplektu saraksts. Tas notiek divos gadījumos: serveris (starpniekserveris) ar konfigurētu sertifikātu (PSK šifru komplekti vienmēr tiek konfigurēti serverī, starpniekserverī, ja kriptogrāfijas bibliotēka atbalsta PSK), aģents, kas konfigurēts pieņemt gan uz sertifikātiem, gan uz PSK balstītus ienākošos savienojumus
- citos gadījumos pietiek ar TLSCipherCert* un/vai TLSCipherPSK*

Tālāk redzamajās tabulās ir parādītas iebūvētās noklusējuma `TLSCipher*` vērtības. Tās var būt labs sākumpunkts jūsu pašu pielāgotajām vērtībām.

Parametrs	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

Parametrs	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] Noklusējuma vērtības atšķiras vecākām OpenSSL versijām (1.0.1, 1.0.2, 1.1.0), LibreSSL un gadījumos, ja OpenSSL ir kompilēts bez PSK atbalsta.

Lietotāja konfigurētu šifru komplektu piemēri

Tālāk skatiet šādus lietotāja konfigurētu šifru komplektu piemērus:

- Šifru virkņu testēšana un tikai PFS šifru komplektu atļaušana
- Pārslēgšanās no AES128 uz AES256

Šifru komplektu virkņu testēšana un tikai PFS šifru komplektu atļaušana

Lai redzētu, kuri šifru komplekti ir atlasīti, konfigurācijas failā ir jāiestata 'DebugLevel=4' vai jāizmanto `-vv` opcija zabbix_sender.

Pirms iegūstat vajadzīgos šifru komplektus, var būt nepieciešama neliela eksperimentēšana ar `TLSCipher*` parametriem. Nav ērti vairākkārt restartēt Zabbix serveri, starpniekserveri vai aģentu tikai tādēļ, lai pielāgotu `TLSCipher*` parametrus. Ērtākas iespējas ir izmantot zabbix_sender vai `openssl` komandu. Parādīsim abus variantus.

1. Izmantojot zabbix_sender.

Izveidosim testa konfigurācijas failu, piemēram, `/home/zabbix/test.conf`, ar `zabbix_agentd.conf` faila sintaksi:

```
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
```

Šim piemēram ir nepieciešami derīgi CA un aģenta sertifikāti, kā arī PSK. Pielāgojiet sertifikātu un PSK failu ceļus un nosaukumus savai videi.

Ja neizmantojat sertifikātus, bet tikai PSK, varat izveidot vienkāršāku testa failu:

```
Hostname=nonexisting
ServerActive=nonexisting

TLSConnect=psk
TLSPSKIdentity=nonexisting
TLSPSKFile=/home/zabbix/agentd.psk
```

Atlasītos šifru komplektus var redzēt, palaižot zabbix_sender (piemērs kompilēts ar 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
```

Šeit redzat pēc noklusējuma atlasītos šifru komplektus. Šīs noklusējuma vērtības ir izvēlētas, lai nodrošinātu savietojamību ar Zabbix aģentiem, kas darbojas sistēmās ar vecākām OpenSSL versijām (sākot no 1.0.1).

Jaunākās sistēmās varat izvēlēties pastiprināt drošību, atļaujot tikai dažus šifru komplektus, piemēram, tikai šifru komplektus ar PFS (Perfect Forward Secrecy). Mēģināsim atļaut tikai šifru komplektus ar PFS, izmantojot `TLSCipher*` parametrus.

> **Attention:**
> Rezultāts nebūs savietojams ar sistēmām, kas izmanto OpenSSL 1.0.1 un 1.0.2, ja tiek izmantots PSK. Uz sertifikātiem balstīta šifrēšana būtu jādarbojas.

Pievienojiet `test.conf` konfigurācijas failam divas rindas:

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
```

un testējiet vēlreiz:

```
$ 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
```

"certificate ciphersuites" un "PSK ciphersuites" saraksti ir mainījušies — tie ir īsāki nekā iepriekš un satur tikai TLS 1.3 šifru komplektus un TLS 1.2 ECDHE-* šifru komplektus, kā bija sagaidāms.

2. TLSCipherAll un TLSCipherAll13 nevar testēt ar zabbix_sender; tie neietekmē iepriekšējā piemērā parādīto "certificate and PSK ciphersuites" vērtību. Lai pielāgotu TLSCipherAll un TLSCipherAll13, jums jāeksperimentē ar aģentu, starpniekserveri vai serveri.

Tātad, lai atļautu tikai PFS šifru komplektus, var būt nepieciešams pievienot līdz trim parametriem

```
TLSCipherCert=EECDH+aRSA+AES128
TLSCipherPSK=kECDHEPSK+AES128
TLSCipherAll=EECDH+aRSA+AES128:kECDHEPSK+AES128
```

failiem zabbix_agentd.conf, zabbix_proxy.conf un zabbix_server_conf, ja katram no tiem ir konfigurēts sertifikāts un aģentam ir arī PSK.

Ja jūsu Zabbix vide izmanto tikai uz PSK balstītu šifrēšanu un neizmanto sertifikātus, tad nepieciešams tikai viens:

```
TLSCipherPSK=kECDHEPSK+AES128
```

Tagad, kad saprotat, kā tas darbojas, varat testēt šifru komplektu atlasi pat ārpus Zabbix, izmantojot `openssl` komandu. Pārbaudīsim visas trīs `TLSCipher*` parametru vērtības:

```
$ 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
```

Iespējams, jūs dosiet priekšroku `openssl ciphers` ar `-V` opciju, lai iegūtu detalizētāku izvadi:

```
$ 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
```

Līdzīgi varat testēt GnuTLS prioritāšu virknes:

```
$ 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
```

Pārslēgšanās no AES128 uz AES256

Zabbix datiem kā iebūvēto noklusējuma algoritmu izmanto AES128. Pieņemsim, ka jūs izmantojat sertifikātus un vēlaties pārslēgties uz AES256, izmantojot OpenSSL 1.1.1.

To var panākt, pievienojot attiecīgos parametrus `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:
Lai gan tiks izmantoti tikai ar sertifikātiem saistītie šifru komplekti, ir definēti arī `TLSCipherPSK*` parametri, lai izvairītos no to noklusējuma vērtībām, kas plašākas savietojamības dēļ ietver mazāk drošus šifrus. PSK šifru komplektus nevar pilnībā atspējot serverī/starpniekserverī.

Un `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 Sertifikātu izmantošana

Pārskats

Zabbix var izmantot RSA sertifikātus PEM formātā, ko parakstījusi publiska vai iekšēja sertifikātu iestāde (CA).

Sertifikātu verifikācija tiek veikta, salīdzinot ar iepriekš konfigurētu CA sertifikātu. Pēc izvēles var izmantot sertifikātu atsaukšanas sarakstus (CRL).

Katram Zabbix komponentam var būt konfigurēts tikai viens sertifikāts.

Plašāku informāciju par iekšējas CA iestatīšanu un uzturēšanu, sertifikātu pieprasījumu ģenerēšanu un parakstīšanu, kā arī sertifikātu atsaukšanu skatiet pamācībās, piemēram, OpenSSL PKI Tutorial v2.0.

Rūpīgi izvērtējiet un pārbaudiet savus sertifikātu paplašinājumus. Plašāku informāciju skatiet sadaļā X.509 v3 sertifikātu paplašinājumu izmantošanas ierobežojumi.

Sertifikātu konfigurācijas parametri

Tālāk norādītie konfigurācijas parametri tiek atbalstīti sertifikātu iestatīšanai Zabbix komponentos.

Parametrs	Obligāts	Apraksts
TLSCAFile	jā	Pilns ceļš līdz failam, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei. Ja tiek izmantota sertifikātu ķēde ar vairākiem elementiem, sakārtojiet sertifikātus tā, lai vispirms būtu zemāka līmeņa CA sertifikāti, pēc tam augstāka līmeņa CA sertifikāti. Viena faila sastāvā var iekļaut vairāku CA sertifikātus.

Parametrs	Obligāts	Apraksts
TLSCRLFile	nē	Pilns ceļš līdz failam, kas satur sertifikātu atsaukšanas sarakstus (CRL).
TLSCertFile	jā	Pilns ceļš līdz failam, kas satur sertifikātu. Ja tiek izmantota sertifikātu ķēde ar vairākiem elementiem, sakārtojiet sertifikātus tā, lai vispirms būtu servera, starpniekservera vai aģenta sertifikāts, pēc tam zemāka līmeņa CA sertifikāti, un noslēgumā augstāka līmeņa CA sertifikāti.
TLSKeyFile	jā	Pilns ceļš līdz failam, kas satur privāto atslēgu. Nodrošiniet, lai šo failu varētu lasīt tikai Zabbix lietotājs, iestatot atbilstošas piekļuves tiesības.
TLSServerCertIssuer	nē	Atļautais servera sertifikāta izdevējs.
TLSServerCertSubject	nē	Atļautais servera sertifikāta subjekts.

Konfigurācijas piemēri

Pēc nepieciešamo sertifikātu iestatīšanas konfigurējiet Zabbix komponentus, lai izmantotu uz sertifikātiem balstītu šifrēšanu.

Tālāk ir sniegti detalizēti soļi, kā konfigurēt:

- Zabbix serveris
- Zabbix starpniekserveris
- Zabbix aģents

Zabbix serveris

1. Sagatavojiet CA sertifikāta failu.

Lai verificētu otras puses sertifikātus, Zabbix serverim jābūt piekļuvei failam, kas satur augstākā līmeņa, pašparakstītos saknes CA sertifikātus. Piemēram, ja ir nepieciešami sertifikāti no divām neatkarīgām saknes CA, ievietojiet tos failā `/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. Ievietojiet Zabbix servera sertifikātu/sertifikātu ķēdi failā, piemēram, `/home/zabbix/zabbix_server.crt`. Pirmais sertifikāts ir Zabbix servera sertifikāts, kam seko starpniek-CA sertifikāts:

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1 (0x1)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Zabbix server
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Basic Constraints:
                CA:FALSE
            ...
-----BEGIN CERTIFICATE-----
MIIECDCCAvCgAwIBAgIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixk
...
h02u1GHiy46GI+xfR3LsPwFKlkTaaLaL/6aaoQ==
-----END CERTIFICATE-----
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Root1 CA
        ...
        Subject: DC=com, DC=zabbix, O=Zabbix SIA, OU=Development group, CN=Signing CA
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
            ...
        X509v3 extensions:
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Basic Constraints: critical
                CA:TRUE, pathlen:0
        ...
-----BEGIN CERTIFICATE-----
MIID4TCCAsmgAwIBAgIBAjANBgkqhkiG9w0BAQUFADB+MRMwEQYKCZImiZPyLGQB
...
dyCeWnvL7u5sd6ffo8iRny0QzbHKmQt/wUtcVIvWXdMIFJMOHw==
-----END CERTIFICATE-----
```

Note:
Izmantojiet tikai iepriekš minētos atribūtus gan klienta, gan servera sertifikātiem, lai neietekmētu sertifikātu verificēšanas procesu. Piemēram, OpenSSL var neizdoties izveidot šifrētu savienojumu, ja tiek izmantoti *X509v3 Subject Alternative Name* vai *Netscape Cert Type* paplašinājumi. Plašāku informāciju skatiet sadaļā Ierobežojumi X.509 v3 sertifikātu paplašinājumu izmantošanā.

3. Ievietojiet Zabbix servera privāto atslēgu failā, piemēram, `/home/zabbix/zabbix_server.key`:

```
-----BEGIN PRIVATE KEY-----
MIIEwAIBADANBgkqhkiG9w0BAQEFAASCBKowggSmAgEAAoIBAQC9tIXIJoVnNXDl
...
IJLkhbybBYEf47MLhffWa7XvZTY=
-----END PRIVATE KEY-----
```

4. Rediģējiet TLS konfigurācijas parametrus Zabbix servera konfigurācijas failā:

```
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_server.crt
TLSKeyFile=/home/zabbix/zabbix_server.key
```

Zabbix starpniekserveris

1. Sagatavojiet failus ar augstākā līmeņa CA sertifikātiem, Zabbix starpniekservera sertifikātu/sertifikātu ķēdi un privāto atslēgu, kā aprakstīts sadaļā Zabbix serveris. Pēc tam attiecīgi rediģējiet parametrus `TLSCAFile`, `TLSCertFile` un `TLSKeyFile` Zabbix starpniekservera konfigurācijas failā.

2. Rediģējiet papildu TLS parametrus Zabbix starpniekservera konfigurācijas failā:

- Aktīvam starpniekserverim: `TLSConnect=cert`
- Pasīvam starpniekserverim: `TLSAccept=cert`

Note:
Lai uzlabotu starpniekservera drošību, varat iestatīt arī parametrus `TLSServerCertIssuer` un `TLSServerCertSubject`. Papildinformāciju skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.

TLS parametri galīgajā starpniekservera konfigurācijas failā var izskatīties šādi:

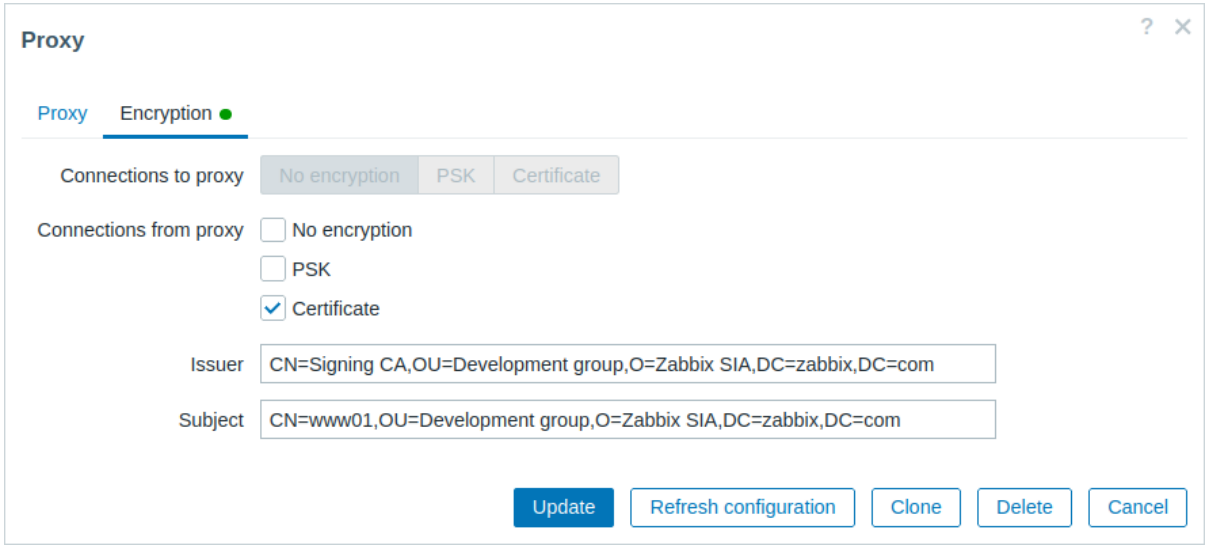
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix server,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_proxy.crt
TLSKeyFile=/home/zabbix/zabbix_proxy.key
```

3. Konfigurējiet šim starpniekserverim šifrēšanu Zabbix lietotāja saskarnē:

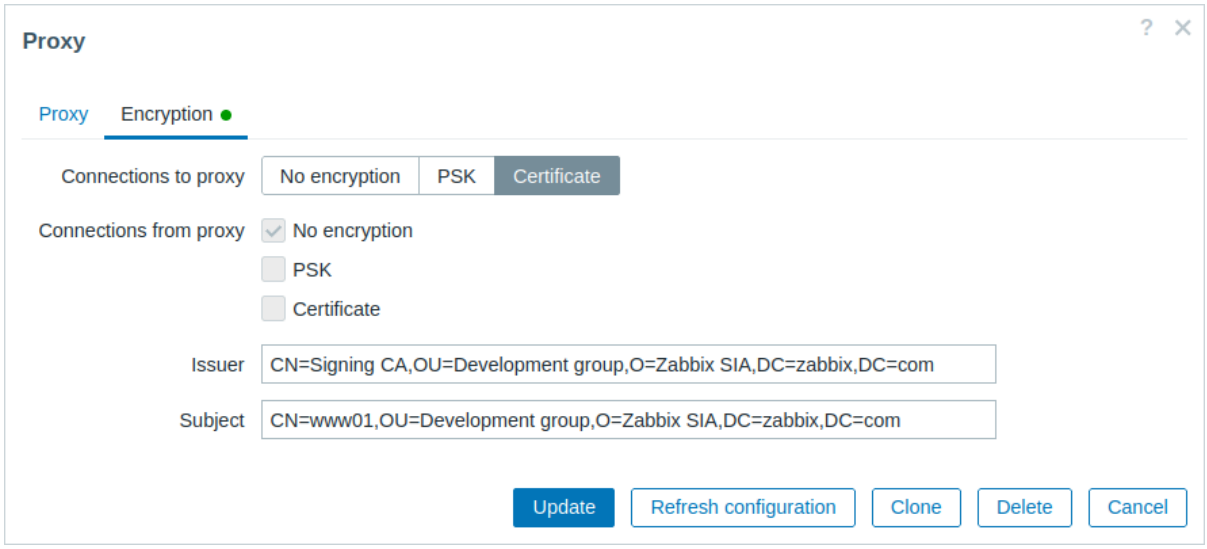
- Dodieties uz: *Administration* → *Proxies*.
- Atlasiet starpniekserveri un noklikšķiniet uz cilnes *Encryption*.

Tālāk redzamajos piemēros lauki *Issuer* un *Subject* ir aizpildīti. Papildinformāciju par to, kāpēc un kā izmantot šos laukus, skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.

Aktīvam starpniekserverim:



Pasīvam starpniekserverim:



Zabbix aģents

1. Sagatavojiet failus ar augstākā līmeņa CA sertifikātiem, Zabbix aģenta sertifikātu/sertifikātu ķēdi un privāto atslēgu, kā aprakstīts sadaļā Zabbix serveris. Pēc tam attiecīgi rediģējiet parametrus `TLSCAFile`, `TLSCertFile` un `TLSKeyFile` Zabbix aģenta konfigurācijas failā.

2. Rediģējiet papildu TLS parametrus Zabbix aģenta konfigurācijas failā:

- Aktīvajam aģentam: `TLSConnect=cert`
- Pasīvajam aģentam: `TLSAccept=cert`

Note:
Lai uzlabotu aģenta drošību, varat iestatīt parametrus `TLSServerCertIssuer` un `TLSServerCertSubject`. Plašāku informāciju skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.

TLS parametri galīgajā aģenta konfigurācijas failā var izskatīties šādi. Ņemiet vērā, ka piemērā ir pieņemts, ka hosts tiek uzraudzīts ar starpniekserveri, tāpēc tas ir norādīts kā sertifikāta subjekts:

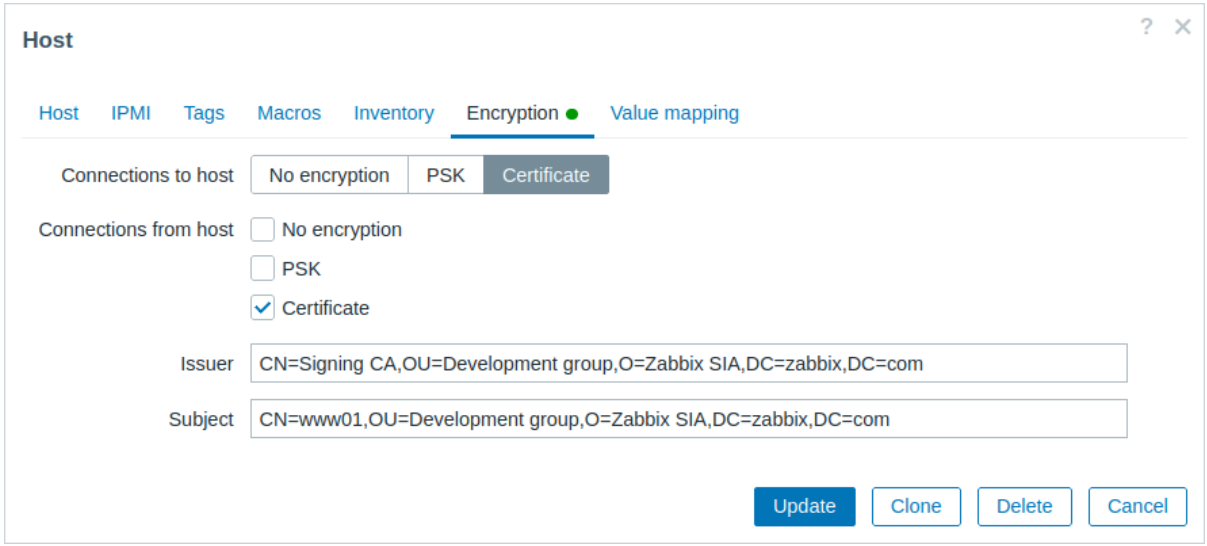
```
TLSConnect=cert
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSServerCertIssuer=CN=Signing CA,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSServerCertSubject=CN=Zabbix proxy,OU=Development group,O=Zabbix SIA,DC=zabbix,DC=com
TLSCertFile=/home/zabbix/zabbix_agentd.crt
TLSKeyFile=/home/zabbix/zabbix_agentd.key
```

3. Konfigurējiet šifrēšanu Zabbix lietotāja saskarnē hostam, kuru uzrauga šis aģents.

- Dodieties uz: *Datu ievākšana* → *Hosti*.

- Atlasiet hostu un noklikšķiniet uz cilnes *Šifrēšana*.

Tālāk redzamajā piemērā lauki *Izdevējs* un *Subjekts* ir aizpildīti. Plašāku informāciju par to, kāpēc un kā izmantot šos laukus, skatiet sadaļā Atļautā sertifikāta izdevēja un subjekta ierobežošana.



Zabbix tīmekļa serviss

1. Sagatavojiet failus ar augstākā līmeņa CA sertifikātiem, Zabbix tīmekļa servisa sertifikātu/sertifikātu ķēdi un privāto atslēgu, kā aprakstīts sadaļā Zabbix serveris. Pēc tam attiecīgi rediģējiet parametrus `TLSCAFile`, `TLSCertFile` un `TLSKeyFile` Zabbix tīmekļa servisa konfigurācijas failā.

2. Rediģējiet papildu TLS parametru Zabbix tīmekļa servisa konfigurācijas failā: `TLSAccept=cert`

TLS parametri gala tīmekļa servisa konfigurācijas failā var izskatīties šādi:

```
TLSAccept=cert
TLSCAFile=/home/zabbix/zabbix_ca_file.crt
TLSCertFile=/home/zabbix/zabbix_web_service.crt
TLSKeyFile=/home/zabbix/zabbix_web_service.key
```

3. Konfigurējiet Zabbix serveri, lai tas izveidotu savienojumu ar TLS konfigurēto Zabbix tīmekļa servisu, rediģējot parametru `WebServiceURL` Zabbix servera konfigurācijas failā:

```
WebServiceURL=https://example.com:443/report
```

Atļautā sertifikāta izdevēja un subjekta ierobežošana

Kad divi Zabbix komponenti (piemēram, serveris un aģents) izveido TLS savienojumu, tie pārbauda viens otra sertifikātus. Ja otras puses sertifikātu ir parakstījis uzticams CA (ar iepriekš konfigurētu augstākā līmeņa sertifikātu `TLSCAFile`), tas ir derīgs, tam nav beidzies derīguma termiņš un tas iztur citas pārbaudes, tad saziņa starp komponentiem var turpināties. Šajā vienkāršākajā gadījumā sertifikāta izdevējs un subjekts netiek pārbaudīti.

Tomēr tas rada risku: ikviens, kam ir derīgs sertifikāts, var uzdoties par jebkuru citu (piemēram, hosta sertifikātu var izmantot, lai uzdotos par serveri). Lai gan tas var būt pieņemami nelielās vidēs, kur sertifikātus paraksta īpašs iekšējs CA un uzdošanās risks ir zems, lielākās vai pret drošību jutīgākās vidēs ar to var nepietikt.

Ja jūsu augstākā līmeņa CA izsniedz sertifikātus, kurus Zabbix nevajadzētu pieņemt, vai ja vēlaties samazināt uzdošanās risku, varat ierobežot atļautos sertifikātus, norādot to izdevēju un subjektu.

Piemēram, Zabbix starpniekservera konfigurācijas failā var norādīt:

```
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
```

Izmantojot šos iestatījumus, aktīvs starpniekserveris nesazināsies ar Zabbix serveri, kura sertifikātam ir cits izdevējs vai subjekts. Līdzīgi arī pasīvs starpniekserveris nepieņems pieprasījumus no šāda servera.

`Issuer` un `Subject` virkņu atbilstības noteikumi

`Issuer` un `Subject` virkņu atbilstības noteikumi ir šādi:

- `Issuer` un `Subject` virknes tiek pārbaudītas neatkarīgi. Abas ir neobligātas.
- Nenorādīta virkne nozīmē, ka tiek pieņemta jebkura virkne.
- Virknes tiek salīdzinātas *kā ir*, un tām ir jāsakrīt precīzi.

- Tiek atbalstītas UTF-8 rakstzīmes. Tomēr aizstājējzīmes (*) vai regulārās izteiksmes netiek atbalstītas.
- Ir ieviestas šādas RFC 4514 prasības — rakstzīmes, kurām nepieciešama atsoļošana (ar atpakaļslīpsvītru '\', U+005C):
  - jebkurā virknes vietā: '"' (U+0022), '+' (U+002B), ',' (U+002C), ';' (U+003B), '<' (U+003C), '>' (U+003E), '\\' (U+005C);
  - virknes sākumā: atstarpe (' ', U+0020) vai numura zīme ('#', U+0023);
  - virknes beigās: atstarpe (' ', U+0020).
- Nulles rakstzīmes (U+0000) netiek atbalstītas. Ja tiek konstatēta nulles rakstzīme, atbilstības pārbaude neizdosies.
- RFC 4517 un RFC 4518 standarti netiek atbalstīti.

Piemēram, ja `Issuer` un `Subject` organizācijas (`O`) virknēs ir atstarpes beigās un `Subject` organizatoriskās vienības (`OU`) virknē ir dubultpēdiņas, šīs rakstzīmes ir jāatsoļo:

```
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
```

Lauku secība un formatējums

Zabbix ievēro RFC 4514 ieteikumus, kas šiem laukiem nosaka "apgrieztu" secību, sākot ar zemākā līmeņa laukiem (`CN`), turpinot ar vidējā līmeņa laukiem (`OU`, `O`) un noslēdzot ar augstākā līmeņa laukiem (`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
```

Turpretī OpenSSL pēc noklusējuma attēlo `Issuer` un `Subject` virknes secībā no augstākā līmeņa uz zemāko līmeni. Tālāk redzamajā piemērā `Issuer` un `Subject` lauki sākas ar augstākā līmeņa (`DC`) lauku un beidzas ar zemākā līmeņa (`CN`) lauku. Arī formatējums ar atstarpēm un lauku atdalītājiem atšķiras atkarībā no izmantotajām opcijām, tādēļ tas neatbildīs Zabbix prasītajam formātam.

```
$ 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
```

Lai *Issuer* un *Subject* virknes pareizi noformatētu lietošanai Zabbix, izsauciet OpenSSL ar šādām opcijām:

```
$ 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
```

Tad izvade būs apgrieztā secībā, ar komatiem atdalīta, un to varēs izmantot Zabbix konfigurācijas failos un lietotāja saskarnē:

```
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
```

X.509 v3 sertifikātu paplašinājumu izmantošanas ierobežojumi

Ieviešot X.509 v3 sertifikātus Zabbix vidē, daži paplašinājumi var netikt pilnībā atbalstīti vai var izraisīt nekonsekventu darbību.

Subject Alternative Name paplašinājums

Zabbix neatbalsta *Subject Alternative Name* paplašinājumu, ko izmanto alternatīvu DNS nosaukumu, piemēram, IP adrešu vai e-pasta adrešu, norādīšanai. Zabbix var pārbaudīt tikai sertifikāta *Subject* laukā esošo vērtību (skatiet Atļautā sertifikāta izdevēja un subjekta ierobežošana). Ja sertifikāti ietver `subjectAltName` lauku, sertifikāta validācijas rezultāts var atšķirties atkarībā no konkrētajiem kriptogrāfiskajiem rīkkopām, kas izmantotas Zabbix komponentu kompilēšanai. Rezultātā Zabbix var vai nu pieņemt, vai noraidīt sertifikātus atkarībā no šo kombināciju īpatnībām.

Extended Key Usage paplašinājums

Zabbix atbalsta *Extended Key Usage* paplašinājumu. Tomēr, ja tas tiek izmantots, parasti ir nepieciešams norādīt abus atribūtus — *clientAuth* (TLS WWW klienta autentifikācijai) un *serverAuth* (TLS WWW servera autentifikācijai). Piemēram:

- Pasīvajās pārbaudēs, kur Zabbix aģents darbojas kā TLS serveris, aģenta sertifikātā jābūt iekļautam atribūtam *serverAuth*.
- Aktīvajām pārbaudēm, kur aģents darbojas kā TLS klients, aģenta sertifikātā jābūt iekļautam atribūtam *clientAuth*.

Lai gan GnuTLS var izdot brīdinājumu par atslēgas lietojuma pārkāpumiem, tas parasti ļauj saziņai turpināties, neskatoties uz šiem brīdinājumiem.

Name Constraints paplašinājums

Name Constraints paplašinājuma atbalsts dažādās kriptogrāfiskajās rīkkopās atšķiras. Pārliecinieties, ka jūsu izvēlētā rīkkopa atbalsta šo paplašinājumu. Šis paplašinājums var ierobežot Zabbix iespēju ielādēt CA sertifikātus, ja šī sadaļa ir atzīmēta kā kritiska, atkarībā no izmantotās rīkkopas.

Sertifikātu atsaukšanas saraksti (CRL)

Ja sertifikāts ir kompromitēts, sertifikācijas iestāde (CA) var to atsaukt, iekļaujot sertifikātu sertifikātu atsaukšanas sarakstā (CRL). CRL tiek pārvaldīti, izmantojot konfigurācijas failus, un tos var norādīt ar parametru `TLSCRLFile` servera, starpniekservera un aģenta konfigurācijas failos. Piemēram:

```
TLSCRLFile=/home/zabbix/zabbix_crl_file.crt
```

Šajā gadījumā `zabbix_crl_file.crt` var saturēt CRL no vairākām CA, un tas var izskatīties šādi:

```
-----BEGIN X509 CRL-----
MIIB/DCB5QIBATANBgkqhkiG9w0BAQUFADCBgTETMBEGCgmSJomT8ixkARkWA2Nv
...
treZeUPjb7LSmZ3K2hpbZN7SoOZcAoHQ3GWd9npuctg=
-----END X509 CRL-----
-----BEGIN X509 CRL-----
MIIB+TCB4gIBATANBgkqhkiG9w0BAQUFADB/MRMwEQYKCZImiZPyLGQBGRYDY29t
...
CAEebS2CND3ShBedZ8YSil59O6JvaDP61lR5lNs=
-----END X509 CRL-----
```

CRL fails tiek ielādēts tikai Zabbix palaišanas laikā. Lai atjauninātu CRL, restartējiet Zabbix.

> **Attention:**
> Ja Zabbix komponenti ir kompilēti ar OpenSSL un tiek izmantoti CRL, pārliecinieties, ka katrai augstākā līmeņa un starpposma CA sertifikātu ķēdēs ir atbilstošs CRL (pat ja tas ir tukšs), kas iekļauts `TLSCRLFile`.

2 Iepriekš koplietotu atslēgu izmantošana

Pārskats

Katra iepriekš koplietotā atslēga (PSK) Zabbix sistēmā faktiski ir pāris no:

- ne-slepenas PSK identitātes virknes,
- slepenas PSK virknes vērtības.

PSK identitātes virkne ir netukša UTF-8 virkne. Piemēram, "PSK ID 001 Zabbix agentd". Tas ir unikāls nosaukums, ar kuru šo konkrēto PSK izmanto Zabbix komponenti. Neievietojiet sensitīvu informāciju PSK identitātes virknē — tā tiek pārraidīta tīklā nešifrētā veidā.

PSK vērtība ir grūti uzminama heksadecimālu ciparu virkne, piemēram, "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d0832

Izmēra ierobežojumi

Zabbix ir noteikti PSK identitātes un vērtības izmēra ierobežojumi; dažos gadījumos kriptogrāfijas bibliotēkai var būt zemāks ierobežojums:

Komponents	PSK identitātes maksimālais izmērs	PSK vērtības minimālais izmērs	PSK vērtības maksimālais izmērs
Zabbix	128 UTF-8 rakstzīmes	128 biti (16 baitu PSK, ievadīts kā 32 heksadecimāli cipari)	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimāli cipari)
GnuTLS	128 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimāli cipari)

Komponents	PSK identitātes maksimālais izmērs	PSK vērtības minimālais izmērs	PSK vērtības maksimālais izmērs
OpenSSL 1.0.x, 1.1.0	127 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimāli cipari)
OpenSSL 1.1.1	127 baiti (var ietvert UTF-8 rakstzīmes)	-	512 biti (64 baitu PSK, ievadīts kā 128 heksadecimāli cipari)
OpenSSL 1.1.1a un jaunākas versijas	127 baiti (var ietvert UTF-8 rakstzīmes)	-	2048 biti (256 baitu PSK, ievadīts kā 512 heksadecimāli cipari)

Attention:
Zabbix lietotāja saskarne ļauj konfigurēt līdz 128 rakstzīmēm garu PSK identitātes virkni un 2048 bitus garu PSK neatkarīgi no izmantotajām kriptogrāfijas bibliotēkām.
 Ja daži Zabbix komponenti atbalsta zemākus ierobežojumus, lietotāja pienākums ir konfigurēt PSK identitāti un vērtību ar šiem komponentiem atļauto garumu.
 Garuma ierobežojumu pārsniegšana izraisa sakaru kļūmes starp Zabbix komponentiem.

Pirms Zabbix serveris izveido savienojumu ar aģentu, izmantojot PSK, serveris datubāzē (faktiski konfigurācijas kešatmiņā) sameklē šim aģentam konfigurēto PSK identitāti un PSK vērtību. Saņemot savienojumu, aģents izmanto savā konfigurācijas failā norādīto PSK identitāti un PSK vērtību. Ja abām pusēm ir vienāda PSK identitātes virkne un PSK vērtība, savienojums var būt veiksmīgs.

Attention:
Katrai PSK identitātei jābūt sasaistītai tikai ar vienu vērtību. Lietotāja pienākums ir nodrošināt, lai nebūtu divu PSK ar vienādu identitātes virkni, bet atšķirīgām vērtībām. Šīs prasības neievērošana var izraisīt neparedzamas kļūdas vai sakaru traucējumus starp Zabbix komponentiem, kas izmanto PSK ar šo PSK identitātes virkni.

PSK ģenerēšana

Piemēram, 256 bitu (32 baitu) PSK var ģenerēt, izmantojot šādas komandas:

- ar *OpenSSL*:

```
$ openssl rand -hex 32
af8ced32dfe8714e548694e2d29e1a14ba6fa13f216cb35c19d0feb1084b0429
```

- ar *GnuTLS*:

```
$ psktool -u psk_identity -p database.psk -s 32
Tiek ģenerēta nejauša atslēga lietotājam 'psk_identity'
Atslēga saglabāta database.psk

$ cat database.psk
psk_identity:9b8eafedfaae00cece62e85d5f4792c7d9c9bcc851b23216a1d300311cc4f7cb
```

Ņemiet vērā, ka iepriekš minētais "psktool" ģenerē datubāzes failu ar PSK identitāti un tai piesaistīto PSK. Zabbix PSK failā sagaida tikai PSK, tāpēc identitātes virkne un kols (':') no faila ir jāizņem.

PSK konfigurēšana servera-aģenta saziņai (piemērs)

Aģenta hostā ierakstiet PSK vērtību failā, piemēram, `/home/zabbix/zabbix_agentd.psk`. Failam pirmajā teksta rindā jāsatur PSK, piemēram:

```
1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952
```

Iestatiet piekļuves tiesības PSK failam — tam jābūt lasāmam tikai Zabbix lietotājam.

Rediģējiet TLS parametrus aģenta konfigurācijas failā `zabbix_agentd.conf`, piemēram, iestatiet:

```
TLSConnect=psk
TLSAccept=psk
TLSPSKFile=/home/zabbix/zabbix_agentd.psk
TLSPSKIdentity=PSK 001
```

Aģents izveidos savienojumu ar serveri (aktīvās pārbaudes) un pieņems no servera un `zabbix_get` tikai savienojumus, kuros tiek izmantots PSK. PSK identitāte būs "PSK 001".

Restartējiet aģentu. Tagad varat pārbaudīt savienojumu, izmantojot `zabbix_get`, piemēram:

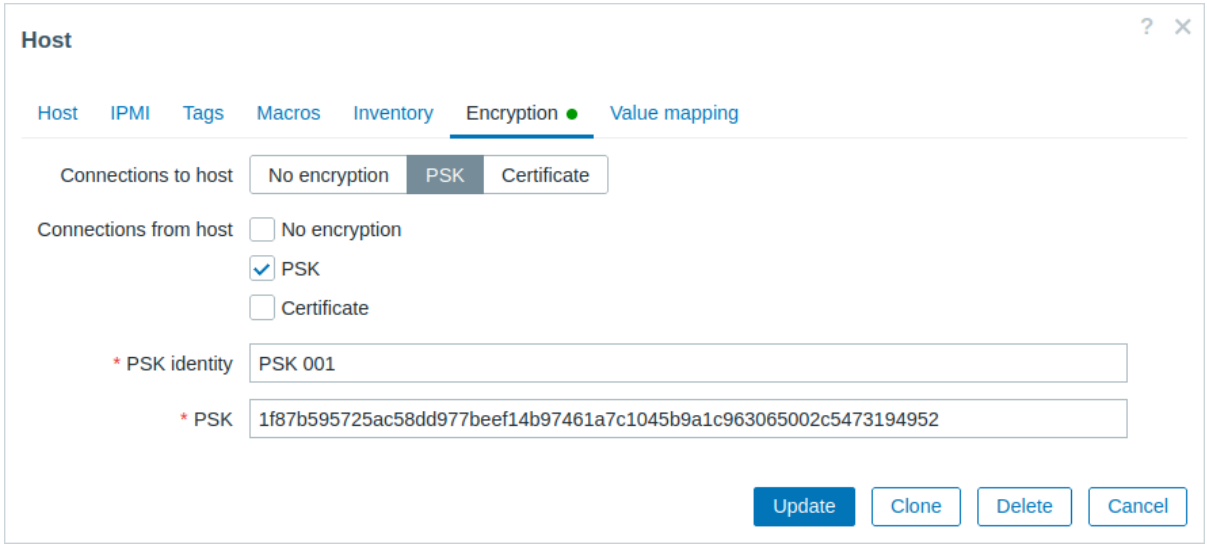
```
zabbix_get -s 127.0.0.1 -k "system.cpu.load[all,avg1]" --tls-connect=psk --tls-psk-identity="PSK 001" --tl
```

(Lai samazinātu dīkstāvi, skatiet, kā mainīt savienojuma tipu sadaļā Savienojuma šifrēšanas pārvaldība).

Konfigurējiet PSK šifrēšanu šim aģentam Zabbix lietotāja saskarnē:

- Dodieties uz: *Datu ievākšana → Hosti*
- Atlasiet hostu un noklikšķiniet uz cilnes **Encryption**

Piemērs:



Visi obligātie ievades lauki ir atzīmēti ar sarkanu zvaigznīti.

Kad konfigurācijas kešatmiņa būs sinhronizēta ar datubāzi, jaunie savienojumi izmantos PSK. Pārbaudiet servera un aģenta žurnālfailus, vai tajos nav kļūdu ziņojumu.

PSK konfigurēšana servera un aktīvā starpniekservera saziņai (piemērs)

Starpniekserverī ierakstiet PSK vērtību failā, piemēram, `/home/zabbix/zabbix_proxy.psk`. Failam pirmajā teksta rindā jāsatur PSK, piemēram:

```
e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9
```

Iestatiet PSK failam piekļuves tiesības — tam jābūt lasāmam tikai Zabbix lietotājam.

Rediģējiet TLS parametrus starpniekservera konfigurācijas failā `zabbix_proxy.conf`, piemēram, iestatiet:

```
TLSConnect=psk
TLSPSKFile=/home/zabbix/zabbix_proxy.psk
TLSPSKIdentity=PSK 002
```

Starpniekserveris izveidos savienojumu ar serveri, izmantojot PSK. PSK identitāte būs "PSK 002".

(Lai samazinātu dīkstāvi, skatiet, kā mainīt savienojuma veidu sadaļā Savienojuma šifrēšanas pārvaldība).

Konfigurējiet PSK šim starpniekserverim Zabbix lietotāja saskarnē. Dodieties uz *Administration→Proxies*, atlasiet starpniekserveri, atveriet cilni "Encryption". Laukā "Connections from proxy" atzīmējiet PSK. Ielīmējiet laukā "PSK identity" "PSK 002" un "e560cb0d918d26d31b4f642181f5f570ad89a390931102e5391d08327ba434e9" laukā "PSK". Noklikšķiniet uz "Update".

Restartējiet starpniekserveri. Tas sāks izmantot ar PSK šifrētus savienojumus ar serveri. Pārbaudiet servera un starpniekservera žurnālfailus, vai tajos nav kļūdu ziņojumu.

Pasīvam starpniekserverim procedūra ir ļoti līdzīga. Vienīgā atšķirība — starpniekservera konfigurācijas failā iestatiet `TLSAccept=psk` un Zabbix lietotāja saskarnē laukā "Connections to proxy" iestatiet PSK.

3 Problēmu novēršana

Vispārīgi ieteikumi

- Sāciet ar izpratni, kurš komponents konkrētajā problēmsituācijā darbojas kā TLS klients un kurš — kā TLS serveris.
 Zabbix serveris, starpniekserveri un aģenti atkarībā no mijiedarbības starp tiem var darboties gan kā TLS serveri, gan kā TLS klienti.
 Piemēram, Zabbix serveris, pieslēdzoties aģentam pasīvai pārbaudei, darbojas kā TLS klients. Aģents ir

TLS servera lomā.
 Zabbix aģents, pieprasot aktīvo pārbaužu sarakstu no starpniekservera, darbojas kā TLS klients. Starpniekserveris ir TLS servera lomā.
 Utilītas `zabbix_get` un `zabbix_sender` vienmēr darbojas kā TLS klienti.

- Zabbix izmanto savstarpēju autentifikāciju.
 Katra puse pārbauda otru pusi un var atteikt savienojumu.
 Piemēram, Zabbix serveris, pieslēdzoties aģentam, var nekavējoties aizvērt savienojumu, ja aģenta sertifikāts nav derīgs. Un otrādi — Zabbix aģents, pieņemot savienojumu no servera, var aizvērt savienojumu, ja serverim aģents neuzticas.
- Pārbaudiet žurnālfailus abās pusēs — TLS klientā un TLS serverī.
 Puse, kas atsaka savienojumu, var reģistrēt precīzu iemeslu, kāpēc tas tika atteikts. Otra puse bieži ziņo tikai par vispārīgu kļūdu (piem., "Connection closed by peer", "connection was non-properly terminated").
- Dažkārt nepareizi konfigurēta šifrēšana rada mulsinošus kļūdu ziņojumus, kas nekādā veidā nenorāda uz patieso cēloni.
 Tālāk esošajās apakšsadaļās mēs cenšamies sniegt (nebūt ne izsmeļošu) ziņojumu un iespējamo cēloņu apkopojumu, kas var palīdzēt problēmu novēršanā.
 Lūdzu, ņemiet vērā, ka dažādi kriptogrāfijas rīkkopas (OpenSSL, GnuTLS) vienādās problēmsituācijās bieži rada atšķirīgus kļūdu ziņojumus.
 Dažkārt kļūdu ziņojumi ir atkarīgi pat no konkrētās kriptogrāfijas rīkkopu kombinācijas abās pusēs.

1 Savienojuma veida vai atļauju problēmas

Serveris ir konfigurēts savienojumam ar aģentu, izmantojot PSK, bet aģents pieņem tikai nešifrētus savienojumus

Servera vai starpniekservera žurnālā (ar *GnuTLS* 3.3.16)

```
Get value from agent failed: zbx_tls_connect(): gnutls_handshake() failed: \
    -110 TLS savienojums tika nepareizi pārtraukts.
```

Servera vai starpniekservera žurnālā (ar *OpenSSL* 1.0.2c)

```
Get value from agent failed: TCP connection successful, cannot establish TLS to [[127.0.0.1]:10050]: \
    Savienojumu aizvēra otra puse. Pārbaudiet atļautos savienojumu veidus un piekļuves tiesības
```

Viena puse izveido savienojumu ar sertifikātu, bet otra puse pieņem tikai PSK, vai otrādi

Jebkurā žurnālā (ar *GnuTLS*):

```
neizdevās pieņemt ienākošo savienojumu: no 127.0.0.1: zbx_tls_accept(): gnutls_handshake() neizdevās:\
    -21 Neizdevās vienoties par atbalstītu šifru kopu.
```

Jebkurā žurnālā (ar *OpenSSL* 1.0.2c):

```
neizdevās pieņemt ienākošo savienojumu: no 127.0.0.1: TLS rokasspiediens atgrieza kļūdas kodu 1:\
    fails .\ssl\s3_srvr.c rinda 1411: error:1408A0C1:SSL routines:ssl3_get_client_hello:no shared cipher:\
    TLS rakstīšanas fatāls brīdinājums "handshake failure"
```

Mēģinājums izmantot Zabbix sender, kas kompilēts ar TLS atbalstu, lai nosūtītu datus uz Zabbix serveri/starpniekserveri, kas kompilēts bez TLS

Savienojošās puses žurnālā:

Linux:

```
...In zbx_tls_init_child()
...OpenSSL bibliotēka (versija OpenSSL 1.1.1  11 Sep 2018) inicializēta
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...End of zbx_tls_connect():FAIL error:'savienojumu aizvēra otra puse'
...vērtības nosūtīšanas kļūda: TCP veiksmīgs, nevar izveidot TLS ar [[localhost]:10051]: savienojumu aizvē
```

Windows:

```
...OpenSSL bibliotēka (versija OpenSSL 1.1.1a  20 Nov 2018) inicializēta
...
...In zbx_tls_connect(): psk_identity:"PSK test sender"
...zbx_psk_client_cb() requested PSK identity "PSK test sender"
...End of zbx_tls_connect():FAIL error:'SSL_connect() I/O kļūda: [0x00000000] Darbība tika veiksmīgi pabei
...vērtības nosūtīšanas kļūda: TCP veiksmīgs, nevar izveidot TLS ar [[192.168.1.2]:10051]: SSL_connect() I
```

Saņemošās puses žurnālā:

```
...neizdevās pieņemt ienākošo savienojumu: no 127.0.0.1: TLS atbalsts netika iekļauts kompilācijā
```

Viena puse izveido savienojumu ar PSK, bet otra puse izmanto LibreSSL vai ir kompilēta bez šifrēšanas atbalsta

LibreSSL neatbalsta PSK.

Savienojumu iniciējošās puses žurnālā:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() I/O error: [0] Success
```

Savienojumu pieņemošās puses žurnālā:

```
...failed to accept an incoming connection: from 192.168.1.2: support for PSK was not compiled in
```

Zabbix lietotāja saskarnē:

```
Get value from aģents failed: TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect()
```

Vienā pusē tiek izveidots savienojums ar PSK, bet otrā pusē tiek izmantots OpenSSL ar atspējotu PSK atbalstu

Savienojumu iniciējošās puses žurnālā:

```
...TCP successful, cannot establish TLS to [[192.168.1.2]:10050]: SSL_connect() set result code to SSL_ERR
```

Savienojumu pieņemošās puses žurnālā:

```
...failed to accept an incoming connection: from 192.168.1.2: TLS handshake set result code to 1: file ssl
```

2 Sertifikātu problēmas

OpenSSL tiek izmantots ar CRL, un kādai CA sertifikātu ķēdē tās CRL nav iekļauts `TLSCRLFile`

TLS servera žurnālā *OpenSSL* partnera gadījumā:

```
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"
```

TLS servera žurnālā *GnuTLS* partnera gadījumā:

```
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 derīguma termiņš ir beidzies vai tas beidzas servera darbības laikā

OpenSSL, servera žurnālā:

- pirms derīguma termiņa beigām:

```
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"
```

- pēc derīguma termiņa beigām:

```
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"
```

Būtība ir tāda, ka ar derīgu CRL atsaukts sertifikāts tiek ziņots kā "certificate revoked". Kad CRL derīguma termiņš beidzas, kļūdas ziņojums mainās uz "certificate expired", kas ir diezgan maldinoši.

GnuTLS, servera žurnālā:

- pirms un pēc derīguma termiņa beigām tas pats:

```
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.
```

Pašparakstīts sertifikāts, nezināms CA

OpenSSL, žurnālā:

```
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"'
```

Tas tika novērots, kad servera sertifikātam kļūdas dēļ bija vienāda Issuer un Subject virkne, lai gan to bija parakstījis CA. Issuer un Subject ir vienādi augšējā līmeņa CA sertifikātā, taču tie nevar būt vienādi servera sertifikātā. (Tas pats attiecas uz starpniekservera un aģenta sertifikātiem.)

Lai pārbaudītu, vai sertifikāts satur vienādus Issuer un Subject ierakstus, izpildiet:

```
openssl x509 -in <yourcertificate.crt> -noout -text
```

Saknes (augšējā līmeņa) sertifikātam ir pieļaujams, ka Issuer un Subject vērtības ir identiskas.

3 PSK problēmas

PSK satur nepāra skaitu heksadecimālo ciparu

Starpniekserveris vai aģents nestartējas, starpniekservera vai aģenta žurnālā ir ziņojums:

```
invalid PSK in file "/home/zabbix/zabbix_proxy.psk"
```

PSK identitātes virkne, kas garāka par 128 baitiem, tiek nodota GnuTLS

TLS klienta puses žurnālā:

```
gnutls_handshake() failed: -110 TLS savienojums tika nepareizi pārtraukts.
```

TLS servera puses žurnālā.

```
gnutls_handshake() failed: -90 Norādītais SRP lietotājvārds nav derīgs.
```

Pārāk gara PSK vērtība, kas izmantota ar OpenSSL 1.1.1

Savienojuma veidojošās puses žurnālā:

```
...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
```

Pieņemošās puses žurnālā:

```
...Message from 123.123.123.123 is missing header. Message ignored.
```

Šī problēma parasti rodas, ja OpenSSL tiek jaunināts no 1.0.x vai 1.1.0 uz 1.1.1 un PSK vērtība ir garāka par 512 bitiem (64 baitu PSK, ievadīta kā 128 heksadecimāli cipari).

Skatiet arī: Vērtību lieluma ierobežojumi

16 Tīmekļa saskarne

Pārskats Lai nodrošinātu ērtu piekļuvi Zabbix no jebkuras vietas un no jebkuras platformas, ir pieejama tīmeklī balstīta lietotāja saskarne.

> **Note:**
> Ja tiek izmantota vairāk nekā viena lietotāja saskarnes instance, pārliecinieties, ka lokalizācijas iestatījumi un bibliotēkas (LDAP, SAML utt.) ir instalētas un konfigurētas vienādi visām lietotāja saskarnēm.

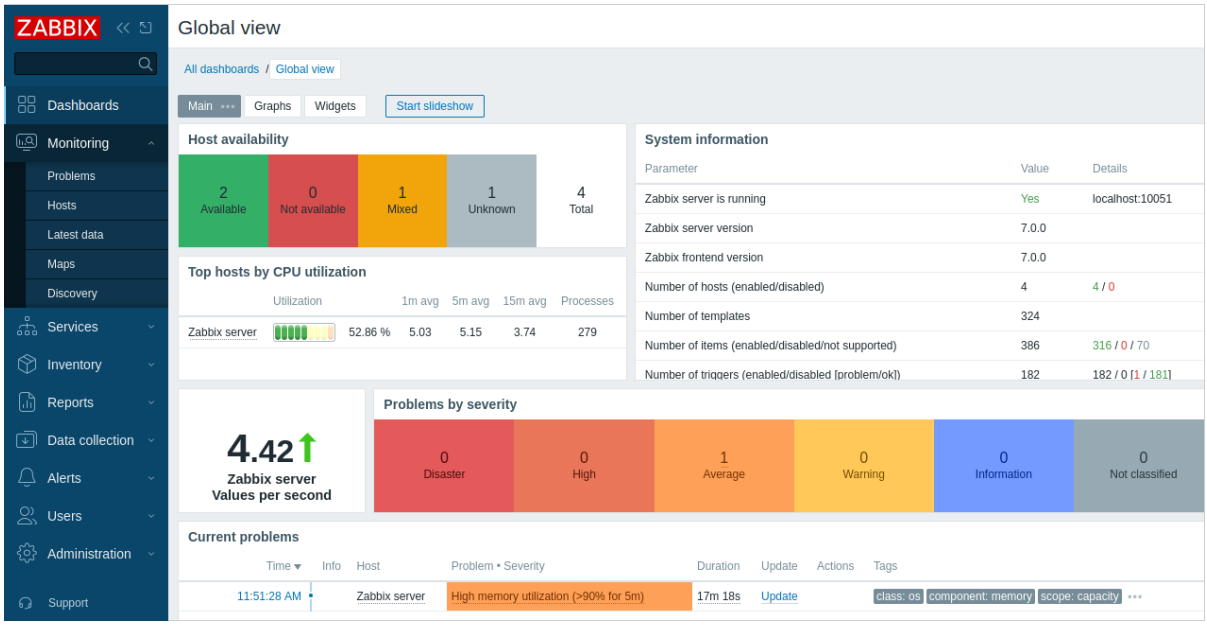
Lietotāja saskarnes palīdzība Palīdzības saite ? ir pieejama Zabbix lietotāja saskarnes formās ar tiešām saitēm uz atbilstošajām dokumentācijas daļām.

1 Izvēlne

Pārskats

Vertikāla izvēlne sānjoslā nodrošina piekļuvi dažādām Zabbix lietotāja saskarne sadaļām.

Izvēlne noklusējuma tēmā ir tumši zila.

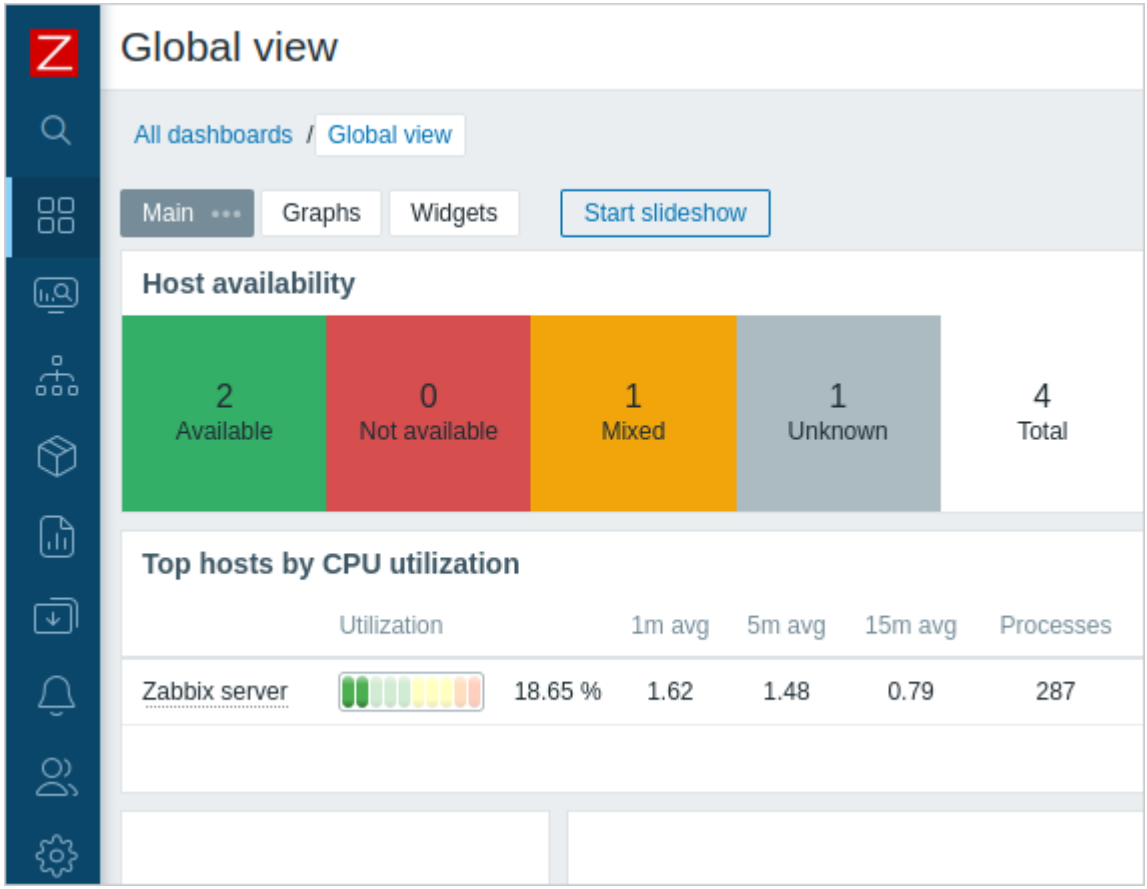


Darbs ar izvēlni

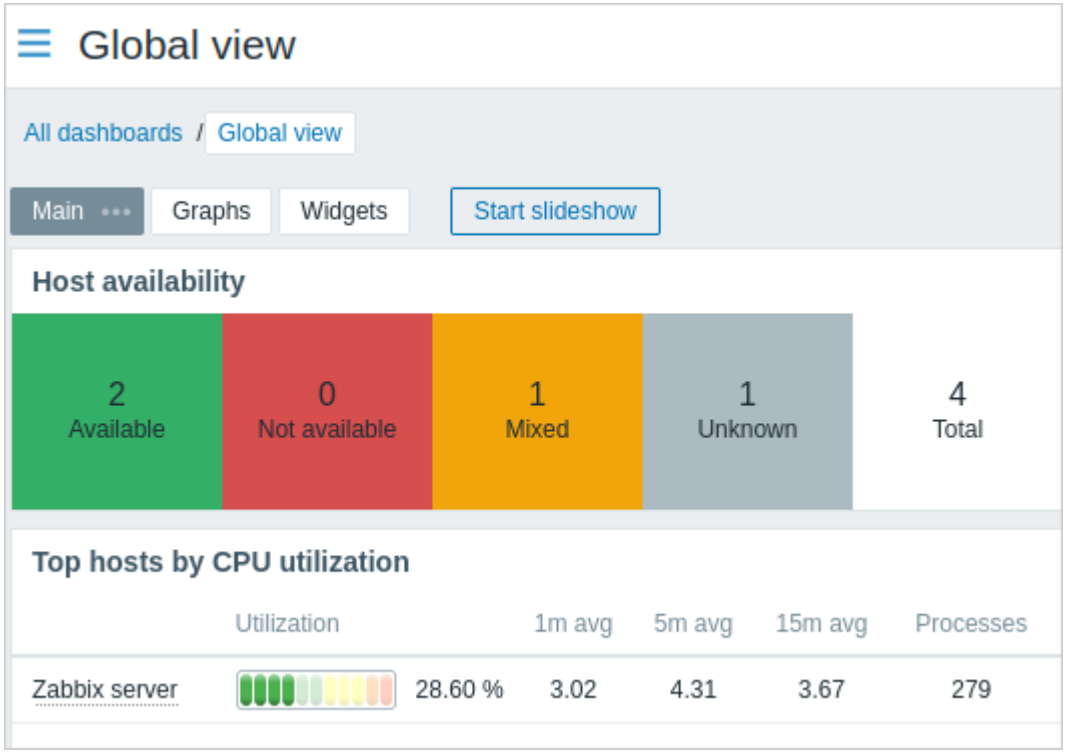
Globālās meklēšanas lodziņš atrodas zem Zabbix logotipa.

Izvēlni var sakļaut vai pilnībā paslēpt:

- Lai sakļautu, noklikšķiniet uz blakus Zabbix logotipam. Sakļautajā izvēlnē ir redzamas tikai ikonas.



- Lai paslēptu, noklikšķiniet uz blakus Zabbix logotipam. Paslēptajā izvēlnē viss ir paslēpts.



Collapsed menu

When the menu is collapsed to icons only, a full menu reappears as soon as the mouse cursor is placed upon it. Note that it reappears over page content; to move page content to the right you have to click on the expand button. If the mouse cursor again is placed outside the full menu, the menu will collapse again after two seconds.

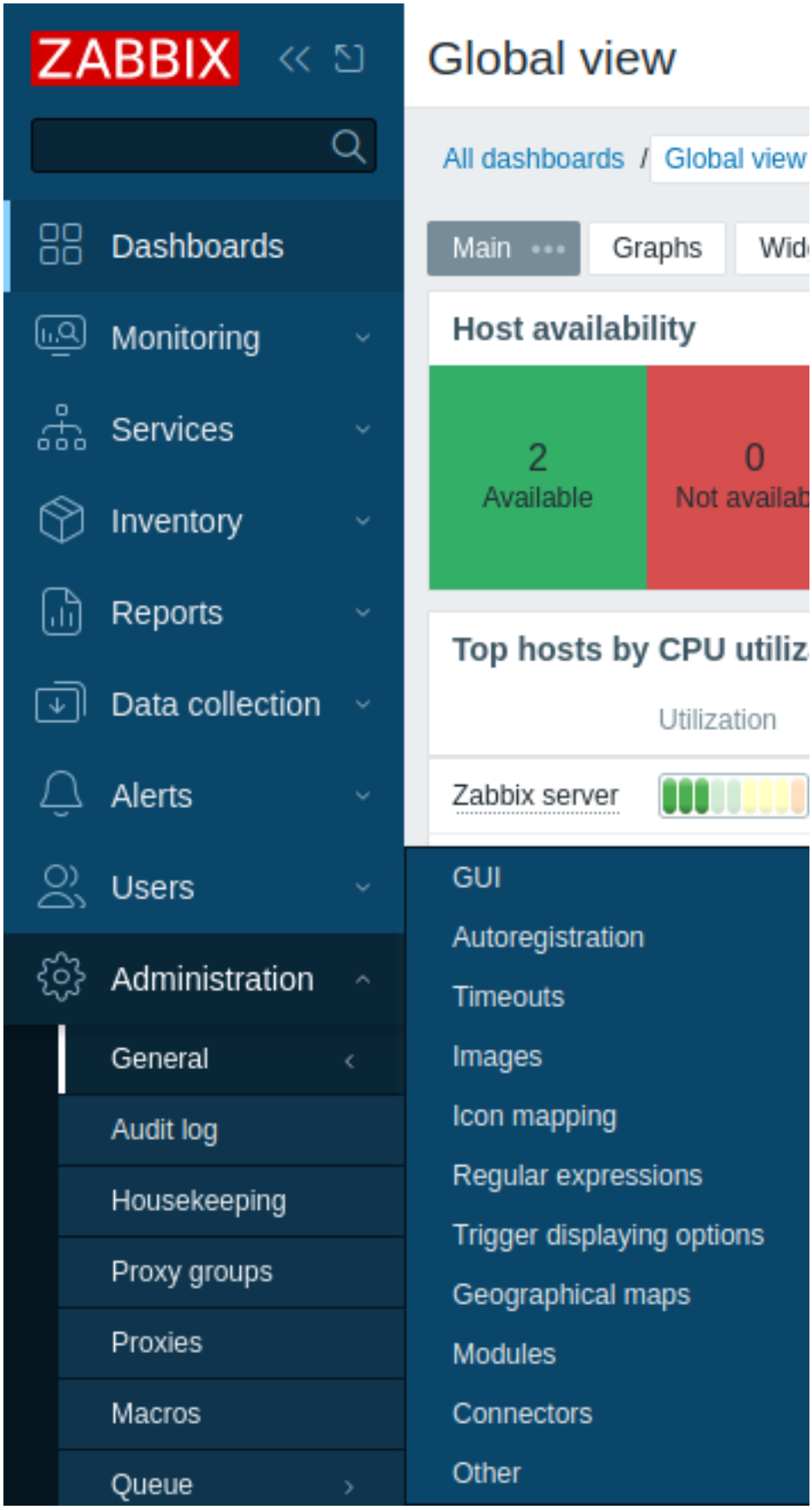
You can also make a collapsed menu reappear fully by hitting the Tab key. Hitting the Tab key repeatedly will allow to focus on the next menu element.

Paslēpta izvēlne

Pat tad, ja izvēlne ir pilnībā paslēpta, pilna izvēlne ir pieejama ar vienu peles klikšķi, noklikšķinot uz burgera ikonas . Ņemiet vērā, ka tā atkal parādās virs lapas satura; lai pārvietotu lapas saturu pa labi, jums ir jāatjauno izvēlnes rādīšana, noklikšķinot uz sānjoslas parādīšanas pogas.

Izvēlnes līmeņi

Izvēlnē var būt līdz trim līmeņiem.



Konteksta izvēlnes

Papildus galvenajai izvēlnei Zabbix nodrošina konteksta izvēlnes hostam, vienumam un notikumam, lai nodrošinātu ātru piekļuvi bieži izmantotām entītijām, piemēram, jaunākajām vērtībām, vienkāršam grafikam, konfigurācijas formai, saistītajiem skriptiem vai ārējām saitēm.

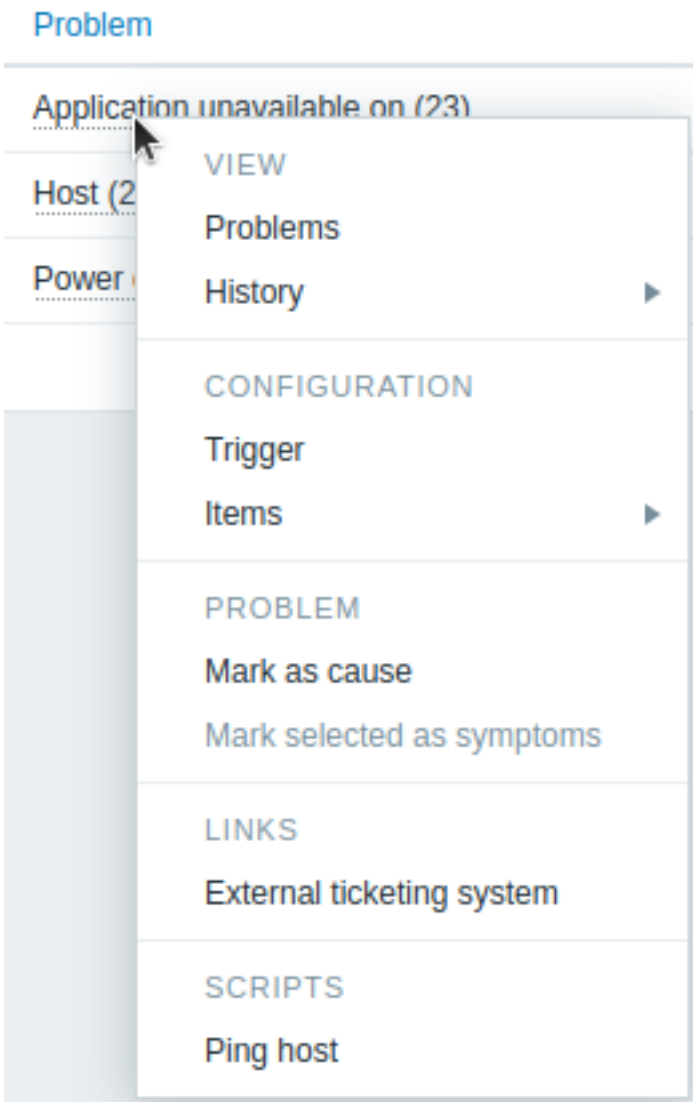
Konteksta izvēlnes ir pieejamas, noklikšķinot uz hosta, vienuma vai problēmas/trigera nosaukuma atbalstītajās vietās.

1 Notikumu izvēlne

Pārskats

Notikuma izvēlnē ir saīsnes uz darbībām vai lietotāja saskarnes sadaļām, kas bieži ir nepieciešamas notikumam.

Notikuma izvēlni var atvērt, noklikšķinot uz notikuma nosaukuma.



Saturs

Notikuma konteksta izvēlnē ir sešas sadaļas: *Skats*, *Darbības*, *Konfigurācija*, *Problēma*, *Saites* un *Skripti*. Entītijām, kas nav konfigurētas, saites ir atspējotas un attēlotas pelēkā krāsā. Sadaļas *Skripti* un *Saites* tiek attēlotas, ja to entītijas ir konfigurētas.

Sadaļā *Skats* ir saites uz:

- **Problēmas** - atver pamatā esošā trigeris neatrisināto problēmu sarakstu;
- **Vēsture** - ved uz pamatā esošā vienums(-u) *Jaunākie dati* grafiku/vienuma vēsturi. Ja trigeris izmanto vairākus vienumus, saites būs pieejamas katram no tiem.

Sadaļa *Darbības* ir pieejama tikai *Trigeru pārskata* logrīkos. Tajā ir saite uz:

- **Atjaunināt problēmu** - atver problēmas atjaunināšanas ekrānu.

Sadaļā *Konfigurācija* ir saites uz šādu objektu konfigurāciju:

- **Trigeris**, kas izraisīja problēmu;
- **Vienumi**, kas izmantoti trigeris izteiksmē.

> **Note:**
> Ņemiet vērā, ka konfigurācijas sadaļa ir pieejama tikai Admin un Super admin lietotājiem.

Sadaļā *Problēma* ir pieejamas šādas opcijas:

- **Atzīmēt kā cēloni** - atzīmēt problēmu kā cēloni;
- **Atzīmēt atlasītos kā simptomus** - atzīmēt atlasītās problēmas kā šīs problēmas simptomus.

Sadaļā *Saites* ir saites, lai:

- piekļūtu konfigurētam trigeris URL;
- piekļūtu pielāgotām saitēm, kas konfigurētas Globālajos skriptos (ar tvērumu 'Manual event action' un tipu 'URL');
- piekļūtu konfigurētai ārējai problēmas biļetei (skatiet opciju *Include event menu entry*, konfigurējot webhooks).

Sadaļā *Skripti* ir saites globāla skripta izpildei (ar tvērumu *Manual event action*). Šī iespēja var būt noderīga, lai palaistu skriptus, kas tiek izmantoti problēmu biļešu pārvaldībai ārējās sistēmās.

Atbalstītās atrašanās vietas

Notikuma konteksta izvēlne ir pieejama, noklikšķinot uz problēmas vai notikuma nosaukuma dažādās lietotāja saskarne sadaļās, piemēram:

- Paneļi widgets, piemēram, *Problems* logrīks, *Trigger overview* logrīks utt.
- Uzraudzība → Problems
- Uzraudzība → Problems → Notikuma informācija
- Pārskati → Top 100 triggers (globālie skripti un piekļuve ārējam pieteikumam šajā vietā netiek atbalstīti)

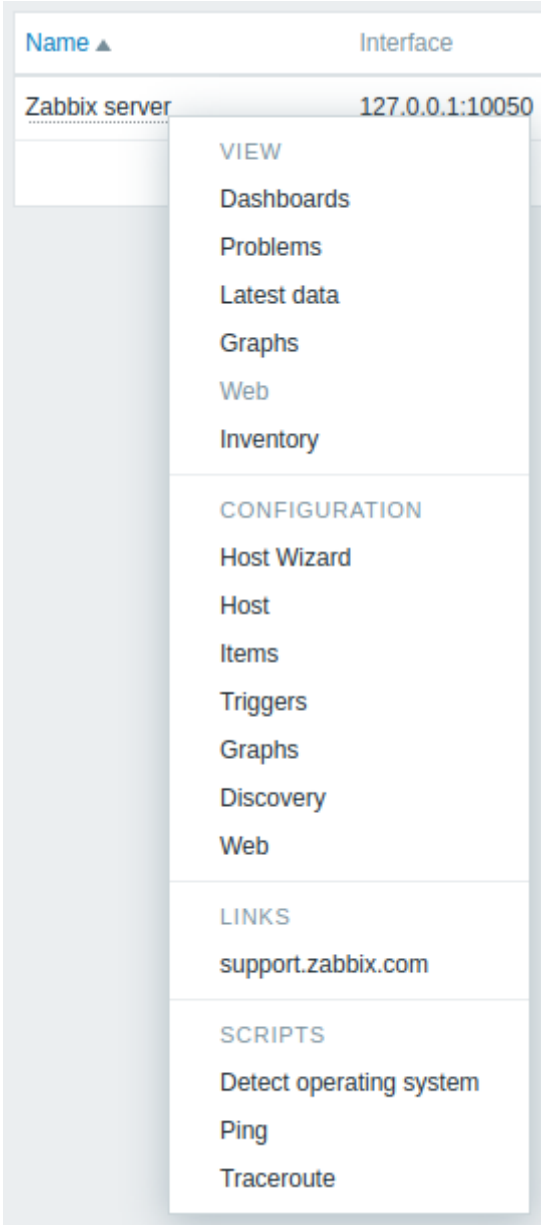
2 Hostu izvēlne

Pārskats

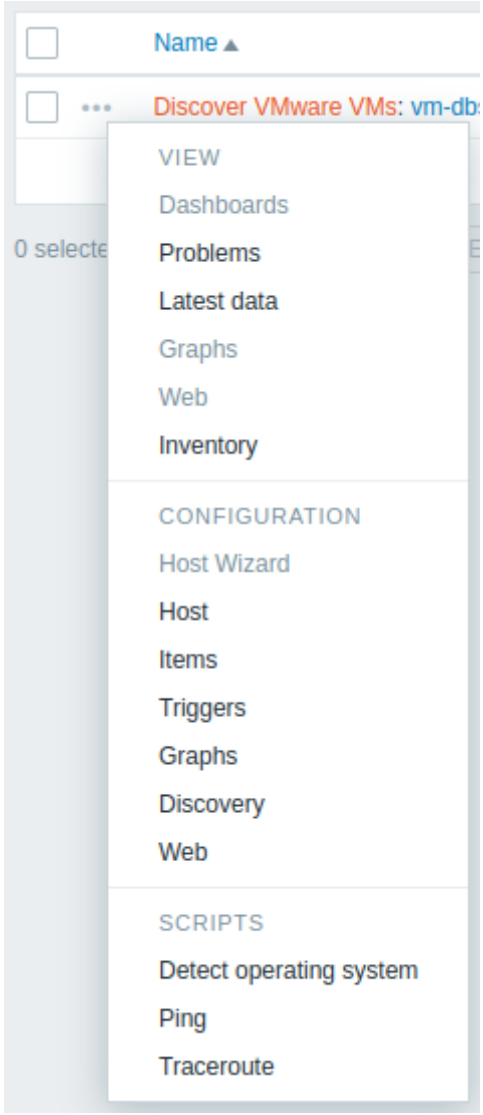
Hosta izvēlnē ir saīsnes uz darbībām vai lietotāja saskarnes sadaļām, kas hostam bieži ir nepieciešamas.

Hosta izvēlni var atvērt, noklikšķinot uz hosta nosaukuma vai trīs punktu ikonas atkarībā no lietotāja saskarnes sadaļas, piemēram:

Monitoring → *Jaunākie dati*



Datu ievākšana → *Hosti*



Saturs

Hosta konteksta izvēlnei ir četras sadaļas: *Skats*, *Konfigurācija*, *Saites* un *Skripti*. Entītijām, kas nav konfigurētas, saites ir atspējotas un tiek rādītas pelēkā krāsā. Sadaļas *Skripti* un *Saites* tiek rādītas, ja to entītijas ir konfigurētas.

Sadaļa *Skats* satur saites uz:

- **Paneļi** - atver logrīkus un grafikus.
- **Problēmas** - atver sadaļu *Problēmas* ar pamatā esošā trigeris neatrisināto problēmu sarakstu.
- **Jaunākie dati** - atver sadaļu *Jaunākie dati* ar visa pašreizējā hosta jaunāko datu sarakstu.
- **Grafiki** - atver pašreizējā hosta vienkāršos grafikus.
- **Tīmeklis** - atver saiti uz konfigurētajiem tīmekļa scenārijiem.
- **Inventārs** - atver saiti uz pašreizējā hosta inventāru.

Sadaļa *Konfigurācija* satur saites uz:

- **Host Wizard** - atver Host Wizard pašreizējam hostam (atspējots atklātajiem hostiem).
- **Hosts** - pašreizējā hosta konfigurācijas forma.
- **Vienumi** - pašreizējā hosta vienumu saraksts.
- **Trigeri** - pašreizējā hosta trigeru saraksts.
- **Grafiki** - pašreizējā hosta vienkāršie grafiki.
- **Atklāšana** - pašreizējā hosta zema līmeņa atklāšanas noteikumu saraksts.
- **Tīmeklis** - pašreizējā hosta tīmekļa scenāriju saraksts.

Note:
Ņemiet vērā, ka konfigurācijas sadaļa ir pieejama tikai Admin un Super admin lietotājiem.

Sadaļa *Saites* satur saites uz:

- piekļuvi konfigurētam trigeris URL.
- piekļuvi pielāgotām saitēm, kas konfigurētas sadaļā Global scripts (ar tvērumu *Manual host action* un tipu 'URL').

Sadaļa *Skripti* ļauj izpildīt globālos skriptus, kas konfigurēti pašreizējam hostam. Lai šie skripti būtu pieejami hosta izvēlnē, to tvērumam jābūt definētam kā *Manual host action*.

Atbalstītās atrašanās vietas

Hosta izvēlne ir pieejama, noklikšķinot uz hosta nosaukuma dažādās lietotāja saskarnes sadaļās, piemēram:

- Paneļu logrīki, piemēram, Problēmas, Populārākie vienumi, Trigeru pārskats u.c.
- Uzraudzība → Problēmas
- Uzraudzība → Problēmas → Notikuma informācija
- Uzraudzība → Hosti
- Uzraudzība → Hosti → Tīmekļa uzraudzība
- Uzraudzība → Jaunākie dati
- Uzraudzība → Kartes
- Inventārs → Hosti
- Pārskati → 100 populārākie trigeri

Hosta izvēlne ir pieejama, noklikšķinot uz trīs punktu ikonas sadaļā Datu ievākšana → Hosti.

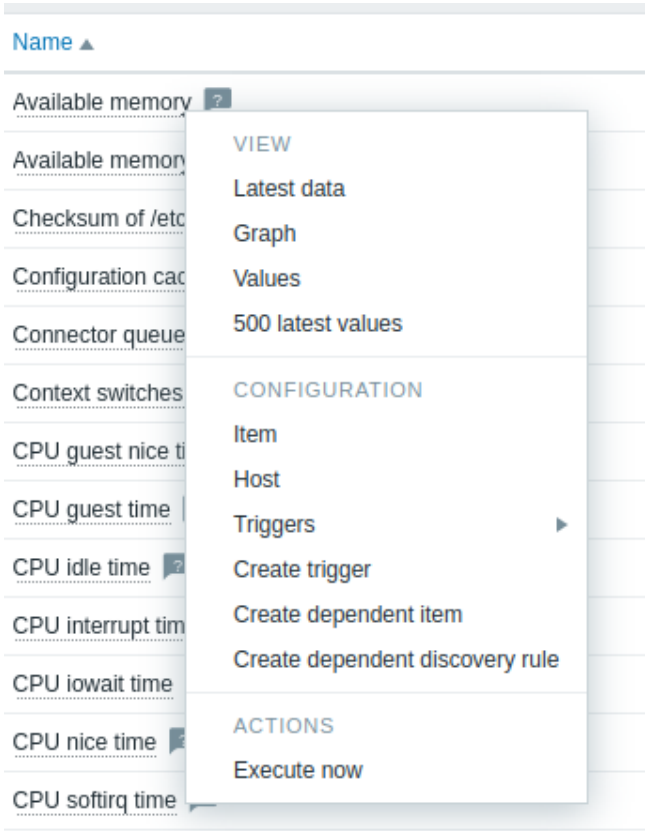
3 Vienuma izvēlne

Pārskats

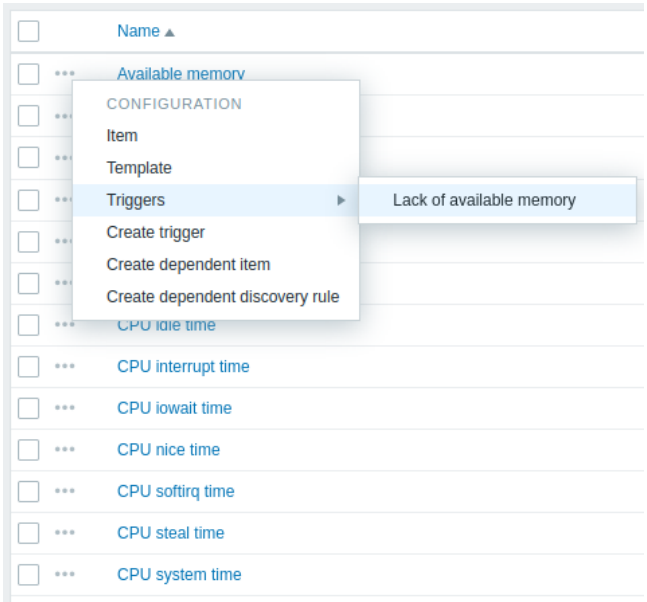
Vienuma izvēlnē ir saīsnes uz darbībām vai lietotāja saskarnes sadaļām, kas bieži ir nepieciešamas vienumam.

Vienuma izvēlni var atvērt, noklikšķinot uz vienuma nosaukuma vai trīs punktu ikonas atkarībā no lietotāja saskarnes sadaļas, piemēram:

Uzraudzība → Jaunākie dati



Datu ievākšana → Veidnes → Vienumi



Saturs

Vienuma izvēlnē ir trīs sadaļas: *Skats*, *Konfigurācija* un *Darbības*.

Sadaļā *Skats* ir pieejamas šādas opcijas:

- **Jaunākie dati** — atver sadaļu *Jaunākie dati*, kas filtrēta pēc pašreizējā hosta un vienuma;
- **Grafiks** — atver pašreizējā vienuma vienkāršo grafiku;
- **Vērtības** — atver visu vērtību sarakstu, kas saņemtas pašreizējam vienumam pēdējo 60 minūšu laikā;
- **500 jaunākās vērtības** — atver pašreizējā vienuma 500 jaunāko vērtību sarakstu.

Sadaļā *Konfigurācija* (pieejama tikai lietotājiem ar tipu *Admin* un *Super admin*) ir pieejamas šādas opcijas:

- **Vienums** — atver pašreizējā vienuma vienuma konfigurācijas formu;
- **Veidne** — atver tās veidnes veidnes konfigurācijas formu, kurai pieder vienums (pieejams tikai tad, ja vienuma izvēlne tiek atvērta no *Datu ievākšana* → *Veidnes* → *Vienumi*);
- **Hosts** — atver tā hosta hosta konfigurācijas formu, kuram pieder vienums;
- **Trigeri** — novietojot peles kursoru virs opcijas, atver vienuma trigeru sarakstu, ja tādi ir; noklikšķinot uz trigeri, tiek atvērta tā trigera konfigurācijas forma;
- **Izveidot trigeri** — atver trigera konfigurācijas formu, lai izveidotu trigeri šim vienumam;
- **Izveidot atkarīgo vienumu** — atver vienuma konfigurācijas formu, lai izveidotu atkarīgo vienumu, kur pašreizējais vienums ir galvenais vienums;
- **Izveidot atkarīgo atklāšanas kārtulu** — atver atklāšanas kārtulas konfigurācijas formu, lai izveidotu atkarīgo atklāšanas kārtulu, kur pašreizējais vienums ir galvenais vienums.

Sadaļā *Darbības* ir pieejama šāda opcija:

- **Izpildīt tagad** — izpilda pārbaudi, lai nekavējoties iegūtu jaunu vienuma vērtību.

Atbalstītās atrašanās vietas

Vienuma izvēlne ir pieejama, noklikšķinot uz vienuma nosaukuma dažādās lietotāja saskarnes sadaļās, piemēram:

- Uzraudzība → Jaunākie dati
- Datu ievākšana → Hosti → Vienumi
- Datu ievākšana → Hosti → Atklāšanas noteikumi → Vienumu prototipi

Vienuma izvēlne ir pieejama, noklikšķinot uz vienuma vērtības paneļa logrīkā Top items.

2 Lietotāja saskarnes sadaļas

Izvēlnes struktūra Zabbix lietotāja saskarnes izvēlnei ir šāda struktūra:

- Paneļi
- Uzraudzība
  - Problēmas
  - Hosti
  - Jaunākie dati
  - Kartes
  - Atklāšana
- Pakalpojumi
  - Pakalpojumi
  - SLA
  - SLA atskaite
- Inventārs
  - Pārskats
  - Hosti
- Atskaites
  - Sistēmas informācija
  - Plānotās atskaites
  - Pieejamības atskaite
  - 100 populārākie trigeri
  - Audita žurnāls
  - Darbību žurnāls
  - Paziņojumi
- Datu vākšana

    - Veidņu grupas
    - Hostu grupas
    - Veidnes
    - Hosti
    - Uzturēšana
    - Notikumu korelācija
    - Atklāšana
- Brīdinājumi
    - Darbības
        * Trigeru darbības
        * Pakalpojumu darbības
        * Atklāšanas darbības
        * Automātiskās reģistrācijas darbības
        * Iekšējās darbības
    - Mediju tipi
    - Skripti
- Lietotāji
    - Lietotāju grupas
    - Lietotāju lomas
    - Lietotāji
    - API marķieri
    - Autentifikācija
- Administrēšana
    - Vispārīgi
        * GUI
        * Automātiskā reģistrācija
        * Noildzes
        * Attēli
        * Ikonu kartēšana
        * Regulārās izteiksmes
        * Trigeru attēlošanas opcijas
        * Ģeogrāfiskās kartes
        * Moduļi
        * Savienotāji
        * Citi
    - Audita žurnāls
    - Uzturēšana
    - Starpniekserveru grupas
    - Starpniekserveri
    - Makrosi
    - Rinda
        * Rindas pārskats
        * Rindas pārskats pēc starpniekservera
        * Rindas detaļas

1 Paneļi

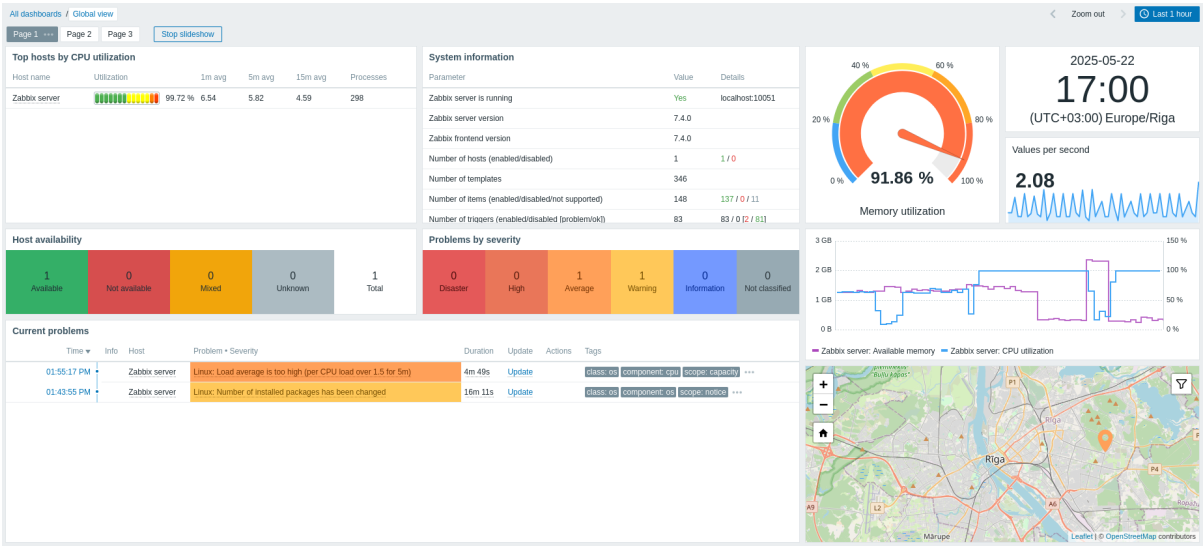
Pārskats

Sadaļa *Paneļi* ir paredzēta, lai **panelī** attēlotu visu svarīgāko informāciju kopsavilkumus.

Lai gan vienlaikus var tikt attēlots tikai viens panelis, ir iespējams konfigurēt vairākus paneļus. Katrs panelis var saturēt vienu vai vairākas lapas, kuras var pārslēgt slaidrādē.

Paneļa lapa sastāv no logrīkiem, un katrs logrīks ir paredzēts noteikta veida un avota informācijas attēlošanai, piemēram, kopsavilkuma, kartes, grafika, pulksteņa u.c.

Piekļuve hosts logrīkos ir atkarīga no host atļaujām.



Lapas un logrīki tiek pievienoti panelim un rediģēti paneļa rediģēšanas režīmā. Lapas var skatīt un pārslēgt skatīšanas režīmā.

Laika periodu, kas tiek attēlots grafiku logrīkos, kontrolē laika perioda atlasītājs, kas atrodas virs logrīkiem. Tā etiķete parāda pašlaik atlasīto laika periodu. Noklikšķinot uz etiķetes, var izvērst un sakļaut laika perioda atlasītāju.

Paneļa izmērs

Paneļa minimālais platums ir 1200 pikseļi. Panelis netiks samazināts zem šī platuma; tā vietā tiks parādīta horizontāla ritjosla, ja pārlūkprogrammas logs būs mazāks par to.

Paneļa maksimālais platums ir pārlūkprogrammas loga platums. Paneļa logrīki horizontāli izstiepjas, lai aizpildītu logu. Vienlaikus paneļa logrīku nevar horizontāli izstiept ārpus loga robežām.

Horizontāli paneli veido 72 kolonnas ar vienmēr vienādu platumu, kas dinamiski izstiepjas/samazinās (bet ne līdz mazāk nekā 1200 pikseļiem kopā).

Vertikāli panelī var būt ne vairāk kā 64 rindas; katras rindas fiksētais augstums ir 70 pikseļi.

Tādējādi logrīks var būt līdz 72 kolonnām plats un 64 rindām augsts.

Paneļu skatīšana

Lai skatītu visus konfigurētos paneļus, noklikšķiniet uz *Visi paneļi* tieši zem sadaļas virsraksta.

Paneļi tiek attēloti ar kopīgošanas atzīmi:

- *Mani* - norāda paneli, kas pieder pašreizējam lietotājam
- *Kopīgoti* - norāda publisku paneli vai privātu paneli, kas kopīgots ar jebkuru lietotāju vai lietotāju grupu

Ņemiet vērā, ka kopīgošanas atzīmes tiek rādītas tikai tiem paneļiem, kas pieder pašreizējam lietotājam. Citiem lietotājiem piederošie paneļi tiek uzskaitīti bez kopīgošanas atzīmēm.

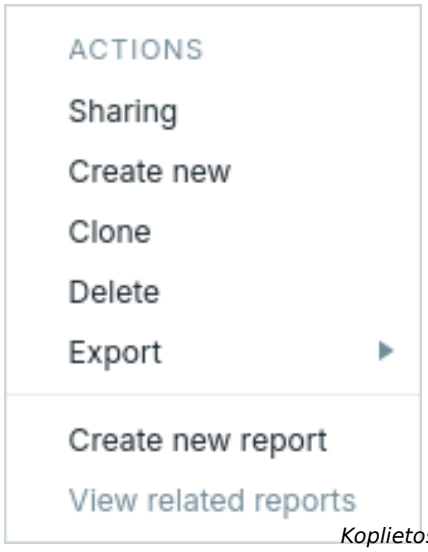
Filtrs, kas atrodas pa labi virs saraksta, ļauj filtrēt paneļus pēc nosaukuma un pēc tiem, ko izveidojis pašreizējais lietotājs.

Lai dzēstu vienu vai vairākus paneļus, atzīmējiet attiecīgo paneļu izvēles rūtiņas un noklikšķiniet uz *Dzēst* zem saraksta.

Informācijas paneļa skatīšana

Lai skatītu vienu informācijas paneli, noklikšķiniet uz tā nosaukuma informācijas paneļu sarakstā.

Skatot informācijas paneli, ir pieejamas šādas opcijas:

Edit dashboard	Pārslēgties uz informācijas paneļa **rediģēšanas** režīmu. Rediģēšanas režīms tiek atvērts arī tad, kad tiek veidots jauns informācijas panelis un kad noklikšķināt uz logrīka rediģēšanas pogas.
	Atvērt darbību izvēlni (skatiet darbību aprakstus zemāk).
	*Koplietošana* - rediģēt informācijas paneļa koplietošanas iestatījumus. *Izveidot jaunu* - izveidot jaunu informācijas paneli. *Klonēt* - izveidot jaunu informācijas paneli, kopējot esošā informācijas paneļa īpašības. Vispirms jums tiks piedāvāts ievadīt informācijas paneļa parametrus. Pēc tam jaunais informācijas panelis tiks atvērts rediģēšanas režīmā ar visiem sākotnējā informācijas paneļa logrīkiem. *Dzēst* - dzēst informācijas paneli. *Izveidot jaunu atskaiti* - atvērt uznirstošo logu ar atskaites konfigurācijas formu. Atspējots, ja lietotājam nav atļaujas pārvaldīt ieplānotās atskaites. *Skatīt saistītās atskaites* - atvērt uznirstošo logu ar esošo atskaišu sarakstu, kas balstītas uz pašreizējo informācijas paneli. Atspējots, ja nav saistītu atskaišu vai lietotājam nav atļaujas skatīt ieplānotās atskaites.
	Rādīt tikai lapas saturu (kioska režīms). Kioska režīmam var piekļūt arī ar šādiem URL parametriem: `/zabbix.php?action=dashboard.view&kiosk=1`. Lai atgrieztos normālajā režīmā: `/zabbix.php?action=dashboard.view&kiosk=0`.

Kopīgošana

Paneļus var padarīt publiskus vai privātus.

Publiskie paneļi ir redzami visiem lietotājiem. Privātie paneļi ir redzami tikai to īpašniekam un lietotājiem/lietotāju grupām, kas pievienotas kopīgošanas iestatījumos.

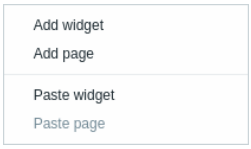
Lai rediģētu paneļa kopīgošanas statusu, skatot vienu paneli, darbību izvēlnē noklikšķiniet uz opcijas *Kopīgošana*:

Parameter	Description
Type	Izvēlieties paneļa tipu: **Privāts** - panelis ir redzams tikai atlasītajām lietotāju grupām un lietotājiem. **Publisks** - panelis ir redzams visiem.
List of user group shares	Atlasiet lietotāju grupas, kurām panelis ir pieejams. Varat atļaut tikai lasīšanas vai lasīšanas-rakstīšanas piekļuvi.
List of user shares	Atlasiet lietotājus, kuriem panelis ir pieejams. Varat atļaut tikai lasīšanas vai lasīšanas-rakstīšanas piekļuvi.

Skatiet Atļaujas paneļiem, lai uzzinātu, kā kopīgošanas opcijas ierobežo lietotājiem pieejamās darbības.

Paneļa rediģēšana

Kad panelis tiek **rediģēts**, ir pieejamas šādas opcijas:

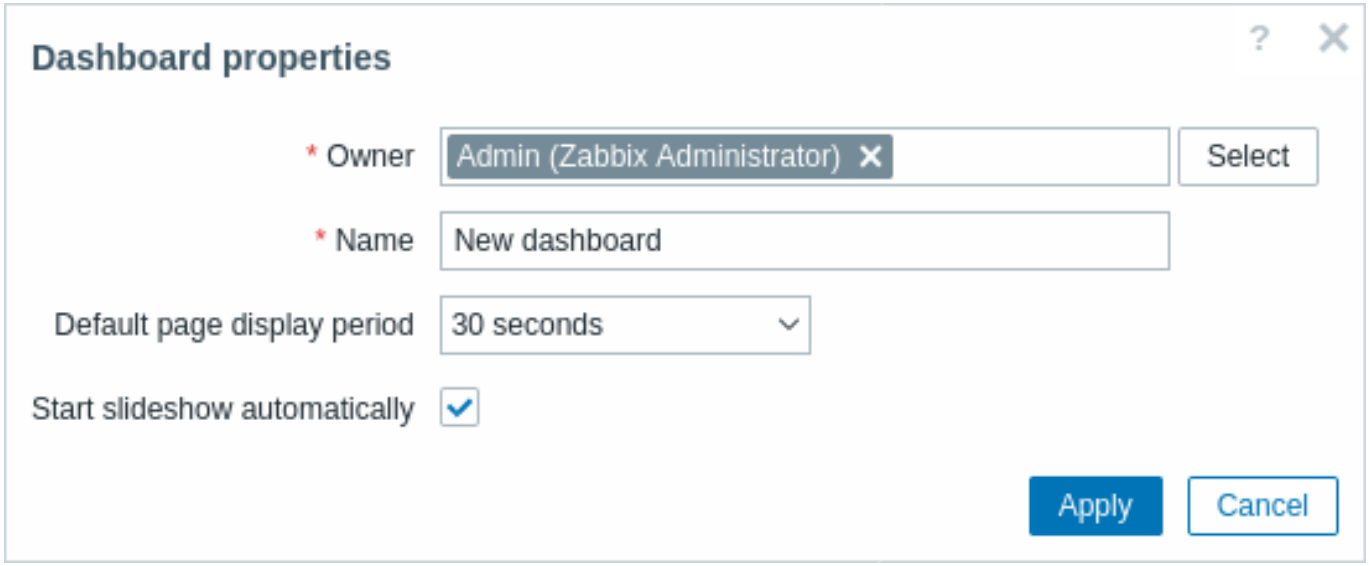
	Rediģēt paneļa vispārīgos parametrus. Pievienot jaunu logrīku. Noklikšķinot uz bultiņas pogas, tiks atvērta darbību izvēlne (skatiet darbību aprakstus zemāk).
	*Pievienot logrīku* — pievienot jaunu logrīku. *Pievienot lapu* — pievienot jaunu lapu. *Ielīmēt logrīku* — ielīmēt nokopētu logrīku. Šī opcija ir pelēkota, ja nav nokopēts neviens logrīks. Vienlaikus var nokopēt tikai vienu vienību (logrīku vai lapu). *Ielīmēt lapu* — ielīmēt nokopētu lapu. Šī opcija ir pelēkota, ja nav nokopēta neviena lapa.
	Saglabāt paneļa izmaiņas.
	Atcelt paneļa izmaiņas.

Informācijas paneļa izveide

Jaunu informācijas paneli ir iespējams izveidot divos veidos:

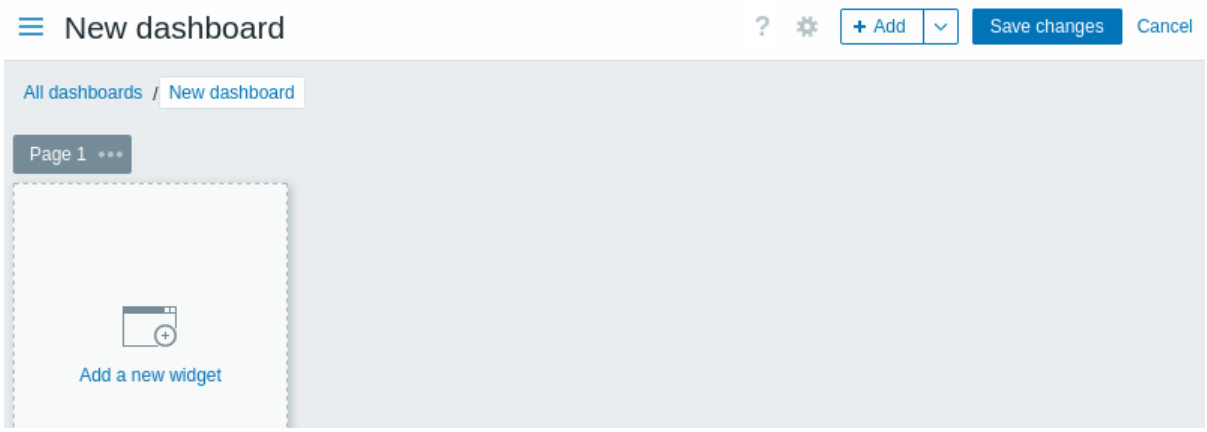
- Noklikšķiniet uz *Create dashboard*, skatot visus informācijas paneļus
- Darbību izvēlnē atlasiet *Create new*, skatot vienu informācijas paneli

Vispirms jums tiks lūgts ievadīt vispārīgos informācijas paneļa parametrus:



Parametrs	Apraksts
Owner	Atlasiet sistēmas lietotāju, kurš būs informācijas paneļa īpašnieks.
Name	Ievadiet informācijas paneļa nosaukumu.
Default page display period	Atlasiet periodu, cik ilgi informācijas paneļa lapa tiks rādīta pirms pārslēgšanās uz nākamo lapu slaidrādē.
Start slideshow automatically	Atzīmējiet šo izvēles rūtiņu, lai slaidrāde tiktu palaista automātiski, ja informācijas panelī ir vairāk nekā viena lapa.

Kad noklikšķināt uz *Apply*, tiek atvērts tukšs informācijas panelis:

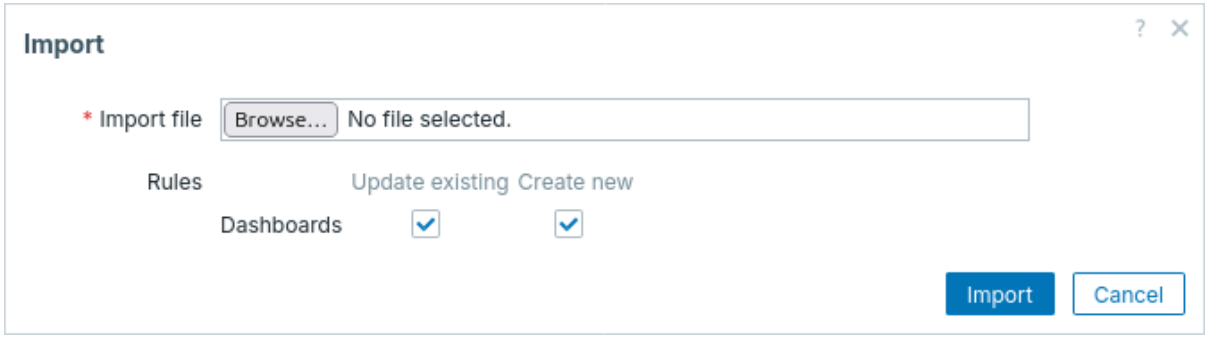


Lai aizpildītu informācijas paneli, varat pievienot logrīkus un lapas.

Noklikšķiniet uz pogas *Save changes*, lai saglabātu informācijas paneli. Ja noklikšķināsiet uz *Cancel*, informācijas panelis netiks izveidots.

Importing a dashboard

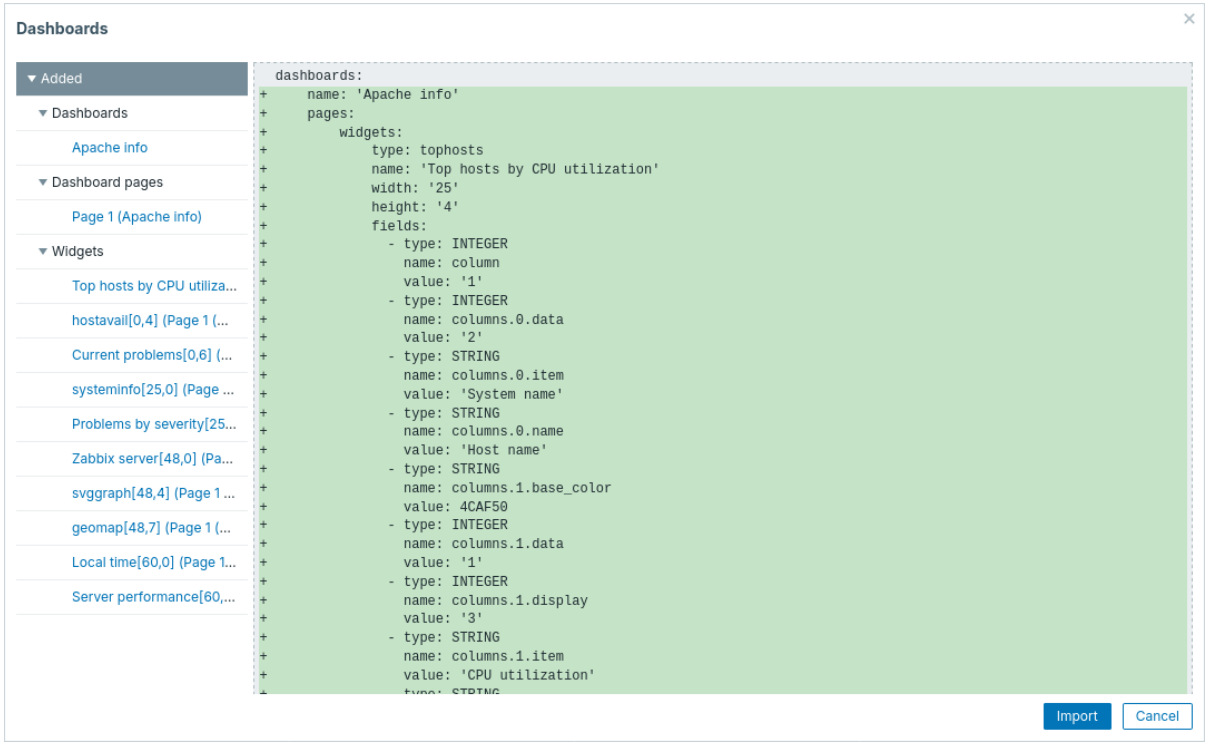
You can import a dashboard by clicking on *Import* when viewing all dashboards. Note that the *Import* button is always visible, but it is disabled if the user does not have permission to *Create and edit dashboards*. A form will open that will allow you to browse dashboard files.



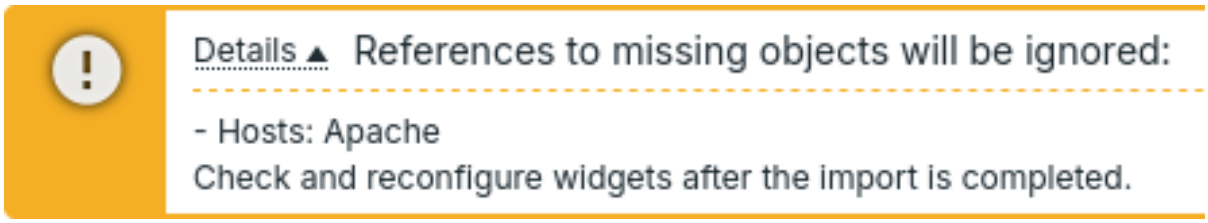
Import rules:

Rule	Description
Update existing	Existing dashboards will be updated using data from the import file. Otherwise, they will not be updated.
Create new	New dashboards will be created using data from the import file. Otherwise, they will not be created.

The import pop-up comparison will appear after clicking on the *Import* button.



If the import file references objects that do not exist on the target system, an alert message appears. This is expected for widgets that refer to specific hosts, items, or other objects. In such cases, Zabbix allows the dashboard to be imported, clears the missing references, and either marks the widget as not fully configured when the data is mandatory or drops the filtering criteria when the fields are used only for filtering.

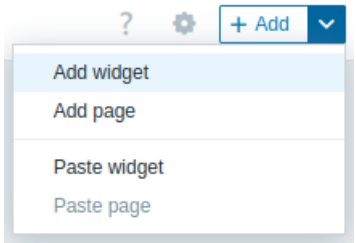


Clicking the *Import* button again will add and/or update the dashboard.

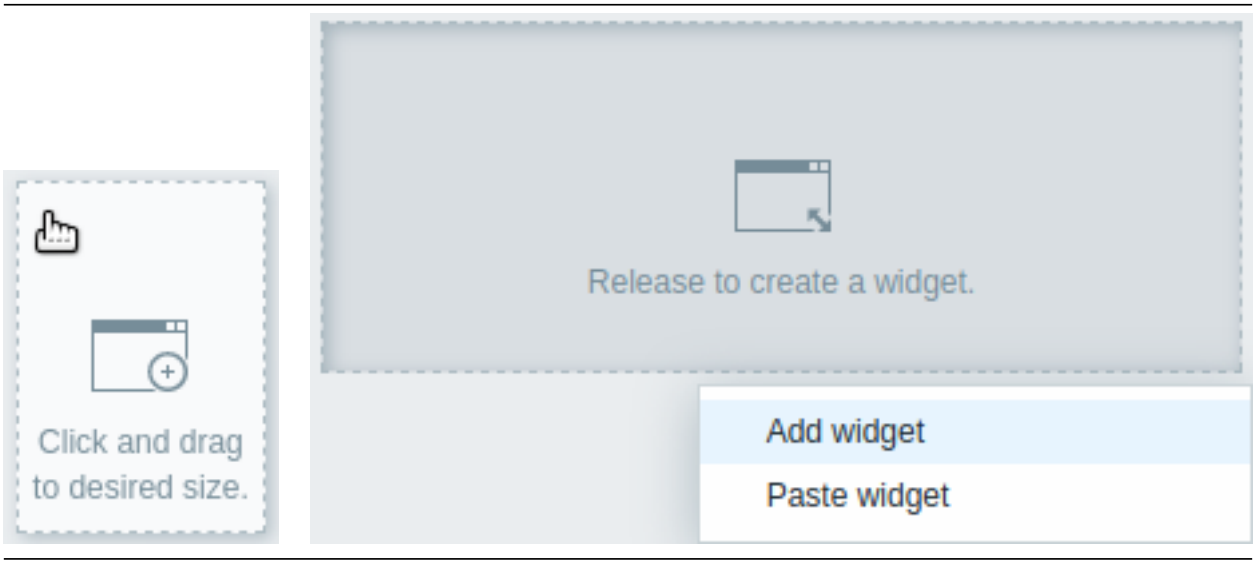
Logrīku pievienošana

Informācijas panelim varat pievienot dažādus logrīkus (darbību žurnālu, pulksteni, atklāšanas statusu u.c.). Logrīkus var pievienot divos veidos:

- Noklikšķiniet uz pogas *Pievienot* vai bultiņas blakus tai, pēc tam darbību izvēlnē atlasiet *Pievienot logrīku*. Pēc logrīka konfigurēšanas tas tiks pievienots noklusējuma izmērā un novietots aiz jebkura jau esoša logrīka.

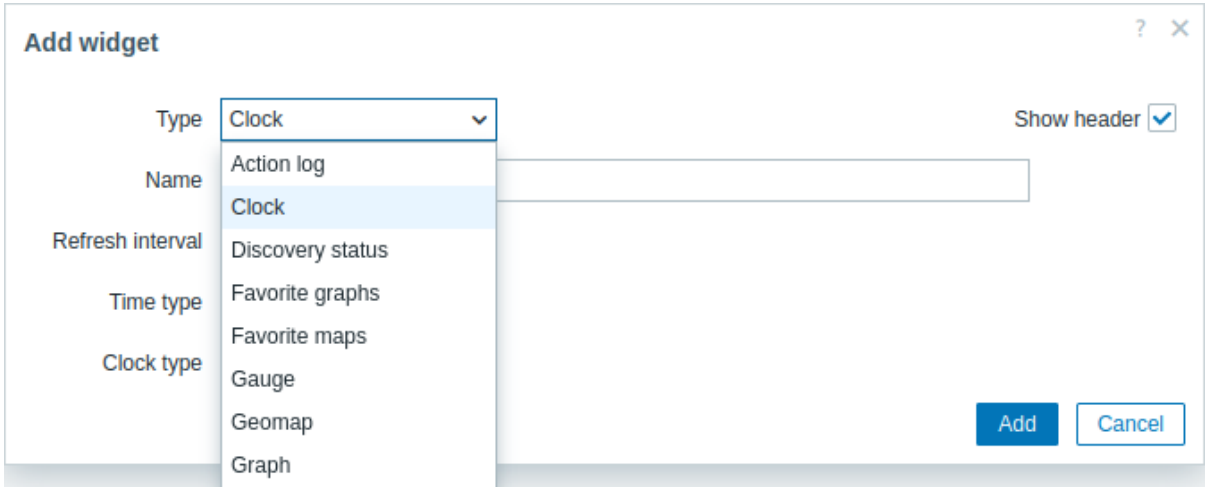


- Pārvietojiet peles kursoru uz tukšu vietu informācijas panelī. Ņemiet vērā, ka parādīsies vietturis; pēc tam noklikšķiniet uz tā, lai atvērtu logrīka konfigurācijas formu. Pēc logrīka konfigurēšanas tas tiks pievienots noklusējuma izmērā vai mainīts tā izmērs, lai tas ietilptu pieejamajā vietā. Varat arī vilkt vietturi, lai noteiktu logrīka izmēru, un atlaist to, lai atvērtu logrīka konfigurācijas formu (ja starpliktuvē ir kopēts logrīks, vispirms jums tiks piedāvāts izvēlēties starp *Pievienot logrīku* vai *Ielīmēt logrīku*).



Logrīka konfigurācijas formā:

1. Atlasiet logrīka tipu.
2. Konfigurējiet logrīka parametrus.
3. Noklikšķiniet uz *Pievienot*, lai pievienotu logrīku informācijas panelim.



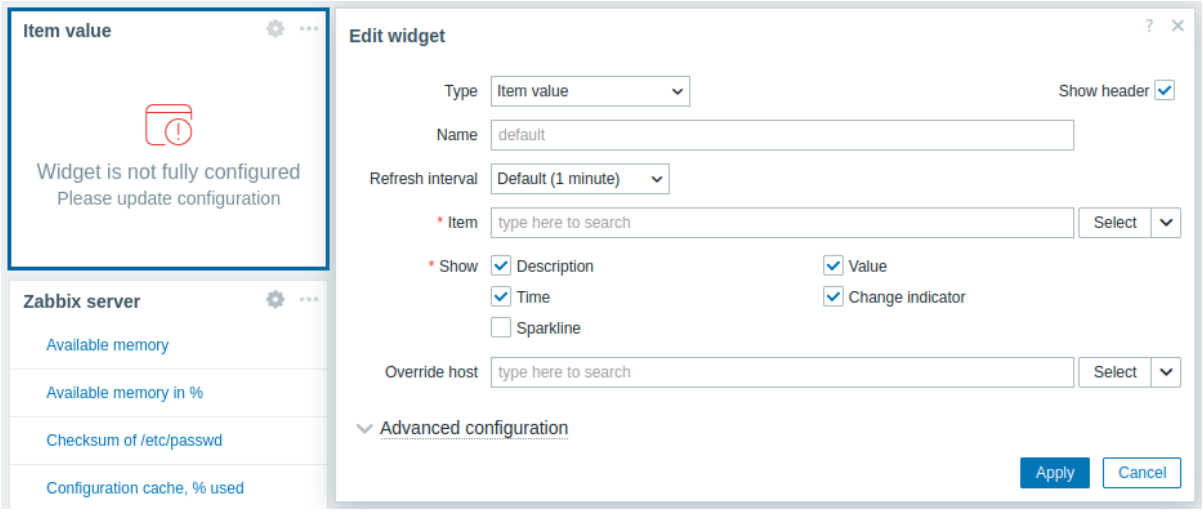
Logrīku rediģēšana

Logrīku izmēru var mainīt un tos var pārvietot informācijas paneļa rediģēšanas režīmā, velkot to virsrakstjoslu. Varat arī pārvietot logrīka konfigurācijas formu, velkot tās virsrakstjoslu.

Katram logrīkam augšējā labajā stūrī ir vadīklas:

- - rediģēt logrīku
- - atvērt logrīka izvēlni

Rediģēšanas laikā logrīks tiek izcelts, un izmaiņu priekšskatījums tiek rādīts reāllaikā. Ja obligātie logrīka parametri nav konfigurēti, logrīks pāriet stāvoklī *Logrīks nav pilnībā konfigurēts*.



Pēc rediģēšanas logrīkā noklikšķiniet uz *Piemērot*, pēc tam informācijas paneļa augšējā labajā stūrī noklikšķiniet uz *Saglabāt izmaiņas*, lai piemērotu izmaiņas.

Logrīku kopēšana/ielīmēšana

Jūs varat kopēt un ielīmēt paneļa logrīkus, lai ātri izveidotu jaunus logrīkus ar tādu pašu konfigurāciju. Logrīkus var kopēt un ielīmēt vienā un tajā pašā panelī vai starp paneļiem, kas atvērti dažādās cilnēs.

Lai kopētu logrīku, izmantojiet logrīka izvēlni. Lai ielīmētu nokopētu logrīku (pieejams paneļa rediģēšanas režīmā):

- Noklikšķiniet uz bultiņas blakus pogai *Pievienot* un atlasiet *Ielīmēt logrīku*.
- Vai arī noklikšķiniet tukšā paneļa apgabalā un atlasiet *Ielīmēt logrīku*.

Jūs varat arī ielīmēt nokopētu logrīku virs esoša logrīka, izmantojot opciju *Ielīmēt* tā logrīka izvēlnē.

Slaidrādes izveide

Slaidrāde tiks palaista automātiski, ja panelī ir divas vai vairāk lapas (skatiet Lapu pievienošana) un ja ir spēkā viens no šiem nosacījumiem:

- Paneļa rekvizītos ir atzīmēta opcija *Automātiski sākt slaidrādi*
- Paneļa URL satur parametru `slideshow=1`

Lapas tiek pārslēgtas atbilstoši intervāliem, kas norādīti paneļa un atsevišķu lapu rekvizītos. Noklikšķiniet uz:

- *Apturēt slaidrādi* - lai apturētu slaidrādi
- *Sākt slaidrādi* - lai sāktu slaidrādi

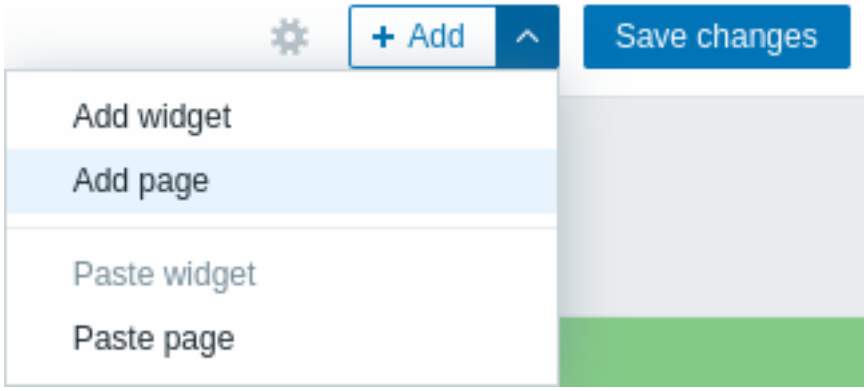
Ar slaidrādi saistītās vadīklas ir pieejamas arī kioska režīmā (kur tiek rādīts tikai lapas saturs):

- - apturēt slaidrādi
- - sākt slaidrādi
- - atgriezties vienu lapu atpakaļ
- - pāriet uz nākamo lapu

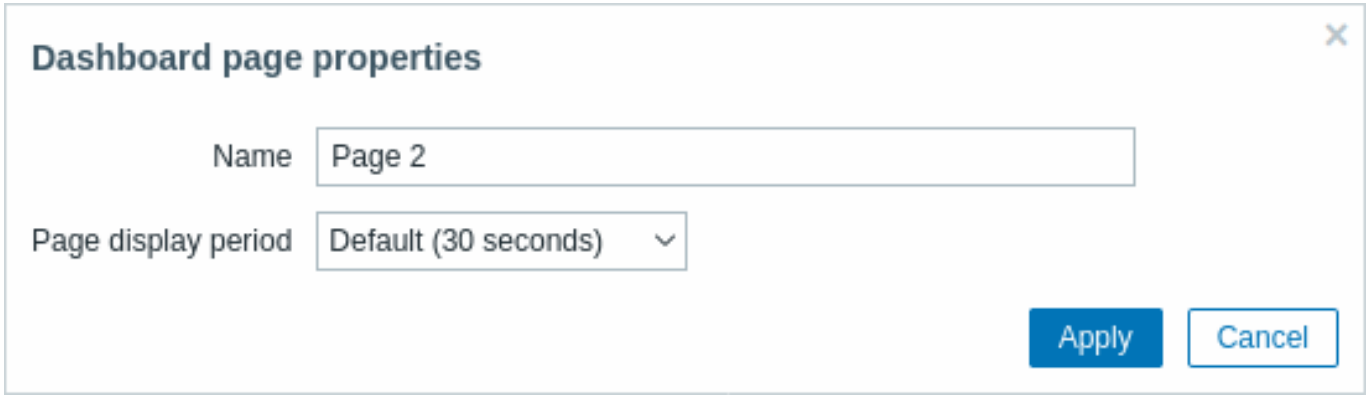
Lapu pievienošana

Lai informācijas panelim pievienotu jaunu lapu:

- Pārliecinieties, ka informācijas panelis ir rediģēšanas režīmā
- Noklikšķiniet uz bultiņas blakus pogai *Pievienot* un atlasiet opciju *Pievienot lapu*



- Aizpildiet lapas vispārīgos parametrus un noklikšķiniet uz *Piemērot*. Ja atstāsiet nosaukumu tukšu, lapa tiks pievienota ar nosaukumu `Lapa N`, kur 'N' ir lapas pieaugošais numurs. Lapas attēlošanas periods ļauj pielāgot, cik ilgi lapa tiek rādīta slaidrādē.



Tiks pievienota jauna lapa, ko norādīs jauna cilne (*Lapa 2*).

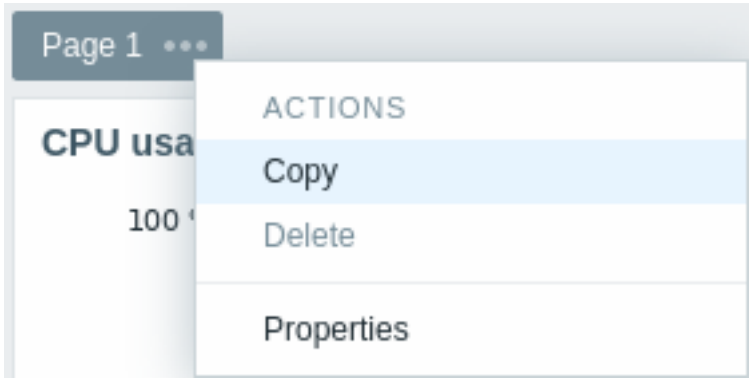
Lapas var pārkārtot, velkot un nometot lapu cilnes. Pārkārtošana saglabā sākotnējo lapu nosaukšanu. Vienmēr ir iespējams pāriet uz jebkuru lapu, noklikšķinot uz tās cilnes.

Kad tiek pievienota jauna lapa, tā ir tukša. Varat tai pievienot logrīkus, kā aprakstīts iepriekš.

Lapu kopēšana/ielīmēšana

Paneļa lapas var kopēt un ielīmēt, ļaujot izveidot jaunu lapu ar esošās lapas īpašībām. Tās var ielīmēt no tā paša paneļa vai cita paneļa.

Lai ielīmētu esošu lapu panelī, vispirms to nokopējiet, izmantojot lapas izvēlni:

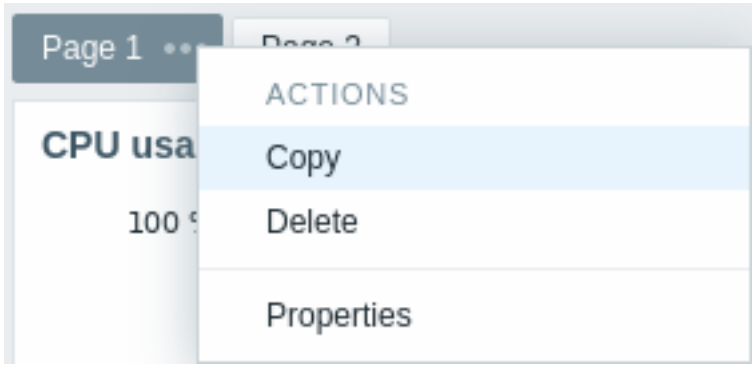


Lai ielīmētu nokopēto lapu:

- Pārliecinieties, ka panelis ir rediģēšanas režīmā
- Noklikšķiniet uz bultiņas blakus pogai *Pievienot* un atlasiet opciju *Ielīmēt lapu*

Lapas izvēlne

Lapas izvēlni var atvērt, noklikšķinot uz trim punktiem blakus lapas nosaukumam:

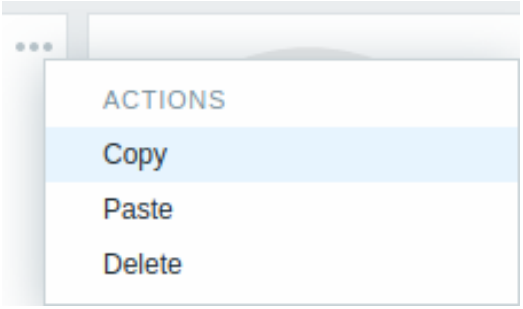
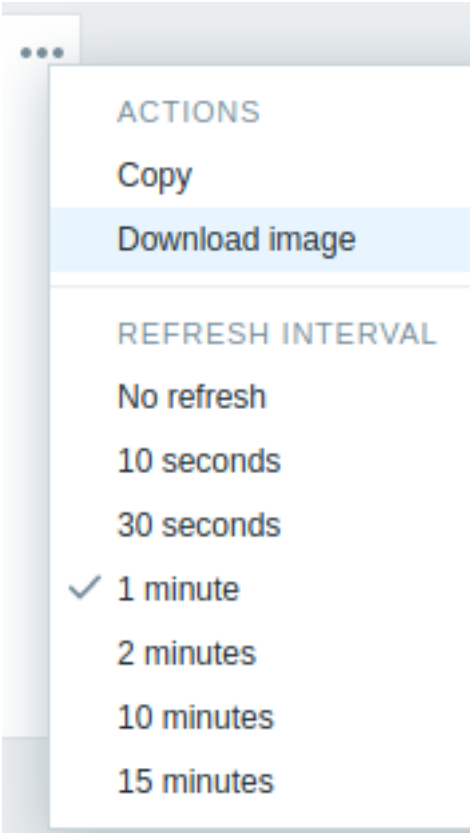


Tā satur šādas opcijas:

- *Kopēt* - kopēt lapu
- *Dzēst* - dzēst lapu (lapas var dzēst tikai paneļa rediģēšanas režīmā)
- *Rekvizīti* - pielāgot lapas parametrus (nosaukumu un lapas attēlošanas periodu slaidrādē)

Logrīka izvēlne

Logrīka izvēlnē ir pieejamas dažādas opcijas atkarībā no tā, vai panelis ir rediģēšanas vai skatīšanas režīmā:

Logrīka izvēlne	Opcijas
Paneļa rediģēšanas režīmā: 	*Kopēt* - kopēt logrīku. *Ielīmēt* - ielīmēt nokopētu logrīku virs šī logrīka. Šī opcija ir pelēkota, ja nav nokopēts neviens logrīks. *Dzēst* - dzēst logrīku.
Paneļa skatīšanas režīmā: 	*Kopēt* - kopēt logrīku. *Lejupielādēt attēlu* - lejupielādēt logrīku kā PNG attēlu. Šī opcija ir pieejama logrīkiem, kas atbalsta šo iespēju (skatiet atsevišķās logrīku lapas). Atkarībā no jūsu pārlūka iestatījumiem attēls tiek vai nu automātiski saglabāts mapē Lejupielādes, vai arī tiek atvērts dialoglogs Saglabāt, kas ļauj izvēlēties atrašanās vietu. *Atsvaidzināšanas intervāls* - izvēlēties logrīka satura atsvaidzināšanas biežumu

Atļaujas informācijas paneļiem

Atļaujas informācijas paneļiem parastajiem lietotājiem un lietotājiem ar tipu "Admin" (ja viņiem piešķirtajai lietotāja lomai ir iespējota piekļuve informācijas paneļiem un opcija *Create and edit dashboards*) ir ierobežotas, kā aprakstīts tālāk.

- Viņi var skatīt un klonēt publisku informācijas paneli pat tad, ja informācijas paneļa kopīgošanas iestatījumos nav pievienoti lietotāji vai lietotāju grupas.
- Viņi var skatīt un klonēt privātu informācijas paneli, ja tam ir iestatītas vismaz *Read* tiesības, izmantojot kopīgošanas iestatījumus.
- Viņi var rediģēt un dzēst informācijas paneli tikai tad, ja tam ir iestatītas *Read-write* tiesības, izmantojot kopīgošanas iestatījumus.
- Viņi nevar mainīt informācijas paneļa īpašnieku.

1 Informācijas paneļa logrīki

Pārskats

Zabbix nodrošina dažādus paneļa logrīkus, kas vizualizē un apkopo uzraudzības datus, sniedzot jums skaidru pārskatu par sistēmas veiktspēju un pieejamību.

Tālāk ir sniegts visu logrīku pārskats, sagrupējot tos pēc to galvenā mērķa.

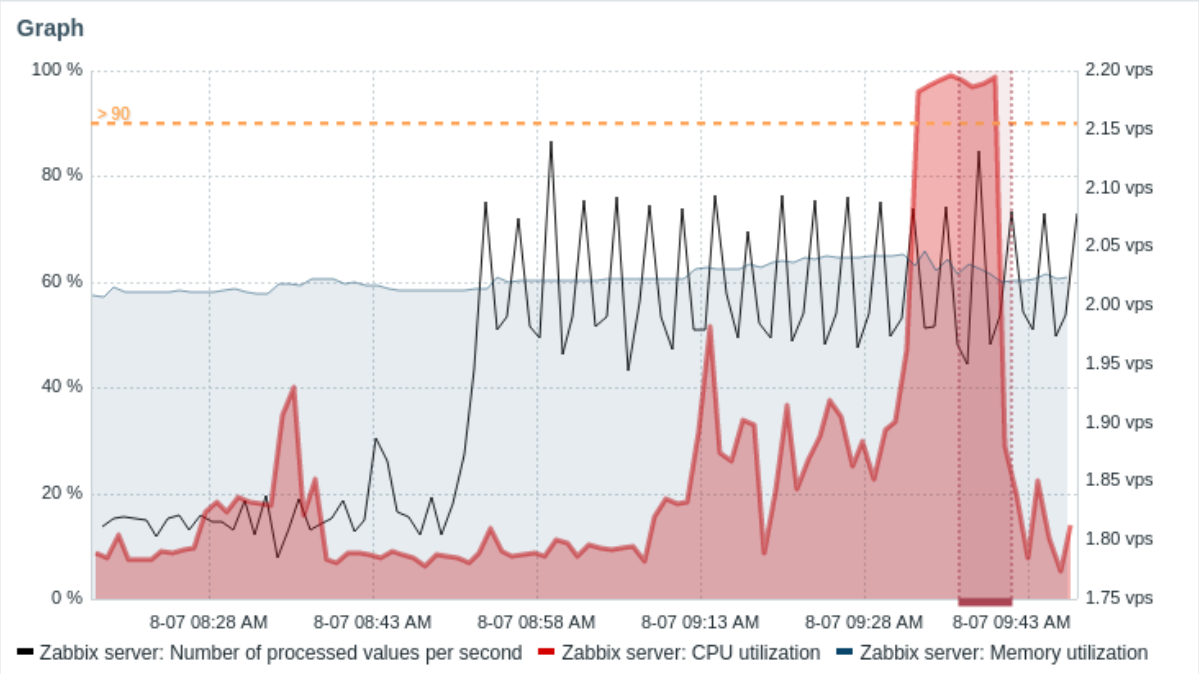
Lai iegūtu detalizētu informāciju par logrīku konfigurāciju, skatiet Paneļa logrīku parametrus.

Datu vizualizācija

Šie logrīki attēlo datus dažādos vizuālos formātos, palīdzot jums pārraudzīt datus un problēmas dažādos veidos.

Grafiks

Grafiks logrīks attēlo skaitlisku vienumu datus kā vektorgrafiku. Tas var palīdzēt jums sekot līdzi metrikām, pamanīt problēmas un salīdzināt vērtības laika gaitā un starp hostiem.

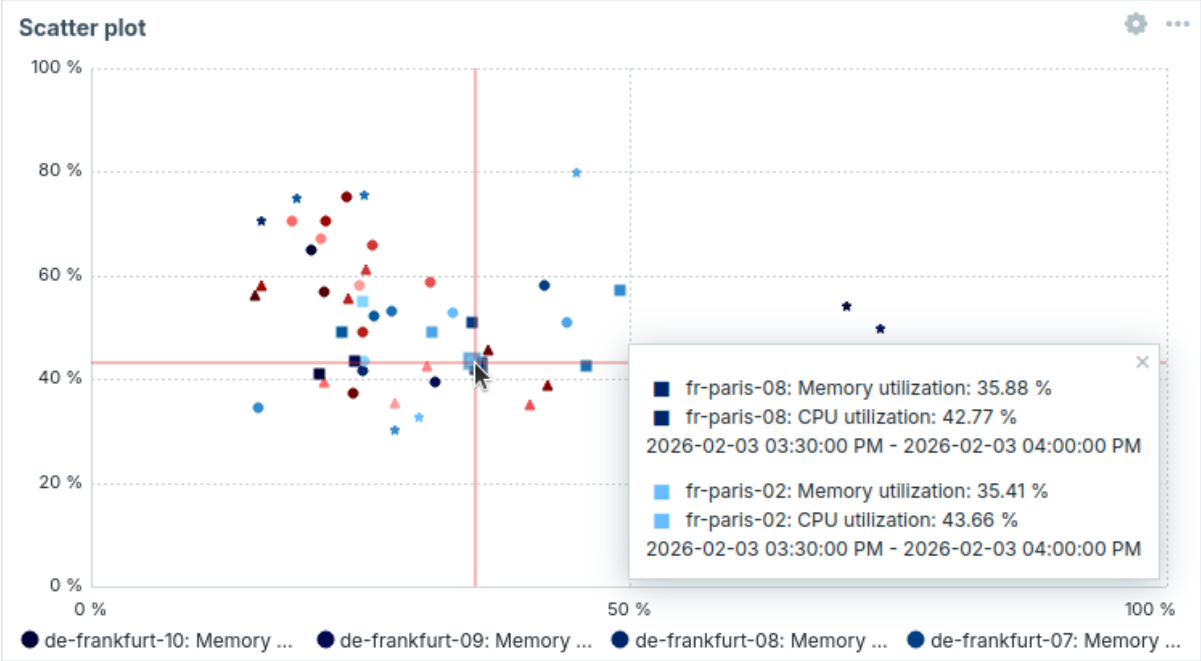


Skatiet arī:

- Grafiks (klasiskais) - mantotais Grafika logrīks, kas attēlo skaitlisku vienumu datus kā uz attēlu balstītu grafiku.
- Grafika prototips - attēlo automātiski izveidotu pielāgotu grafiku režģi, pamatojoties uz zema līmeņa atklāšanas kārtulas grafiku prototipiem vai vienumu prototipiem.
- Iecienītie grafiki - attēlo alfabētisku saīšņu sarakstu uz grafikiem, ko pašreizējais lietotājs atzīmējis kā iecienītus.

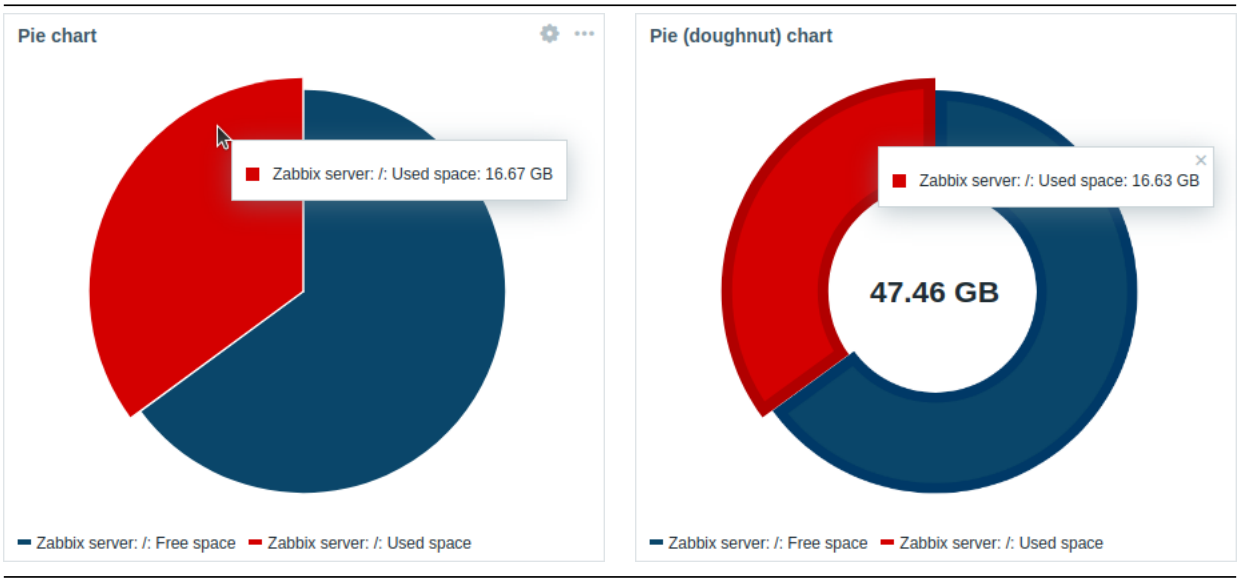
Izkliedes diagramma

Izkliedes diagrammas logrīks attēlo saistību starp diviem rādītājiem, izvietojot atsevišķus datu punktus gar X un Y asi. Tas palīdz atklāt datu kopas modeļus, grupas, korelācijas un izņēmuma vērtības.



Sektoru diagramma

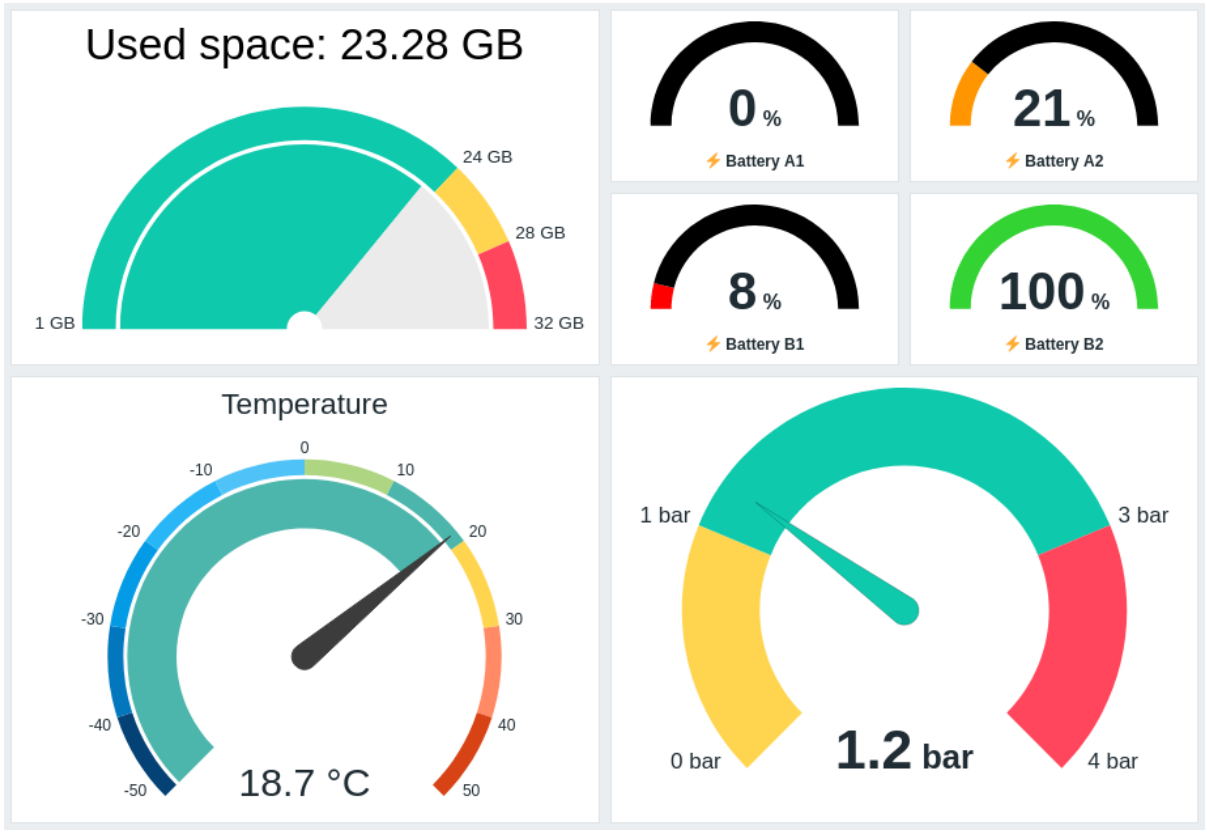
Sektoru diagrammas logrīks attēlo skaitliskus vienumu datus kā vektorgrafikas sektoru vai gredzenveida diagrammu. Šis logrīks var palīdzēt vizualizēt, kā vienumi vai hosti veido kopējo datu kopu.



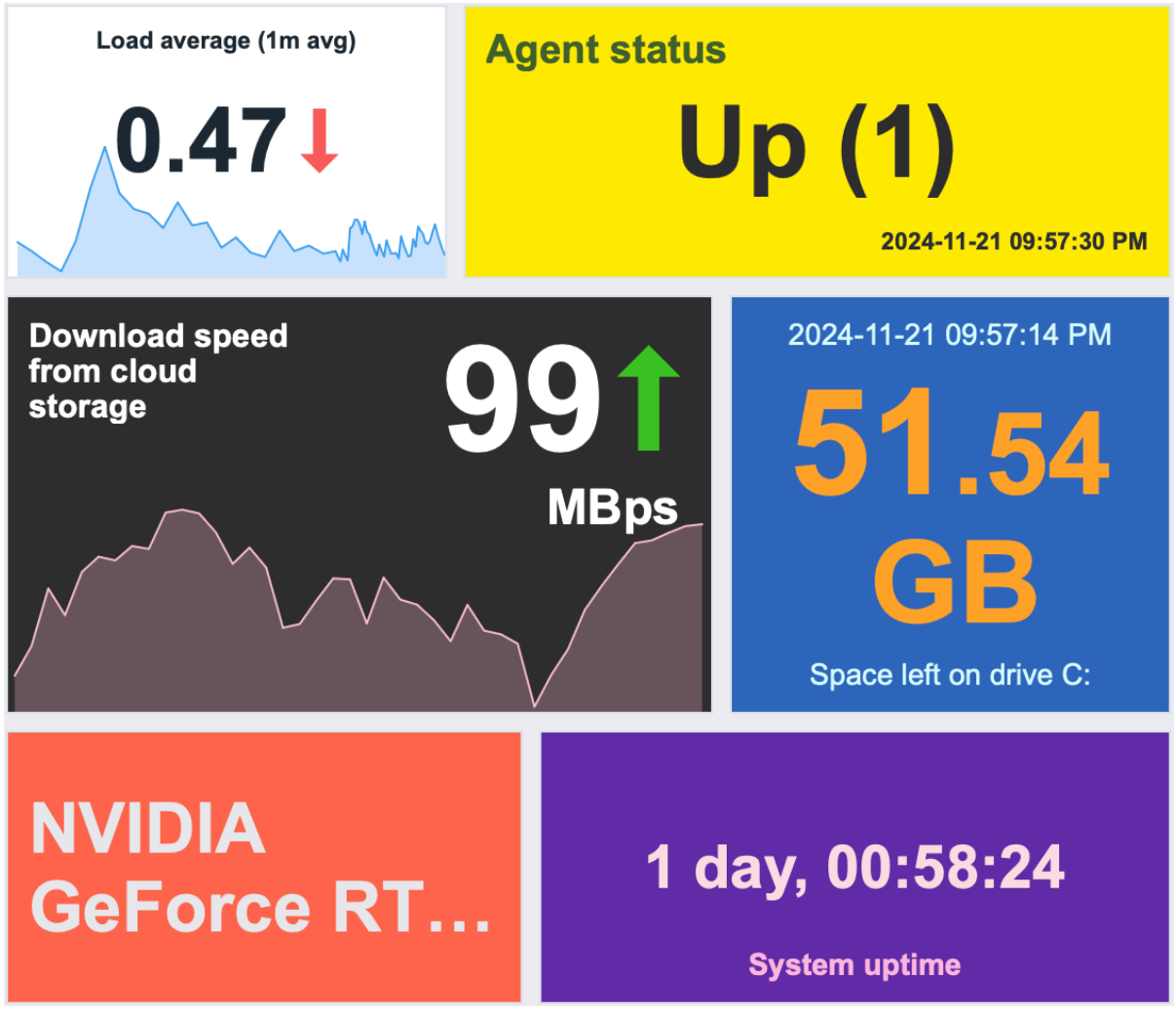
Mērinstruments/vienuma vērtība

Mērinstruments/Vienuma vērtība logrīki attēlo viena vienuma vērtību, kas ir noderīgi, lai sekotu svarīgiem rādītājiem, vizualizētu sliekšņus un noteiktu pēkšņas datu izmaiņas.

Mērinstrumenta logrīks:

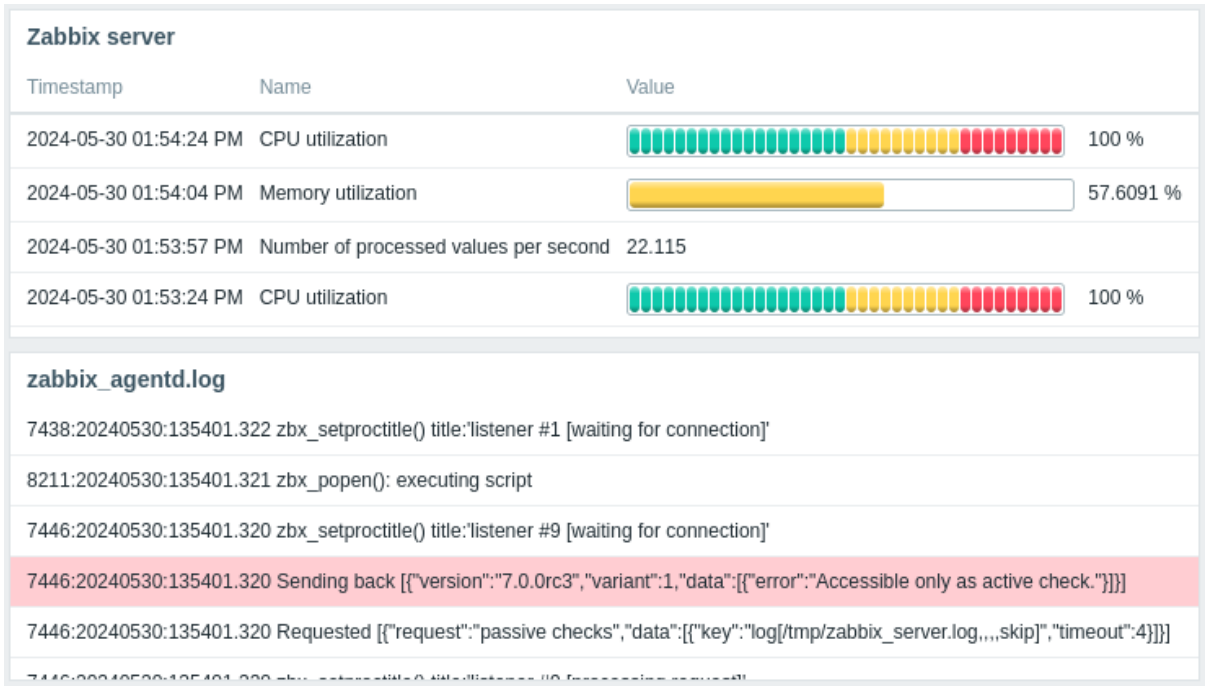


Vienuma vērtības logrīks:



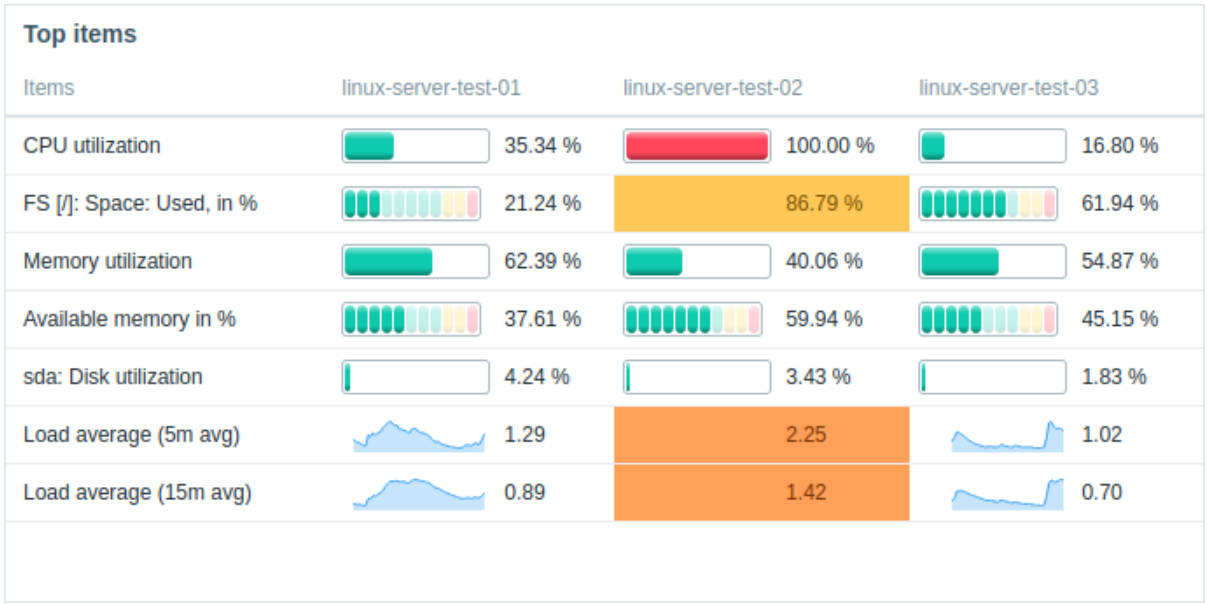
Vienumu vēsture

Logrīks Vienumu vēsture attēlo jaunākos datus visiem vienumu tipiem (skaitliskiem, teksta u.c.) tabulas formātā. Tas var rādīt progresa joslas un attēlus (noderīgi pārlūka vienumiem), kā arī izcelt vērtības (noderīgi žurnālfailu uzraudzībai).



Populārākie vienumi

Logrīks Top items attēlo atlasīto vienumu augstākās/zemākās vērtības, nodrošinot ātru pārskatu par to veiktspēju.

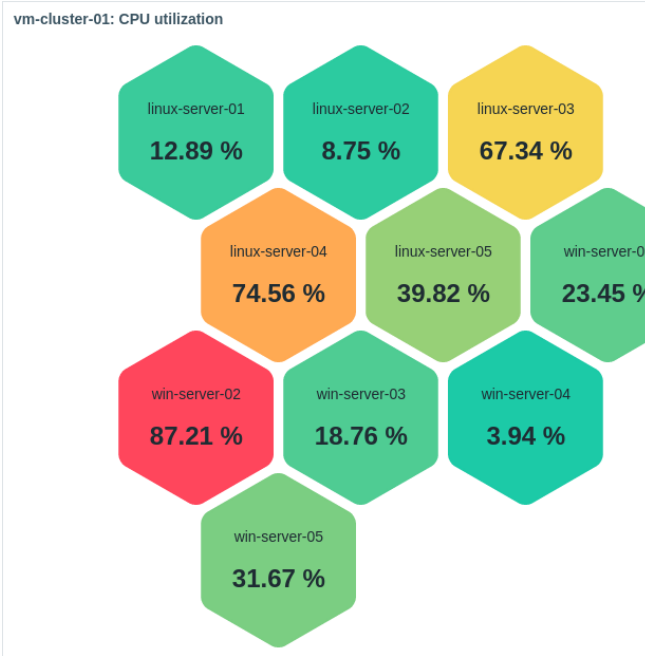


Skatiet arī:

- Top hosts — attēlo atlasīto vienumu augstākās/zemākās vērtības no vairākiem hostiem.
- Top triggers — attēlo trigerus ar lielāko problēmu skaitu.

Honeycomb

Honeycomb logrīks attēlo hostus vai vienumus kā sešstūrainas šūnas režģī. Šāds izkārtojums ļauj viegli noteikt problemātiskus hostus/vienumus, pamanīt problēmu kopas, ātri salīdzināt grupas un daudz ko citu.

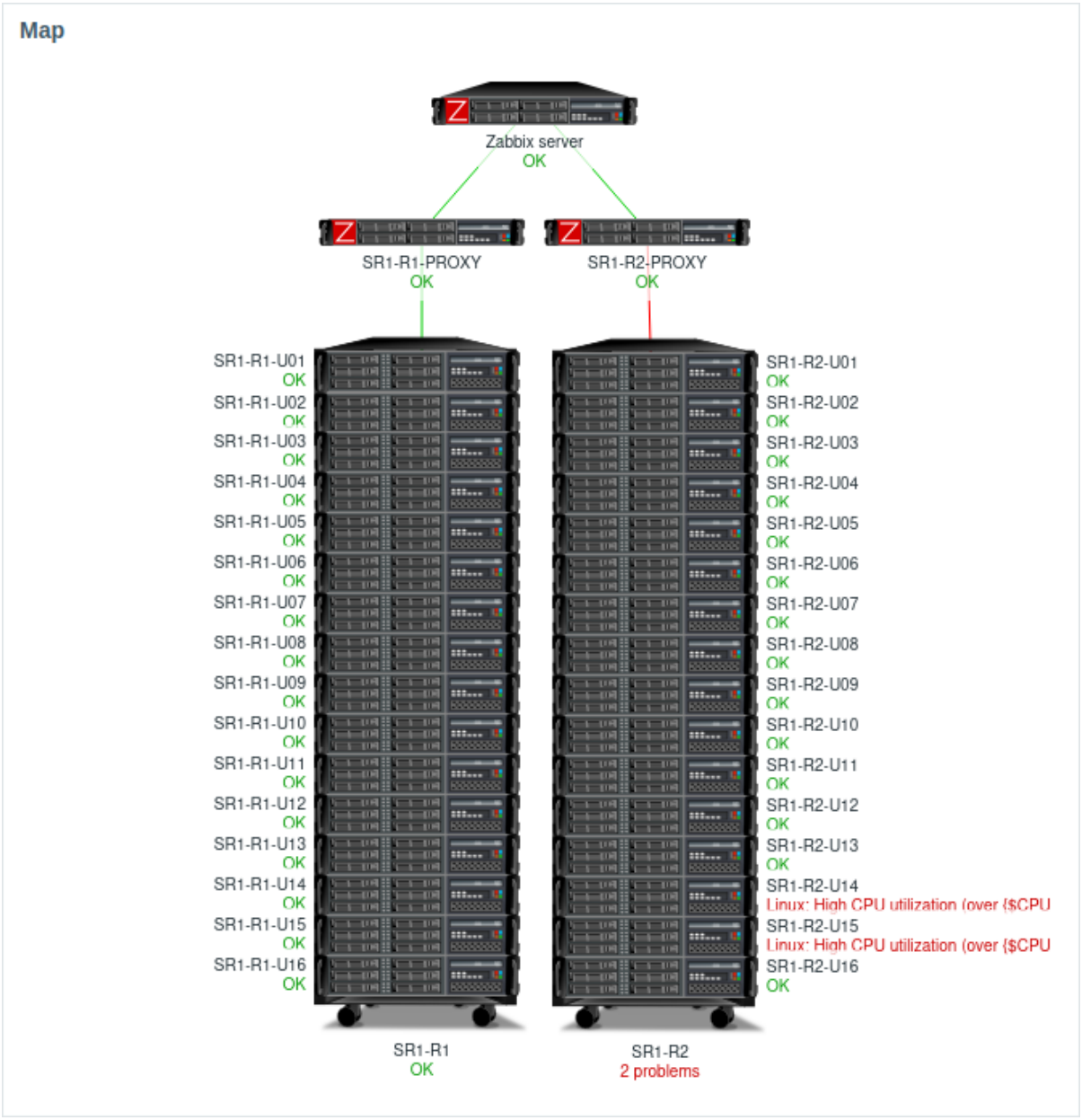


Kartes

Karšu logrīki vizualizē tīkla topoloģiju vai hostu atrašanās vietas, palīdzot novērtēt vispārējo un konkrētai atrašanās vietai raksturīgo stāvokli.

Karte

Logrīks Karte var attēlot tīkla karti vai līdzīgu vizualizāciju, sniedzot jums dinamisku pārskatu par jūsu tīklu.

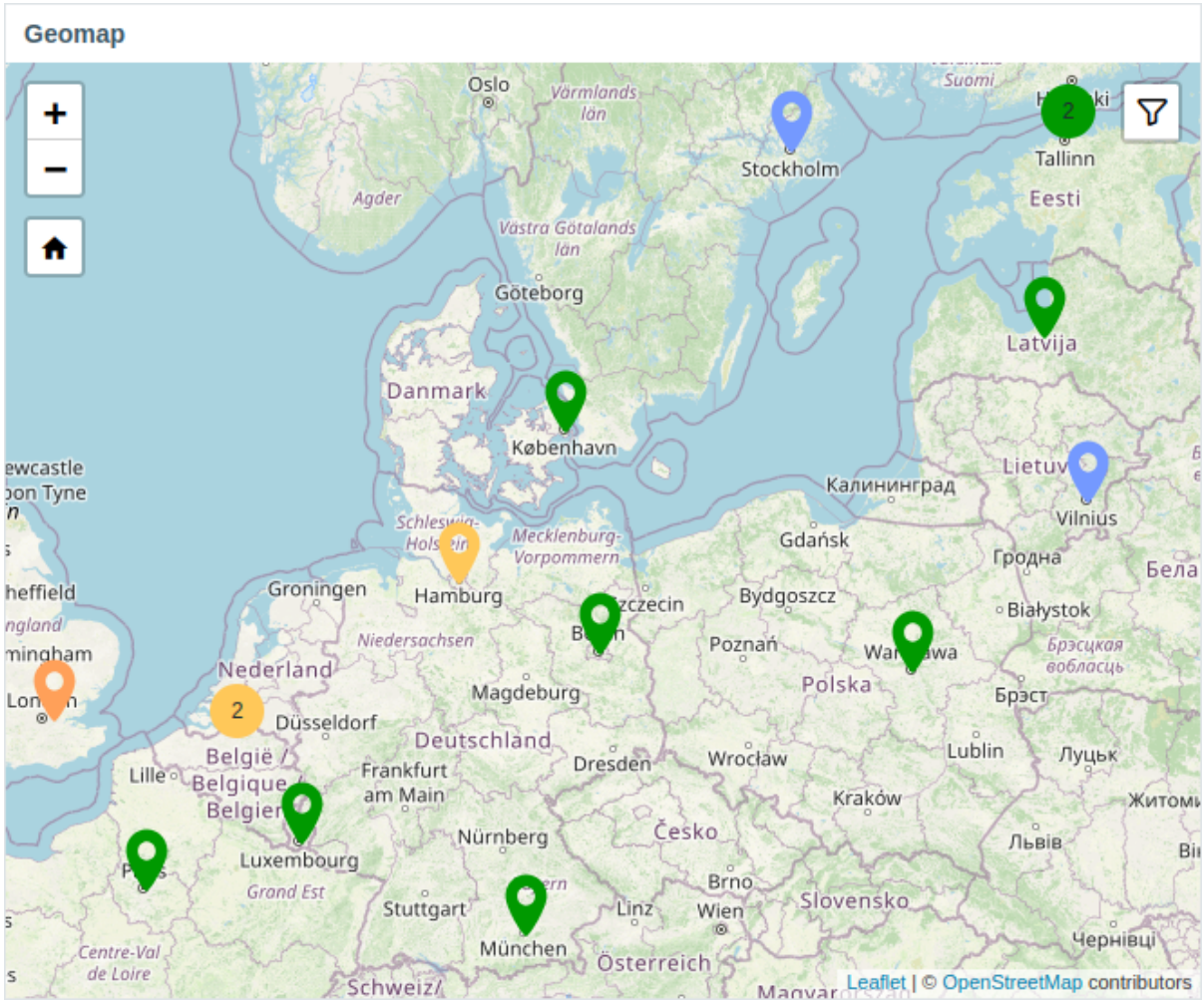


Skatiet arī:

- Kartes navigācijas koks — attēlo esošo karšu hierarhiju un problēmu skaitu katrai kartei un karšu grupai.
- Iecienītās kartes — attēlo pašreizējā lietotāja kā iecienītas atzīmēto karšu īsceļu sarakstu, sakārtotu alfabētiskā secībā.

Ģeokarte

Ģeokartes logrīks attēlo hostus kā marķierus interaktīvā, uz Leaflet balstītā kartē, kur marķieru krāsas norāda katra hosta vissmagāko problēmu.

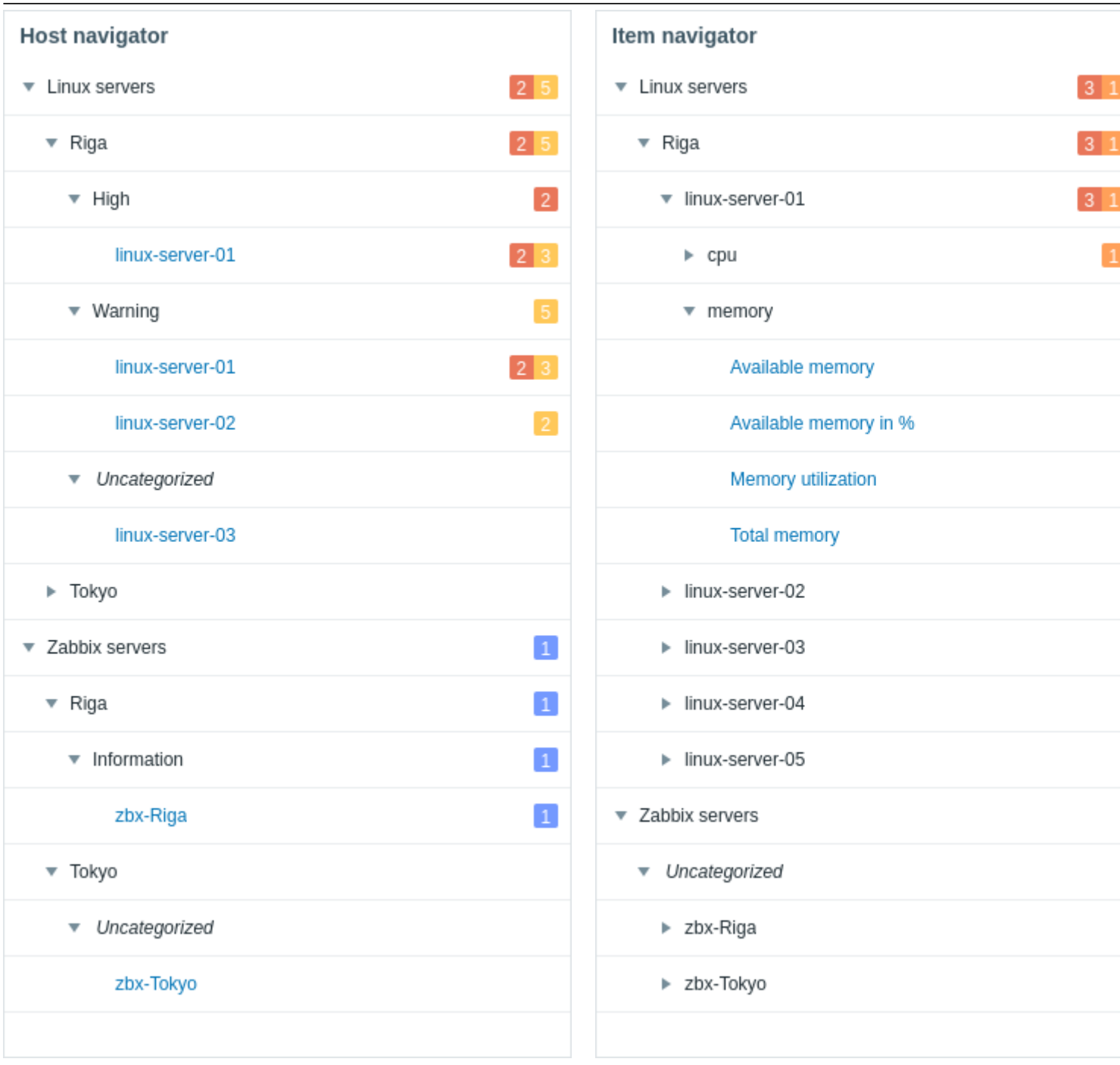


Hosta/vienuma informācija

Šie logrīki nodrošina hostu vai vienumu sarakstus un detalizētus skatus.

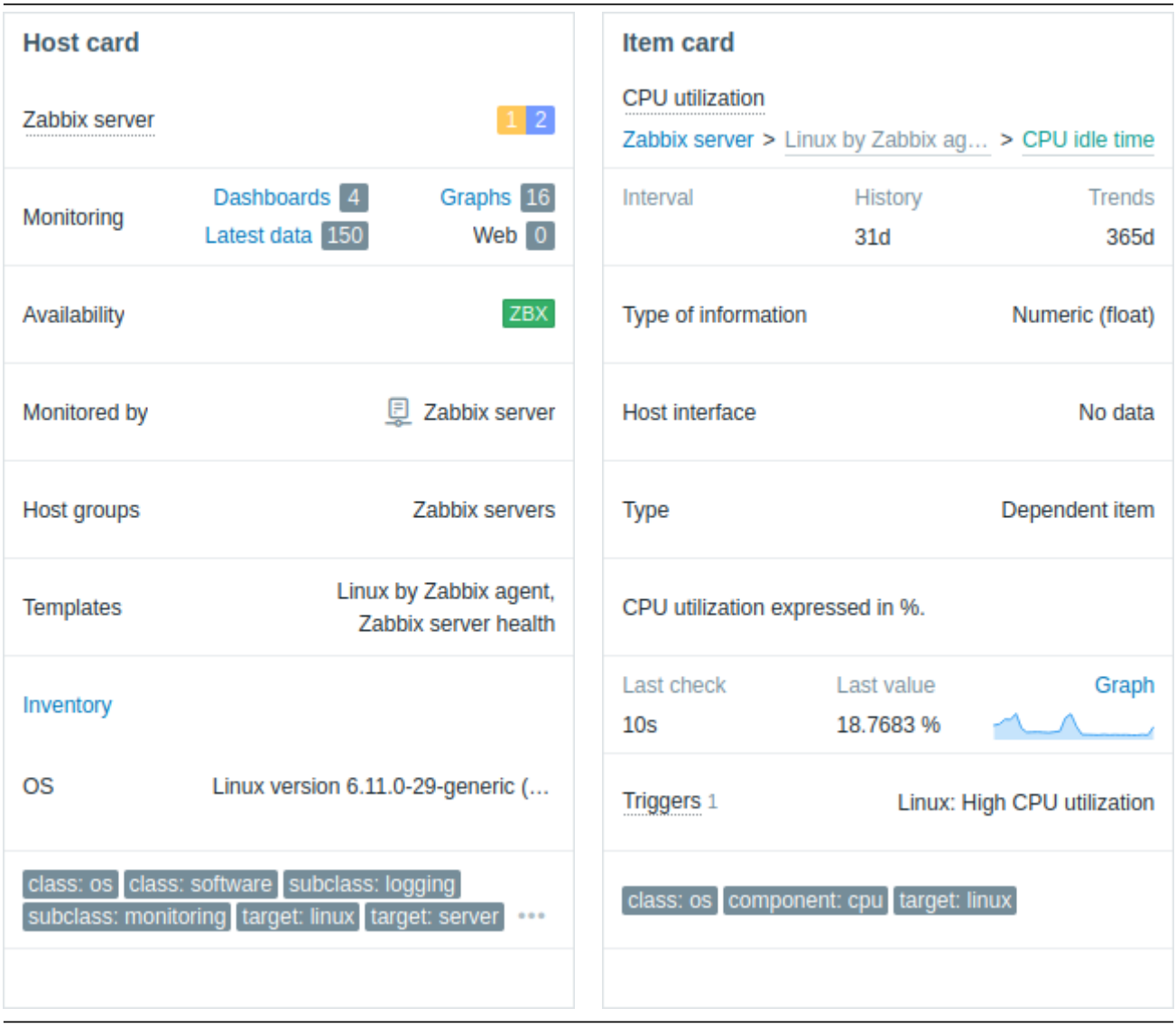
Hostu/vienumu navigators

Host navigator/Item navigator logrīki parāda hostu/vienumu sarakstu, pamatojoties uz dažādām filtrēšanas un grupēšanas opcijām. Šie logrīki ir visnoderīgākie, lai kontrolētu, ko citi logrīki attēlo, pamatojoties uz atlasīto hostu/vienumu.



Hosta/vienuma kartīte

Host card/Item card logrīki attēlo detalizētu informāciju par vienu hostu/vienumu, ļaujot jums ātri novērtēt hosta vai vienuma statusu un konfigurāciju (hosta pieejamību, problēmas, inventāru, vienuma jaunākos datus, kļūdas, aprakstu u.c.).



Statuss un atskaites

Šie logrīki attēlo statusa datus (piemēram, pieejamību, trigerus, problēmas), kā arī atskaites.

Hostu pieejamība

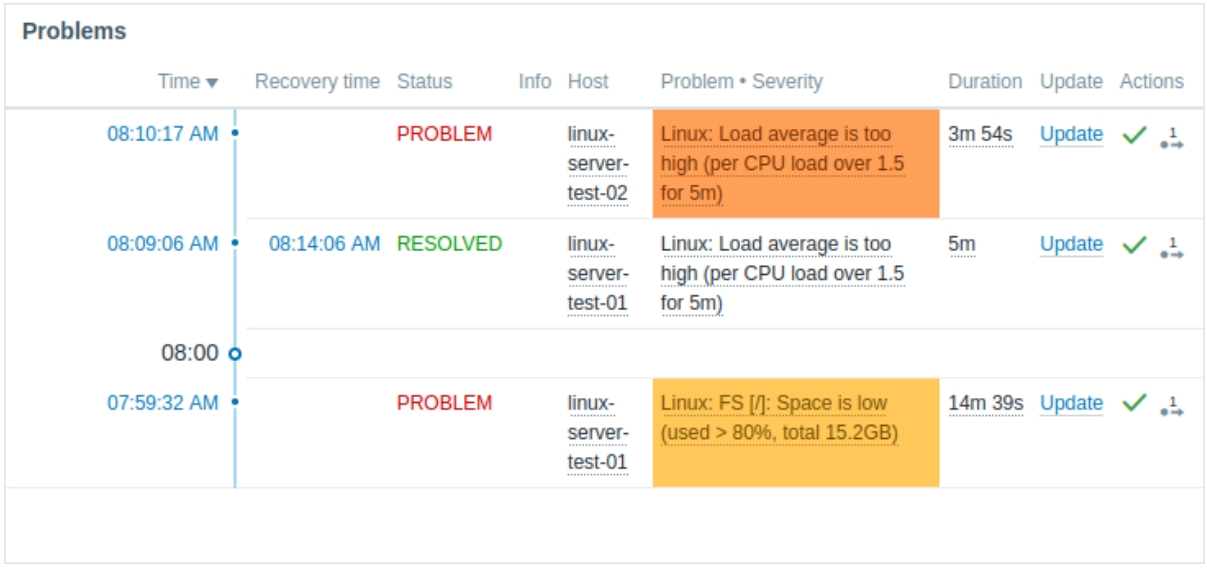
Logrīks Host availability attēlo hostu pieejamību atlasītajās hostu grupās, ļaujot jums uzraudzīt ieslēgto vai izslēgto hostu skaitu.

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

Problēmas

Problēmas logrīks attēlo pašreizējās problēmas, kas filtrētas pēc dažādiem parametriem (hostu grupas, hosti, problēmu nosaukumi utt.), sniedzot skaidru pārskatu par to, kam jāpievērš uzmanība.



Skatiet arī:

- Problēmas pēc smaguma pakāpes - attēlo cēloņproblēmu skaitu, grupētu pēc smaguma pakāpes.
- Problēmu hosti - attēlo cēloņproblēmu skaitu katrā hostu grupā un parāda augstāko problēmas smaguma pakāpi katrā grupā.

Trigeru pārskats

Trigeru pārskats logrīks attēlo pašreizējos hostu trigeru stāvokļus kā krāsainu bloku tabulu. Tas nodrošina ātru vizuālu pārskatu par trigeru veselību un aktivitāti dažādos hostos.

Darbību žurnāls

Logrīks Action log attēlo detalizētu informāciju par darbībām (paziņojumiem, attālinātām komandām), kas izpildītas darbības ietvaros. Detalizētais žurnāls palīdz auditēšanā, problēmu novēršanā un katras darbības izpildes uzraudzībā.

Action log

Time ▾	Action	Media type	Recipient	Message	Status	Info
2025-08-15 07:58:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Resolved in 1m 59s: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem has been resolved at 10:58:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Problem duration: 1m 59s Host: Linux server Severity: Average Original problem ID: 45	Sent	
2025-08-15 07:56:30 AM	Report problems to Zabbix administrators	Email	Admin (Zabbix Administrator) *admin@example.com*	**Problem: Linux: Load average is too high (per CPU load over 1.5 for 5m)** Problem started at 10:56:30 on 2025.08.15 Problem name: Linux: Load average is too high (per CPU load over 1.5 for 5m) Host: Linux server Severity: Average Operational data: Load averages(1m 5m 15m): (6.256836 4.808105 3.398438), # of CPUs: 2 Original problem ID: 45	Sent	
2025-08-15 07:42:30 AM	Report problems to Zabbix administrators		Admin (Zabbix Administrator)		Failed	i

SLA atskaite

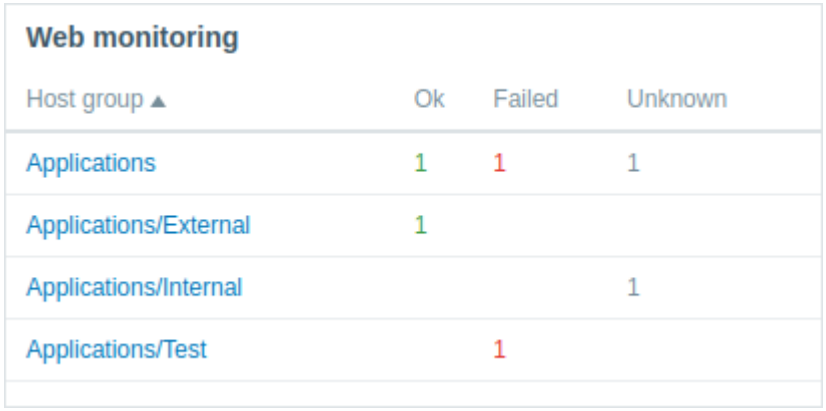
SLA atskaite logrīks attēlo SLA atskaites, palīdzot jums uzraudzīt pakalpojumu veiktspēju attiecībā pret noteiktajiem mērķiem.

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

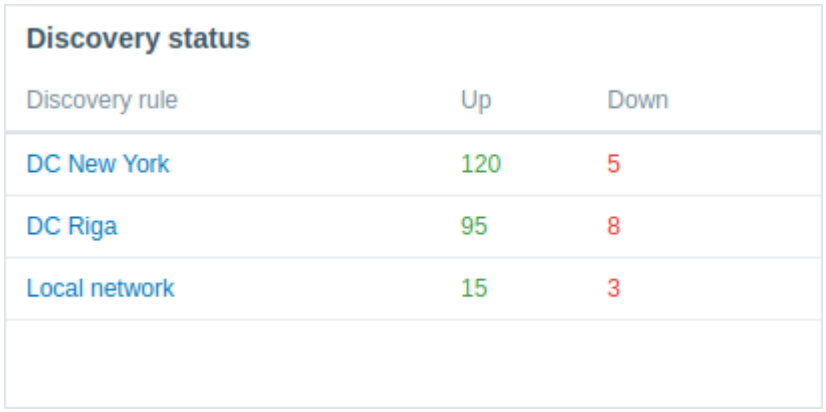
Tīmekļa uzraudzība

Web monitoring logrīks attēlo aktīvo tīmekļa scenāriju statusu.



Atklāšanas statuss

Logrīks Atklāšanas statuss parāda to ierīču statusu, kuras ir atklātas ar iespējotiem tīkla atklāšanas noteikumiem. Tas palīdz uzraudzīt, vai tīkla atklāšana darbojas pareizi, kā arī to, cik ierīču ir pieejamas vai nepieejamas.



Paneļa utilītas un sistēmas informācija

Šie logrīki nodrošina papildu informāciju un rīkus, lai uzlabotu jūsu paneļus.

Sistēmas informācija

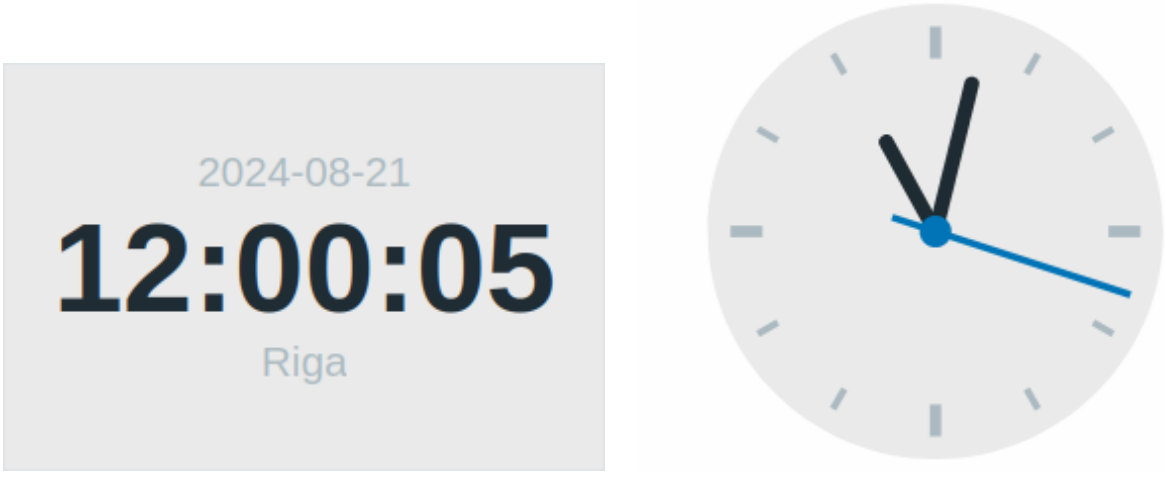
Logrīks Sistēmas informācija parāda galveno Zabbix serveris un sistēmas datu kopsavilkumu vai detalizētu informāciju par augstas pieejamības mezgliem.

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	

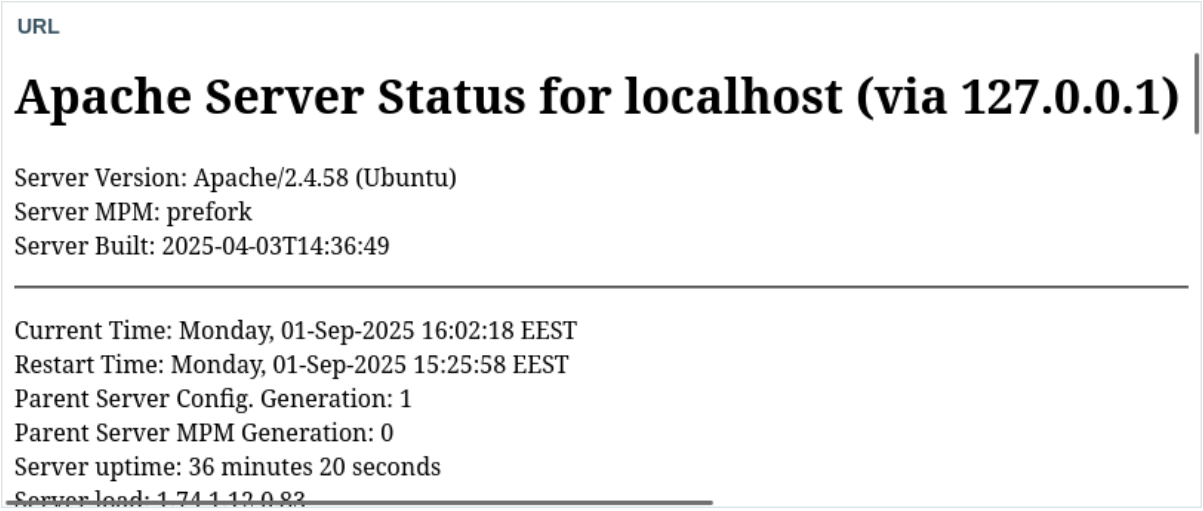
Pulkstenis

Clock logrīks attēlo lokālo, servera vai hosta laiku analogā vai digitālā formātā.



URL

URL logrīks attēlo saturu, kas iegūts no URL.



2 Informācijas paneļa logrīka parametri

Pārskats

Šajā lapā ir aprakstīti parametri, kas ir kopīgi visiem paneļa logrīkiem, kā arī dinamiskie parametri, kas ļauj vairākiem logrīkiem savstarpēji apmainīties ar konfigurācijas datiem vai apmainīties ar tiem ar paneli.

Lai skatītu katra logrīka specifiskos parametrus, lūdzu, izmantojiet sānjoslu, lai pārietu uz attiecīgā logrīka lapu.

Kopīgie parametri

Tālāk norādītie parametri ir kopīgi katram logrīkam:

Nosaukums	Ievadiet logrīka nosaukumu.
Atsvaidzināšanas intervāls	Konfigurējiet noklusējuma atsvaidzināšanas intervālu. Logrīku noklusējuma atsvaidzināšanas intervāli atkarībā no logrīka veida ir diapazonā no *Bez atsvaidzināšanas* līdz *15 minūtes*. Piemēram: - *Bez atsvaidzināšanas* logrīkam *URL*; - *1 minūte* logrīkam *Darbību žurnāls*; - *15 minūtes* logrīkam *Pulkstenis*. Atsvaidzināšanas intervālus var iestatīt kā noklusējuma vērtību visiem lietotājiem. Pārslēdziet paneli uz rediģēšanas režīmu, noklikšķiniet uz pogas rediģēt logrīku un nolaižamajā sarakstā atlasiet vajadzīgo atsvaidzināšanas intervālu. Katrs lietotājs var arī iestatīt savu logrīka atsvaidzināšanas intervālu. Paneļa skatīšanas režīmā logrīkā noklikšķiniet uz pogas ar trim punktiem un nolaižamajā sarakstā atlasiet vajadzīgo atsvaidzināšanas intervālu. Ņemiet vērā, ka lietotāja individuālais atsvaidzināšanas intervāls ir prioritārs pār logrīka iestatījumu un tiek saglabāts pat tad, ja logrīka iestatījums tiek mainīts.
Rādīt galveni	Atzīmējiet izvēles rūtiņu, lai logrīka galvene tiktu rādīta pastāvīgi. Ja izvēles rūtiņa nav atzīmēta, galvene tiek paslēpta, lai ietaupītu vietu, un kļūst redzama tikai tad, kad peles kursors tiek novietots virs logrīka (gan skatīšanas, gan rediģēšanas režīmā). Galvene ir arī daļēji redzama, velkot logrīku uz jaunu vietu.

Dinamiskie parametri

Vairāki logrīka parametri (piemēram, *Hosts*, *Override host*, *Time period*) atbalsta dinamiskus datu avotus. Tā vietā, lai konfigurētu statiskas vērtības, šos parametrus var sasaistīt ar citiem logrīkiem vai pašu paneli. Pēc tam logrīki tiek automātiski atjaunināti, pamatojoties uz atlasēm vai apraidēm no citām paneļa daļām.

Logrīku saderība

Daži logrīki var pārraidīt konfigurācijas datus citiem logrīkiem, daži var saņemt datus, bet daži var darīt abus. Piemēram:

- Logrīks *Action log* var tikai saņemt laika perioda datus no logrīkiem *Graph*, *Graph (classic)* un *Graph prototype*.
- Logrīks *Geomap* var pārraidīt hostu datus logrīkiem, kas tos saņem (*Honeycomb*, *Top items* u.c.), un var arī saņemt hostu grupu un hostu datus no logrīkiem, kas tos pārraida (*Honeycomb*, *Problem hosts* u.c.).

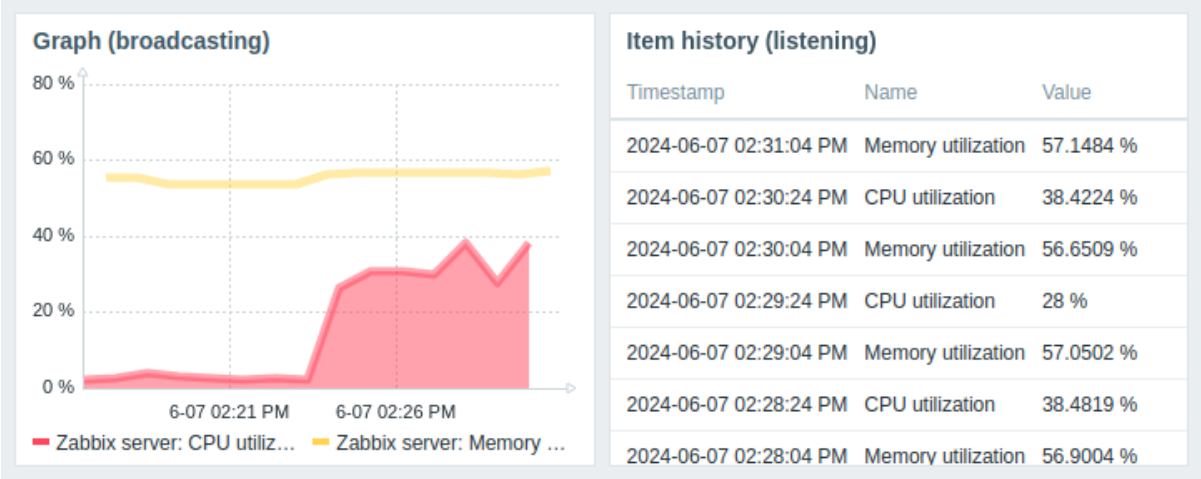
- Logrīks *Clock* nevar ne pārraidīt, ne saņemt nekādus datus.

Tālāk esošajā tabulā ir norādītas katra logrīka pārraidīšanas un datu saņemšanas iespējas.

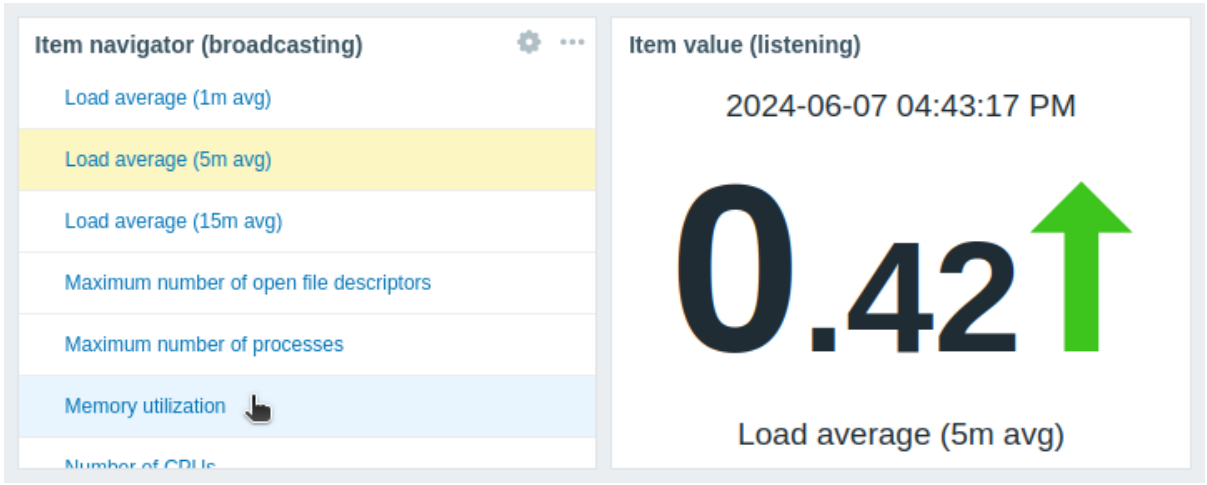
Widget	Broadcasts	Listens
Action log	-	Laika periods
Clock	-	-
Discovery status	-	-
Favorite graphs	-	-
Favorite maps	-	-
Gauge	-	Hosti, vienumi
Geomap	Hosti	Hostu grupas, hosti
Graph	Laika periods, vienumi	Hosti, vienumi, laika periods
Graph (classic)	Laika periods	Hosti, vienumi, grafiki, laika periods
Graph prototype	Laika periods	Hosti, laika periods
Honeycomb	Hosti, vienumi	Hostu grupas, hosti
Host availability	-	Hostu grupas
Host card	-	Hosti
Host navigator	Hosti	Hostu grupas
Item card	-	Hosti, vienumi, laika periods
Item history	Vienumi	Hosti, laika periods
Item navigator	Vienumi	Hostu grupas, hosti
Item value	-	Hosti, vienumi, laika periods
Map	Hostu grupas, hosti	Kartes
Map navigation tree	Kartes	-
Pie chart	-	Vienumi, laika periods
Problem hosts	Hostu grupas	Hostu grupas, hosti
Problems	Notikumi	Hostu grupas, hosti
Problems by severity	Hostu grupas	Hostu grupas, hosti
Scatter plot	Laika periods, vienumi	Hosti, vienumi, laika periods
SLA report	-	-
System information	-	-
Top hosts	Hosti	Hostu grupas, hosti
Top items	-	Hostu grupas, hosti
Top triggers	-	Laika periods
Trigger overview	-	Hostu grupas, hosti
URL	-	Hosti
Web monitoring	Hostu grupas	Hostu grupas, hosti

Logrīki atšķiras pēc tā, kā tie **pārraida** datus citiem logrīkiem.

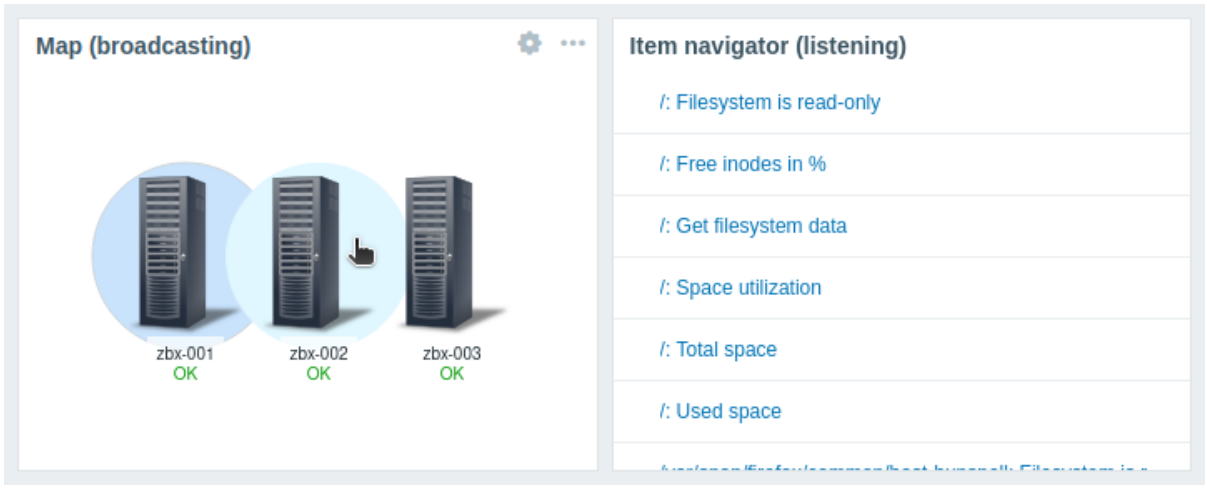
Logrīki, kas atbalsta datu pārraidīšanu, sāk to darīt automātiski pēc izveides. Piemēram, logrīks *Graph* nekavējoties pārraida laika perioda datus klausīšanās režīmā esošajiem logrīkiem.



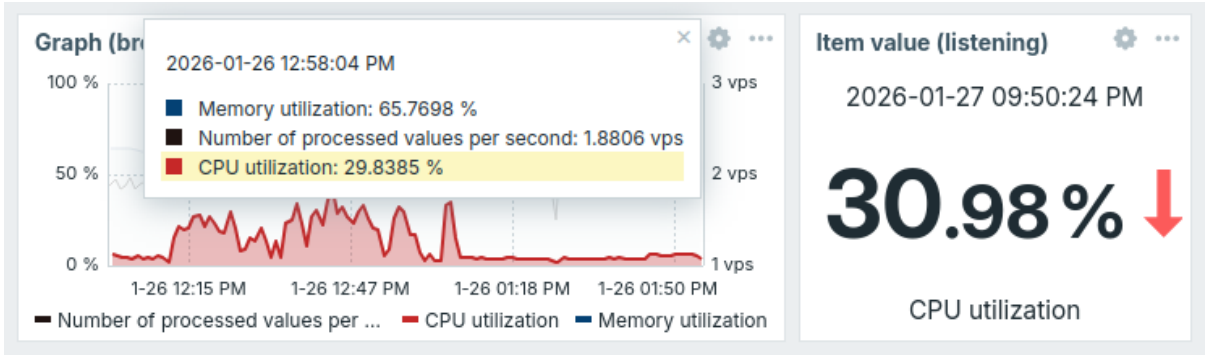
Logrīki, kas atbalsta entītiju atlasi, automātiski pārraida datus par pirmo pieejamo entītiju. Piemēram, logrīks *Item navigator* pārraida datus par pirmo vienumu savā vienumu sarakstā. Kad tiek atlasīts cits vienums, tas pārraida datus par šo vienumu. Novietojot peles kursoru virs vienuma, tas tiek izcelts gaiši zilā krāsā; atlasot to, tas tiek izcelts dzeltenā krāsā.



Logrīks *Map* darbojas līdzīgi, pārraidot datus par pirmo elementu, kas atrodas vistuvāk logrīka redzamās zonas augšējam kreisajam stūrim (logrīkam *Geomap* — vistuvāk centram). Kad tiek atlasīts cits elements, tas pārraida datus par šo elementu. Novietojot peles kursoru virs elementa, tas tiek izcelts gaiši zilā krāsā; atlasot to, tas tiek izcelts tumši zilā krāsā.

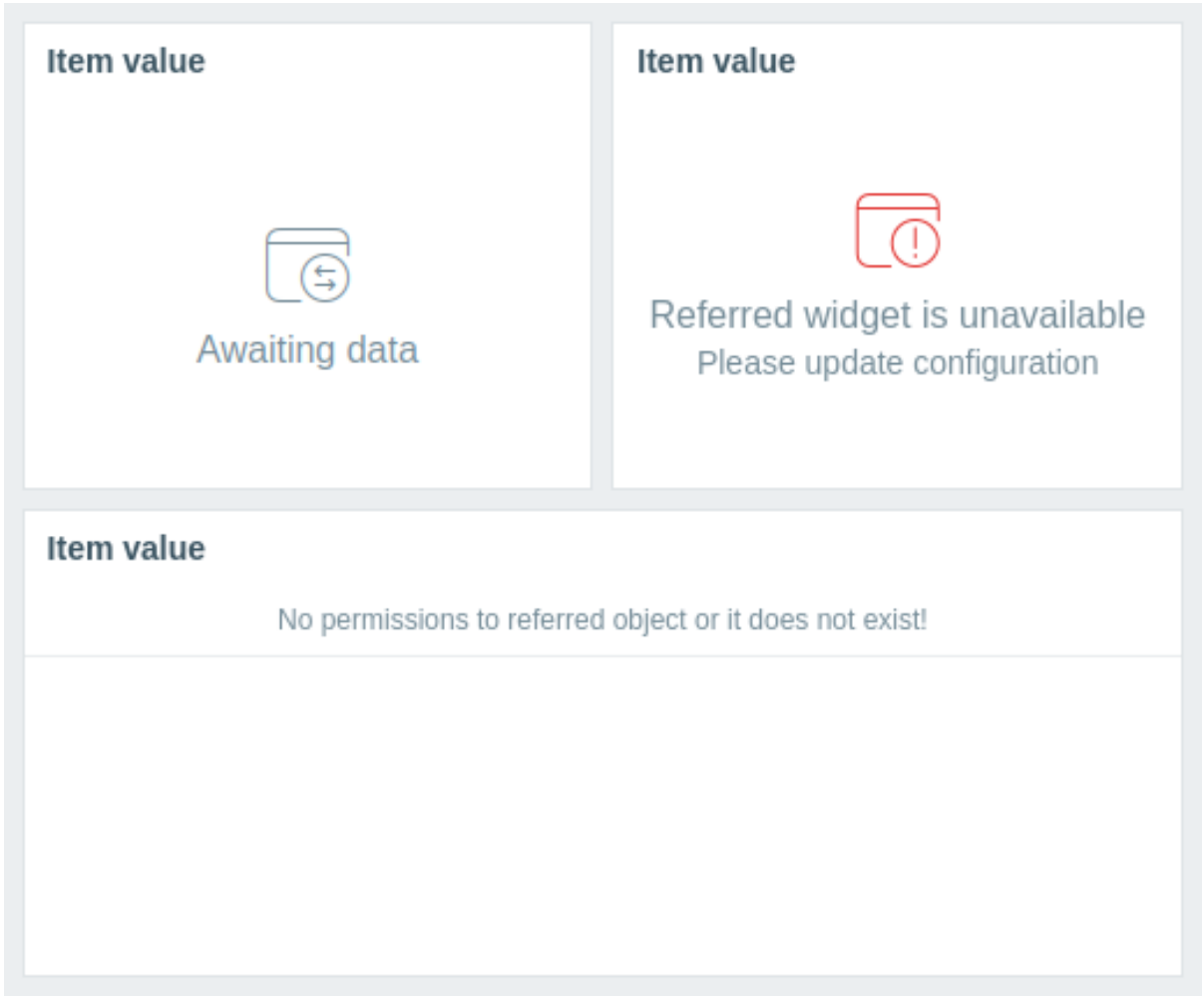


Tādi logrīki kā *Graph* un *Scatter plot* var pārraidīt vienumu datus, kad vienums ir atlasīts to uznirstošajos paskaidrojumos:



Logrīkiem ir arī noteikta darbība, kad tie **saņem** datus no citiem logrīkiem:

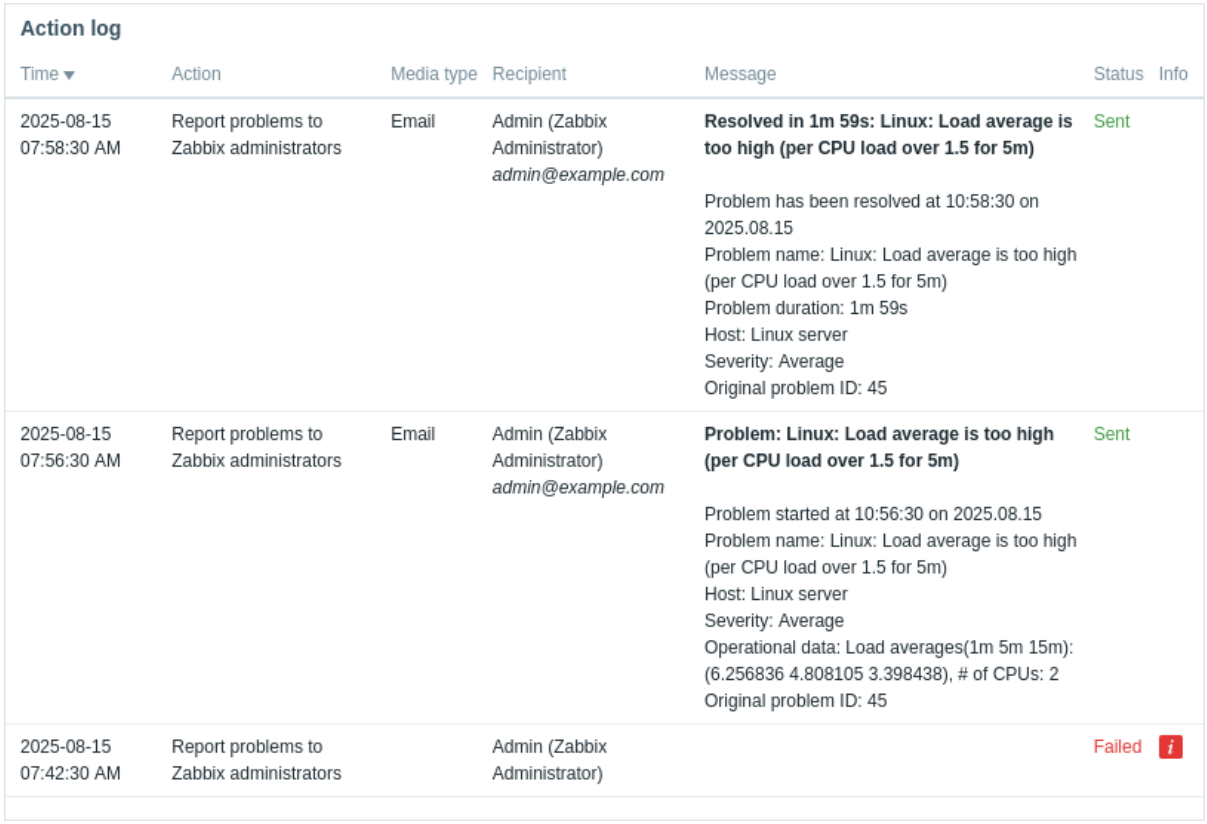
- Ja datu avota logrīks nepārraida datus, klausīšanās režīmā esošais logrīks pāriet stāvoklī *Awaiting data*.
- Ja datu avota logrīks ir dzēsts, aizstāts ar nesaderīgu logrīku vai pārvietots uz citu paneļa lapu, klausīšanās režīmā esošais logrīks pāriet stāvoklī *Referred widget is unavailable*.
- Ja datu avotā (logrīkā vai panelī) norādītajam hostam nav entītijas, kas konfigurēta klausīšanās režīmā esošajā logrīkā (vienums, grafiks, karte utt.), vai arī lietotājam nav atļauju piekļūt hostam, klausīšanās režīmā esošais logrīks parāda šādu ziņojumu: *"No permissions to referred object or it does not exist!"*



1 Darbību žurnāls

Pārskats

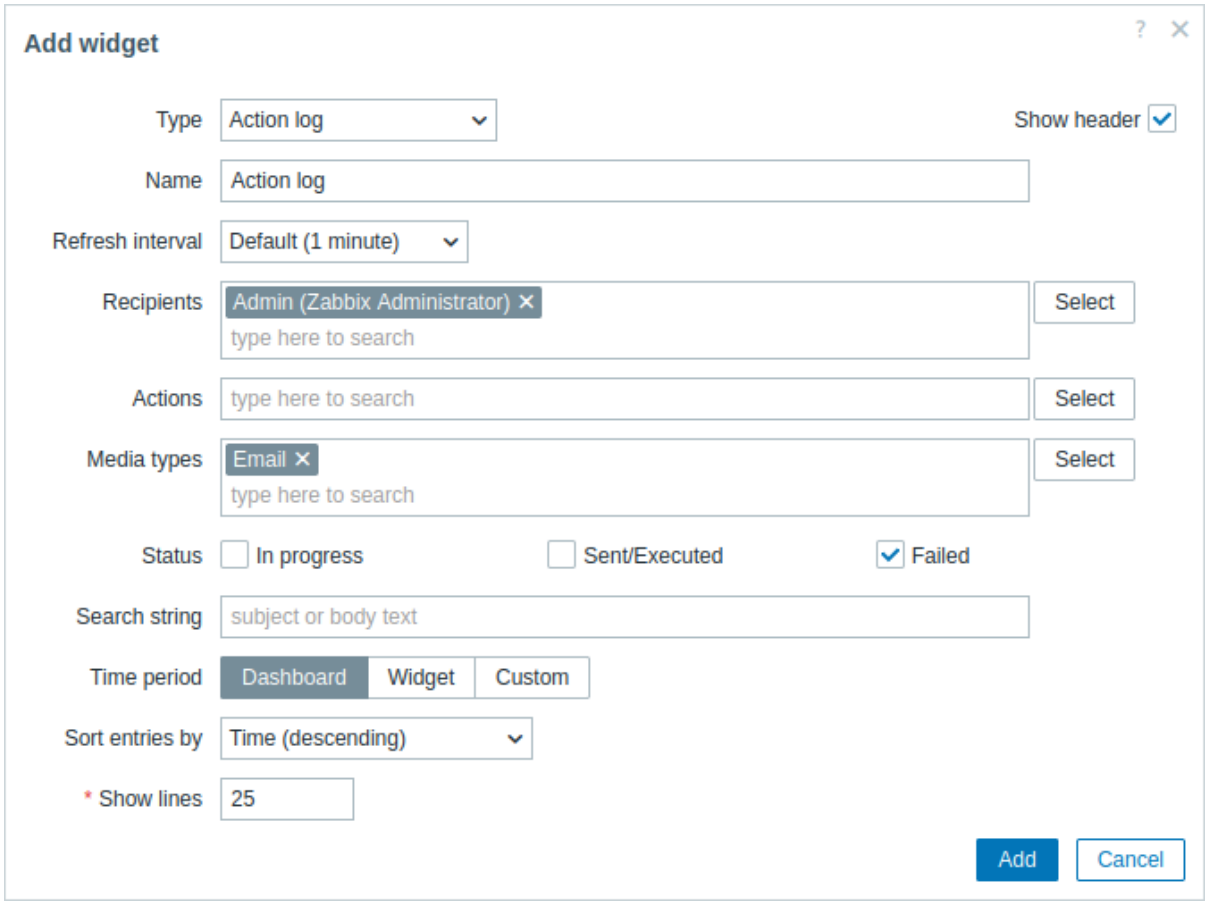
Logrīks *Darbību žurnāls* parāda detalizētu informāciju par operācijām (paziņojumiem, attālinātajām komandām), kas izpildītas darbībā. Detalizētais žurnāls palīdz auditēšanā, problēmu novēršanā un katras darbības izpildes uzraudzībā.



Tas parāda tos pašus datus kā *Atskaites* > *Darbību žurnāls* un var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Action log* kā tipu:



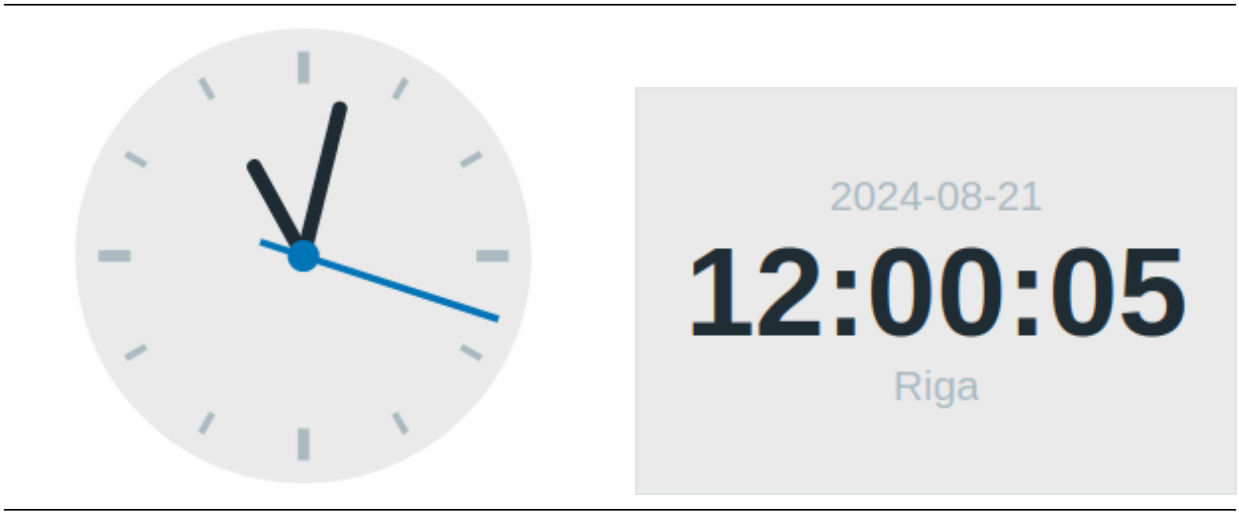
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Recipients	Filtrēt ierakstus pēc saņēmējiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot ievadīt saņēmēja nosaukumu, tiks piedāvāts atbilstošo saņēmēju nolaižamais saraksts. Ja nav atlasīts neviens saņēmējs, tiks parādīta darbību operāciju informācija visiem saņēmējiem.
Actions	Filtrēt ierakstus pēc darbībām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot ievadīt darbības nosaukumu, tiks piedāvāts atbilstošo darbību nolaižamais saraksts. Ja nav atlasīta neviena darbība, tiks parādīta darbību operāciju informācija visām darbībām.
Media types	Filtrēt ierakstus pēc mediju tipiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot ievadīt mediju tipa nosaukumu, tiks piedāvāts atbilstošo mediju tipu nolaižamais saraksts. Ja nav atlasīts neviens mediju tips, tiks parādīta darbību operāciju informācija visiem mediju tipiem.
Status	Atzīmējiet izvēles rūtiņu, lai filtrētu ierakstus pēc attiecīgā statusa: **In progress** - tiek parādītas darbību operācijas, kas ir izpildes procesā; **Sent/Executed** - tiek parādītas darbību operācijas, kas ir nosūtījušas paziņojumu vai ir izpildītas; **Failed** - tiek parādītas darbību operācijas, kas ir neizdevušās.
Search string	Filtrēt ierakstus pēc ziņojuma/attālinātās komandas satura. Ja šeit ievadīsiet virkni, tiks parādītas tikai tās darbību operācijas, kuru ziņojums/attālinātā komanda satur ievadīto virkni. Makrosi netiek atrisināti.
Time period	Filtrēt ierakstus pēc laika perioda. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (norādīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Widget”.
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
Sort entries by	Kārtot ierakstus pēc: **Time** (dilstošā vai augošā secībā); **Type** (dilstošā vai augošā secībā); **Status** (dilstošā vai augošā secībā); **Recipient** (dilstošā vai augošā secībā).
Show lines	Iestatiet, cik darbību žurnāla rindas tiks parādītas logrīkā.

2 Pulkstenis

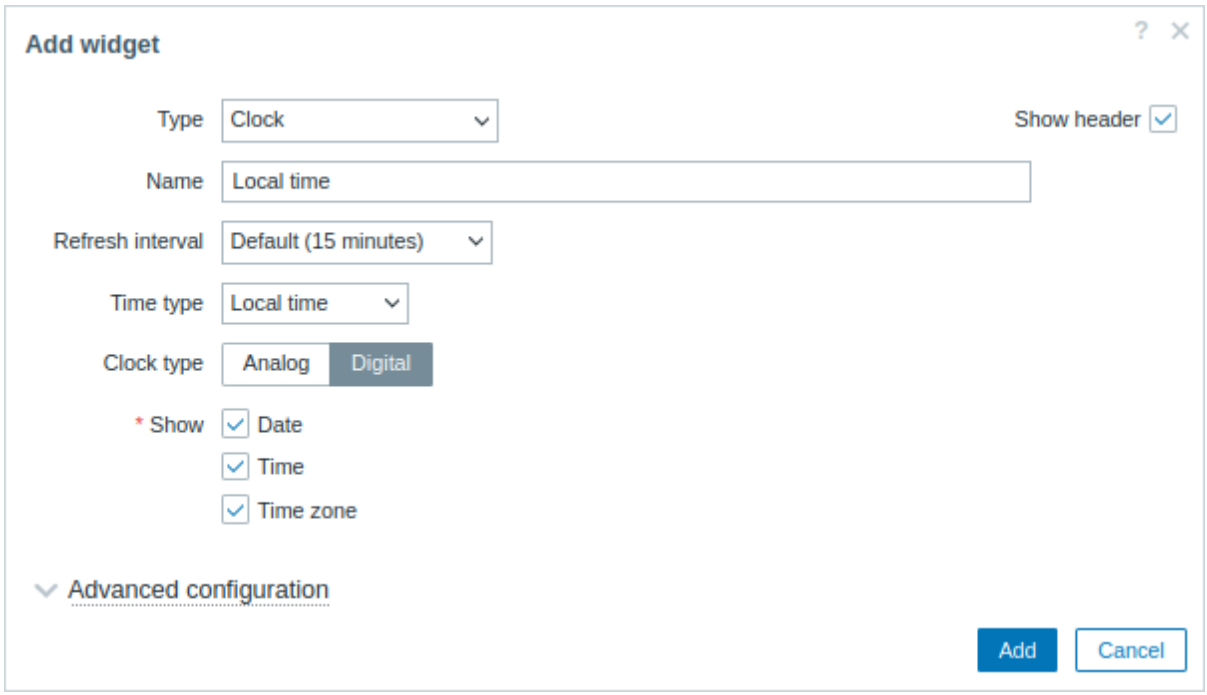
Pārskats

Pulksteņa logrīks attēlo lokālo, servera vai hosta laiku analogā vai digitālā formātā.



Konfigurācija

Lai konfigurētu, atlasiet *Clock* kā tipu:



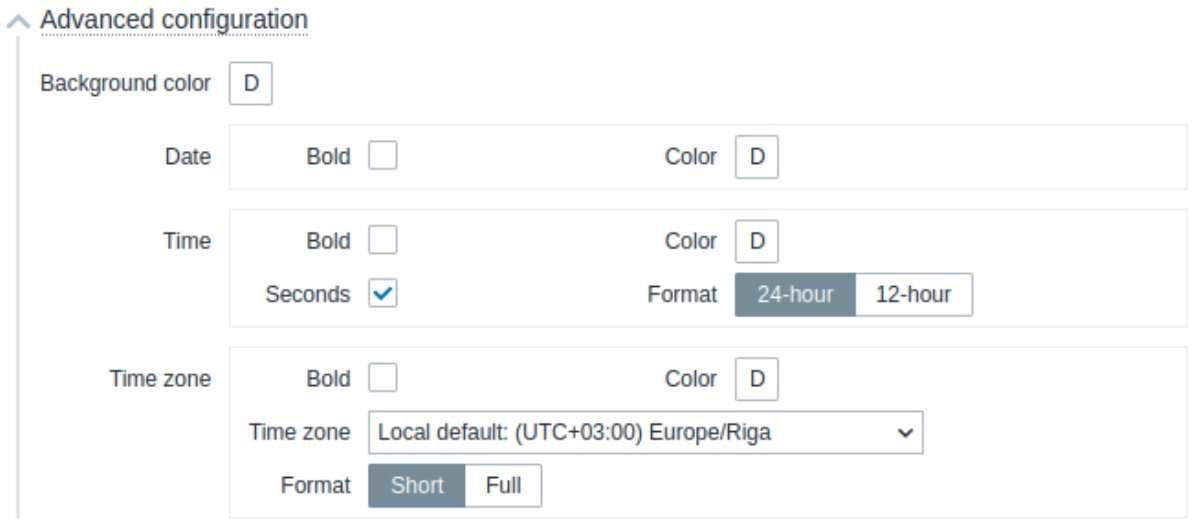
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Laika tips	Atlasiet laika avotu: **Lokāls** — sistēmas laiks ierīcē, kurā darbojas pārlūks; **Serveris** — laika josla, kas iestatīta globāli vai Zabbix lietotājam; **Hosts** — laiks, kas iegūts no hosta vienuma.
Vienums	Atlasiet vienumu hosta laikam (izmantojiet system.localtime[local] vienumu). Šis parametrs ir pieejams, ja *Laika tips* ir iestatīts uz ”Host time”.
Pulksteņa tips	Atlasiet pulksteņa tipu (analogs vai digitāls).
Rādīt	Atlasiet vienu vai vairākas informācijas vienības (datums, laiks, laika josla), ko attēlot digitālajā pulkstenī. Vienības izmērs tiek automātiski pielāgots attēlotajām vienībām, logrīka izmēram un attēlošanas mērogam. Šis parametrs ir pieejams, ja *Pulksteņa tips* ir iestatīts uz ”Digital”.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu digitālā pulksteņa paplašinātās konfigurācijas opcijas. Šī sadaļa ir pieejama, ja *Pulksteņa tips* ir iestatīts uz ”Digital”.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas sakļaujamajā sadaļā *Papildu konfigurācija*, un tikai tiem elementiem, kas ir atlasīti laukā *Rādīt* (skatīt iepriekš).

Papildus tam papildu konfigurācija ļauj mainīt visa logrīka fona krāsu.

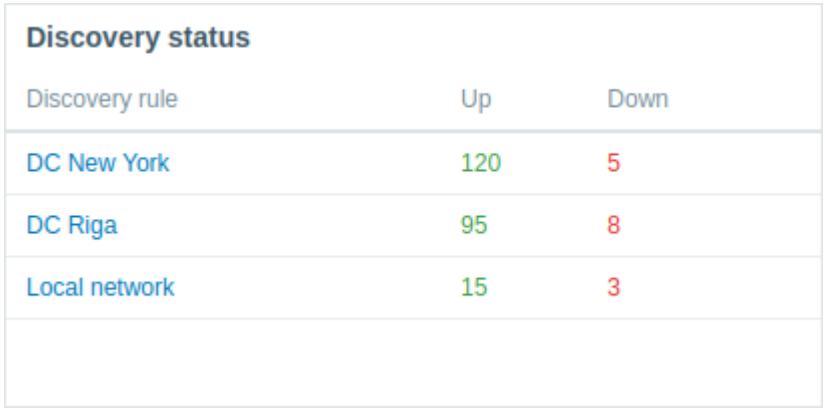


Fona krāsa	Atlasiet fona krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Datums	
Treknraksts	Atzīmējiet izvēles rūtiņu, lai datumu attēlotu treknrakstā.
Krāsa	Atlasiet datuma krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Laiks	
Treknraksts	Atzīmējiet izvēles rūtiņu, lai laiku attēlotu treknrakstā.
Krāsa	Atlasiet laika krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Sekundes	Atzīmējiet izvēles rūtiņu, lai attēlotu sekundes. Pretējā gadījumā tiks attēlotas tikai stundas un minūtes.
Formāts	Atlasiet, vai attēlot laiku 24 stundu vai 12 stundu formātā.
Laika josla	
Treknraksts	Atzīmējiet izvēles rūtiņu, lai laika joslu attēlotu treknrakstā.
Krāsa	Atlasiet laika joslas krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Laika josla	Atlasiet laika joslu.
Formāts	Atlasiet, vai attēlot laika joslu īsajā formātā (piemēram, `New York`) vai pilnajā formātā (piemēram, `(UTC-04:00) America/New York`).

3 Atklāšanas statuss

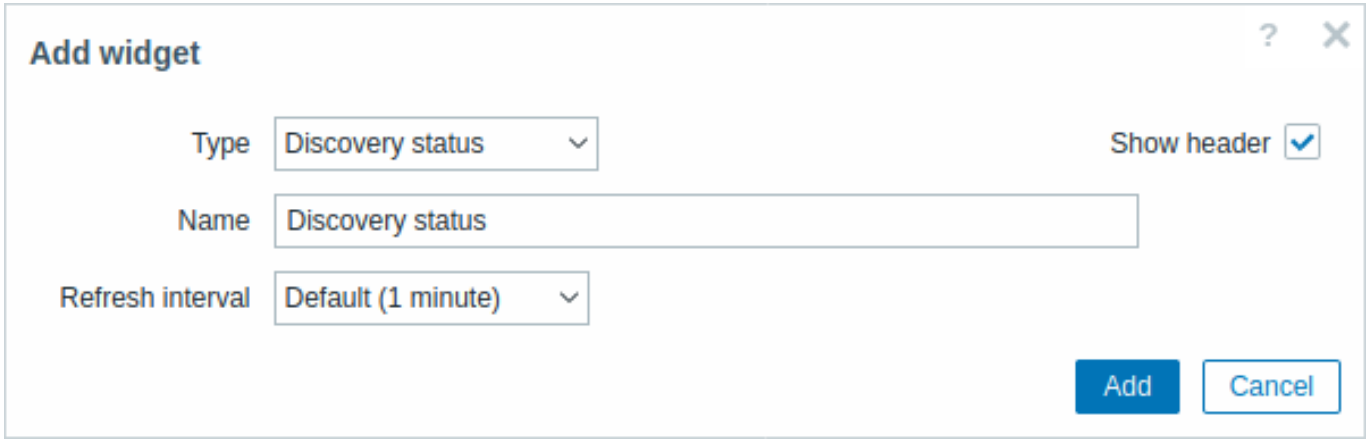
Pārskats

Logrīks *Atklāšanas statuss* parāda to ierīču statusu, kuras ir atklātas ar iespējotiem tīkla atklāšanas noteikumiem. Tas palīdz uzraudzīt, vai tīkla atklāšana darbojas pareizi, kā arī to, cik ierīču ir pieejamas vai nav pieejamas.



Konfigurācija

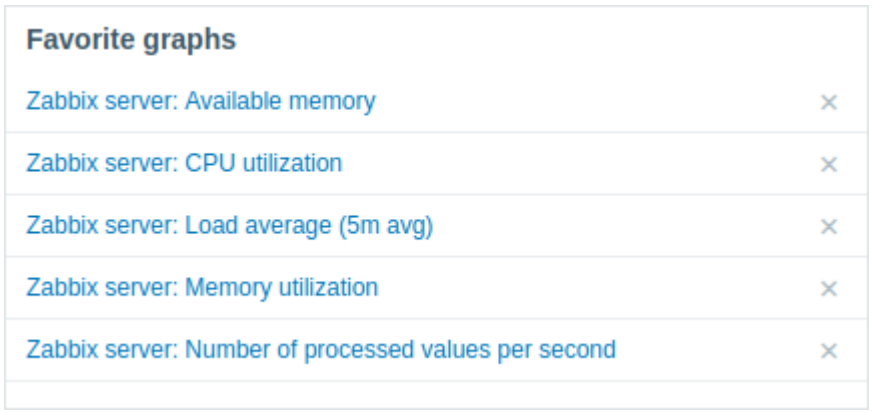
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



4 Iecienītākie grafiki

Pārskats

Logrīks *Iecienītākie grafiki* parāda pašreizējā lietotāja kā iecienītus atzīmēto grafiku saīšņu sarakstu alfabētiskā secībā.

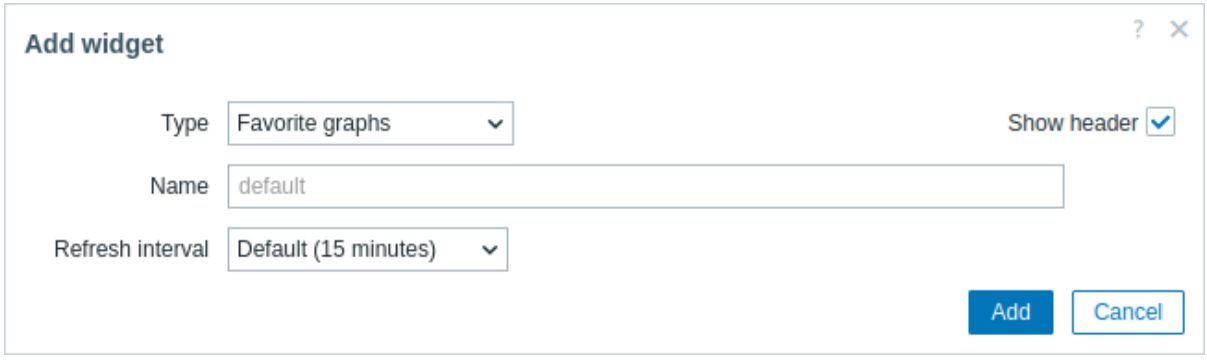


Noklikšķinot uz ✕, grafiks tiek noņemts no saraksta un no iecienītajiem.

Grafiki tiek atzīmēti kā iecienīti, noklikšķinot uz pogas *Pievienot iecienītajiem*, skatot vienuma grafikus sadaļā *Monitoring* > *Latest data*.

Konfigurācija

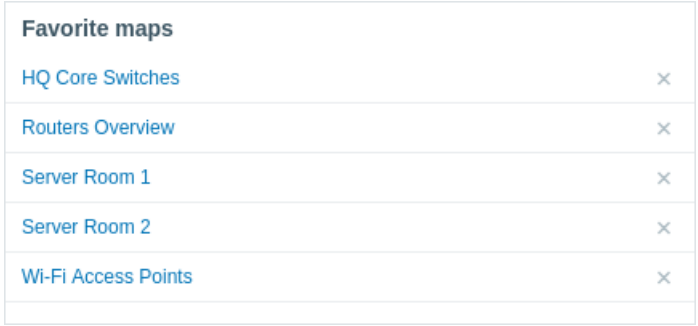
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



5 Iecienītākās kartes

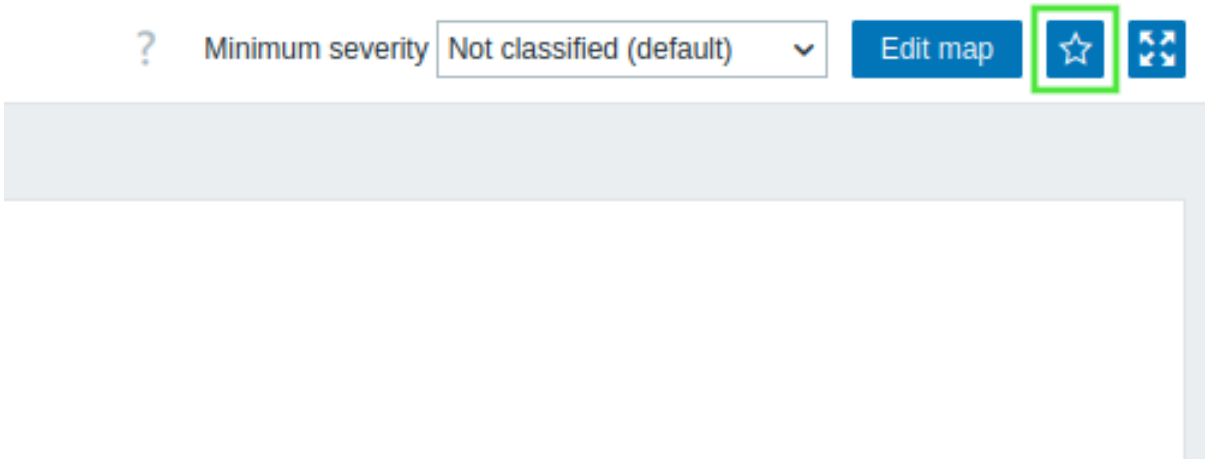
Pārskats

Logrīks *Iecienītās kartes* parāda pašreizējā lietotāja kā iecienītas atzīmēto karšu saīšņu sarakstu, kas sakārtots alfabētiskā secībā.



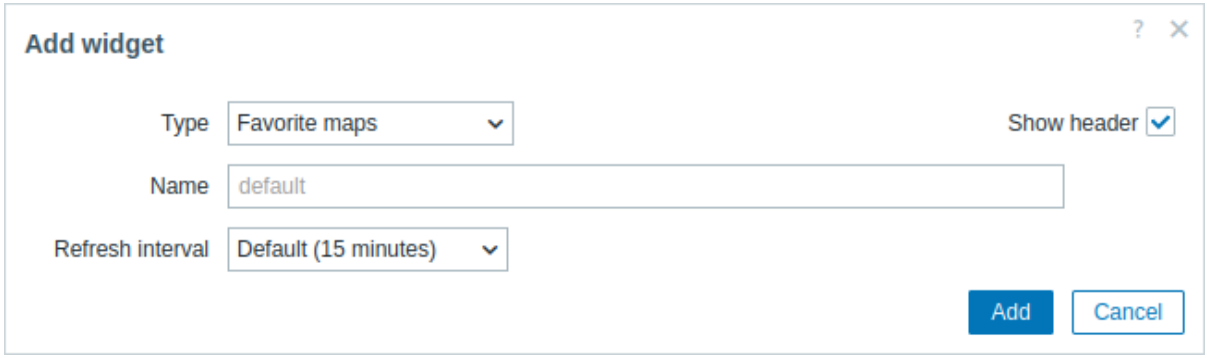
Noklikšķinot uz ✕, karte tiek noņemta no iecienītajām.

Kartes tiek atzīmētas kā iecienītas, noklikšķinot uz pogas *Pievienot iecienītajām*, skatot kartes sadaļā *Uzraudzība* > *Kartes*.



Konfigurācija

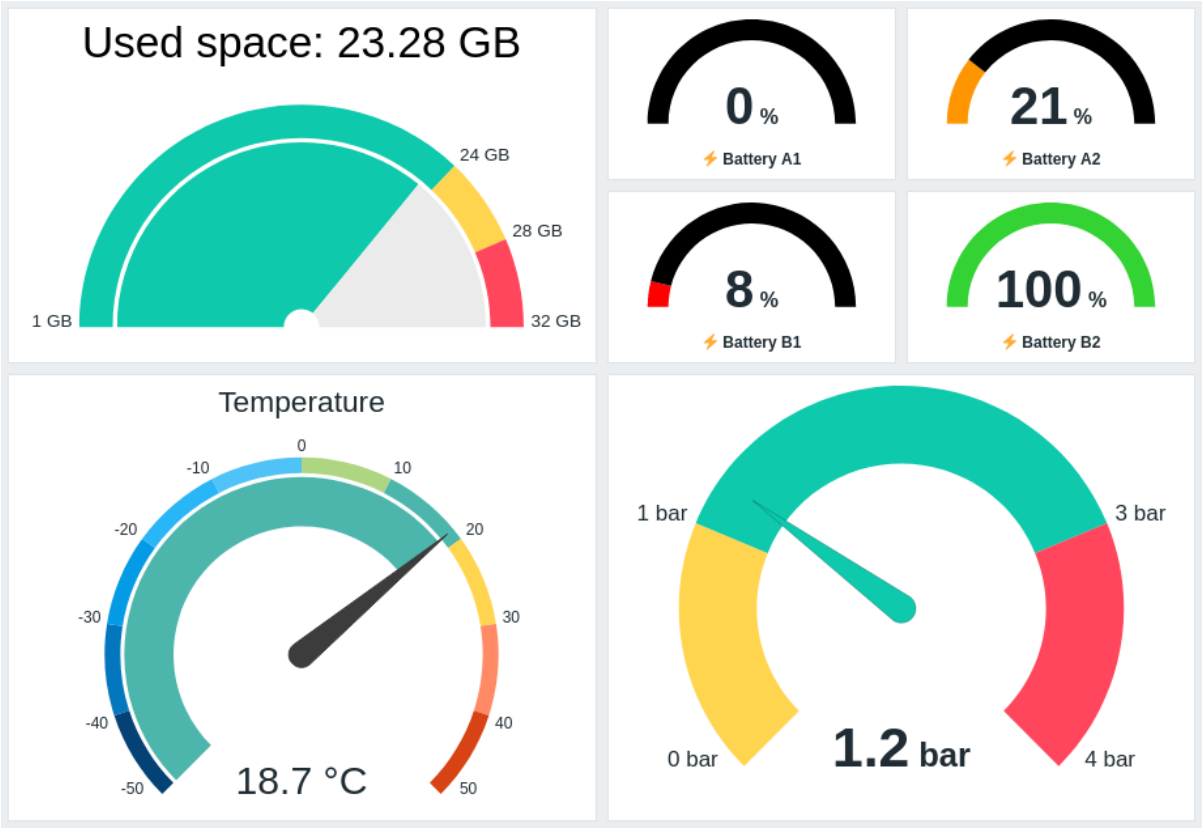
Visi šī logrīka konfigurācijas parametri ir kopīgi visiem logrīkiem.



6 Mērinstruments

Pārskats

Gauge logrīks attēlo viena skaitliska vienuma vērtību kā mēraparāta skalu. Tas ir noderīgs, lai sekotu svarīgiem rādītājiem, vizualizētu sliekšņus un noteiktu pēkšņas izmaiņas.



Varat konfigurēt logrīku, lai attēlotu tālāk norādīto:

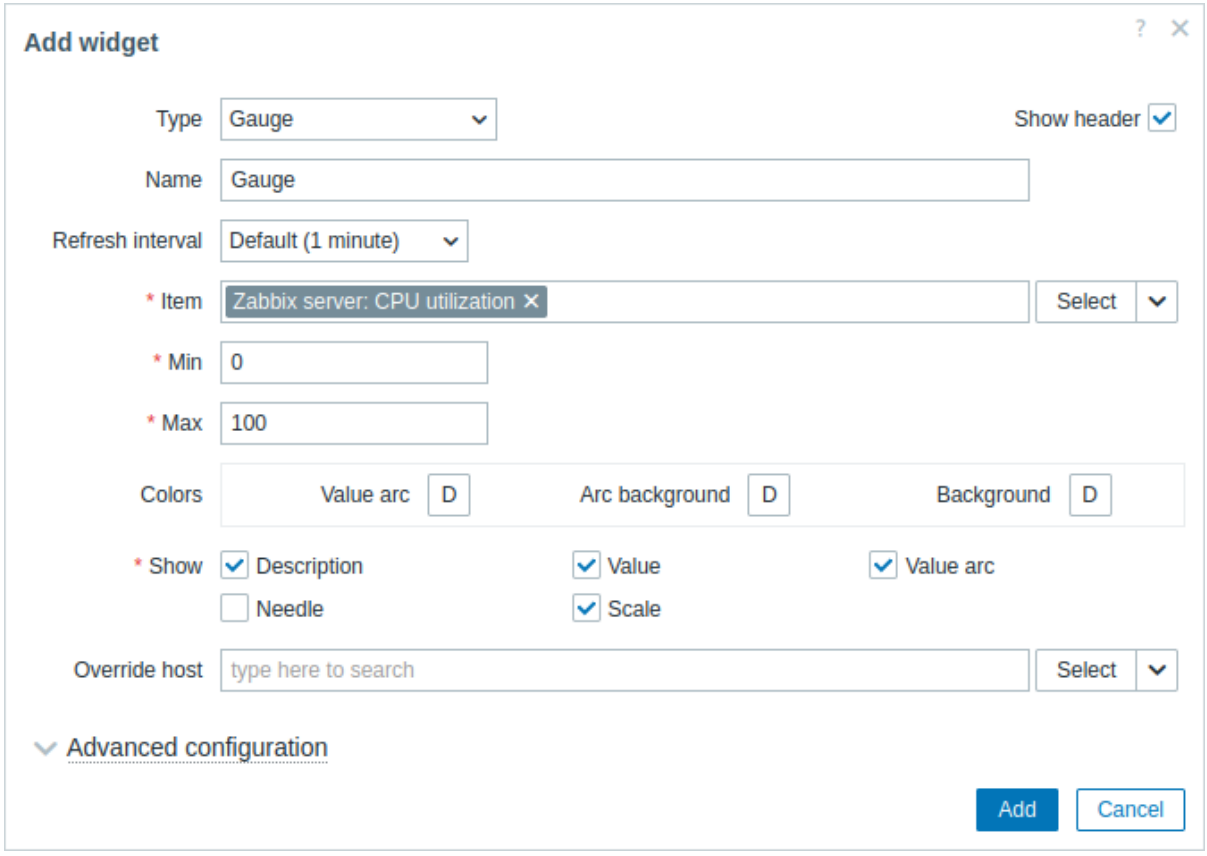
- Vienuma aprakstu (Izmantotā vieta: {ITEM.LASTVALUE}, Temperatūra)
- Vienuma vērtību un mērvienības (21 %, 18.7 °C, 1.2 bar)
- Mēraparāta skalu (1GB/24GB/28GB/32GB, -50/-40/-30/etc.)
- Mēraparāta loku (mēraparāta vērtības loku un mēraparāta sliekšņu loku)
- Mēraparāta rādītāju

Noklikšķinot uz logrīka, tiek atvērts vienuma vienkāršais grafiks.

Gauge logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Lejupielādēt attēlu*.

Konfigurācija

Lai konfigurētu, atlasiet *Gauge* kā tipu:

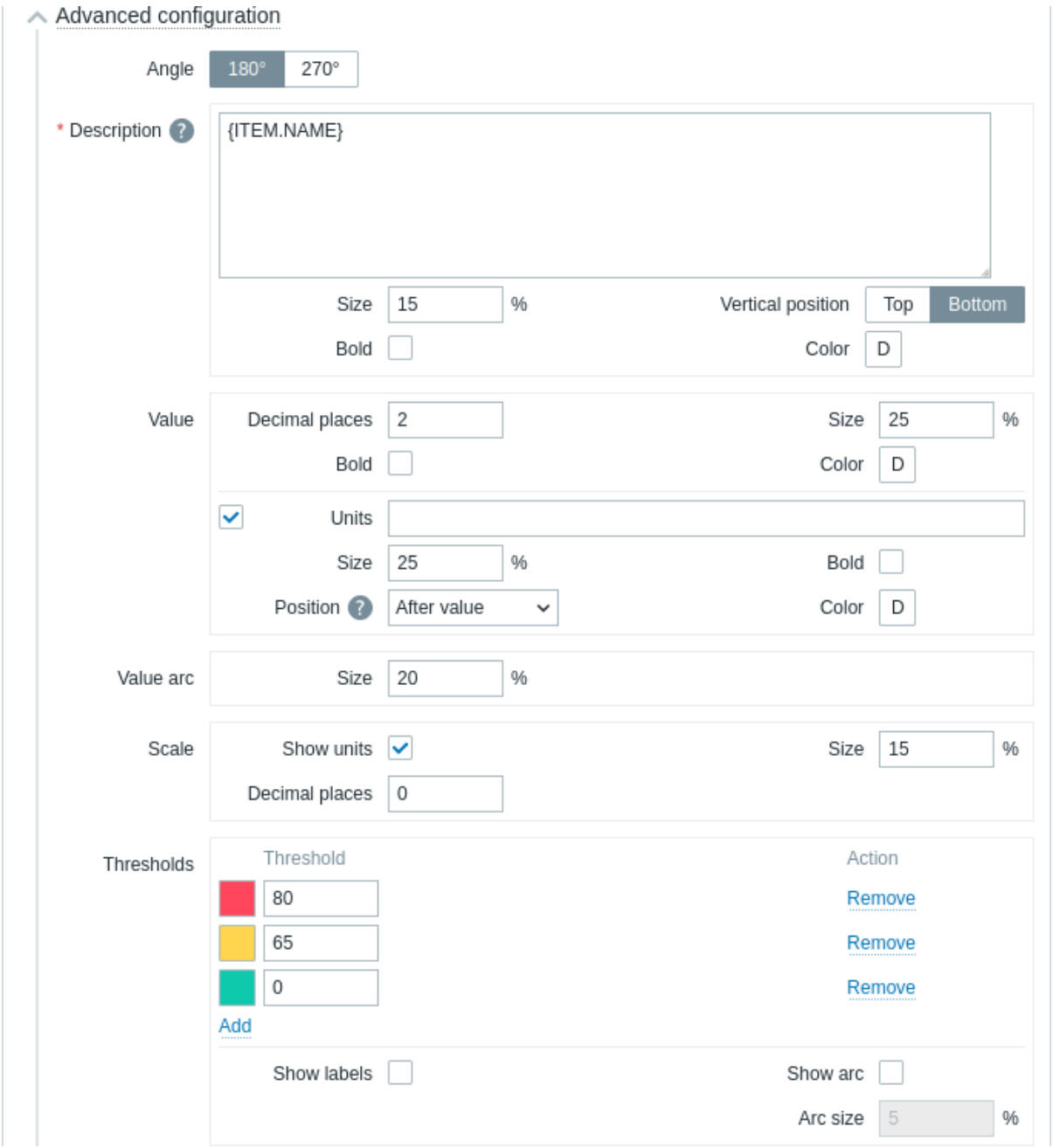


Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Item	Atlasiet vienumu. Varat arī atlasīt saderīgu logrīku kā datu avotu vienumiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt vienuma nosaukumu, tiks piedāvāts atbilstošu vienumu nolaižamais saraksts. Ņemiet vērā, ka varat atlasīt tikai tādus vienumus, kas atgriež skaitliskus datus (izņemot bināros datus).
Min	Ievadiet mērinstrumenta minimālo vērtību. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas arī vērtību kartēšanas.
Max	Ievadiet mērinstrumenta maksimālo vērtību. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas arī vērtību kartēšanas.
Colors	Atlasiet krāsu krāsu atlasītājā: **Value arc** — atlasiet mērinstrumenta vērtības loka krāsu; **Arc background** — atlasiet mērinstrumenta vērtības loka un mērinstrumenta sliekšņu loka fona krāsu; **Background** — atlasiet logrīka fona krāsu. ”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Ja ir iestatīti *Thresholds*, noklusējuma krāsa laukam **Value arc** ir atkarīga no sliekšņa krāsas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Show	Atzīmējiet izvēles rūtiņu, lai parādītu attiecīgo mērinstrumenta elementu — aprakstu, vērtību, vērtības loku, rādītāju, skalu (mērinstrumenta minimālo un maksimālo vērtību mērinstrumenta loka sākumā un beigās). Noņemiet atzīmi, lai paslēptu. Jābūt atlasītam vismaz vienam elementam. Ņemiet vērā, ka mērinstrumenta rādītāju un skalu var parādīt, ja ir parādīts mērinstrumenta vērtības loks vai mērinstrumenta sliekšņu loks (skatiet paplašinātās konfigurācijas opcijas). Ņemiet vērā arī to, ka, ja mērinstrumenta rādītājs ir parādīts, vērtība tiek novietota zem rādītāja; ja rādītājs ir paslēpts, vērtība tiek līdzināta ar mērinstrumenta loka apakšu.
Override host	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Advanced configuration	Noklikšķiniet uz etiķetes *Advanced configuration*, lai parādītu paplašinātās konfigurācijas opcijas. Šeit arī varēsiet pielāgot laukā *Show* atlasītos mērinstrumenta elementus.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas salokāmajā sadaļā *Papildu konfigurācija*:



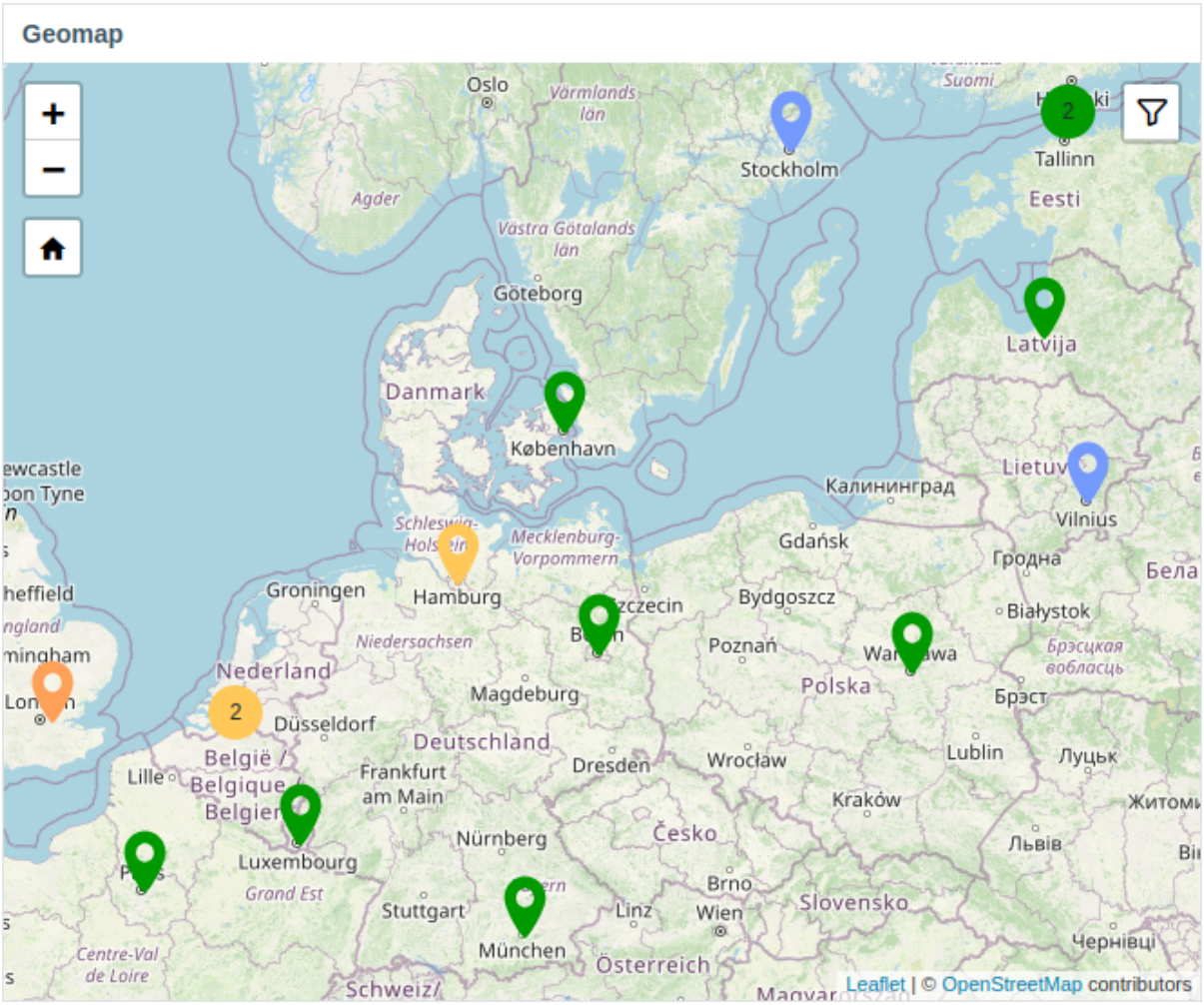
Leņķis	Izvēlieties mērierīces leņķi (180° vai 270°).
Apraksts	
Apraksts	Ievadiet vienuma aprakstu. Šis apraksts var aizstāt noklusējuma vienuma nosaukumu. Tiek atbalstīti vairākrindu apraksti. Ir iespējama teksta un atbalstīto makrosu kombinācija. Tiek atbalstīti {HOST.*}, {ITEM.*}, {INVENTORY.*} un lietotāja makrosi.
Izmērs	Ievadiet vienuma apraksta fonta izmēra augstumu (procentos attiecībā pret kopējo logrīka augstumu).
Vertikālā pozīcija	Izvēlieties vienuma apraksta vertikālo pozīciju (augšā vai apakšā attiecībā pret mērierīces loku).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai vienuma aprakstu attēlotu treknrakstā.
Krāsa	Izvēlieties vienuma apraksta krāsu krāsu atlasītājā. "D" apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Vērtība	
Decimālzīmes	Ievadiet ar vērtību attēlojamo zīmju skaitu aiz komata. Šī opcija ietekmē tikai vienumus, kas atgriež numeric (float) datus.
Izmērs	Ievadiet vērtības fonta izmēra augstumu (procentos attiecībā pret mērierīces loka augstumu).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai vērtību attēlotu treknrakstā.

Krāsa	Izvēlieties vērtības krāsu krāsu atlasītājā. ”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Mērvienības	
Mērvienības	Atzīmējiet izvēles rūtiņu, lai kopā ar vienuma vērtību attēlotu mērvienības. Ja ievadīsiet mērvienības nosaukumu, tas aizstās vienuma konfigurācijā iestatītās mērvienības.
Izmērs	Ievadiet vienuma mērvienību fonta izmēra augstumu (procentos attiecībā pret mērierīces loka augstumu).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai vienuma mērvienības attēlotu treknrakstā.
Pozīcija	Izvēlieties vienuma mērvienību pozīciju (virs, zem, pirms vai pēc attiecībā pret vienuma vērtību). Šī opcija tiek ignorēta šādām ar laiku saistītām mērvienībām: unixtime, uptime, s.
Krāsa	Izvēlieties vienuma mērvienību krāsu krāsu atlasītājā. ”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Vērtības loks	
Loka izmērs	Ievadiet mērierīces vērtības loka izmēra augstumu (procentos attiecībā pret mērierīces loka rādiusu).
Rādītājs	
Krāsa	Izvēlieties mērierīces rādītāja krāsu krāsu atlasītājā. ”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarnes tēmas. Ja ir iestatīti *Sliekšņi*, rādītāja noklusējuma krāsa ir atkarīga no sliekšņa krāsas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Skala	
Rādīt mērvienības	Atzīmējiet izvēles rūtiņu, lai kopā ar mērierīces minimālo un maksimālo vērtību attēlotu mērvienības.
Izmērs	Ievadiet mērierīces minimālās un maksimālās vērtības fonta izmēra augstumu (procentos attiecībā pret mērierīces loka augstumu).
Decimālzīmes	Ievadiet ar mērierīces minimālo un maksimālo vērtību attēlojamo zīmju skaitu aiz komata. Šī opcija ietekmē tikai vienumus, kas atgriež numeric (float) datus.
Sliekšņi	
Sliekšņi	Noklikšķiniet uz *Add*, lai pievienotu slieksni, izvēlieties sliekšņa krāsu krāsu atlasītājā un norādiet skaitlisku vērtību. Saglabājot sliekšņu saraksts tiks sakārtots augošā secībā. Ņemiet vērā, ka kā sliekšņi konfigurētās krāsas tiks attēlotas pareizi tikai skaitliskiem vienumiem. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas vērtību kartēšanas.
Rādīt etiķetes	Atzīmējiet izvēles rūtiņu, lai sliekšņu vērtības attēlotu kā etiķetes uz mērierīces skalas.
Rādīt loku	Atzīmējiet izvēles rūtiņu, lai attēlotu mērierīces sliekšņu loku.
Loka izmērs	Ievadiet mērierīces sliekšņu loka izmēra augstumu (procentos attiecībā pret mērierīces loka rādiusu).

7 Geomap

Pārskats

Geomap logrīks attēlo hostus kā marķierus interaktīvā uz Leaflet balstītā kartē, kur marķieru krāsas norāda katra hosta vissmagāko problēmu.

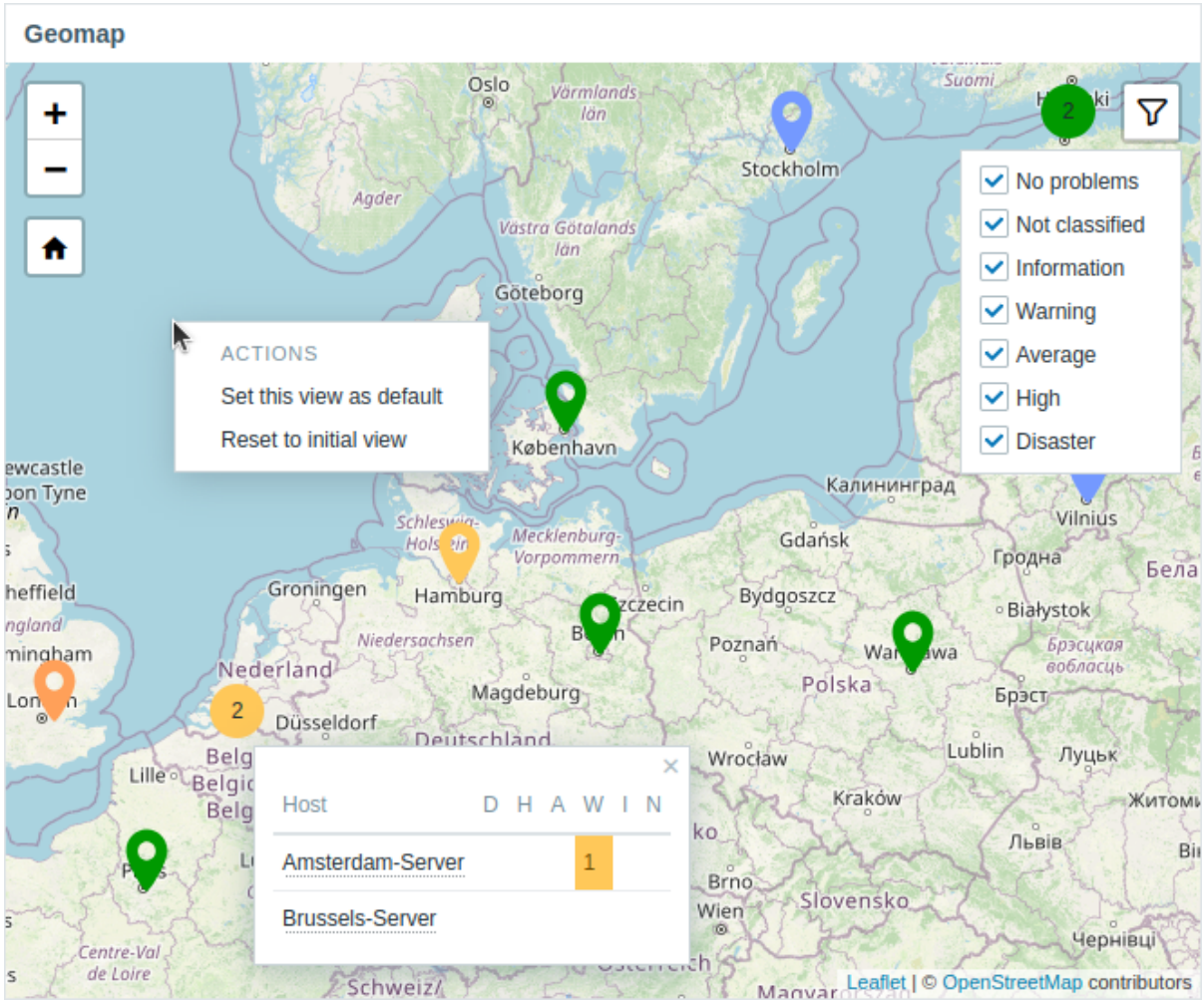


Note:
Zabbix piedāvā vairākus iepriekš definētus karšu flīžu nodrošinātājus un ļauj pievienot pielāgotus nodrošinātājus sadaļā *Administration* > *General* > *Geographical maps*.

Pēc noklusējuma logrīks attēlo visus iespējotos hostus, kuriem to inventarizācijā ir derīgas koordinātas (platums no -90 līdz 90, garums no -180 līdz 180).

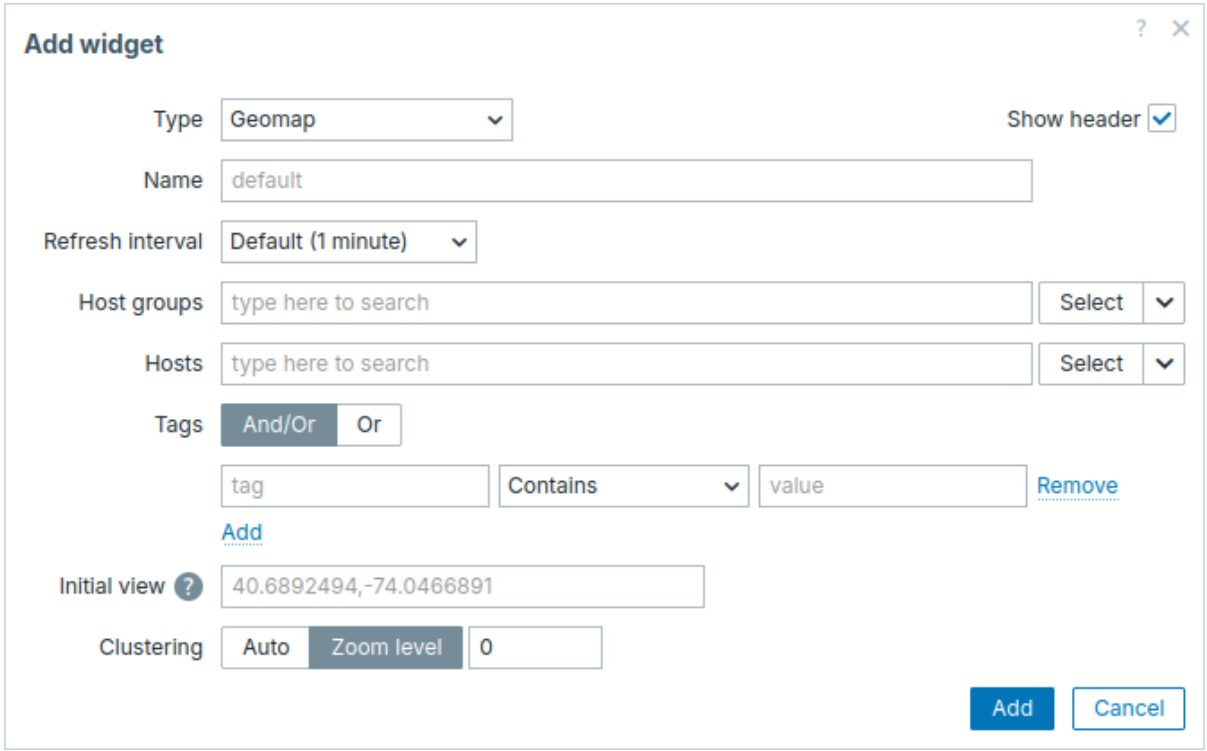
Izmantojiet tālāk norādītās opcijas, lai mijiedarbotos ar logrīku:

- Noklikšķinot uz marķiera, tiek parādīts hosta nosaukums un neatrisinātās problēmas; noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne.
- Filtrējiet hostus pēc problēmas smaguma, izmantojot filtra ikonu augšējā labajā stūrī.
- Ar peles labo pogu noklikšķiniet uz kartes, lai iestatītu vai atiestatītu noklusējuma skatu; izmantojiet sākuma ikonu, lai atgrieztos noklusējuma skatā.



Konfigurācija

Lai pievienotu logrīku, atlasiet *Geomap* kā tipu.



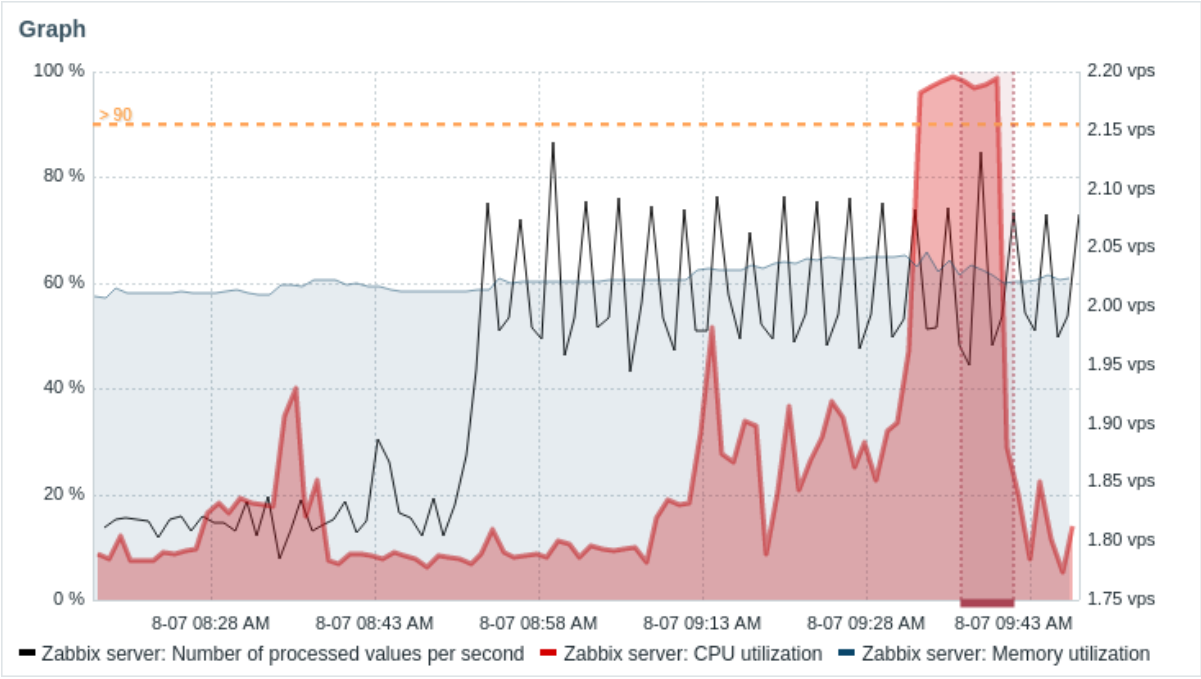
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas, kas tiks attēlotas kartē. Varat arī atlasīt saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Ja laukos *Hostu grupas* un *Hosti* nekas nav atlasīts, tiks attēloti visi hosti ar derīgām koordinātām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, kas tiks attēloti kartē. Varat arī atlasīt saderīgu logrīku vai paneli kā hostu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošo hostu nolaižamais saraksts. Ja laukos *Hostu grupas* un *Hosti* nekas nav atlasīts, tiks attēloti visi hosti ar derīgām koordinātām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto hostu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** - pietiek, ja izpildās viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Sākotnējais skats	Ar komatiem atdalītas centra koordinātas un neobligāts tālummaiņas līmenis, kas tiks attēlots, kad logrīks tiek sākotnēji ielādēts, formātā `<latitude>,<longitude>,<zoom>` Ja ir norādīts sākotnējais tālummaiņas līmenis, Geomap logrīks tiek ielādēts ar norādīto tālummaiņas līmeni. Pretējā gadījumā sākotnējais tālummaiņas līmenis tiek aprēķināts kā puse no konkrētā flīžu nodrošinātāja maksimālās tālummaiņas. Sākotnējais skats tiek ignorēts, ja ir iestatīts noklusējuma skats (skatiet tālāk). Piemēri: 40.6892494,-74.0466891 40.6892494,-122.0466891
Klasterizācija	Norādiet, kā tuvumā esošie hostu marķieri tiek apvienoti vienā marķierī ar skaitu: **Auto** - marķieri tiek apvienoti automātiski; **Tālummaiņas līmenis** - marķieri tiek apvienoti, pamatojoties uz norādīto tālummaiņas līmeni (0 - maksimāla tālināšana, 30 - maksimāla pietuvināšana). Piemēram, ja parametrs ir iestatīts uz 0, marķieri netiek apvienoti nevienā tālummaiņas līmenī; ja iestatīts uz 15, marķieri tiek apvienoti, sākot no maksimālas tālināšanas līdz 15. tālummaiņas līmenim; ja iestatīts uz 30, marķieri tiek apvienoti visos tālummaiņas līmeņos. Ņemiet vērā, ka maksimālo pietuvināšanas līmeni nosaka kartes flīžu nodrošinātājs.

8 Graph

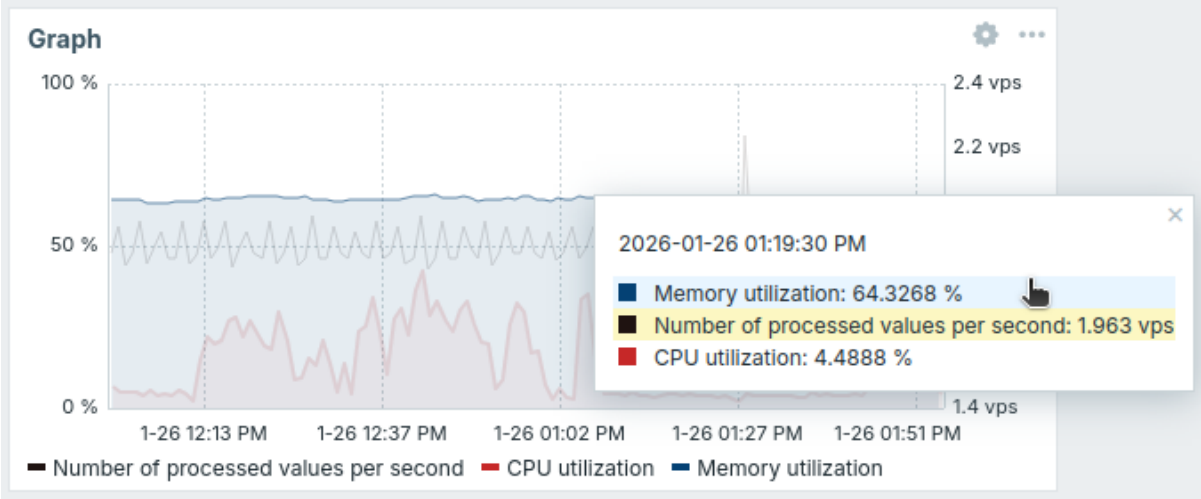
Pārskats

Graph logrīks attēlo skaitlisku vienumu datus kā vektorgrafiku. Tas var palīdzēt jums izsekot tendencēm, pamanīt problēmas un salīdzināt vērtības laika gaitā un starp hostiem.

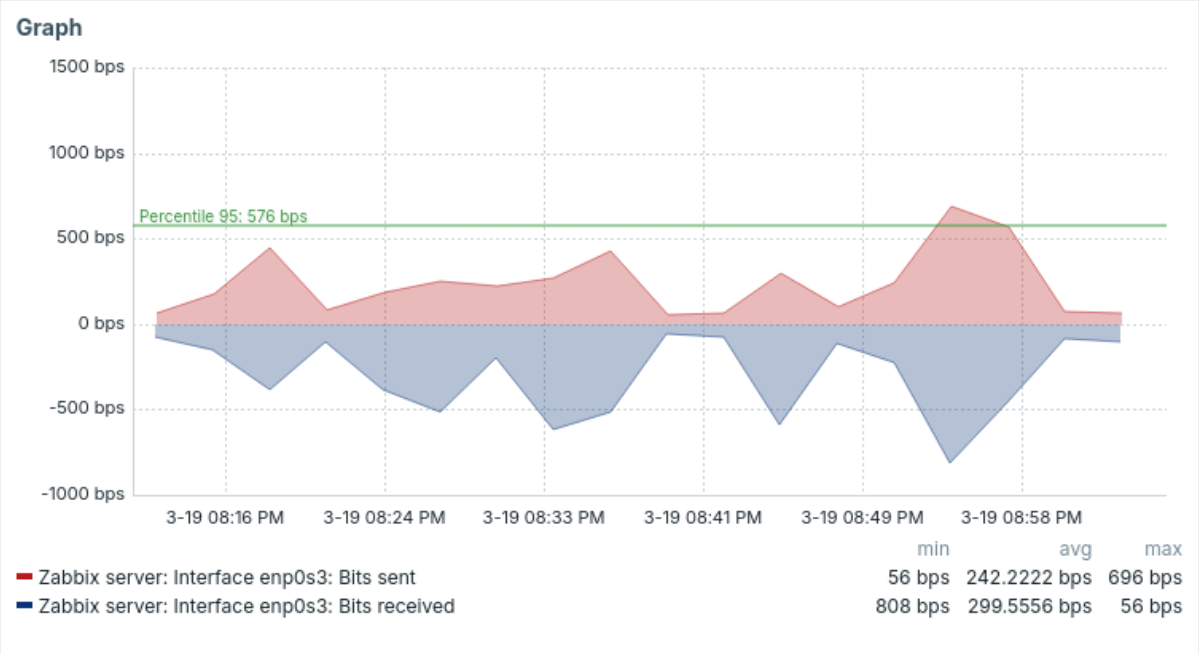


Novietojot kursoru virs grafika, tiek parādīts uznirstošais paskaidrojums ar vienumu sarakstu, kas sakārtots dilstošā secībā pēc vērtības.

Novietojot kursoru virs vienuma uznirstošajā paskaidrojumā, tiek izcelts tā grafiks (pārējie tiek aptumšoti); atlasot vienumu, tā dati tiek pārraidīti uz citiem logrīkiem.



Grafiku var attēlot ar invertētām Y ass vērtībām; tas ietekmē tikai vizuālo attēlojumu un nemaina sākotnējās vērtības.

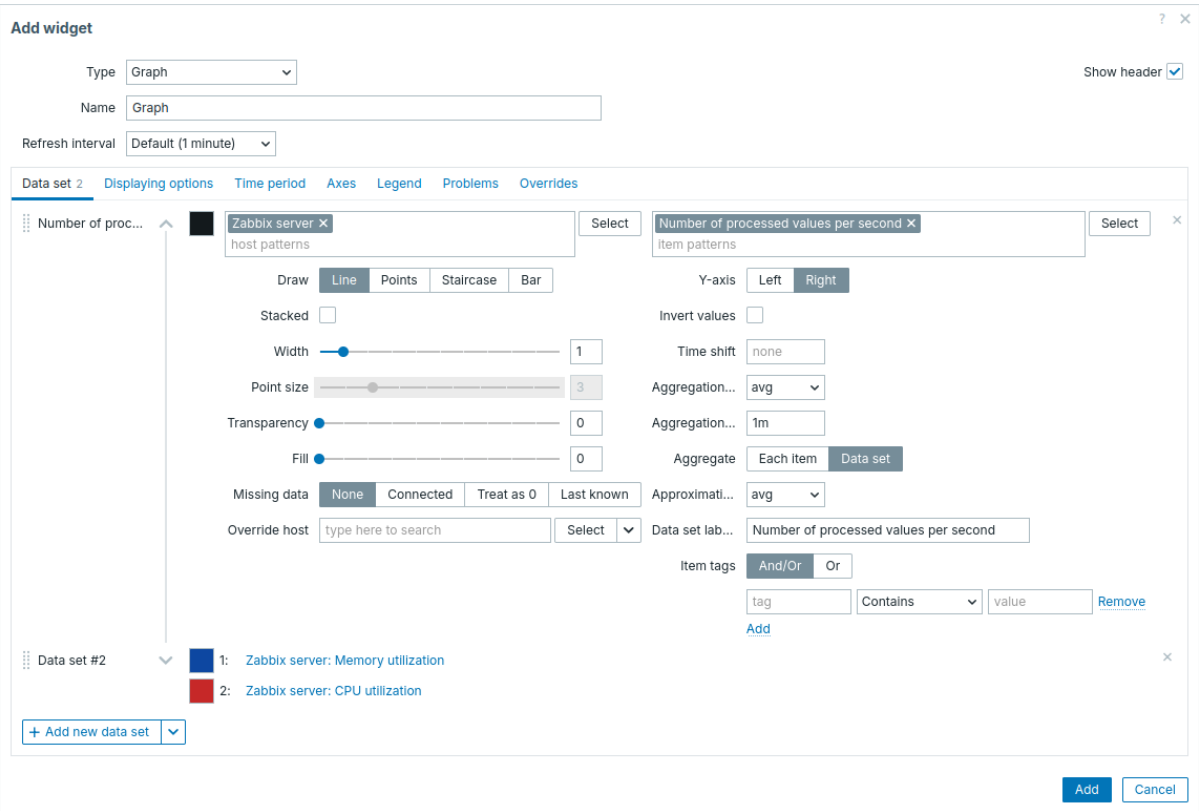


Ir pieejams arī mantotais Graph (classic) logrīks.

Graph logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Download image*.

Konfigurācija

Lai konfigurētu, atlasiet *Graph* kā tipu:



Datu kopa

Cilne **Datu kopa** ļauj atlasīt datus grafikam, pievienojot datu kopas. Var pievienot divu veidu datu kopas:

- *Vienumu šabloni* — tiek attēloti dati no atbilstošajiem vienumiem. Varat izvēlēties vienu pamatkrāsu vai atlasīt paletes rindu, lai katram atbilstošajam vienumam piešķirtu atšķirīgu krāsu.
- *Vienumu saraksts* — tiek attēloti dati no atlasītajiem vienumiem. Katra vienuma krāsu varat izvēlēties atsevišķi, izmantojot krāsu atlasītāju.

Pēc noklusējuma tiek pievienota *Vienumu šablonu* datu kopa.

Datu kopa	**Vienumu šablonu** datu kopai: Atlasiet vai ievadiet hosta un vienuma šablonus; grafikā tiks attēloti to vienumu dati, kas atbilst šiem šabloniem; var tikt attēloti līdz 50 vienumiem. Atlasei var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item**, ja ir citi atbilstoši vienumi (piemēram, *item2*, *item3*). Hosta un vienuma šablonu norādīšana ir obligāta ”Vienumu šablonu” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksta** datu kopai: Atlasiet vienumus grafikam, noklikšķinot uz pogas *Pievienot vienumu*. Varat arī atlasīt saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*. Vienumu vai logrīku norādīšana ir obligāta ”Vienumu saraksta” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitlisku tipu vienumi. Konfigurējot logrīku veidnes panelī, parametrs hosta šablonu norādīšanai nav pieejams, un parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.
Zīmēšana	Izvēlieties metrikas attēlošanas veidu. Iespējamie attēlošanas veidi: *Līnija* (iestatīts pēc noklusējuma), *Punkti*, *Kāpnes* un *Josla*. Ņemiet vērā, ka, ja līnijas/kāpņu grafikā ir tikai viens datu punkts, tas tiek attēlots kā punkts neatkarīgi no attēlošanas veida. Punkta izmērs tiek aprēķināts no līnijas platuma, taču tas nevar būt mazāks par 3 pikseļiem, pat ja līnijas platums ir mazāks.
Sakrauts	Atzīmējiet izvēles rūtiņu, lai attēlotu datus sakrauti (tiek rādīti aizpildīti apgabali). Šī opcija ir atspējota, ja ir atlasīts attēlošanas veids *Punkti*.
Platums	Iestatiet līnijas platumu. Šī opcija ir pieejama, ja ir atlasīts attēlošanas veids *Līnija* vai *Kāpnes*.
Punkta izmērs	Iestatiet punkta izmēru. Šī opcija ir pieejama, ja ir atlasīts attēlošanas veids *Punkti*.
Caurspīdīgums	Iestatiet caurspīdīguma līmeni.
Aizpildījums	Iestatiet aizpildījuma līmeni. Šī opcija ir pieejama, ja ir atlasīts attēlošanas veids *Līnija* vai *Kāpnes*.
Trūkstošie dati	Atlasiet trūkstošo datu attēlošanas opciju: **Nav** — atstarpe tiek atstāta tukša; **Savienots** — abas robežvērtības tiek savienotas; **Uzskatīt par 0** — trūkstošie dati tiek attēloti kā 0 vērtības; **Pēdējā zināmā vērtība** — trūkstošie dati tiek attēloti ar tādu pašu vērtību kā pēdējā zināmā vērtība; nav piemērojams attēlošanas veidam *Punkti* un *Josla*.
Aizstāt hostu	Atlasiet saderīgu logrīku vai paneļa hosta atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Y ass	Atlasiet grafika pusi, kurā tiks attēlota Y ass.
Invertēt vērtības	Atzīmējiet izvēles rūtiņu, lai pirms attēlošanas grafikā visas Y ass vērtības reizinātu ar -1. Tas ietekmē tikai vizuālo attēlojumu; sākotnējie dati paliek nemainīti. Šī opcija neietekmē uznirstošo paskaidri, leģendu vai procentiles līniju. Ja asij ir atlasīta opcija *Vienkāršie trigeri*, trigeru sliekšņi tiks attēloti invertēti.
Laika nobīde	Ja nepieciešams, norādiet laika nobīdi. Šajā laukā var izmantot laika sufiksus. Ir atļautas negatīvas vērtības.

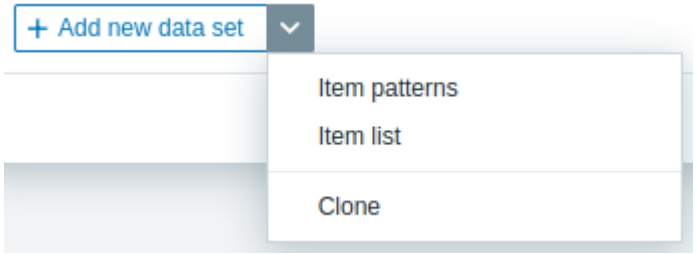
Agregācijas funkcija	Norādiet, kuru agregācijas funkciju izmantot katram vienumam datu kopā atlasītajā *Laika periodā*: **min** — attēlot mazāko vērtību; **max** — attēlot lielāko vērtību; **avg** — attēlot vidējo vērtību; **sum** — attēlot vērtību summu; **count** — attēlot vērtību skaitu; **first** — attēlot pirmo vērtību; **last** — attēlot pēdējo vērtību. Ja ir atlasīts **not used**, tiek attēlotas visas vērtības (bez agregācijas). Agregācija ļauj attēlot agregētu vērtību izvēlētajam intervālam (5 minūtes, stunda, diena), nevis visas vērtības. *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Opcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem. Skatiet arī: Agregācija grafikos.
Agregācijas intervāls	Norādiet intervālu vērtību agregēšanai. Šajā laukā var izmantot laika sufiksus. Skaitliska vērtība bez sufiksa tiks uzskatīta par sekundēm. Ņemiet vērā: ja logrīks ir konfigurēts vēsturisko datu attēlošanai, pamatojoties uz trendiem (*Vēsturisko datu atlase* ir iestatīta uz *Trendi* vai *Auto*), ieteicams izmantot agregācijas intervālu, kas ir 1 stundas reizinājums (piemēram, 3600, 60m, 1h, 3h utt.). Trendi glabā pa stundām agregētas vērtības, tāpēc agregācijas intervāla izmantošana, kas nav 1 stundas reizinājums (piemēram, 100s, 7min, 15min, 90min utt.), var radīt grūti interpretējamus rezultātus.
Agregēt	Norādiet, ko agregēt: **Katru vienumu** — katrs vienums datu kopā tiks agregēts un attēlots atsevišķi; **Datu kopu** — visi datu kopas vienumi tiks agregēti un attēloti kā viena vērtība.
Aproksimācija	Norādiet, kuru vērtību attēlot, ja uz vienu vertikālo grafika pikseli attiecas vairāk nekā viena vērtība: **all** — attēlot mazāko, lielāko un vidējo vērtību; **min** — attēlot mazāko vērtību; **max** — attēlot lielāko vērtību; **avg** — attēlot vidējo vērtību. Šis iestatījums ir noderīgs, attēlojot grafiku lielam laika periodam ar biežu atjaunināšanas intervālu (piemēram, vienam gadam ar vērtībām, kas ievāktas ik pēc 10 minūtēm).
Datu kopas etiķete	Norādiet datu kopas etiķeti, kas tiek attēlota grafika *Datu kopas* konfigurācijā un grafika *Leģendā* (agregētām datu kopām). Visas datu kopas tiek numurētas, ieskaitot tās, kurām ir norādīta *Datu kopas etiķete*. Ja etiķete nav norādīta, datu kopa tiks automātiski marķēta atbilstoši tās numuram (piemēram, ”Datu kopa #2”, ”Datu kopa #3” utt.). Datu kopu numerācija tiek pārrēķināta pēc datu kopu pārkārtošanas/vilkšanas. Pārāk garas datu kopu etiķetes tiks saīsinātas, lai ietilptu attēlošanas vietā (piemēram, ”Procesu skaits...”).

Vienumu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos vienumus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **And/Or** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Or; **Or** — pietiek, ja ir izpildīts viens nosacījums.

Datu kopas konfigurācijas informācija

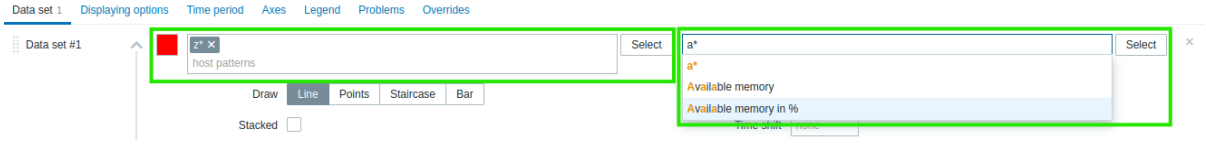
Esošās datu kopas tiek parādītas sarakstā. Jūs varat:

- Noklikšķināt uz pārvietošanas ikonas un vilkt datu kopu uz jaunu vietu sarakstā.
- Noklikšķināt uz izvēršanas ikonas, lai izvērstu datu kopas informāciju. Kad tā ir izvērsta, šī ikona pārvēršas par sakļaušanas ikonu.
- Noklikšķināt uz krāsas ikonas, lai atvērtu atlasītāju. Jūs varat ievadīt heksadecimālo kodu, izvēlēties vienkrāsainu paraugu vai pārslēgties uz cilni *Palette* un atlasīt iepriekš definētu krāsu rindu. Izvēlētā krāsa tiek piemērota tieši *Item list* datu kopām vai izmantota kā pamats ģenerētajiem toņiem sadaļā *Item patterns*. Izmantojiet Tab, lai pārvietotos starp dialoga vadīklām, bulttaustiņus, lai pārvietotos starp paraugiem vai paletes rindām, Enter, lai atlasītu, un Esc, lai atceltu.
- Noklikšķināt uz pogas *Add new data set*, lai pievienotu tukšu datu kopu, kas ļauj atlasīt hostu un vienumu šablonus. Ja noklikšķināt uz lejupvērstās ikonas blakus pogai *Add new data set*, tiek parādīta nolaižamā izvēlne, kas ļauj pievienot jaunu *Item patterns* vai *Item list* datu kopu vai *Clone* pašlaik atvērto datu kopu. Ja visas datu kopas ir sakļautas, opcija *Clone* nav pieejama.

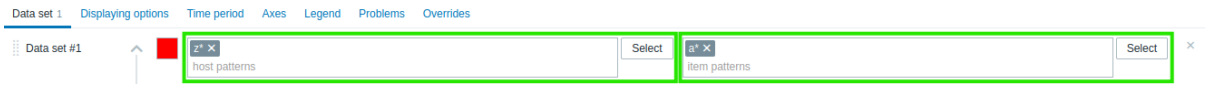


Datu kopā **Vienumu šabloni** ir lauki *Hostu šabloni* un *Vienumu šabloni*, kas abi atpazīst pilnus nosaukumus vai šablonus, kuros ir aizstājējzīme (*). Šī funkcionalitāte ļauj atlasīt visus hostu nosaukumus un vienumu nosaukumus, kas satur izvēlēto šablonu. Rakstot vienuma nosaukumu vai vienuma šablonu laukā *Vienumu šabloni*, nolaižamajā sarakstā tiek parādīti tikai tie vienumi, kas pieder atlasītā(-o) hosta(-u) nosaukumam(-iem).

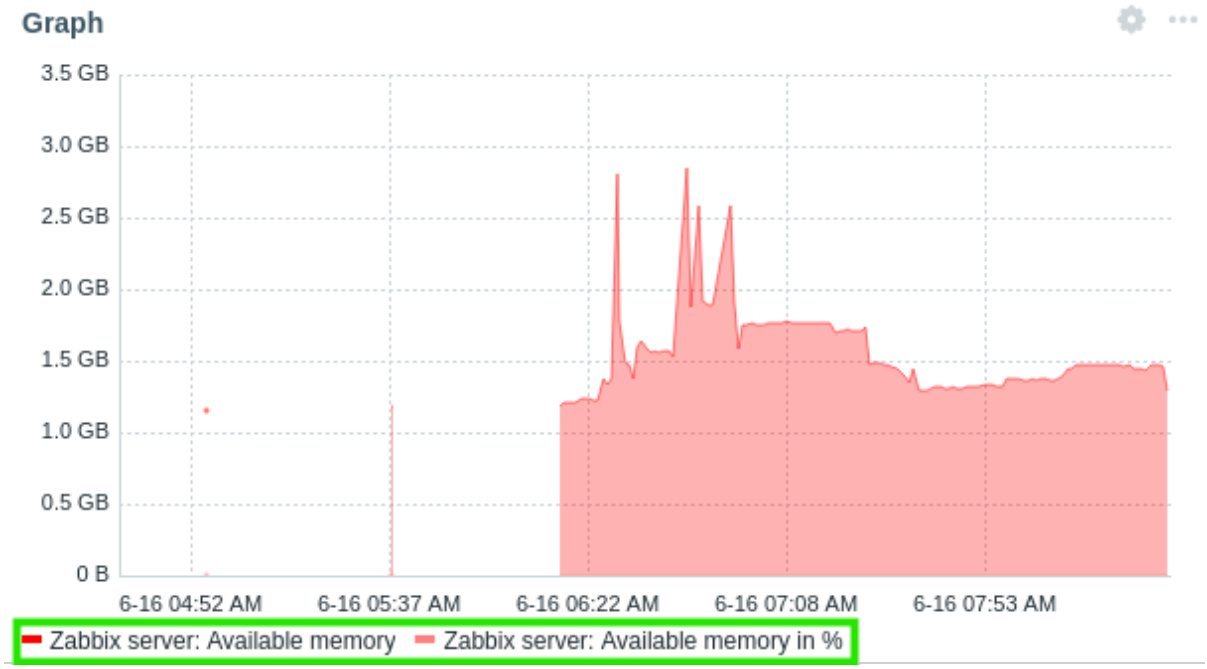
Piemēram, ievadot šablonu **z*** laukā *Hostu šabloni*, nolaižamajā sarakstā tiek parādīti visi hostu nosaukumi, kas satur šo šablonu: **z*, Zabbix serveris, Zabbix starpniekserveris**. Pēc *Enter* nospiešanas šis šablons tiek pieņemts un parādīts kā **z***. Līdzīgi, ievadot šablonu **a*** laukā *Vienumu šabloni*, nolaižamajā sarakstā tiek parādīti visi vienumu nosaukumi, kas satur šo šablonu: **a*, Pieejamā atmiņa, Pieejamā atmiņa %**.



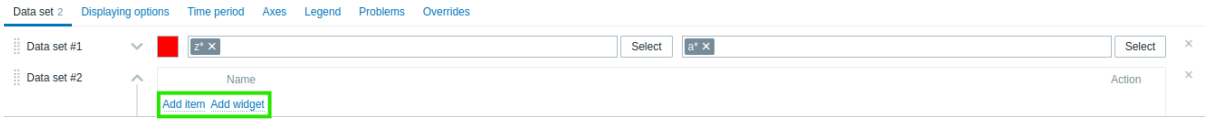
Pēc *Enter* nospiešanas šablons tiek pieņemts un parādīts kā **a***.



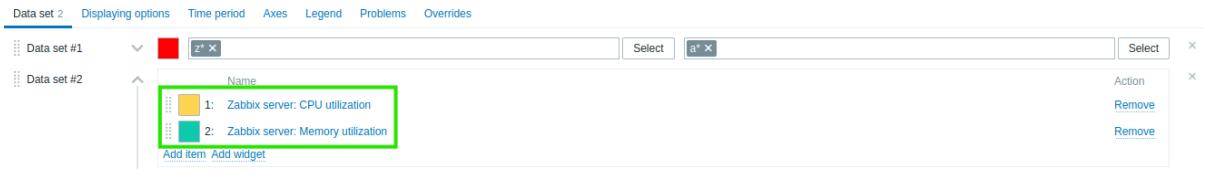
Pēc tam grafikā tiek parādīti visi vienumi, kas pieder atlasītā(-o) hosta(-u) nosaukumam(-iem).



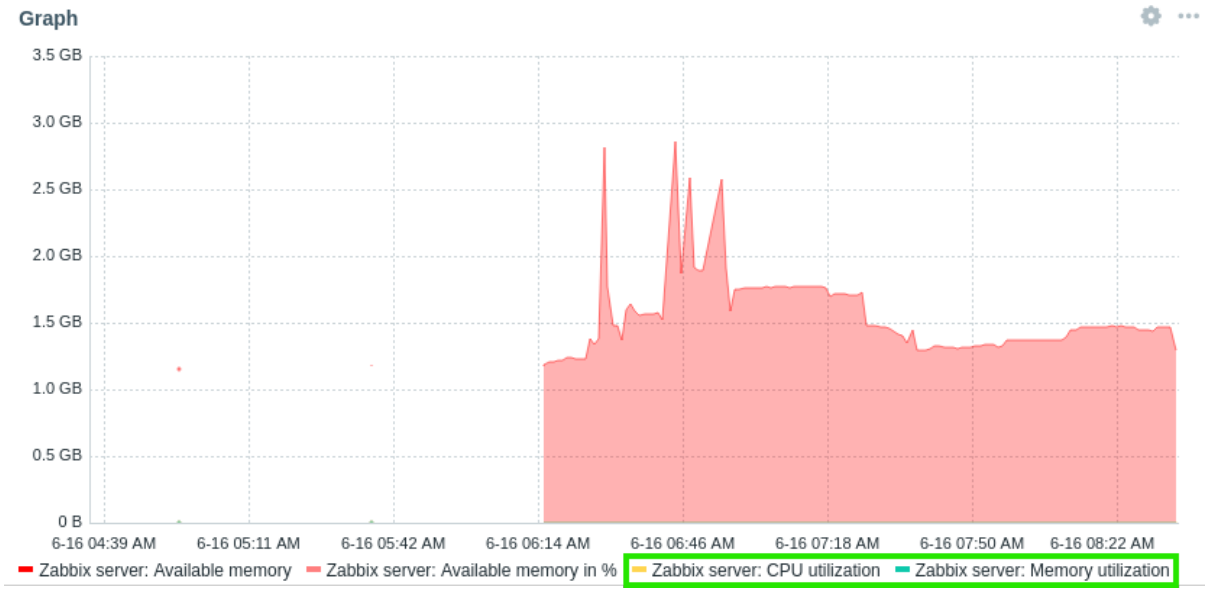
Datu kopa **Vienumu saraksts** satur pogu *Pievienot vienumu*, kas ļauj pievienot vienumus attēlošanai grafikā. Varat arī pievienot saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*.



Piemēram, noklikšķinot uz pogas *Pievienot vienumu*, tiek atvērts uznirstošais logs ar parametru *Hosts*. Pēc hosta atlasīšanas sarakstā tiek parādīti visi tā vienumi, kas ir pieejami atlasei.

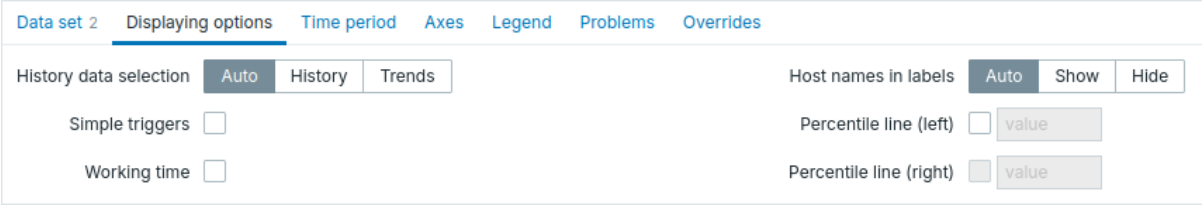


Pēc viena vai vairāku vienumu atlasīšanas tie tiks parādīti datu kopas vienumu sarakstā un grafikā.



Attēlošanas opcijas

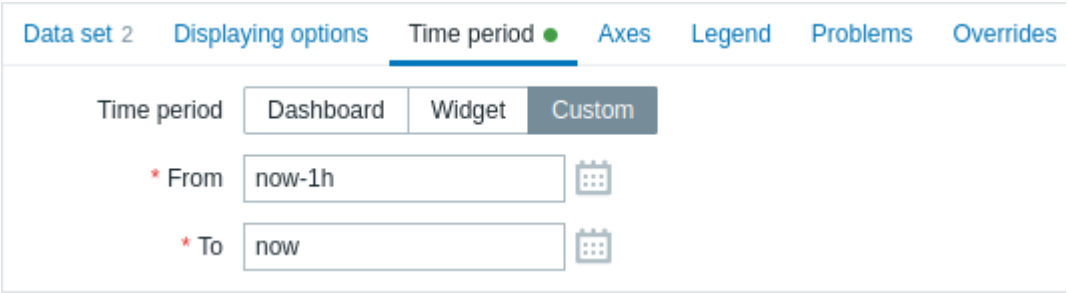
Cilne **Attēlošanas opcijas** ļauj definēt vēstures datu atlasi:



Vēstures datu atlase	Iestatiet grafika datu avotu: **Auto** - dati tiek iegūti saskaņā ar klasiskā grafika algoritmu (noklusējums); **History** - dati no vēstures; **Trends** - dati no tendenču datiem.
Vienkāršie trigeri	Atzīmējiet izvēles rūtiņu, lai parādītu trigeru sliekšņus vienkāršajiem trigeriem. Sliekšņi tiks attēloti kā pārtrauktas līnijas, izmantojot trigera svarīguma krāsu. Vienkāršs trigeris ir trigeris ar vienu funkciju (tikai `last`, `max`, `min`, `avg`) vienam vienumam izteiksmē. Var attēlot ne vairāk kā trīs trigerus. Ņemiet vērā, ka trigerim jāatrodas attēlotajā diapazonā, lai tas būtu redzams.
Darba laiks	Atzīmējiet izvēles rūtiņu, lai grafikā parādītu darba laiku. Darba laiks (darba dienas) grafikos tiek attēlots ar baltu fonu, savukārt ārpus darba laiks tiek attēlots pelēkā krāsā (ar noklusējuma lietotāja saskarnes tēmu *Original blue*).
Hostu nosaukumi etiķetēs	Izvēlieties, vai grafika leģendā un uznirstošajā paskaidrojumā, kas parādās, novietojot kursoru virs grafika datiem, attēlot hostu nosaukumus: **Auto** - hostu nosaukumi tiek attēloti tikai tad, ja datu kopās ir vairāk nekā viens hosts (noklusējums). **Show** - hostu nosaukumi tiek attēloti. **Hide** - hostu nosaukumi tiek paslēpti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Percentiles līnija (kreisā)	Atzīmējiet izvēles rūtiņu un ievadiet percentiles vērtību, lai norādīto percentili parādītu kā līniju grafika kreisajā Y asī. Ja, piemēram, ir iestatīta 95% percentile, tad percentiles līnija atradīsies līmenī, zem kura ietilpst 95 procenti vērtību.
Percentiles līnija (labā)	Atzīmējiet izvēles rūtiņu un ievadiet percentiles vērtību, lai norādīto percentili parādītu kā līniju grafika labajā Y asī. Ja, piemēram, ir iestatīta 95% percentile, tad percentiles līnija atradīsies līmenī, zem kura ietilpst 95 procenti vērtību.

Laika periods

Cilne **Laika periods** ļauj iestatīt laika periodu, par kuru attēlot datus grafikā:

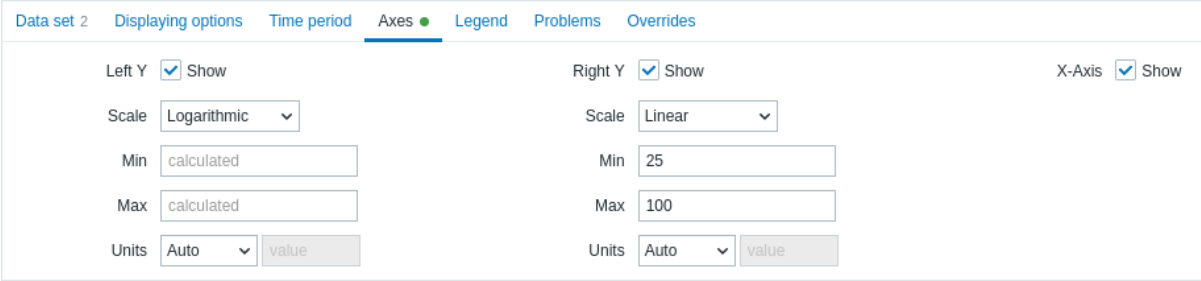


Laika periods	Atlasiet datu avotu laika periodam: **Panelis** - izmantot paneļa laika perioda atlasītāju; **Logrīks** - izmantot saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks attēlota pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki joprojām var to izmantot kā datu avotu laika periodam.

Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā datu avotu laika periodam. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Logrīks".
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".

Asis

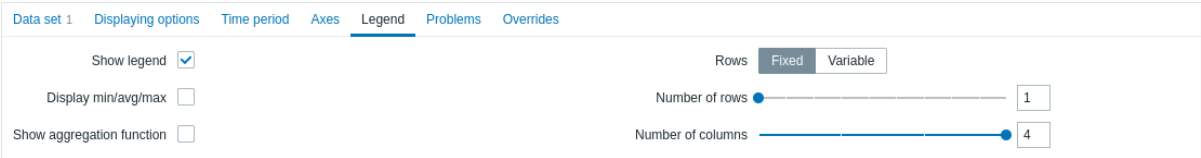
Cilne **Asis** ļauj pielāgot, kā tiek attēlotas asis:



Kreisā Y	Atzīmējiet šo izvēles rūtiņu, lai kreisā Y ass būtu redzama. Izvēles rūtiņa var būt atspējota, ja tā nav atlasīta cilnē *Datu kopa* vai *Pārrakstīšana*.
Labā Y	Atzīmējiet šo izvēles rūtiņu, lai labā Y ass būtu redzama. Izvēles rūtiņa var būt atspējota, ja tā nav atlasīta cilnē *Datu kopa* vai *Pārrakstīšana*.
X ass	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu X asi (pēc noklusējuma atzīmēta).
Mērogs	Izvēlieties grafika ass vērtību mērogu nolaižamajā sarakstā: **Lineārs** — ass vērtības palielinās par fiksētu lielumu (piemēram, 10, 20, 30), piemērots datiem, kas mainās vienmērīgi vai aptver nelielu līdz vidēju diapazonu; **Logaritmisks** — ass vērtības palielinās eksponenciāli (piemēram, 10, 100, 1000), piemērots datiem, kas mainās strauji vai aptver lielu diapazonu.
Min	Iestatiet atbilstošās ass minimālo vērtību. Tiek norādīta Y ass redzamā diapazona minimālā vērtība.
Max	Iestatiet atbilstošās ass maksimālo vērtību. Tiek norādīta Y ass redzamā diapazona maksimālā vērtība.
Vienības	Izvēlieties grafika ass vērtību vienību nolaižamajā sarakstā: **Auto** — ass vērtības tiek attēlotas, izmantojot pirmā vienuma vienību datu kopā; **Static** — ass vērtības tiek attēlotas, izmantojot ievades laukā *value* norādīto vienību; ja lauks ir atstāts tukšs, tiek attēlotas tikai skaitliskās vērtības.

Leģenda

Cilne **Leģenda** ļauj pielāgot grafika leģendu:

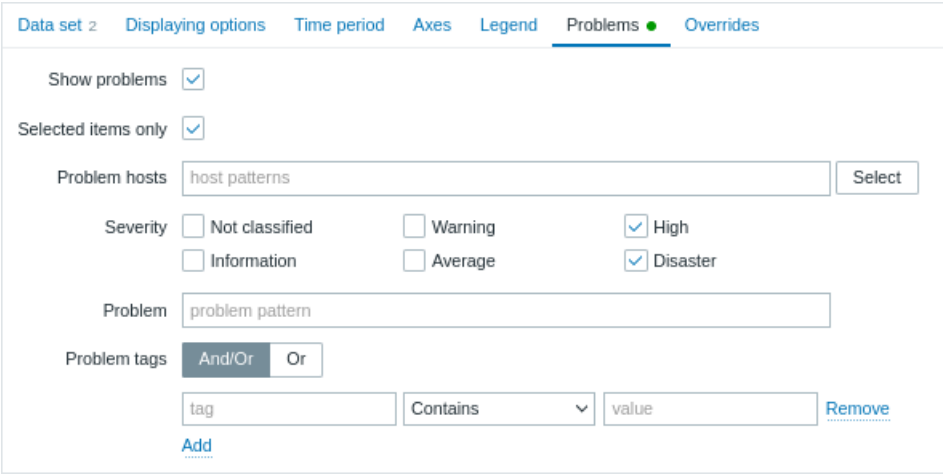


Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu leģendu grafikā (pēc noklusējuma atzīmēta).
Rādīt min/vid./maks.	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu vienuma minimālās, vidējās un maksimālās vērtības.
Rādīt agregācijas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu agregācijas funkciju.
Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** — attēloto rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** — attēloto rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.

Rindu skaits/ Maksimālais rindu skaits	Ja *Rindas* ir iestatīts uz ”Fiksēts”, iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja *Rindas* ir iestatīts uz ”Mainīgs”, iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja *Rādīt min/vid./maks.* nav atzīmēts.

Problēmas

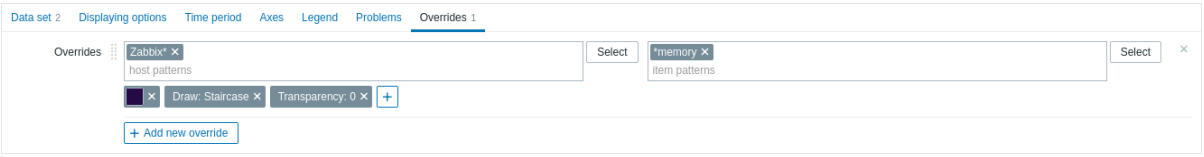
Cilne **Problēmas** ļauj pielāgot problēmu attēlošanu:



Rādīt problēmas	Atzīmējiet šo izvēles rūtiņu, lai iespējotu problēmu attēlošanu grafikā (pēc noklusējuma nav atzīmēta, t.i., ir atspējota).
Tikai atlasītie vienumi	Atzīmējiet šo izvēles rūtiņu, lai grafikā tiktu attēlotas tikai ar atlasītajiem vienumiem saistītās problēmas.
Problēmu hosti	Atlasiet problēmu hostus, kas tiks attēloti grafikā. Var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmju šablonu, vienkārši ievadiet virkni manuāli un nospiediet *Enter*. Rakstīšanas laikā ņemiet vērā, ka nolaižamajā sarakstā tiek parādīti visi atbilstošie hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Svarīgums	Atzīmējiet problēmu svarīguma līmeņus, lai filtrētu grafikā attēlojamās problēmas. Ja nav atzīmēts neviens svarīguma līmenis, tiks attēlotas visas problēmas.
Problēma	Norādiet problēmas nosaukumu, kas tiks attēlots grafikā.
Problēmu birkas	Norādiet problēmu birkas, lai ierobežotu logrīkā attēloto problēmu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** — pietiek, ja ir izpildīts viens nosacījums.

Pārrakstīšana

Cilne **Pārrakstīšana** ļauj pievienot pielāgotas pārrakstīšanas datu kopām:



Pārrakstīšanas ir noderīgas, ja datu kopai, izmantojot aizstājējzīmi ∗, ir atlasīti vairāki vienumi un vēlaties mainīt to, kā vienumi pēc noklusējuma tiek attēloti (piemēram, noklusējuma pamatkrāsu vai jebkuru citu īpašību).

Esošās pārrakstīšanas (ja tādas ir) tiek parādītas sarakstā. Lai pievienotu jaunu pārrakstīšanu:

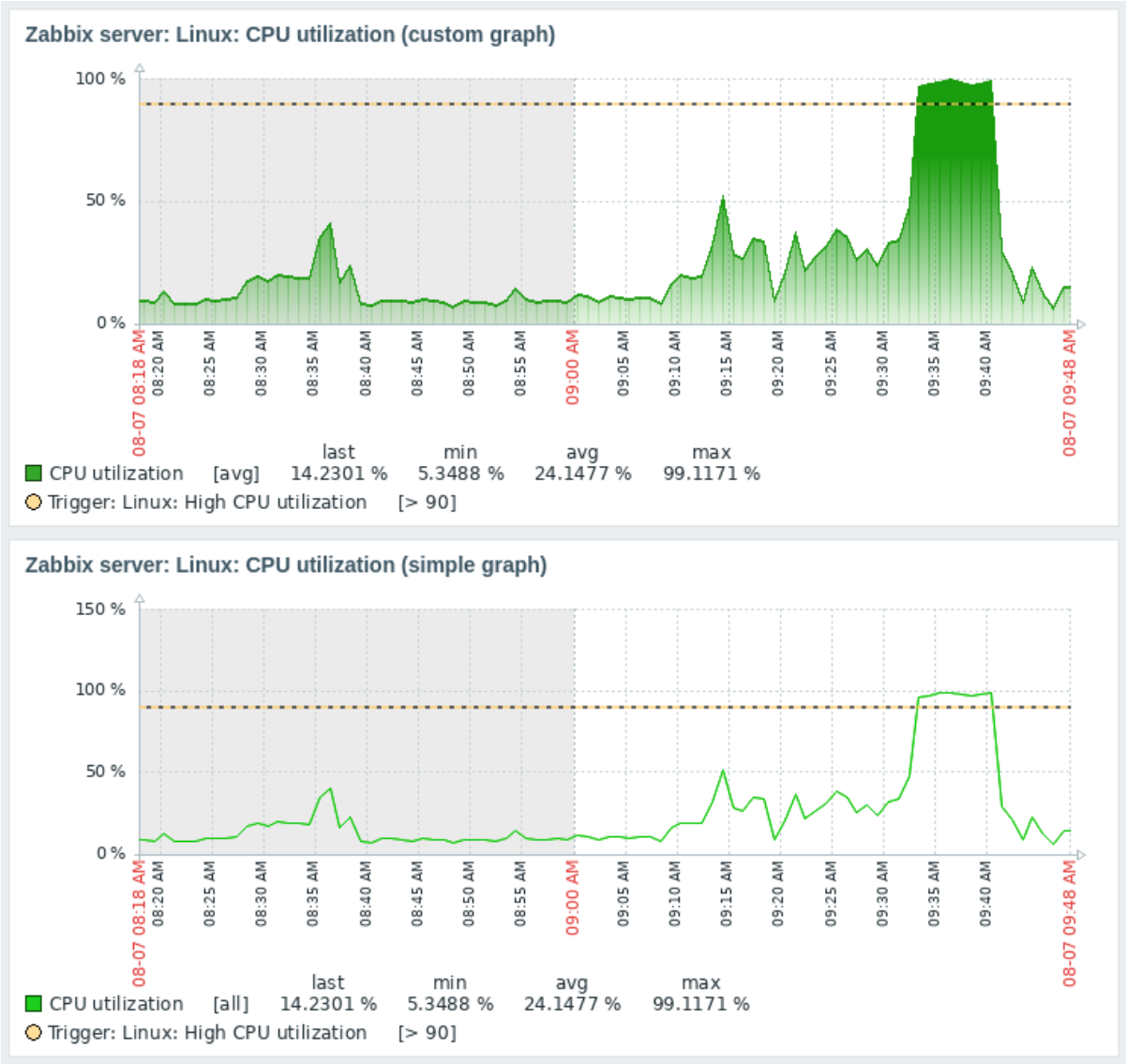
- Noklikšķiniet uz pogas

- Atlasiet pārrakstīšanas hosts un vienumus. Alternatīvi varat ievadīt hostu un vienumu šablonus. Var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, vienkārši ievadiet virkni manuāli un nospiediet *Enter*. Rakstīšanas laikā pievērsiet uzmanību tam, kā nolaižamajā sarakstā tiek parādīti visi atbilstošie hosti. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams, piemēram, atsevišķi pievienot vienumu ar nosaukumu ”item*”, ja ir citi atbilstoši vienumi (piemēram, item2, item3). Hosta šablona un vienuma šablona parametri ir obligāti. Parametrs hostu šablonu norādīšanai nav pieejams, konfigurējot logrīku veidnes panelī. Parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus, konfigurējot logrīku veidnes panelī.
- Noklikšķiniet uz , lai atlasītu pārrakstīšanas parametrus. Jāatlasa vismaz viens pārrakstīšanas parametrs. Parametru aprakstus skatiet iepriekš esošajā cilnē *Datu kopa*.

9 Graph (classic)

Pārskats

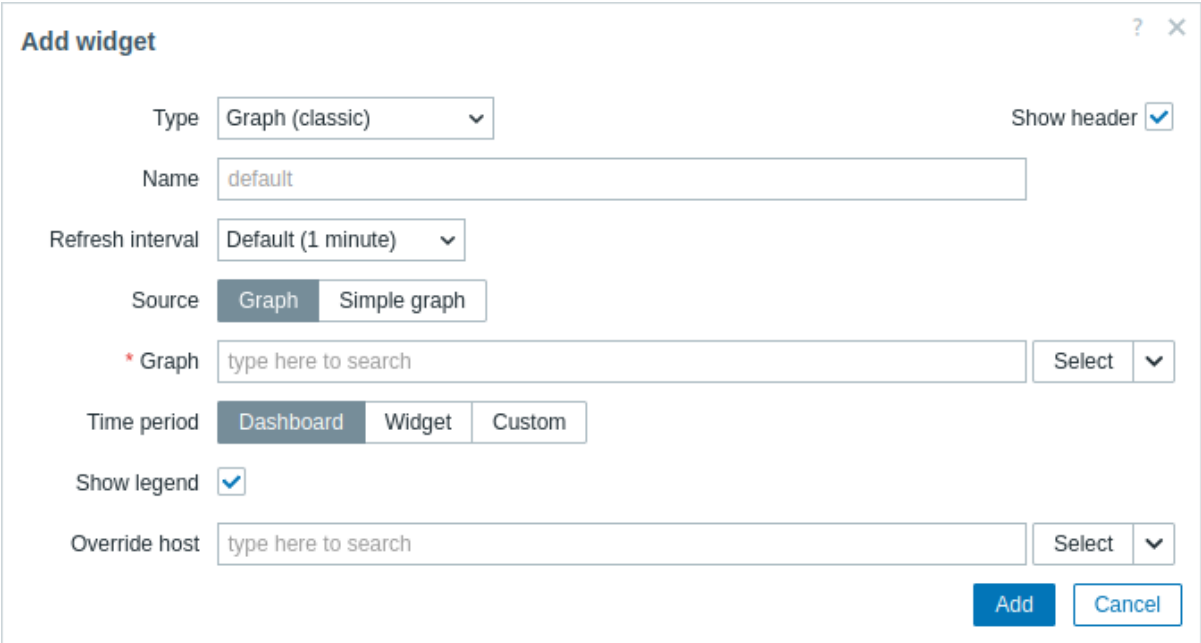
Logrīks *Graph (classic)* attēlo skaitliskus vienumu datus kā uz attēla balstītu pielāgotu grafiku vai vienkāršu grafiku. Tas atbalsta laika perioda atlasi, trigeru līnijas un izvēles leģendas attēlošanu.



Logrīkā *Graph (classic)* attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Download image*.

Konfigurācija

Lai konfigurētu, atlasiet *Graph (classic)* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Source	Atlasiet grafika tipu: **Graph** - pielāgots grafiks; **Simple graph** - vienkāršs grafiks.
Graph	Atlasiet pielāgoto grafiku, ko attēlot. Alternatīvi atlasiet saderīgu logrīku kā grafiku datu avotu. Šis parametrs ir pieejams, ja *Source* ir iestatīts uz ”Graph”.
Item	Atlasiet vienums, ko attēlot vienkāršajā grafikā. Alternatīvi atlasiet saderīgu logrīku kā vienumu datu avotu. Šis parametrs ir pieejams, ja *Source* ir iestatīts uz ”Simple graph”.
Time period	Iestatiet laika periodu, par kuru attēlot datus grafikā. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (iestatīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Widget”.
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
Show legend	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu grafika leģendu (pēc noklusējuma atzīmēta).
Override host	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Grafika leģenda sastāv no trim sadaļām:

- Vienumi un to agregētās vērtības
- Procentiles (ja konfigurētas)
- Trigeri (ja tie ir saistīti ar attēlotajiem vienumiem)

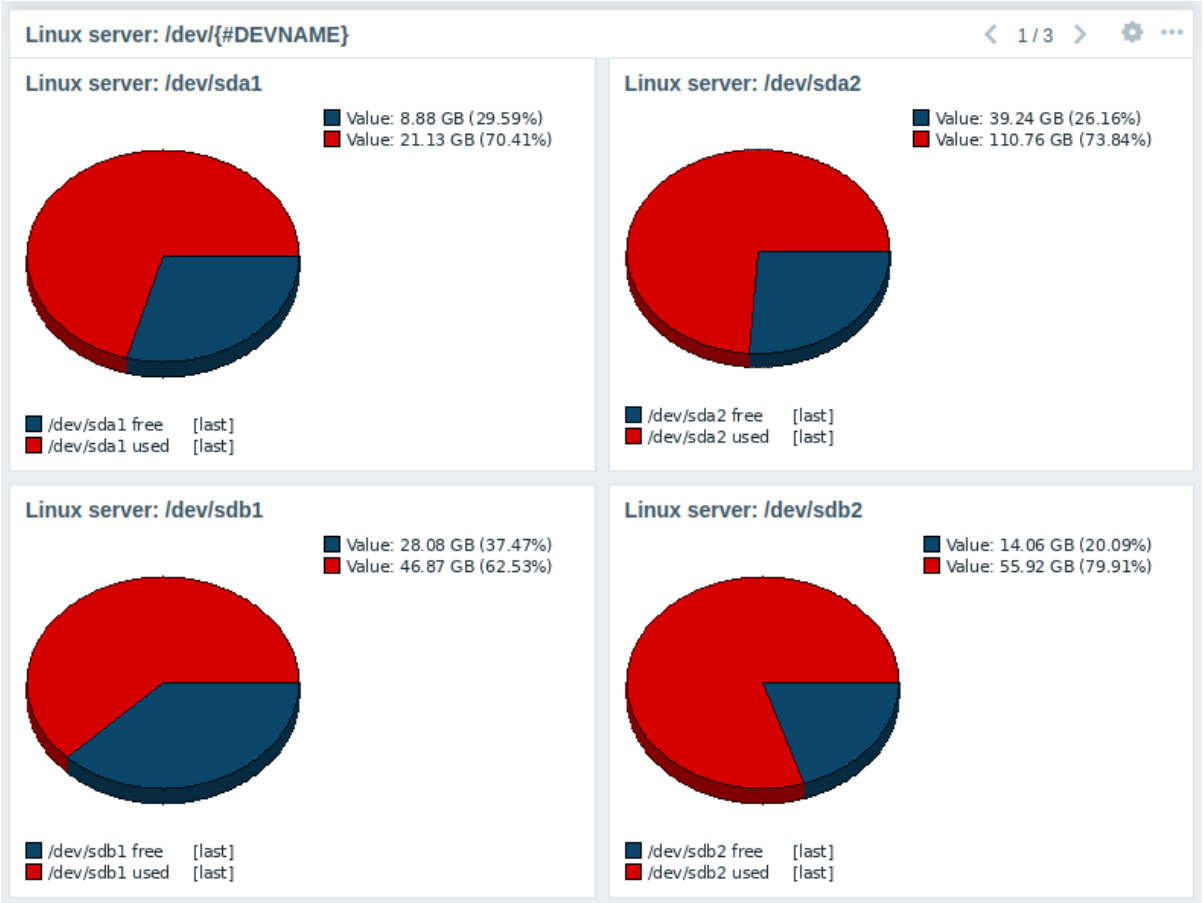
Ja grafika augstums logrīkā ir nepietiekams, leģenda var netikt attēlota vai tikt attēlota tikai daļēji. Vispirms tiek paslēpti trigeri un procentiles, pēc tam vienumu leģenda. Lai attēlotu pilnu leģendu, palieliniet logrīka vertikālo izmēru.

Note:
Var attēlot ne vairāk kā 3 trigeru līnijas. Ja trigeru ir vairāk, attēlošanai prioritāri tiek izvēlēti trigeri ar zemāku svarīgumu.

10 Graph prototype

Pārskats

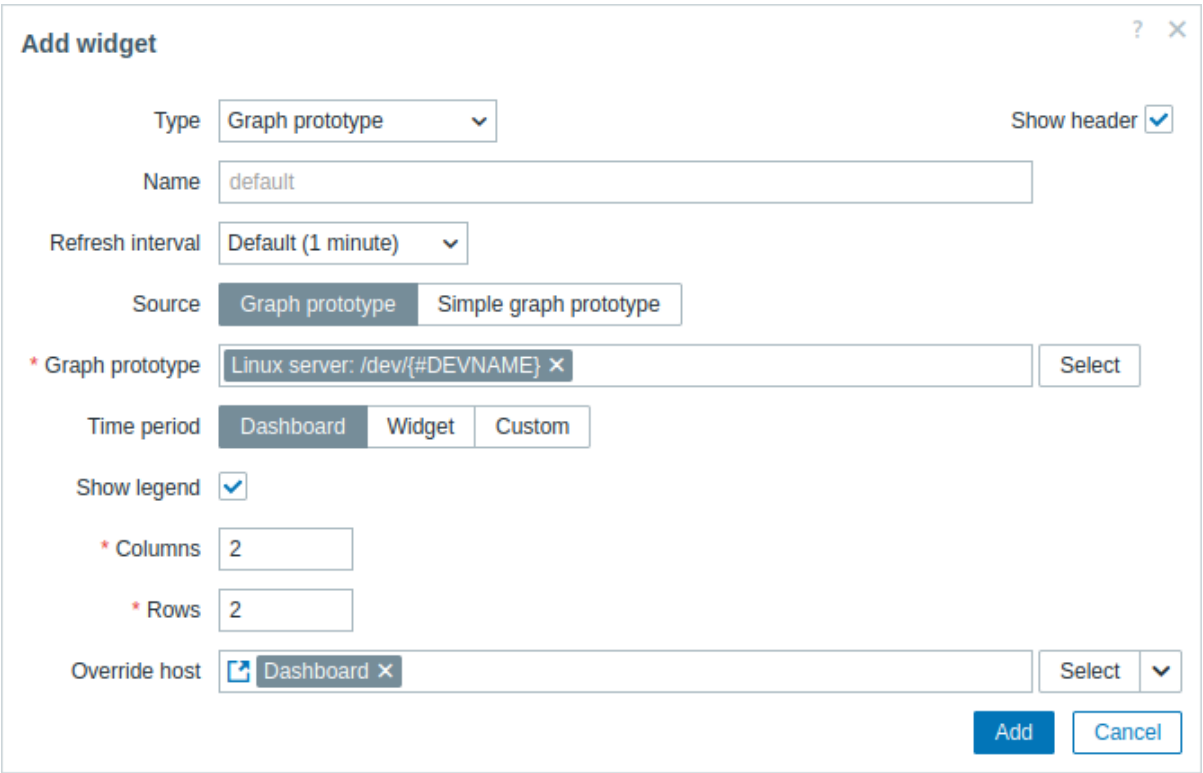
Grafika prototipa logrīks attēlo režģi ar automātiski izveidotiem pielāgotiem grafikiem, kas balstīti uz zema līmeņa atklāšanas kārtulas grafiku prototipiem vai vienumu prototipiem.



Grafiki ir sakārtoti konfigurējamā režģa izkārtojumā; ja grafiku skaits pārsniedz izkārtojuma ietilpību, logrīkā tiek iespējota lapošana.

Konfigurācija

Lai konfigurētu, atlasiet *Graph prototype* kā logrīka tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Source	Atlasiet grafiku avotu: **Graph prototype** vai **Simple graph prototype**.
Graph prototype	Atlasiet grafika prototipu, lai parādītu grafikus, ko atklājis grafika prototips. Šis parametrs ir pieejams, ja *Source* ir iestatīts uz ”Graph prototype”.

Item prototype	Atlasiet vienuma prototipu, lai parādītu vienkāršos grafikus vienumiem, ko atklājis vienuma prototips. Šis parametrs ir pieejams, ja *Source* ir iestatīts uz "Simple graph prototype".
Time period	Iestatiet laika periodu, par kuru attēlot datus grafikos. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (iestatīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Widget".
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
Show legend	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu grafiku leģendu (pēc noklusējuma atzīmēta).
Override host	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Columns	Ievadiet grafiku kolonnu skaitu, ko attēlot grafika prototipa logrīkā.
Rows	Ievadiet grafiku rindu skaitu, ko attēlot grafika prototipa logrīkā.

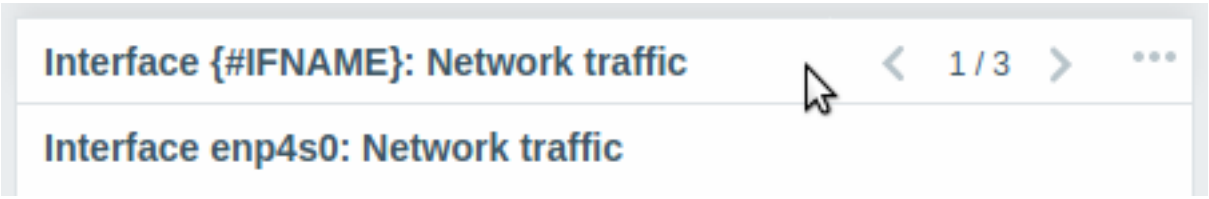
Grafika leģenda sastāv no trim sadaļām:

- Vienumi un to agregētās vērtības
- Percentiles (ja konfigurētas)
- Trigeri (ja tādi ir saistīti ar attēlotajiem vienumiem)

Ja grafika augstums logrīkā ir nepietiekams, leģenda var netikt parādīta vai tikt parādīta tikai daļēji. Vispirms tiek paslēpti trigeri un percentiles, pēc tam vienumu leģenda. Lai parādītu pilnu leģendu, palieliniet logrīka vertikālo izmēru.

Note:
Var attēlot ne vairāk kā 3 trigeru līnijas. Ja trigeru ir vairāk, attēlošanai prioritāri tiek izvēlēti trigeri ar zemāku svarīguma pakāpi.

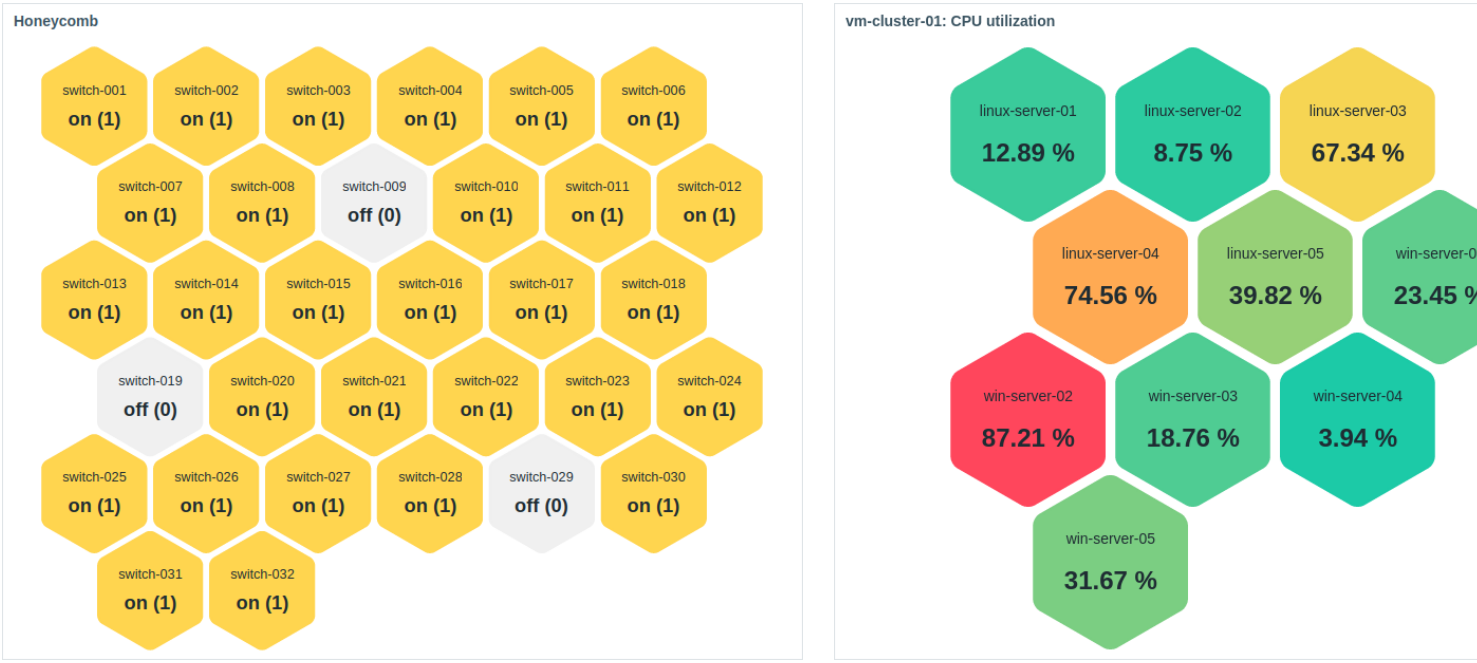
Lai gan parametri *Columns* un *Rows* ļauj logrīkā ievietot vairāk nekā vienu grafiku, atklāto grafiku joprojām var būt vairāk nekā logrīkā pieejamo kolonnu/rindu. Šādā gadījumā logrīkā kļūst pieejama lapošana, un iznirstoša galvene ļauj pārslēgties starp lapām, izmantojot kreiso un labo bultiņu:



11 Honeycomb

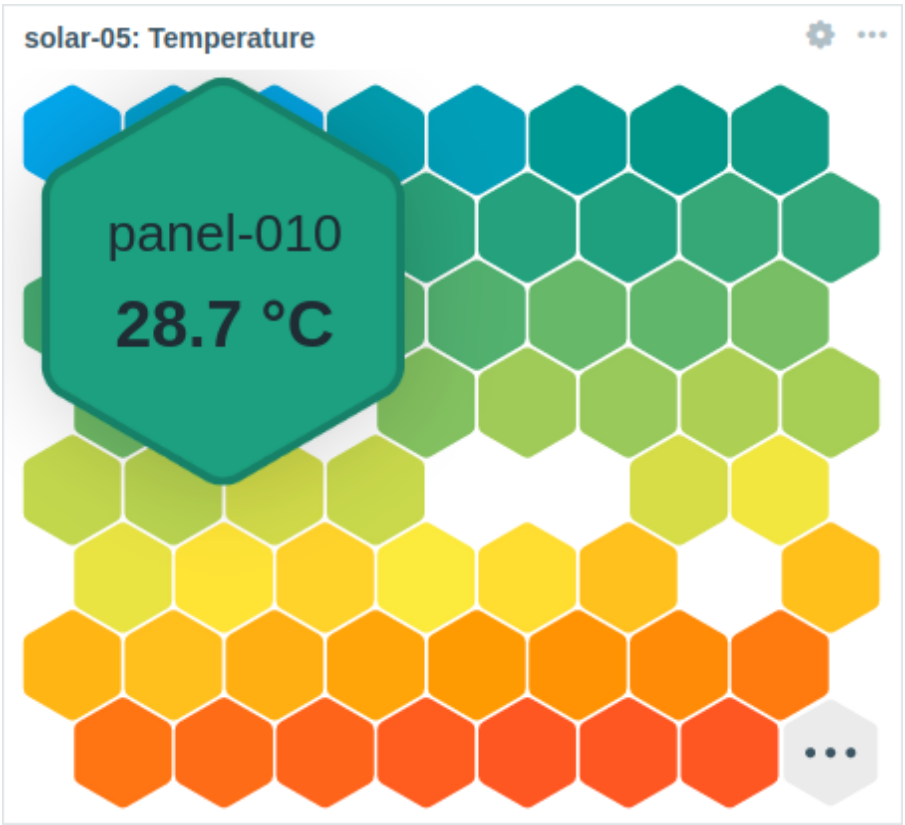
Pārskats

Honeycomb logrīks attēlo hostus vai vienumus kā sešstūrainas šūnas režģī. Šāds izkārtojums ļauj viegli noteikt problemātiskus hostus/vienumus, pamanīt problēmu kopas, ātri salīdzināt grupas un veikt citas darbības.



Attēloto šūnu skaitu ierobežo logrīka izmērs un minimālais šūnas izmērs (32px). Ja visas šūnas neietilpst, pēdējā šūnā tiek parādīta daudzpunkte. Logrīka izmēru var mainīt, un šūnu skaits, izmērs un novietojums tiek dinamiski pielāgoti.

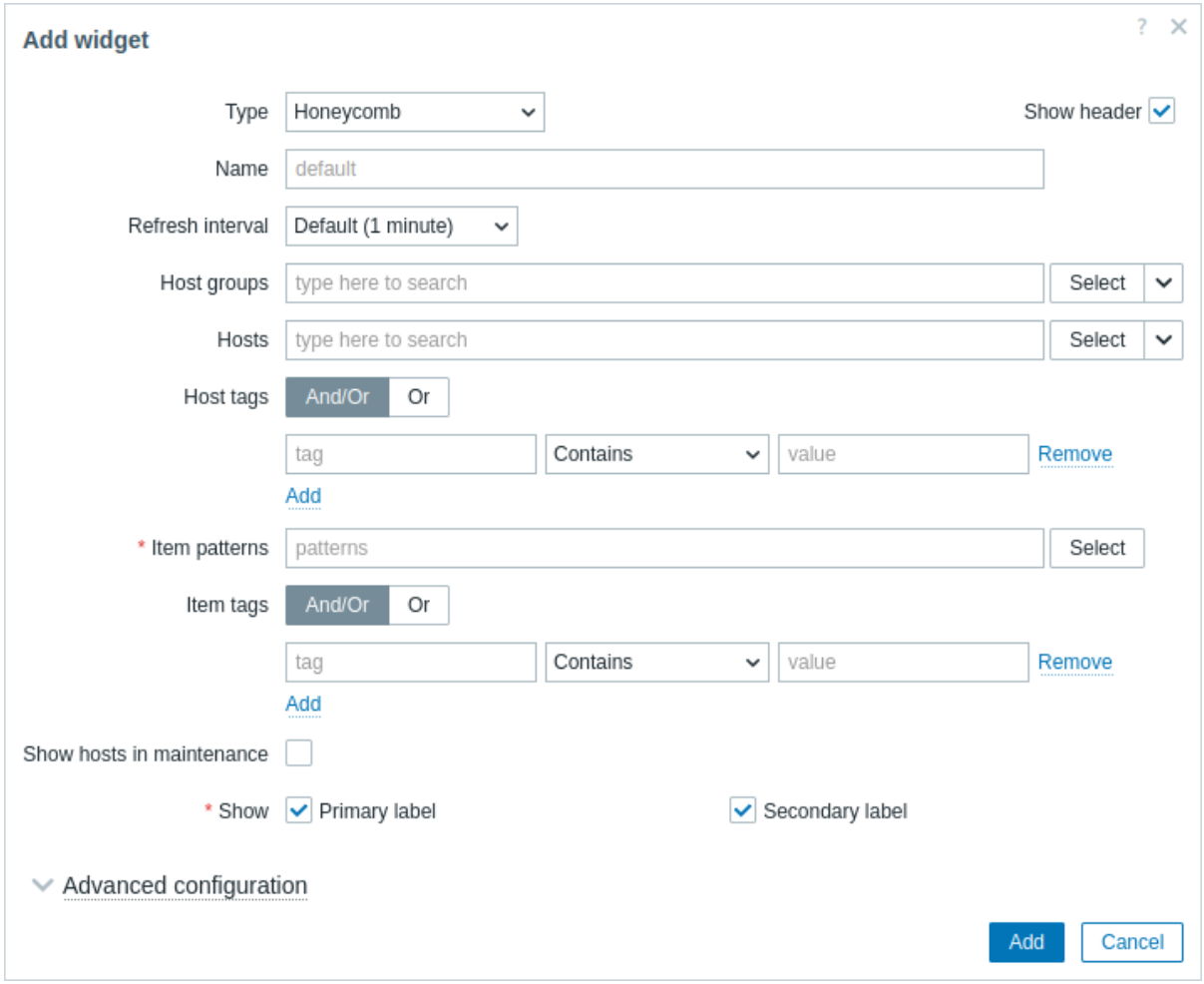
Novietojot peles kursoru virs šūnas, fokusētā šūna tiek palielināta, lai uzlabotu redzamību. Noklikšķinot uz šūnas, tās apmale tiek izcelta, līdz tiek atlasīta cita šūna.



Honeycomb logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Download image*.

Konfigurācija

Lai konfigurētu, atlasiet *Honeycomb* kā tipu:



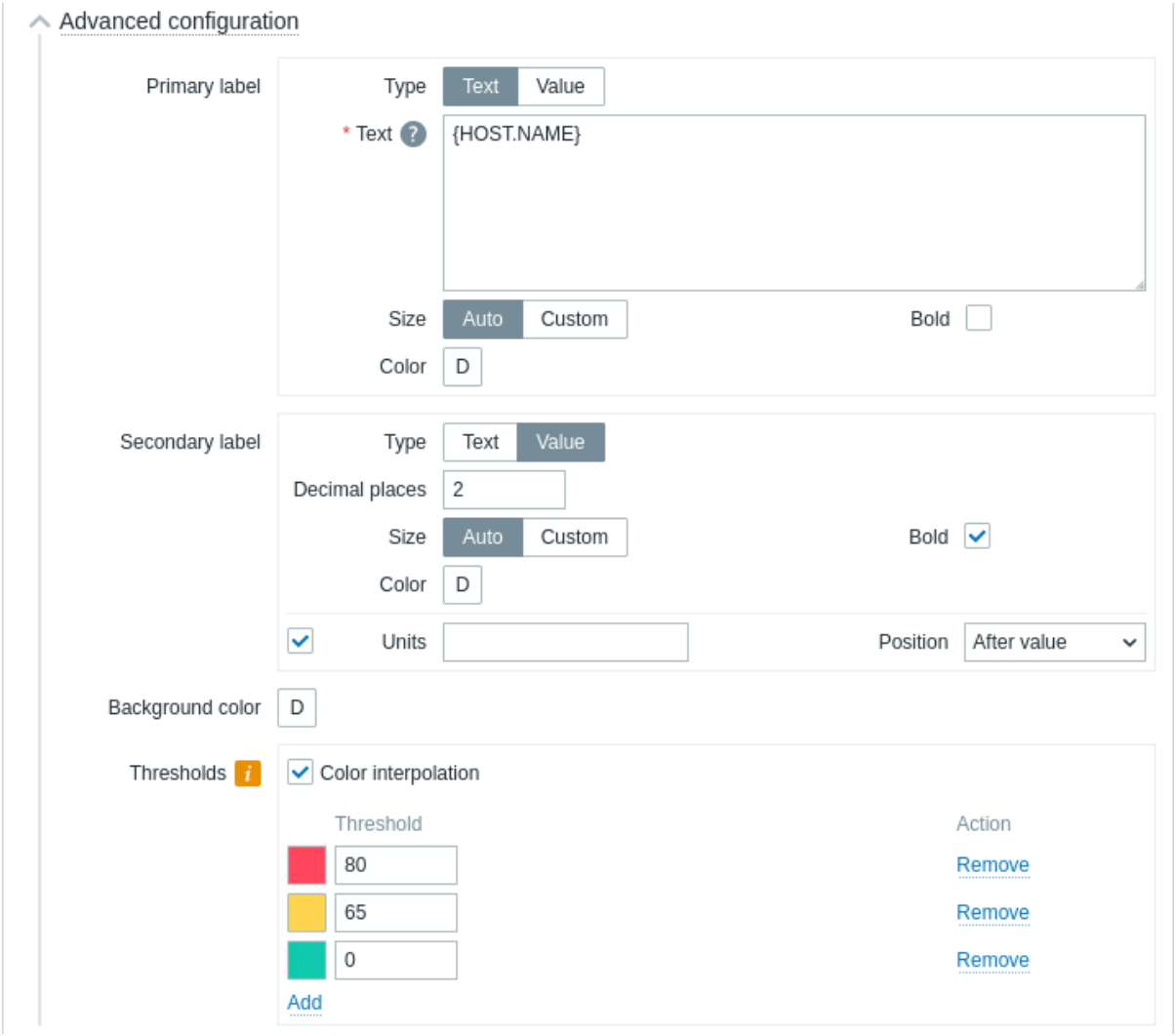
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Atlasot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas; ja nav atlasīta neviena hostu grupa, logrīks attēlos visas hostu grupas, kurās ir hosti ar vienumiem, kas atbilst atlasītajam *Vienumu šablonam* (skatiet zemāk). Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Ja nav atlasīts neviens hosts, logrīks attēlos visus hostus ar vienumiem, kas atbilst atlasītajam *Vienumu šablonam* (skatiet zemāk). Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Hostu birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto hostu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** — pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Vienumu šabloni	Ievadiet vienumu šablonus vai atlasiet esošos vienumus kā vienumu šablonus. Honeycomb tiks attēloti to vienumu dati, kas atbilst ievadītajiem vai atlasītajiem šabloniem. Parametrs *Vienumu šabloni* ir obligāts. Atlasei var izmantot aizstājējzīmju šablonus (piemēram, `*` atgriezīs vienumus, kas atbilst nullei vai vairāk rakstzīmēm; `Zabbix*` atgriezīs vienumus, kas sākas ar "Zabbix"). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Kad sāksiet rakstīt, nolaižamajā sarakstā tiks rādīti atbilstoši vienumi, kas ierobežoti ar tiem, kuri pieder atlasītajiem *Hostiem* vai hostiem atlasītajās *Hostu grupās*, ja tādas ir. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams, piemēram, atsevišķi pievienot vienumu ar nosaukumu *item**, ja ir citi atbilstoši vienumi (piemēram, item2, item3). Konfigurējot logrīku veidnes panelī, šis parametrs ļauj atlasīt tikai vienumus, kas konfigurēti veidnē.
Vienumu birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto vienumu skaitu. Papildinformāciju skatiet iepriekš sadaļā *Hostu birkas*.
Rādīt hostus uzturēšanā	Atzīmējiet šo izvēles rūtiņu, lai attēlotu hostus uzturēšanā (šajā gadījumā blakus hosta nosaukumam tiks rādīta uzturēšanas ikona). Šis parametrs ir apzīmēts kā *Rādīt datus uzturēšanā*, konfigurējot logrīku veidnes panelī.
Rādīt	Atzīmējiet šo izvēles rūtiņu, lai attēlotu attiecīgo honeycomb šūnas elementu — primāro etiķeti, sekundāro etiķeti. Jābūt atlasītam vismaz vienam elementam.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas sakļaujamajā sadaļā *Papildu konfigurācija* un tikai tiem elementiem, kas atlasīti laukā *Rādīt* (skatīt iepriekš), kā arī šūnveida režģa šūnu fona krāsai vai sliekšņiem.



Primārā/sekundārā etiķete

Tips — Atlasiet etiķetes tipu:
Teksts — etiķetē tiks parādīts teksts, kas norādīts parametrā *Teksts*;
Vērtība — etiķetē tiks parādīta vienuma vērtība ar decimālzīmju skaitu, kas norādīts parametrā *Decimālzīmes*.

Teksts — Ievadiet etiķetes tekstu. Šis teksts var aizstāt noklusējuma vienuma nosaukumu.
Tiek atbalstīts vairākrindu teksts. Iespējama teksta un atbalstīto makrosu kombinācija.
Tiek atbalstīti {HOST.*}, {ITEM.*}, {INVENTORY.*} un lietotāja makrosi.
Šūnveida režģa šūnas tiek kārtotas alfabētiski pēc hosta nosaukuma un katra hosta ietvaros — pēc vienuma nosaukuma.
Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz ”Teksts”.

Decimālzīmes — Ievadiet ar vērtību parādāmo decimālzīmju skaitu.
Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz ”Vērtība”, un ietekmē tikai vienumus, kas atgriež skaitliskus (peldošā komata) datus.

Izmērs — Atlasiet etiķetes izmēru:
Automātiski — izmantot automātiski pielāgotu etiķetes izmēru;
Pielāgots — ievadīt pielāgotu etiķetes izmēru (procentos attiecībā pret šūnveida režģa šūnas izmēru).
Ņemiet vērā, ka etiķetes, kas neietilpst šūnveida režģa šūnas izmērā, tiek saīsinātas.

Treknraksts — Atzīmējiet izvēles rūtiņu, lai vienuma mērvienības tiktu parādītas treknrakstā.

Krāsa — Atlasiet vienuma mērvienību krāsu krāsu atlasītājā.
”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarne tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.

Mērvienības

Mērvienības — Atzīmējiet izvēles rūtiņu, lai kopā ar vienuma vērtību tiktu parādītas mērvienības.
Ja ievadīsiet mērvienības nosaukumu, tas aizstās vienuma konfigurācijā iestatītās mērvienības.
Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz ”Teksts”.

Pozīcija	Atlasiet vienuma mērvienību novietojumu (pirms vai pēc vienuma vērtības). Šis parametrs tiek ignorēts šādām ar laiku saistītām mērvienībām: unixtime, uptime, s. Šis parametrs ir pieejams, ja *Tips* ir iestatīts uz ”Teksts”.
Fona krāsa	
Fona krāsa	Atlasiet šūnveida režģa šūnu fona krāsu krāsu atlasītājā. ”D” apzīmē noklusējuma krāsu, kas ir atkarīga no lietotāja saskarne tēmas. Lai atgrieztos pie noklusējuma krāsas, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Sliekšņi	
Krāsu interpolācija	Atzīmējiet izvēles rūtiņu, lai iespējotu vienmērīgu pāreju starp sliekšņu krāsām šūnveida režģa šūnām. Šis parametrs ir pieejams, ja ir iestatīti divi vai vairāki sliekšņi.
Slieksnis	Noklikšķiniet uz *Pievienot*, lai pievienotu slieksni, atlasiet sliekšņa krāsu krāsu atlasītājā un norādiet skaitlisku vērtību. Saglabājot sliekšņu saraksts tiks sakārtots augošā secībā. Ņemiet vērā, ka kā sliekšņi konfigurētās krāsas tiks pareizi parādītas tikai skaitliskiem vienumiem. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas vērtību kartēšanas.

12 Hosta pieejamība

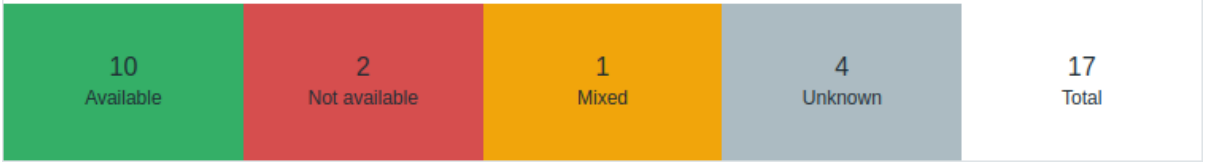
Pārskats

Logrīks *Host availability* attēlo pieejamību atlasītajās hostu grupās, ļaujot jums uzraudzīt, cik hostu ir pieejami vai nav pieejami.

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

Host availability

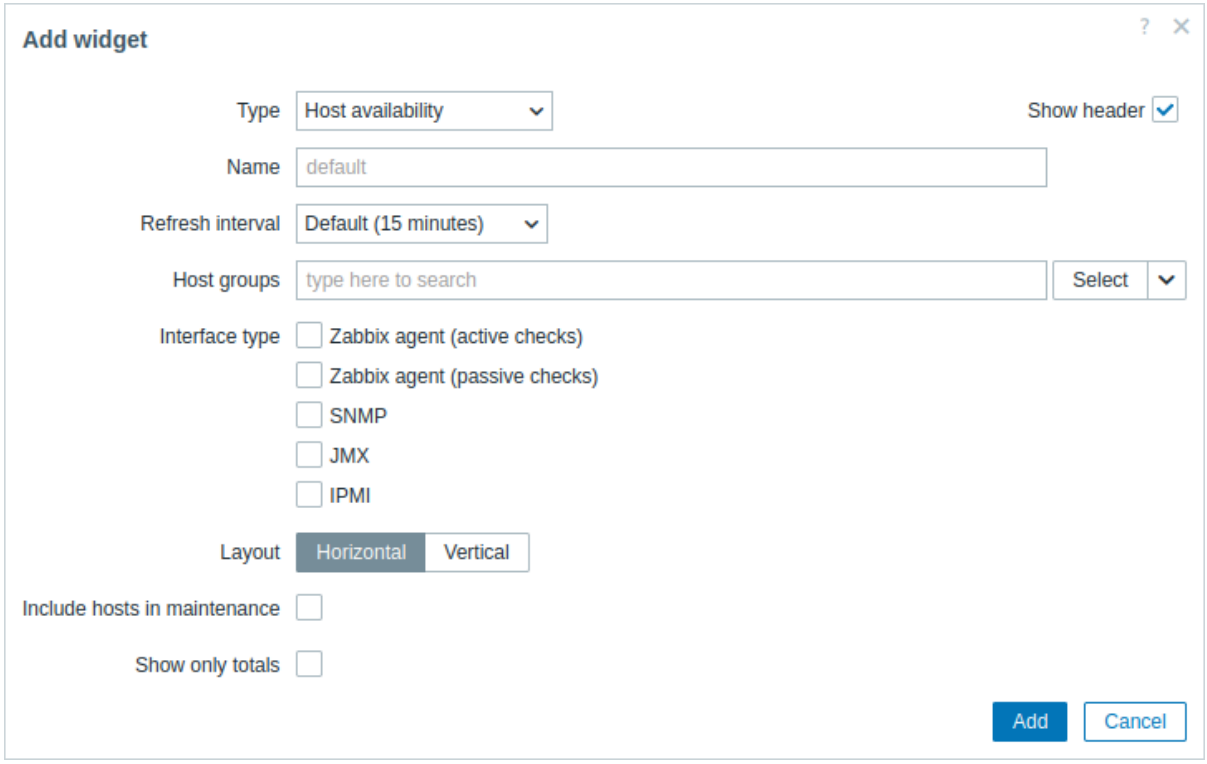


Hostu pieejamība tiek skaitīta šādi:

- *Pieejams* — hosti, kuriem visas saskarnes ir pieejamas
- *Nav pieejams* — hosti, kuriem visas saskarnes nav pieejamas
- *Jaukts* — hosti, kuriem vismaz viena saskarne nav pieejama un vismaz viena ir pieejama vai nezināma
- *Nezināms* — hosti, kuriem vismaz viena saskarne ir nezināma, bet neviena nav nav pieejama
- *Kopā* — visu hostu kopskaits

Konfigurācija

Lai konfigurētu, atlasiet *Host availability* kā tipu:



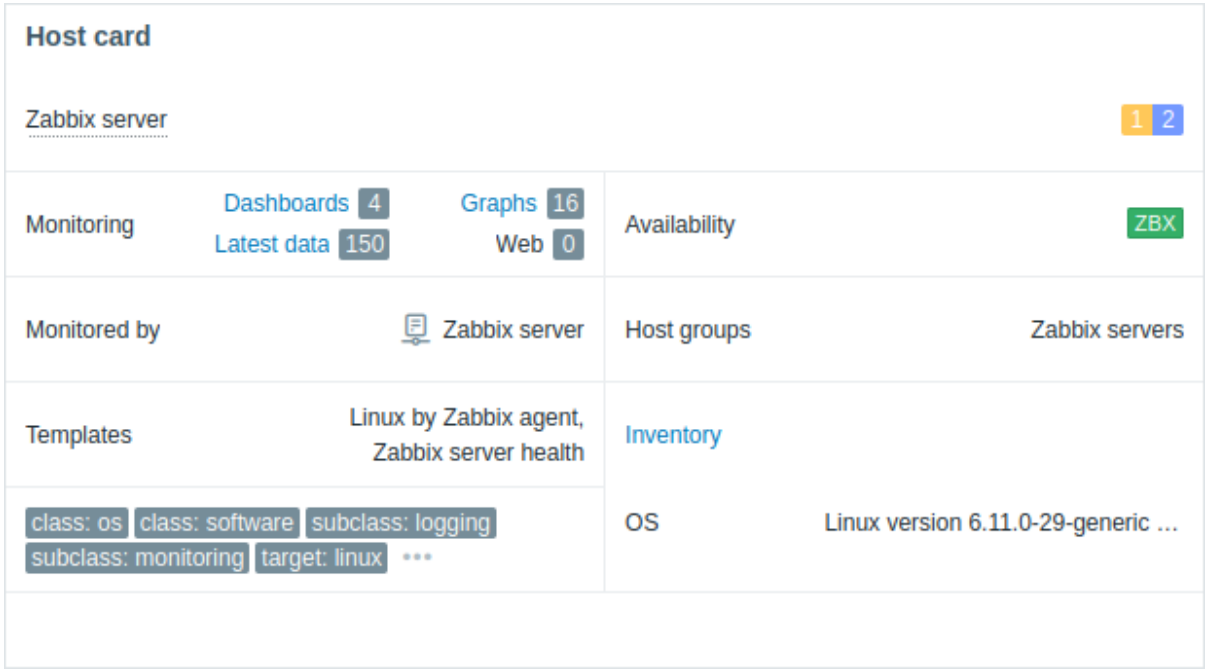
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Host groups	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Interface type	Atlasiet, kuru hosta saskarņu pieejamības datus vēlaties redzēt — *Zabbix aģents (aktīvās pārbaudes)*, *Zabbix aģents (pasīvās pārbaudes)*, *SNMP*, *JMX*, *IPMI*. Opcijai *Zabbix aģents (aktīvās pārbaudes)* šūna *Mixed* vienmēr būs tukša, jo šī tipa vienumiem nevar būt vairākas saskarnes. Pēc noklusējuma tiek attēlota visu saskarņu pieejamība, ja nekas nav atlasīts.
Layout	Atlasiet horizontālu attēlojumu (kolonnas) vai vertikālu attēlojumu (rindas).
Include hosts in maintenance	Iekļaut hostus, kuriem ir iestatīts apkopes režīms, statistikā. Šis parametrs ir apzīmēts kā *Show data in maintenance*, konfigurējot logrīku veidnes panelī.
Show only totals	Ja atzīmēts, tiek attēlots kopējais hostu skaits bez sadalījuma pa saskarnēm. Šī opcija ir atspējota, ja ir atlasīta tikai viena saskarne.

13 Hosta kartīte

Pārskats

Logrīks *Host card* attēlo informāciju par vienu hostu, ļaujot ātri novērtēt hosta statusu un konfigurāciju (pieejamību, problēmas, inventāru u.c.).

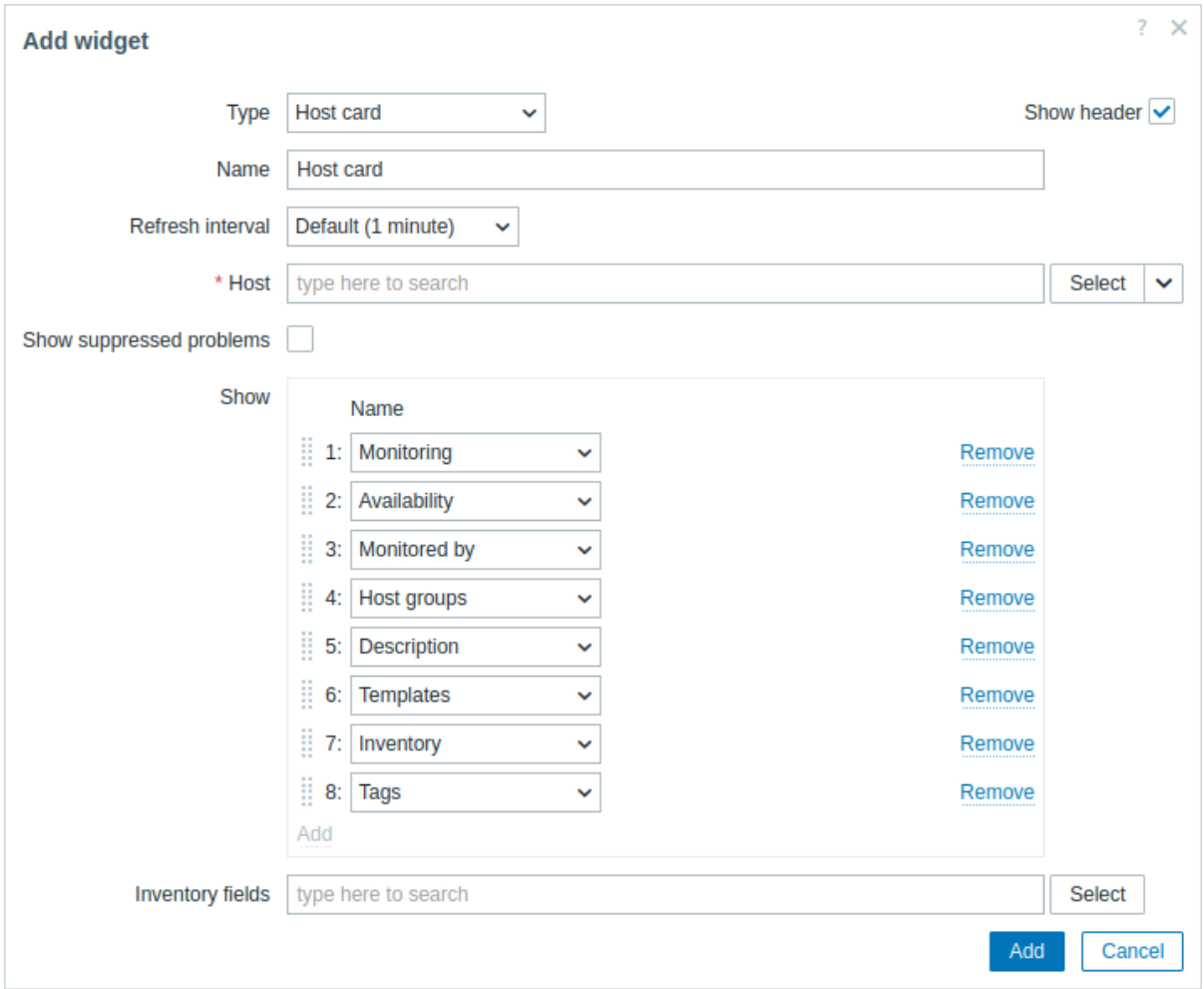


Hostu var iestatīt tieši logrīka konfigurācijā vai atlasīt no saderīga logrīka vai hosta atlasītāja informācijas panelī.

Logrīks var attēlot vairākas sadaļas, no kurām katra rāda atšķirīgu informāciju. Sadaļu izkārtojums tiek automātiski pielāgots atkarībā no logrīka platuma; paplašinot to horizontāli, sadaļas tiek automātiski pārkārtotas vairākās kolonnās.

Konfigurācija

Lai konfigurētu, atlasiet *Host card* kā tipu:



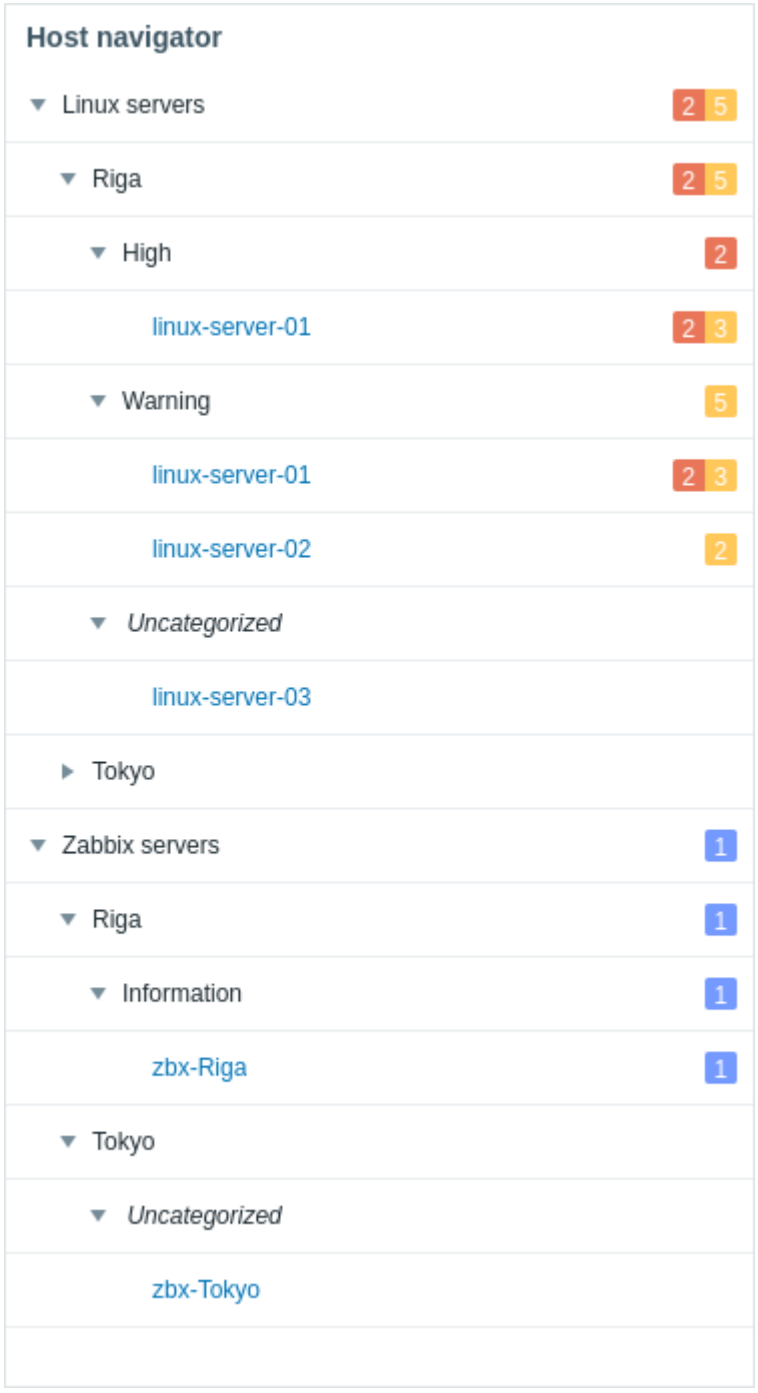
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādus specifiskos parametrus:

Hosts	Atlasiet hostu. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Ievadot tekstu, tiek parādīti atbilstoši ieteikumi. Logrīka augšējā sadaļā vienmēr tiek parādīts **hosta nosaukums** (noklikšķinot uz tā, tiek atvērta hosta izvēlne): - Ja hosts ir atspējots, blakus hosta nosaukumam tiek parādīta sarkana *Disabled* etiķete. - Ja hosts atrodas uzturēšanā, blakus hosta nosaukumam tiek parādīta uzturēšanas ikona. Logrīka augšējā sadaļā tiek parādīts arī **hosta problēmu** skaits, grupēts pēc svarīguma pakāpes (noklikšķinot uz tiem, tiek atvērtas hosta problēmas). Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Rādīt apspiestās problēmas	Atzīmējiet šo izvēles rūtiņu, lai problēmu aprēķinā iekļautu apspiestās problēmas.
Rādīt	Pievienojiet sadaļas, kas tiks parādītas logrīkā. Sadaļas var pārkārtot, velkot un nometot.
Hostu grupas	Parāda hostu grupas, kurām hosts pieder.
Apraksts	Parāda hosta aprakstu.
Uzraudzība	Parāda ātrās saites pārejai uz hosta Paneļiem, Jaunākajiem datiem, Grafikiem un Tīmekļa scenārijiem. Skaitlis blakus katrai saitei norāda attiecīgo vienību skaitu.
Pieejamība	Parāda hosta pieejamību pēc saskarnes.
Uzrauga	Parāda, vai hostu uzrauga Zabbix serveris vai konkrēts starpniekserveris vai starpniekserveru grupa.
Veidnes	Parāda ar hostu saistītās veidnes.
Inventārs	Parāda hosta inventāra laukus.
Birkas	Parāda hosta birkas.
Inventāra lauki	Atlasiet inventāra laukus, kas tiks parādīti. Ievadot tekstu, tiek parādīti atbilstoši ieteikumi. Ja nav norādīts neviens inventāra lauks, tiks parādīti visi aizpildītie inventāra lauki. Šis parametrs ir pieejams, ja *Rādīt* satur *Inventārs*.

14 Hostu navigators

Pārskats

Logrīks *Host navigator* parāda hostu sarakstu, pamatojoties uz dažādām filtrēšanas un grupēšanas opcijām.



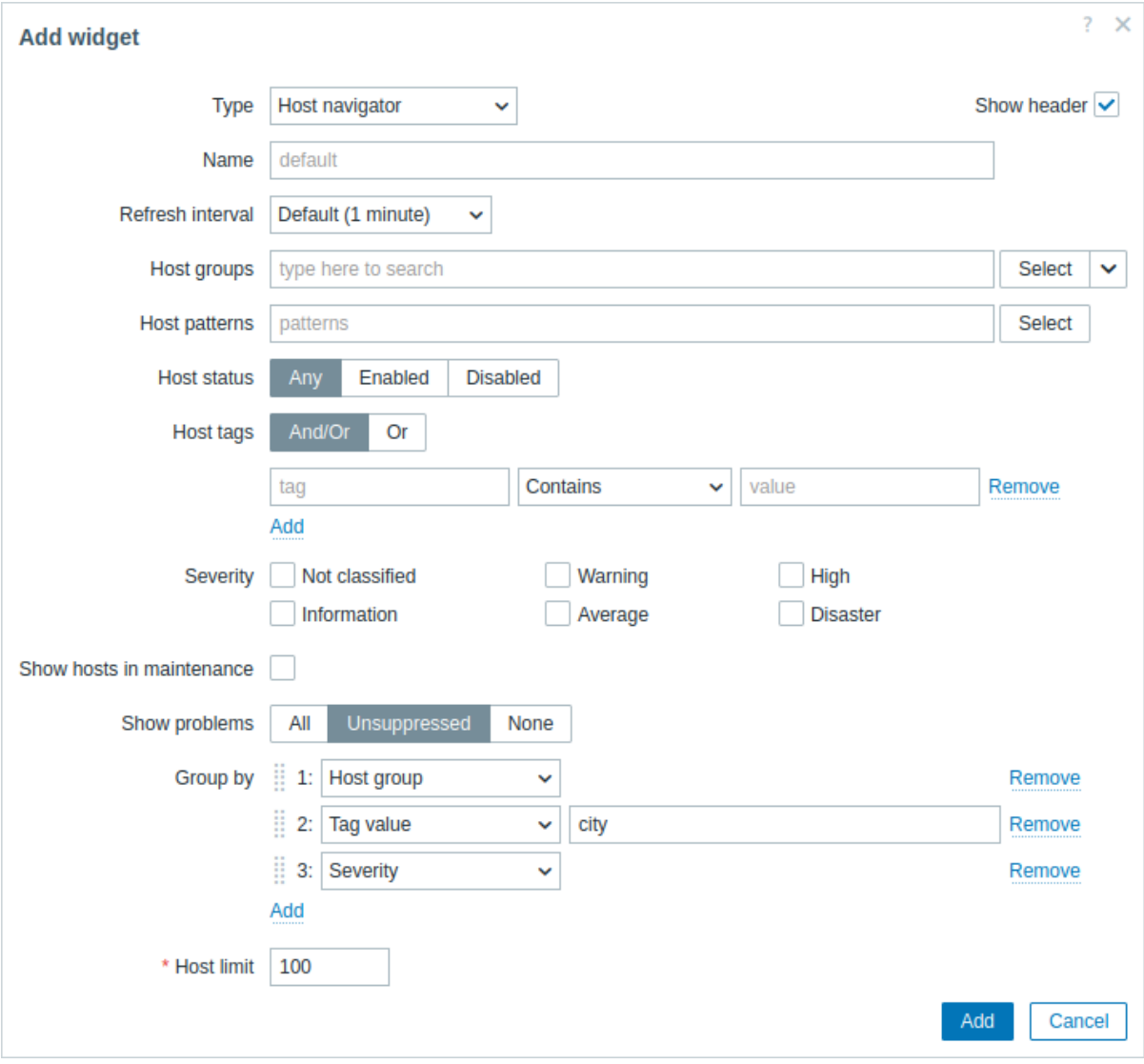
Grupas, pēc kurām hosti ir organizēti, var izvērst vai sakļaut.

Grupām, problēmām un uzturēšanā esošajiem hostiem papildu informācija ir pieejama, novietojot peles kursoru virs attiecīgā elementa.

Logrīks ir visnoderīgākais, lai kontrolētu, ko citi logrīki attēlo, pamatojoties uz atlasīto hostu.

Konfigurācija

Lai konfigurētu, atlasiet *Host navigator* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

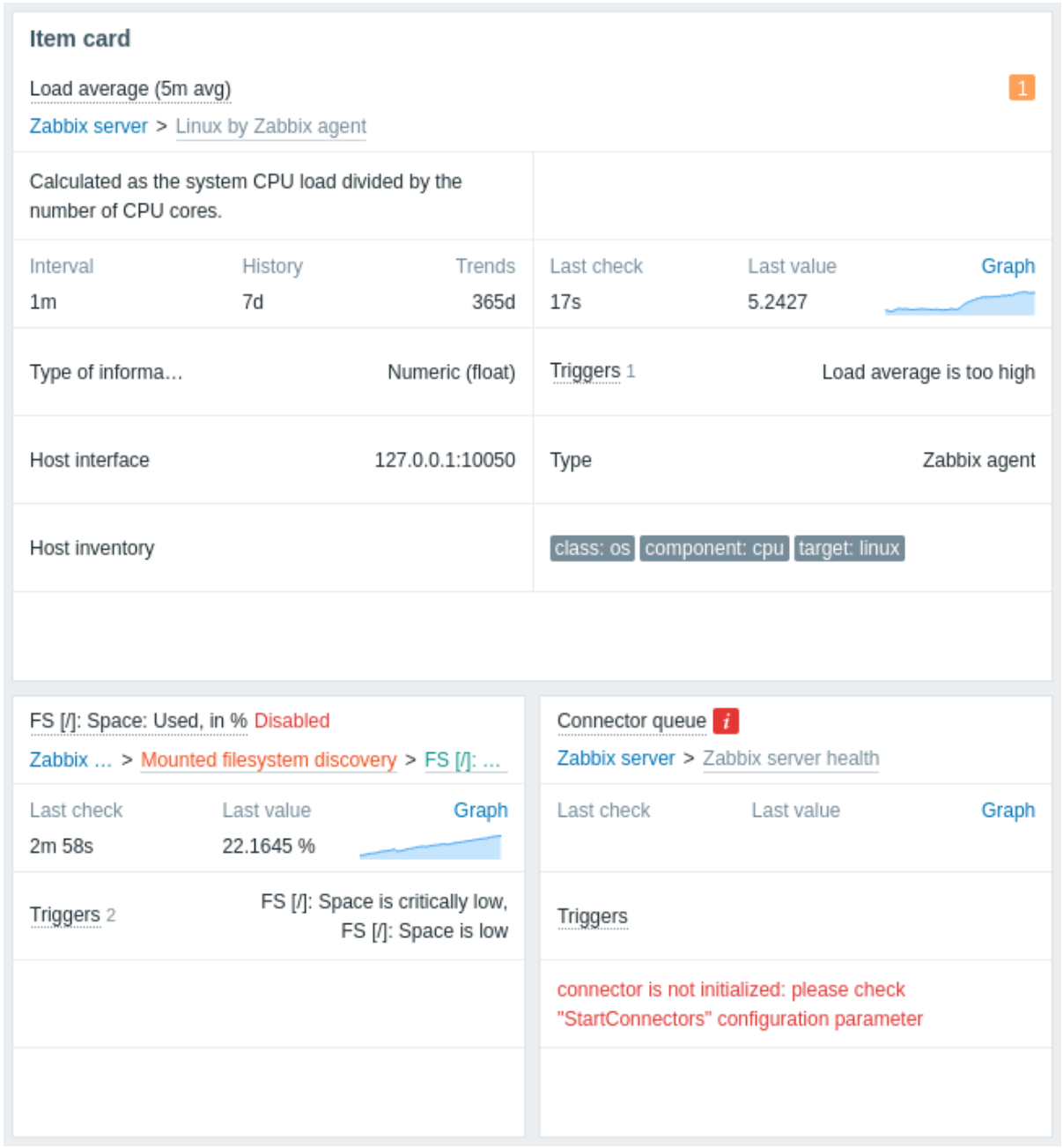
Hostu grupas	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Atlasot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas; ja nav atlasīta neviena hostu grupa, logrīks attēlos visus hostus no visām hostu grupām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hostu raksti	Ievadiet hostu rakstus vai atlasiet esošos hostus kā hostu rakstus. Hosti, kas atbilst norādītajiem rakstiem, tiks attēloti hostu navigatorā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Ja nav atlasīts neviens hosts, logrīks attēlos visus hostus. Atlasei var izmantot aizstājējzīmju rakstus (piemēram, * atgriezīs hostus, kas atbilst nullei vai vairāk rakstzīmēm; `Zabbix*` atgriezīs hostus, kuru nosaukums sākas ar "Zabbix"). Lai norādītu aizstājējzīmes rakstu, ievadiet virkni manuāli un nospiediet *Enter*. Kad sāksiet rakstīt, nolaižamajā sarakstā tiks rādīti atbilstošie hosti, ierobežoti ar tiem, kas pieder hostiem atlasītajās *Hostu grupās*, ja tādas ir. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosta statuss	Filtrējiet, kurus hostus attēlot, pamatojoties uz to statusu (jebkurš, iespējots, atspējots). Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hostu birkas	Norādiet birkas, lai filtrētu logrīkā attēlojamos hostus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **And/Or** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Or nosacījuma; **Or** - pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Svarīgums	Atzīmējiet problēmu svarīguma līmeņus, lai filtrētu logrīkā attēlojamos hostus ar problēmām. Ja nav atzīmēts neviens svarīguma līmenis, tiks attēloti visi hosti ar visām problēmām.
Rādīt hostus apkopes režīmā	Atzīmējiet šo izvēles rūtiņu, lai attēlotu hostus apkopes režīmā (šajā gadījumā blakus hosta nosaukumam tiks rādīta apkopes ikona). Šis parametrs ir apzīmēts kā *Show data in maintenance*, konfigurējot logrīku veidnes panelī.
Rādīt problēmas	

Grupēt pēc	Pievienojiet grupēšanas atribūtu, pēc kura grupēt atlasītos hostus: **Host group** - grupēt hostus pēc to hostu grupas; **Tag value** - ievadiet birkas nosaukumu, lai grupētu hostus pēc šīs birkas vērtībām (piemēram, ievadiet ”city”, lai grupētu hostus pēc vērtībām ”Riga”, ”Tokyo” utt.); **Severity** - grupēt hostus pēc to problēmu svarīguma līmeņiem. Ja *Show problems* ir konfigurēts problēmu attēlošanai, tās tiek attēlotas šādi: - katrai svarīguma līmeņa grupai tiek attēlots tikai atbilstošais problēmu skaits; - katram hostam tiek attēloti visi tā problēmu skaiti. Ņemiet vērā, ka hosti tiks grupēti tikai pēc tiem svarīguma līmeņiem, kas atzīmēti parametrā *Severity*; ja nav atzīmēts neviens svarīguma līmenis, visi hosti tiks grupēti pēc visiem svarīguma līmeņiem. Grupēšanas atribūtus var pārkārtot, velkot tos uz augšu vai uz leju aiz turiņa pirms grupas nosaukuma. Ņemiet vērā, ka grupēšanas atribūtu secība nosaka grupu ligzdošanas secību. Piemēram, norādot vairākus birku nosaukumus (1: color, 2: city), hosti vispirms tiks grupēti pēc krāsas (red, blue utt.) un pēc tam pēc pilsētas (Riga, Tokyo utt.). Hosts var tikt attēlots vairākās grupās atkarībā no konfigurētajiem grupēšanas atribūtiem (piemēram, grupējot pēc hostu grupas un hostam piederot vairākām hostu grupām). Noklikšķinot uz šādiem hostiem, tie tiek atlasīti un izcelti visās grupās. Hosti, kas neatbilst konfigurētajiem grupēšanas atribūtiem, tiek attēloti grupā *Uncategorized*. Var norādīt līdz 10 grupēšanas atribūtiem, un tiem visiem jābūt unikāliem. Ja nav norādīts neviens grupēšanas atribūts, hosti netiks grupēti.
Hostu limits	Ievadiet maksimālo attēlojamo hostu skaitu. Iespējamās vērtības ir diapazonā no 1 līdz 9999. Ja attēlošanai ir pieejams vairāk hostu nekā iestatītais limits, zem attēlotajiem hostiem tiks parādīts atbilstošs ziņojums (piemēram, ”100 of 100+ hosts are shown”). Ņemiet vērā, ka konfigurētais hostu limits ietekmē arī konfigurēto grupu attēlošanu; piemēram, ja hostu limits ir iestatīts uz 100 un hosti ir grupēti pēc birku vērtībām (vairāk nekā 200), logrīkā tiks attēlotas tikai pirmās 100 birku vērtības ar atbilstošajiem hostiem. Šo parametru neietekmē parametrs *Limit for search and filter results* sadaļā *Administration* → *General* → *GUI*. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

15 Vienuma kartīte

Pārskats

Logrīks *Vienuma kartīte* parāda detalizētu informāciju par vienu vienumu, ļaujot ātri novērtēt vienuma statusu un konfigurāciju (jaunākie dati, kļūdas, apraksts u.c.).

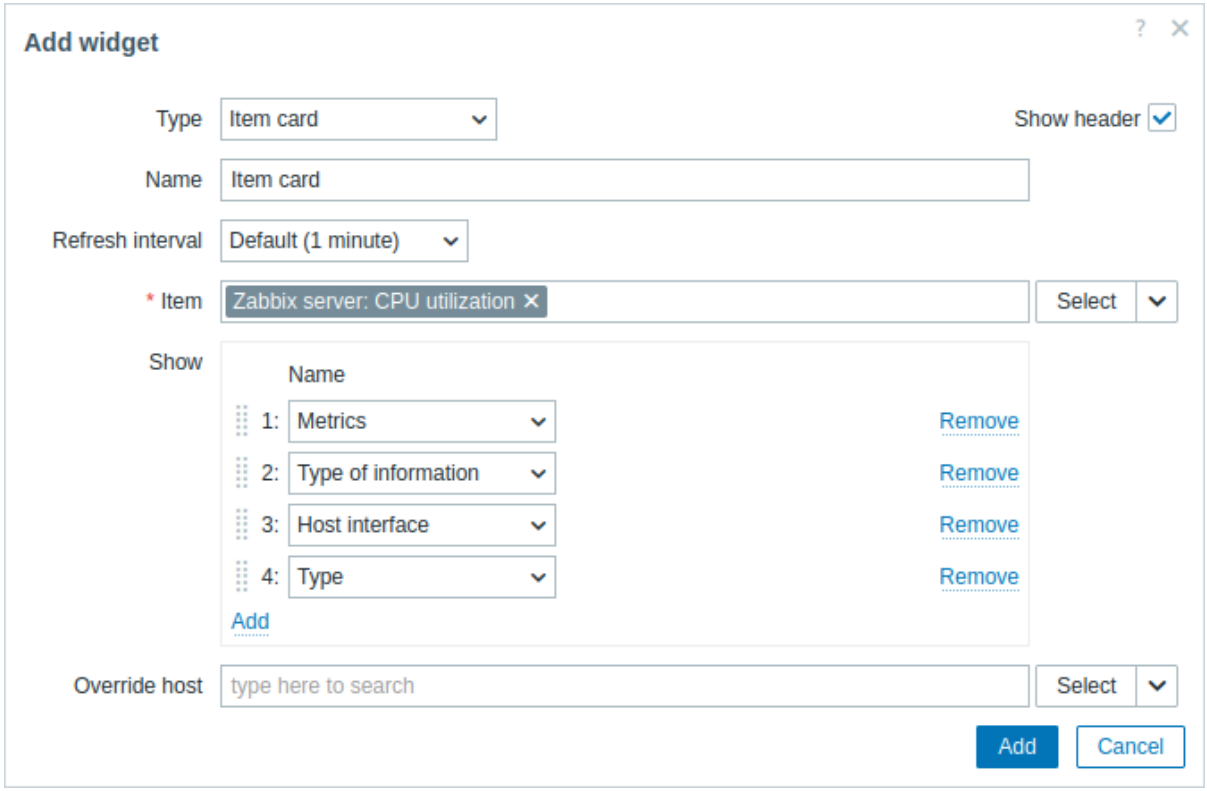


Vienumu var iestatīt tieši logrīka konfigurācijā vai atlasīt no saderīga logrīka informācijas panelī.

Logrīks var parādīt vairākas sadaļas, un katra no tām attēlo atšķirīgu informāciju. Sadaļu izkārtojums tiek automātiski pielāgots atkarībā no logrīka platuma; paplašinot to horizontāli, sadaļas tiek automātiski pārkārtotas vairākās kolonnās.

Konfigurācija

Lai konfigurētu, atlasiet *Vienuma kartīte* kā tipu:



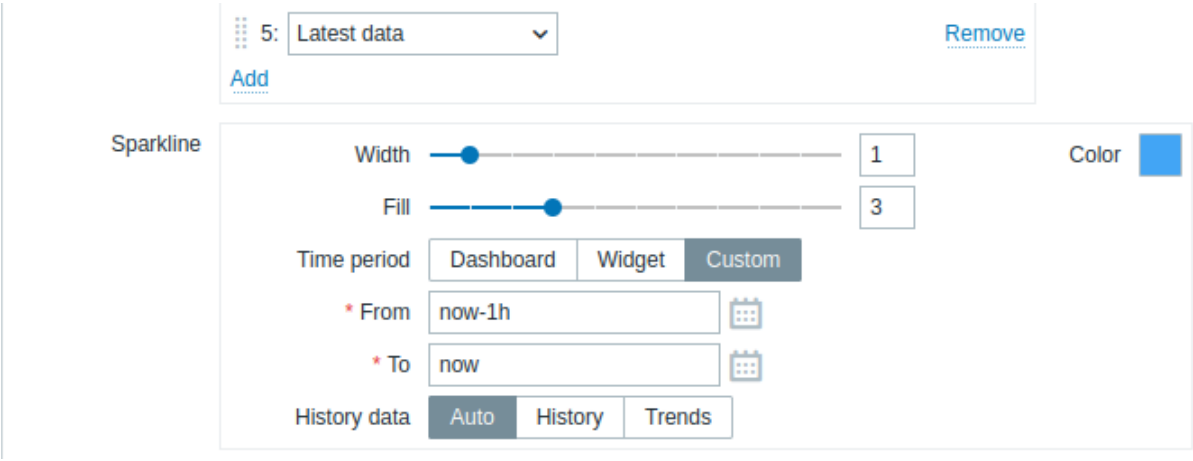
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Vienums	Atlasiet vienumu. Alternatīvi atlasiet saderīgu logrīku kā datu avotu vienumiem. Atbilstoši ieteikumi tiek parādīti rakstīšanas laikā. Logrīka augšējā sadaļā vienmēr tiek parādīts **vienuma nosaukums** (noklikšķinot uz tā, tiek atvērta vienuma izvēlne): - Ja vienums ir atspējots, blakus vienuma nosaukumam tiek parādīta sarkana etiķete *Atspējots*. - Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Problēmu gadījumā katrai problēmas nopietnības pakāpei tiek parādīta kvadrātveida ikona ar problēmu skaitu. Novietojot kursoru virs ikonas, tiek atvērts uznirstošais logs ar detalizētu informāciju. - Veidnes panelī tiek parādīts vienuma tehniskais nosaukums (redzamā nosaukuma vietā). Logrīka augšējā sadaļā tiek parādīts arī **vienuma ceļš** — tā hosts un, ja piemērojams, veidne, zema līmeņa atklāšanas (LLD) kārtula vai galvenais vienums (noklikšķinot uz entītijas, tiek atvērta tās konfigurācija). Vienuma ceļš tiek parādīts tikai lietotājiem ar atļaujām attiecīgajām entītijām.
Rādīt	Pievienojiet sadaļas, kas tiks parādītas logrīkā. Sadaļas var pārkārtot, velkot un nometot.
Apraksts	Parāda vienuma aprakstu. Makrosi tiek atrisināti.
Kļūdas teksts	Parāda vienuma kļūdu (ja tāda ir).
Metrikas	Parāda šādu informāciju: - Vienuma atjaunināšanas intervālu (izņemot trapper vienumus); - Vienuma vēstures konfigurāciju (ja tāda ir); - Vienuma tendenču konfigurāciju (ja tāda ir). Makrosi tiek atrisināti. Ja vienums izmanto pielāgotus intervālus, blakus vienuma atjaunināšanas intervālam tiek parādīta palīdzības ikona. Novietojot kursoru virs ikonas, tiek atvērts uznirstošais logs ar detalizētu informāciju par pielāgotajiem intervāliem.

Jaunākie dati	Parāda šādu informāciju: - Laiku kopš pēdējās vienuma pārbaudes; - Vienuma pēdējo vērtību; - Saiti uz vienuma vērtību vienkāršo grafiku/vēsturi (ja tiek glabāta vēsture vai tendences); - Konfigurējamu sparkline diagrammu (ja vienums ir skaitlisks). Binära tipa vienumiem pēdējās vērtības vietā tiek parādīts sīktēls vai opcija *Rādīt*. Novietojot kursoru virs sīktēla, tiek atvērts uznirstošais logs ar attēlu. Novietojot kursoru virs *Rādīt*, tiek atvērts uznirstošais logs ar vienuma vērtību (Base64 virkne). Veidnes panelī vienuma pēdējās vērtības vietā tiek parādīts *Nav datu*.
Informācijas tips	Parāda vienuma informācijas tipu.
Trigeri	Parāda šādu informāciju: - Etiķeti *Trigeri* ar trigeru skaitu (novietojot kursoru virs *Trigeri*, tiek atvērts uznirstošais logs ar detalizētu informāciju par trigeriem); - Trigeru nosaukumu sarakstu (makrosi tiek atrisināti).
Hosta saskarne	Parāda hosta saskarni, ko izmanto vienums, vai *Nav datu*, ja vienumam nav hosta saskarnes (piemēram, atkarīgajiem vienumiem, vienkāršajām pārbaudēm). Veidnes panelī tiek parādīts *Nav datu*.
Tips	Parāda vienuma tipu.
Hosta inventārs	Parāda hosta inventāra lauku, kuru aizpilda vienuma vērtība.
Birkas	Parāda vienuma birkas.
Pārrakstīt hostu	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Dzirksteles diagramma

Dzirksteles diagrammas konfigurācijas opcijas ir pieejamas tikai tad, ja logrīkam ir pievienota sadaļa *Jaunākie dati*.



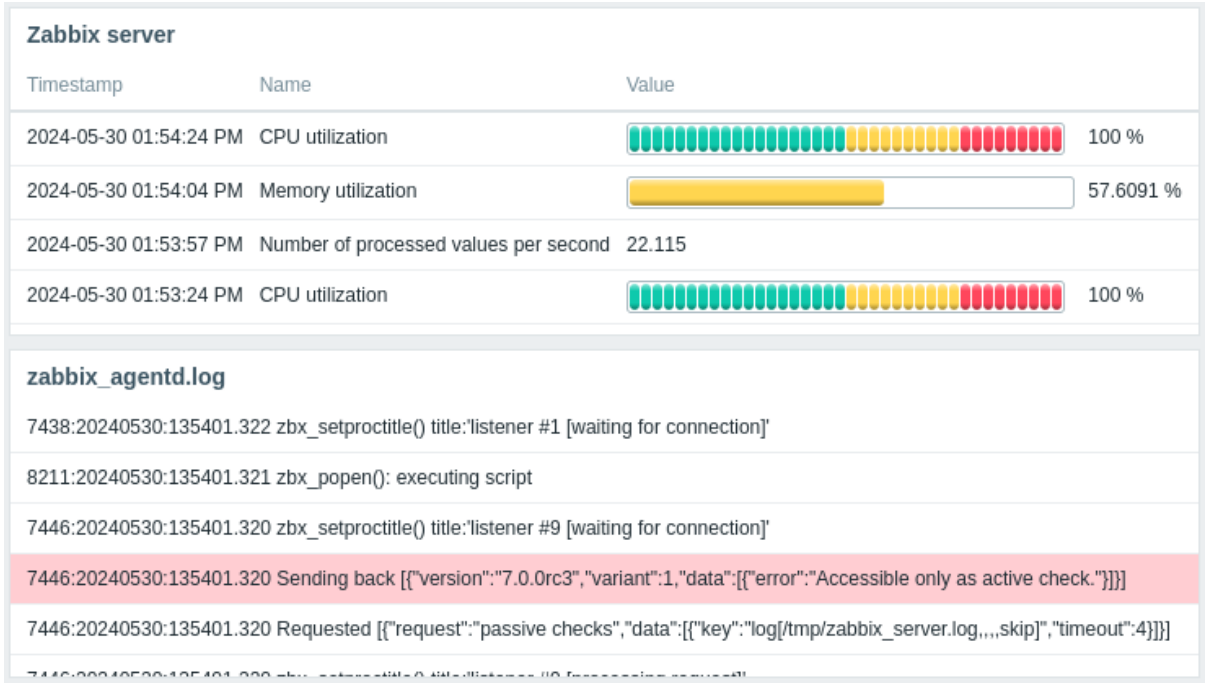
Platums	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Krāsa	Atlasiet līnijas un aizpildījuma krāsu.
Aizpildījums	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Laika periods	Atlasiet laika perioda datu avotu: **Panelis** - izmantot paneļa laika perioda atlasītāju; **Logrīks** - izmantot saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Logrīks”.

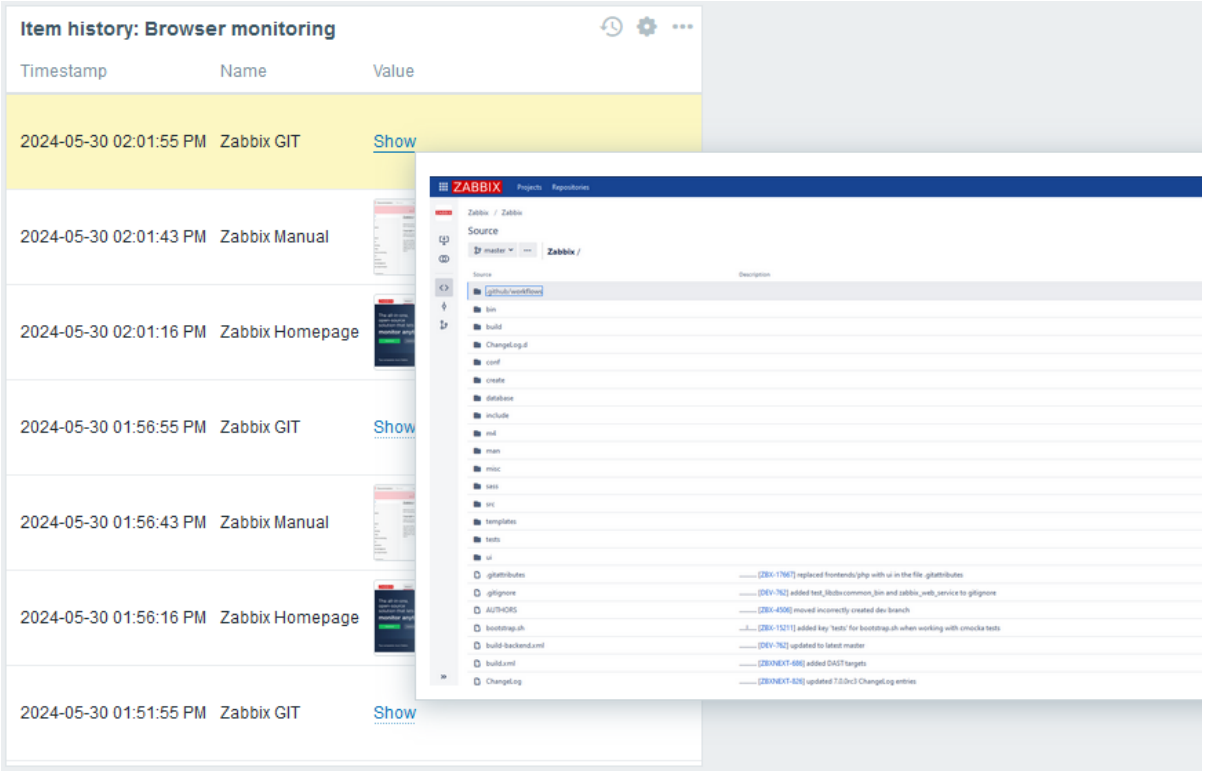
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".
Vēsturiskie dati	Ņemt datus no vēstures vai tendencēm: **Automātiski** - automātiska atlase; **Vēsture** - ņemt vēstures datus; **Tendences** - ņemt tendenču datus. Šis parametrs attiecas tikai uz skaitliskiem datiem. Neskaitliskie dati vienmēr tiks ņemti no vēstures.

16 Vienumu vēsture

Pārskats

Logrīks *Vienumu vēsture* tabulas formātā attēlo jaunākos datus visiem vienumu tipiem (skaitliskiem, teksta u.c.). Tas var rādīt progresa joslas un attēlus (noderīgi pārlūka vienumiem) un var izcelt vērtības (noderīgi žurnālfailu uzraudzībai).

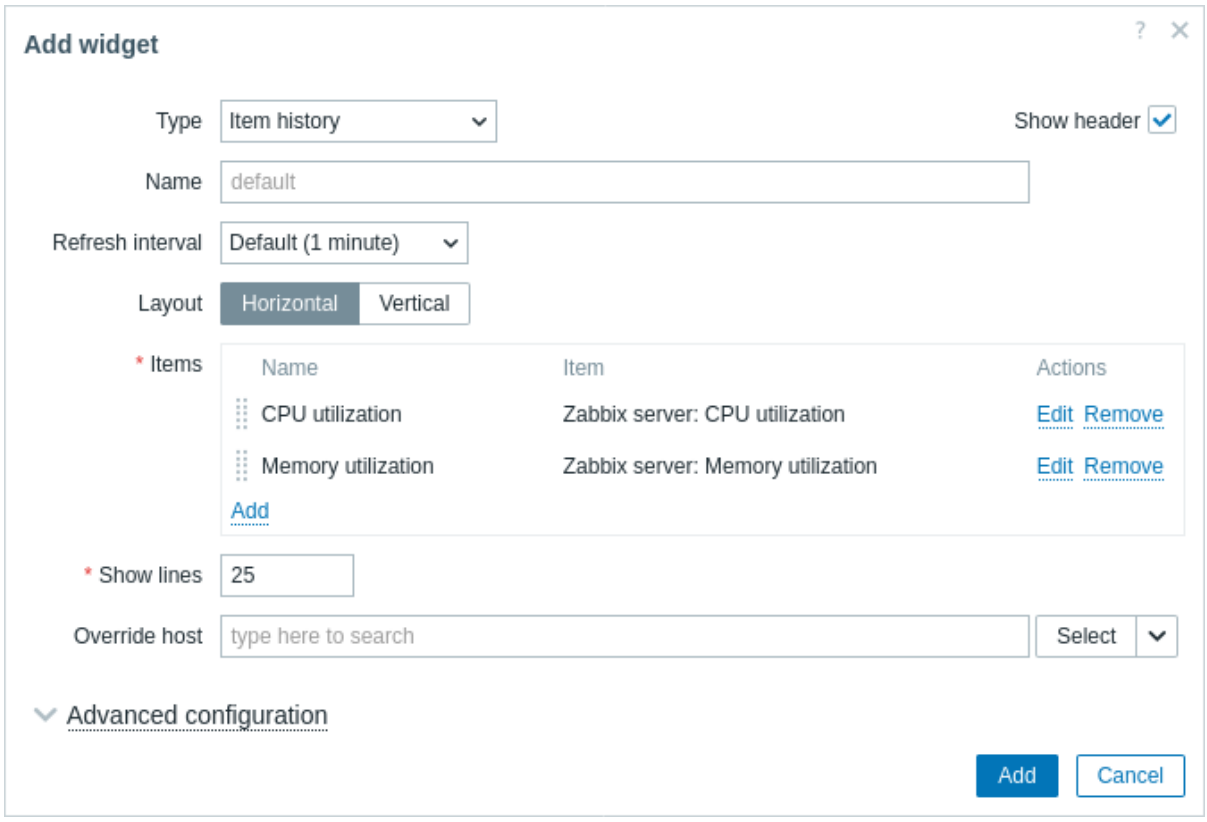




Var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Vienuma vēsture* kā tipu:



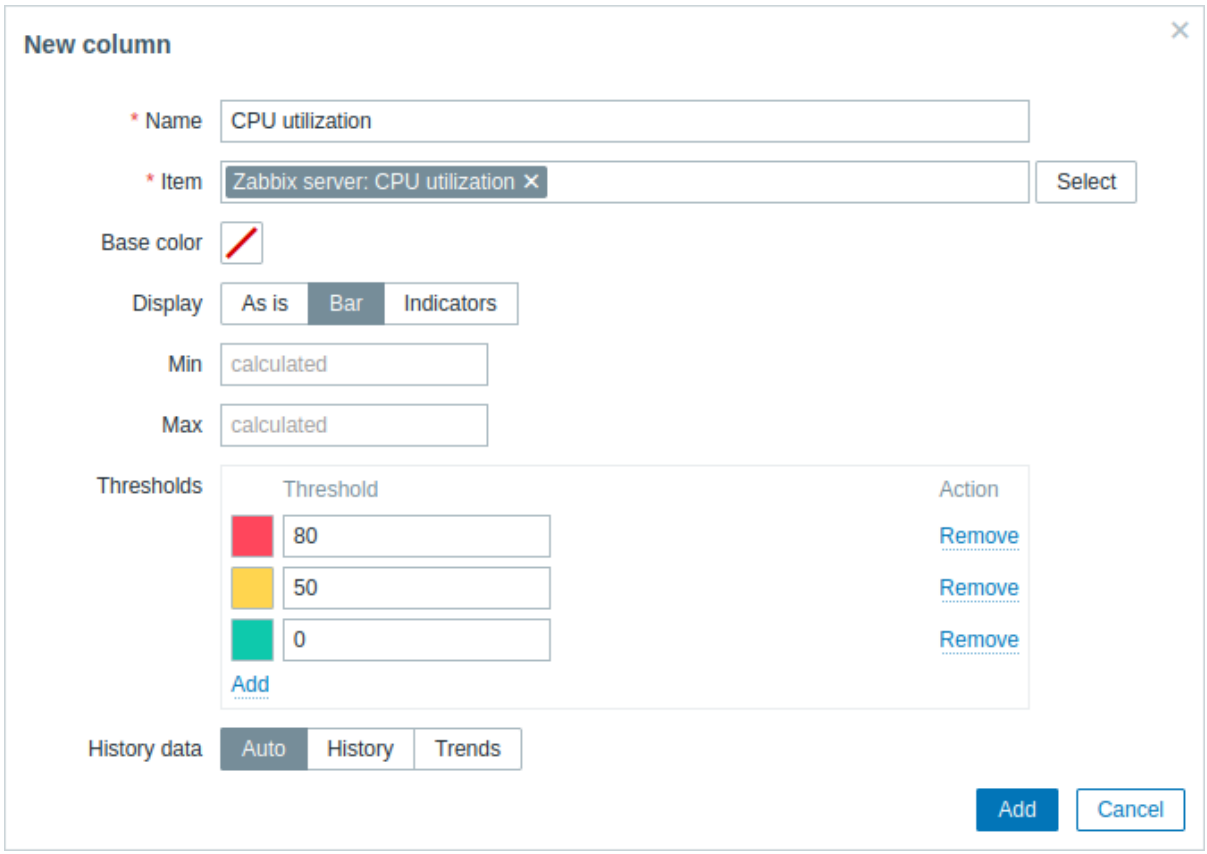
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Izkārtojums	Atlasiet vienumu kolonnu izkārtojuma opciju: **Horizontāli** - vienumi tiks attēloti horizontāli, vērtības vertikāli; **Vertikāli** - vienumi tiks attēloti vertikāli, vērtības horizontāli.
Vienumi	Pievienojiet attēlojamos vienumu kolonnas. Vienumu secība nosaka to attēlošanas secību. Vienumus var pārkārtot, velkot tos uz augšu vai uz leju aiz roktura pirms vienuma nosaukuma.

Rādīt rindas	Norādiet attēlojamo vienuma vērtību rindu skaitu.
Aizstāt hostu	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

Kolonnu konfigurācija

Lai konfigurētu vienumu kolonnas, parametrā *Items* noklikšķiniet uz *Add*:



Biežāk lietotie kolonnas parametri:

Nosaukums	Ievadiet kolonnas nosaukumu. Ja tas ir atstāts tukšs, tiek izmantots vienuma nosaukums no parametra *Vienums*.
Vienums	Atlasiet vienumu. Ņemiet vērā, ka kolonnas konfigurācijas parametri atšķiras atkarībā no atlasītā vienuma informācijas tipa; plašāku informāciju skatiet atsevišķajos parametros zemāk. Konfigurējot logrīku veidnes panelī, var atlasīt tikai vienumus, kas konfigurēti veidnē.
Pamatkrāsa	Atlasiet kolonnas fona krāsu vai aizpildījuma krāsu, ja parametrs *Attēlojums* ir iestatīts uz ”Bar” vai ”Indicators”. Ņemiet vērā, ka pamatkrāsu var pārrakstīt sliekšņa vai izcelšanas krāsas.

Kolonnu parametri, kas raksturīgi skaitliska tipa vienumiem:

Attēlojums	Izvēlieties, kā jāattēlo vienuma vērtība: **Kā ir** — kā parasts teksts; **Josla** — kā vienlaidus, ar krāsu aizpildīta josla; **Indikatori** — kā segmentēta, ar krāsu aizpildīta josla.
Min	Ievadiet minimālo vērtību joslai/indikatoriem. Ja lauks tiek atstāts tukšs, logrīks izmantos vienuma minimālo vērtību. Šis parametrs ir pieejams tikai tad, ja *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”.
Max	Ievadiet maksimālo vērtību joslai/indikatoriem. Ja lauks tiek atstāts tukšs, logrīks izmantos vienuma maksimālo vērtību. Šis parametrs ir pieejams tikai tad, ja *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”.

Sliekšņi	Noklikšķiniet uz *Pievienot*, lai pievienotu slieksni, izvēlieties sliekšņa krāsu no krāsu atlasītāja un norādiet skaitlisku vērtību. Saglabājot sliekšņu saraksts tiks sakārtots augošā secībā. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas arī vērtību kartēšanas.
Vēstures dati	Izvēlieties, vai ņemt datus no vēstures vai tendencēm: **Auto** — automātiska atlase; **Vēsture** — ņemt vēstures datus; **Tendences** — ņemt tendenču datus.

Kolonnu parametri, kas raksturīgi rakstzīmju, teksta, žurnāla un JSON tipa vienumiem:

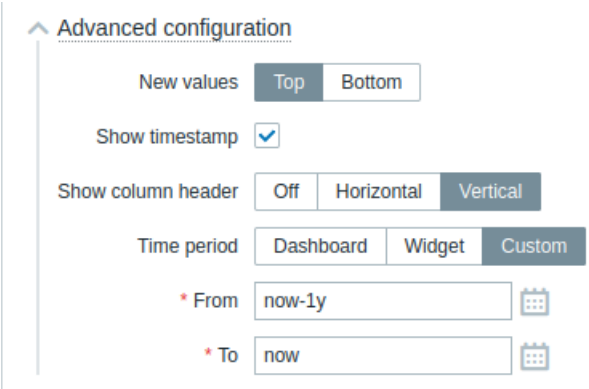
Izcēlumi	Noklikšķiniet uz *Pievienot*, lai pievienotu izcēlumu, atlasiet izcēluma krāsu no krāsu izvēlnes un norādiet regulāro izteiksmi. Atlasītā krāsa tiks izmantota kā fona krāsa vienuma vērtībām, kur norādītā regulārā izteiksme atbilst tekstam.
Attēlošana	Atlasiet, kā jāattēlo vienuma vērtība: **Kā ir** — tieši tā, kā saņemta. Rindu pārtraukumi tiek saglabāti, ja vērtība satur rindas pārneses. Vārdu aplaušana tiek lietota, ja *Izkārtojums* ir iestatīts uz ”Vertikāls” un ir konfigurēta vairāk nekā viena kolonna; **HTML** — kā HTML formatēts teksts; **Viena rinda** — kā viena rinda, saīsināta līdz norādītajam garumam (1–500 rakstzīmes). Novietojot kursoru virs saīsinātās vērtības vai noklikšķinot uz tās, tiek atvērts uznirstošais logs ar pilno vērtību.
Izmantot vienplatuma fontu	Atzīmējiet šo izvēles rūtiņu, lai attēlotu vienuma vērtību vienplatuma fontā (pēc noklusējuma nav atzīmēta).
Attēlot vietējo laiku	Atzīmējiet šo izvēles rūtiņu, lai laikspiedola kolonnā laikspiedola vietā attēlotu vietējo laiku. Ņemiet vērā, ka arī izvēles rūtiņai *Rādīt laikspiedolu* sadaļā paplašinātā konfigurācija jābūt atzīmētai. Šis parametrs ir pieejams tikai žurnāla tipa vienumiem.

Kolonnu parametri, kas raksturīgi binārā tipa vienumiem:

Rādīt sīktēlu	Atzīmējiet šo izvēles rūtiņu, lai attēlotu sīktēlu attēlu binārajiem datiem vai opciju ”Rādīt” binārajiem datiem, kas nav attēli. Noņemiet atzīmi no šīs izvēles rūtiņas, lai attēlotu opciju ”Rādīt” visām bināro vienumu vērtībām. Novietojot kursoru virs opcijas ”Rādīt” vai noklikšķinot uz tās, tiek atvērts uznirstošais logs ar vienuma vērtību (attēlu vai Base64 virkni). Ja vienuma vērtība ir tukša virkne, tiek attēlota opcija ”Rādīt”; novietojot kursoru virs tās vai noklikšķinot uz tās, tiek atvērts uznirstošais logs ar tekstu ”Tukša virkne”.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas sakļaujamajā sadaļā *Papildu konfigurācija*:

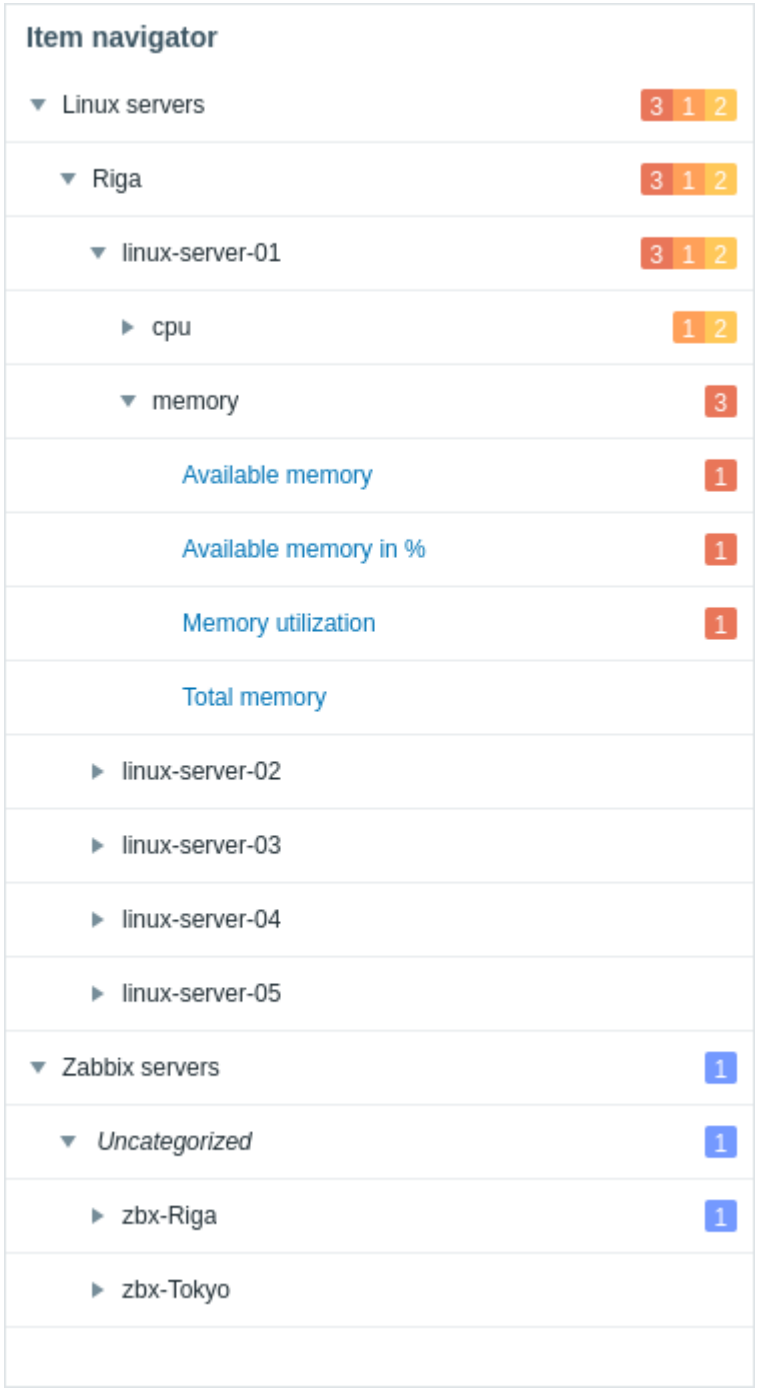


Jaunās vērtības	Izvēlieties, kur jāpievieno jaunās vienumu vērtības: **Augšā** — kolonnu augšdaļā; **Apakšā** — kolonnu apakšdaļā.
Rādīt laikspiedolu	Atzīmējiet šo izvēles rūtiņu, lai parādītu laikspiedola kolonnu (pēc noklusējuma nav atzīmēta).
Rādīt kolonnas galveni	Izvēlieties kolonnas galvenes orientāciju: **Izslēgts** — paslēpt galveni; **Horizontāli** — parādīt galveni horizontāli; **Vertikāli** — parādīt galveni vertikāli.
Laika periods	Izvēlieties datu avotu laika periodam: **Panelis** — izmantot paneļa laika perioda atlasītāju; **Logrīks** — izmantot saderīgu logrīku (norādīts parametrā *Logrīks*); **Pielāgots** — izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki joprojām var to izmantot kā datu avotu laika periodam.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku (*Grafiks*, *Grafiks (klasisks)*, *Grafika prototips*) kā datu avotu laika periodam. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Logrīks”.
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Pielāgots”.
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Pielāgots”.

17 Vienumu navigators

Pārskats

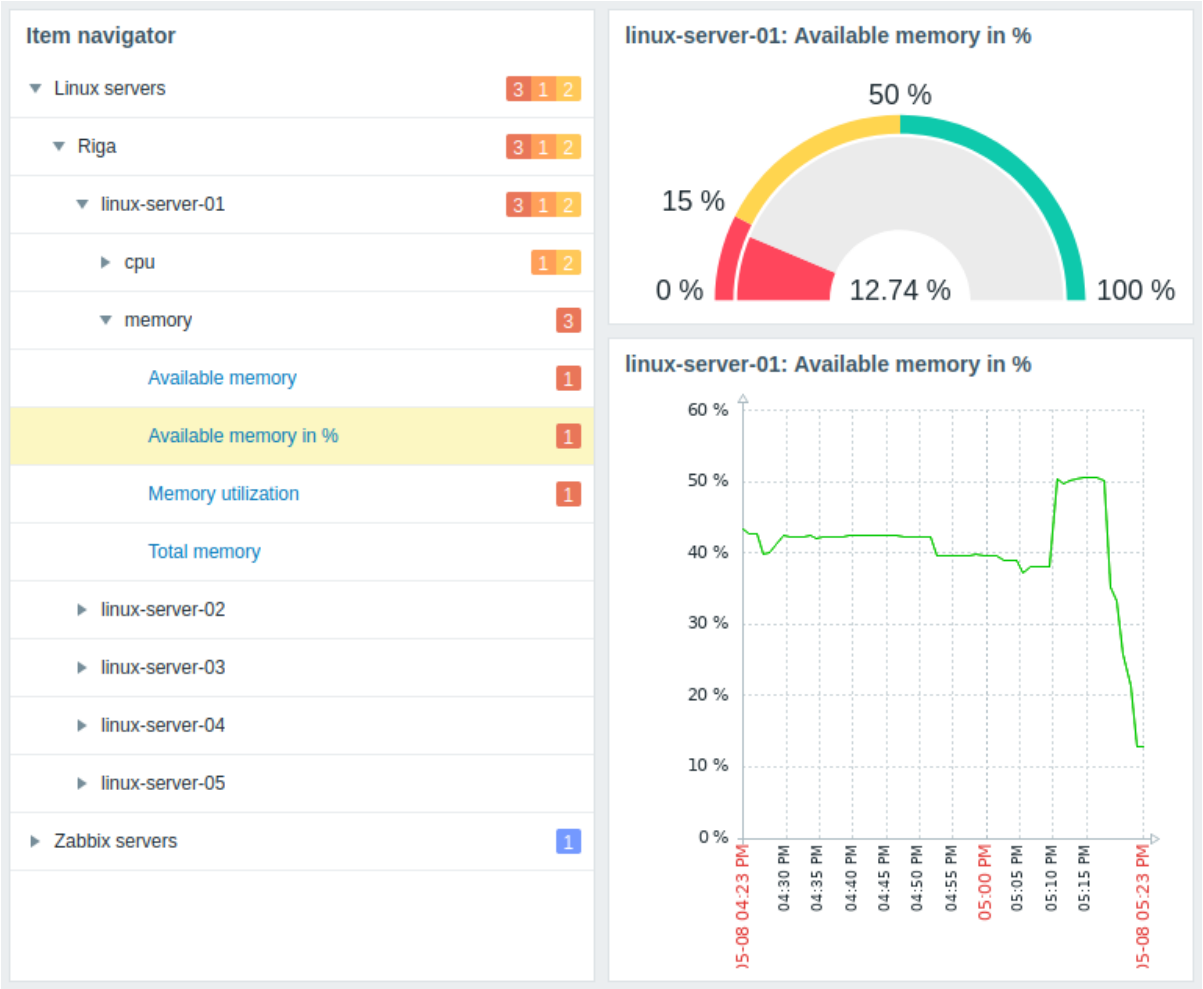
Logrīks *Vienumu navigators* attēlo vienumu sarakstu, pamatojoties uz dažādām filtrēšanas un grupēšanas opcijām.



Grupas, pēc kurām vienumi ir organizēti, var izvērst vai sakļaut.

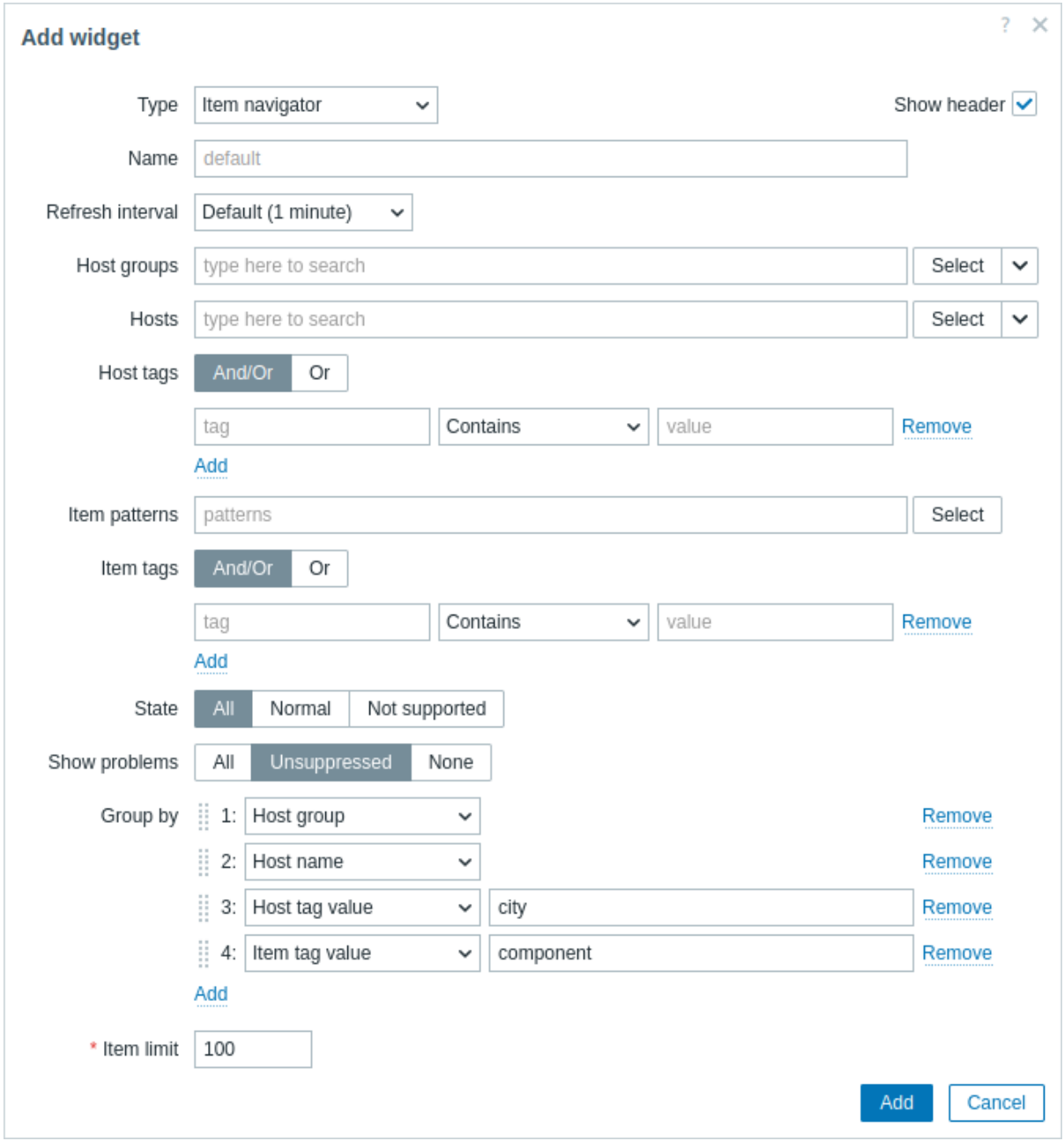
Grupām un problēmām papildu informācija ir pieejama, novietojot peles kursoru virs tām.

Logrīks ir visnoderīgākais, lai kontrolētu, ko attēlo citi logrīki, pamatojoties uz atlasīto vienumu.



Konfigurācija

Lai konfigurētu, atlasiet *Vienumu navigators* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Atlasot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas; ja nav atlasīta neviena hostu grupa, logrīks attēlos vienumus, kas pieder visiem hostiem no visām hostu grupām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošo hostu nolaižamais saraksts. Ja nav atlasīts neviens hosts, logrīks attēlos vienumus, kas pieder visiem hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

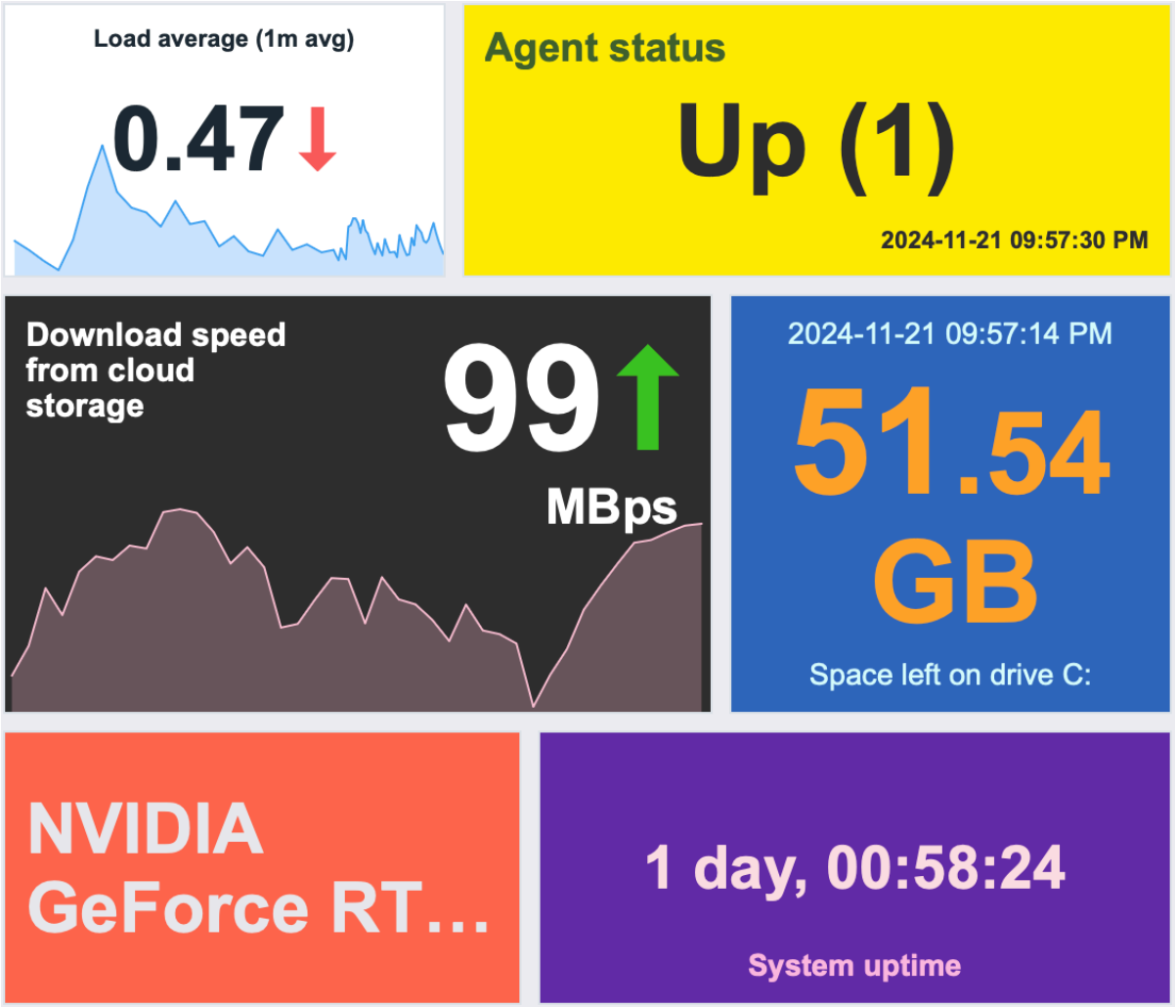
Hostu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos vienumus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** — pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Vienumu šabloni	Ievadiet vienumu šablonus vai atlasiet esošos vienumus kā vienumu šablonus. Vienumi, kas atbilst norādītajiem šabloniem, tiks attēloti vienumu navigatorā. Atlasei var izmantot aizstājējzīmju šablonus (piemēram, `*` atgriezīs vienumus, kas atbilst nullei vai vairāk rakstzīmēm; `Zabbix*` atgriezīs vienumus, kas sākas ar "Zabbix"). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Kad sāksiet rakstīt, nolaižamajā sarakstā tiks parādīti atbilstošie vienumi, aprobežojoties ar tiem, kas pieder atlasītajiem *Hostiem* vai hostiem atlasītajās *Hostu grupās*, ja tādas ir. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item**, ja ir citi atbilstoši vienumi (piemēram, item2, item3). Konfigurējot logrīku veidnes panelī, šis parametrs ļauj atlasīt tikai vienumus, kas konfigurēti veidnē.
Vienumu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos vienumus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** — pietiek, ja ir izpildīts viens nosacījums.
Stāvoklis	Filtrējiet, kurus vienumus attēlot, pamatojoties uz to stāvokli (visi, normāli, neatbalstīti).
Rādīt problēmas	Filtrējiet, kuras problēmas attēlot, pamatojoties uz to statusu (visas, neapspiestas, neviena). Problēmas tiek norādītas ar krāsainiem blokiem vienuma rindas beigās. Krāsas ir balstītas uz problēmas nopietnību, ko var pielāgot sadaļā *Administrēšana* > *Vispārīgi* > *Trigeru attēlošanas opcijas*. Problēmu nopietnības var mainīt, atjauninot problēmas.

Grupēt pēc	Pievienojiet grupēšanas atribūtu, pēc kura grupēt vienumus: **Hostu grupa** — grupēt vienumus pēc to hostu hostu grupām; **Hosta nosaukums** — grupēt vienumus pēc to hostiem; **Hosta birkas vērtība** — ievadiet birkas nosaukumu, lai grupētu vienumus pēc šīs hosta birkas vērtībām (piemēram, ievadiet ”city”, lai grupētu vienumus pēc vērtībām ”Riga”, ”Tokyo” utt.); **Vienuma birkas vērtība** — ievadiet birkas nosaukumu, lai grupētu vienumus pēc šīs vienuma birkas vērtībām (piemēram, ievadiet ”component”, lai grupētu vienumus pēc vērtībām ”cpu”, ”memory” utt.). Grupēšanas atribūtus var pārkārtot, velkot uz augšu vai uz leju aiz roktura pirms grupas nosaukuma. Ņemiet vērā, ka grupēšanas atribūtu secība nosaka grupu ligzdošanas secību. Piemēram, norādot vairākus hosta birku nosaukumus (1: color, 2: city), vienumi vispirms tiks grupēti pēc color (red, blue utt.) un pēc tam pēc city (Riga, Tokyo utt.). Vienums var tikt attēlots vairākās grupās atkarībā no konfigurētajiem grupēšanas atribūtiem (piemēram, grupējot pēc hostu grupas un vienuma hosts pieder vairākām hostu grupām). Noklikšķinot uz šādiem vienumiem, tie tiek atlasīti un izcelti visās grupās. Vienumi, kas neatbilst konfigurētajiem grupēšanas atribūtiem, tiek attēloti grupā *Nekategorizēts*. Ja *Rādīt problēmas* ir konfigurēts problēmu attēlošanai, tās tiek attēlotas šādi: - katrai grupai tiek attēlots visu pakārtoto vienumu problēmu skaits; - katram vienumam tiek attēlots tikai tā problēmu skaits. Var norādīt līdz 10 grupēšanas atribūtiem, un tiem visiem jābūt unikāliem. Ja nav norādīts neviens grupēšanas atribūts, vienumi netiks grupēti.
Vienumu limits	Ievadiet maksimālo attēlojamo vienumu skaitu. Iespējamās vērtības ir diapazonā no 1 līdz 9999. Ja attēlošanai ir pieejams vairāk vienumu nekā iestatītais limits, zem attēlotajiem vienumiem tiek parādīts atbilstošs ziņojums (piemēram, ”100 no 100+ vienumiem ir parādīti”). Ņemiet vērā, ka konfigurētais vienumu limits ietekmē arī konfigurēto grupu attēlojumu; piemēram, ja vienumu limits ir iestatīts uz 100 un vienumi ir grupēti pēc to hostiem (katrs satur 200 vienumus), logrīkā tiks attēlots tikai pirmais hosts ar saviem 100 vienumiem. Šo parametru neietekmē parametrs *Meklēšanas un filtrēšanas rezultātu limits* sadaļā *Administrēšana* → *Vispārīgi* → *GUI*.

18 Vienuma vērtība

Pārskats

Logrīks *Vienuma vērtība* parāda viena skaitliska vai virknes vienuma vērtību. Tas ir noderīgs, lai sekotu svarīgiem rādītājiem, vizualizētu sliekšņus un noteiktu pēkšņas datu izmaiņas.



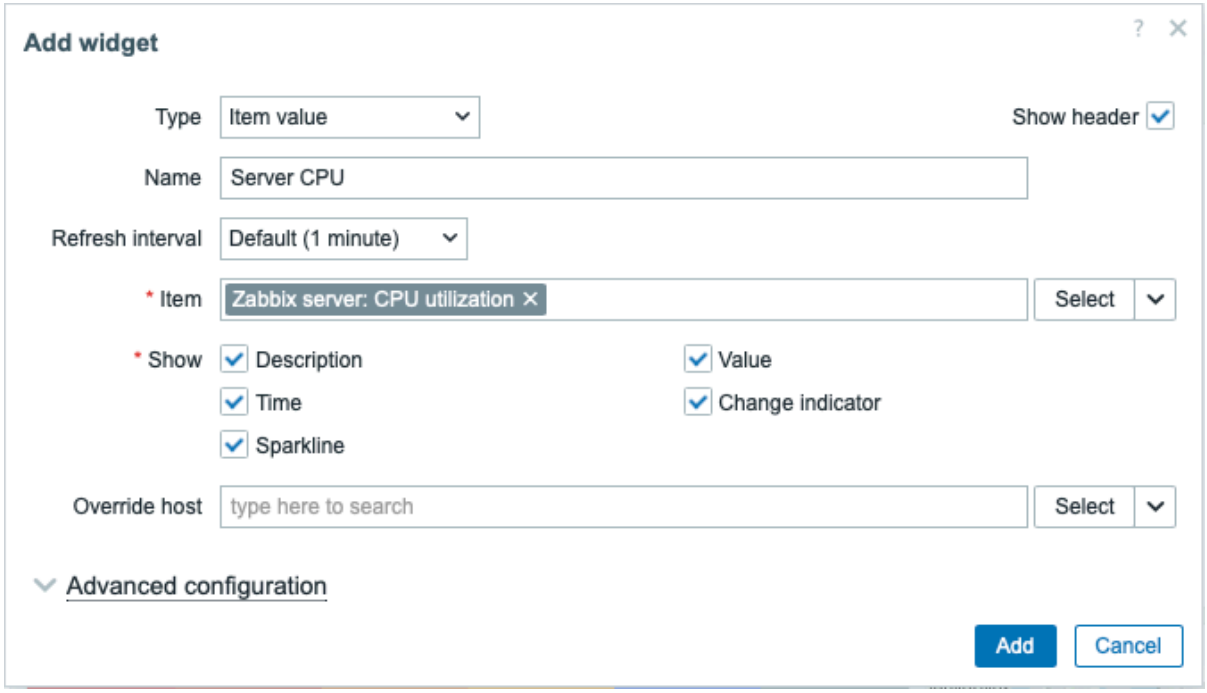
Varat konfigurēt logrīku, lai tas parādītu tālāk norādīto:

- Metrikas laikspiedolu (2024-11-21 09:57:30)
- Vienuma aprakstu (Vidējā slodze (1m), Aģenta statuss)
- Vienuma vērtību, izmaiņu indikatoru un mērvienības (0.47 ↓, 99 ↑ Mbps)
- Fona krāsu
- Dzirksteļlīnijas diagrammu vērtībām no norādītā laika perioda

Noklikšķinot uz logrīka, skaitliskajiem vienumiem tiek atvērts ad-hoc graph, bet virknes vienumiem — jaunākie dati.

Konfigurācija

Lai konfigurētu, atlasiet *Vienuma vērtība* kā logrīka tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Vienums	Atlasiet vienumu. Alternatīvi atlasiet saderīgu logrīku kā vienuma datu avotu. Vienumi, kas atgriež bināros datus, netiek atbalstīti.
Rādīt	Atzīmējiet izvēles rūtiņu, lai parādītu attiecīgo elementu: apraksts, vērtība, laiks, izmaiņu indikators, dzirksteļlīnija. Noņemiet atzīmi, lai paslēptu. Jābūt atlasītam vismaz vienam elementam.
Aizstāt hostu	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas.

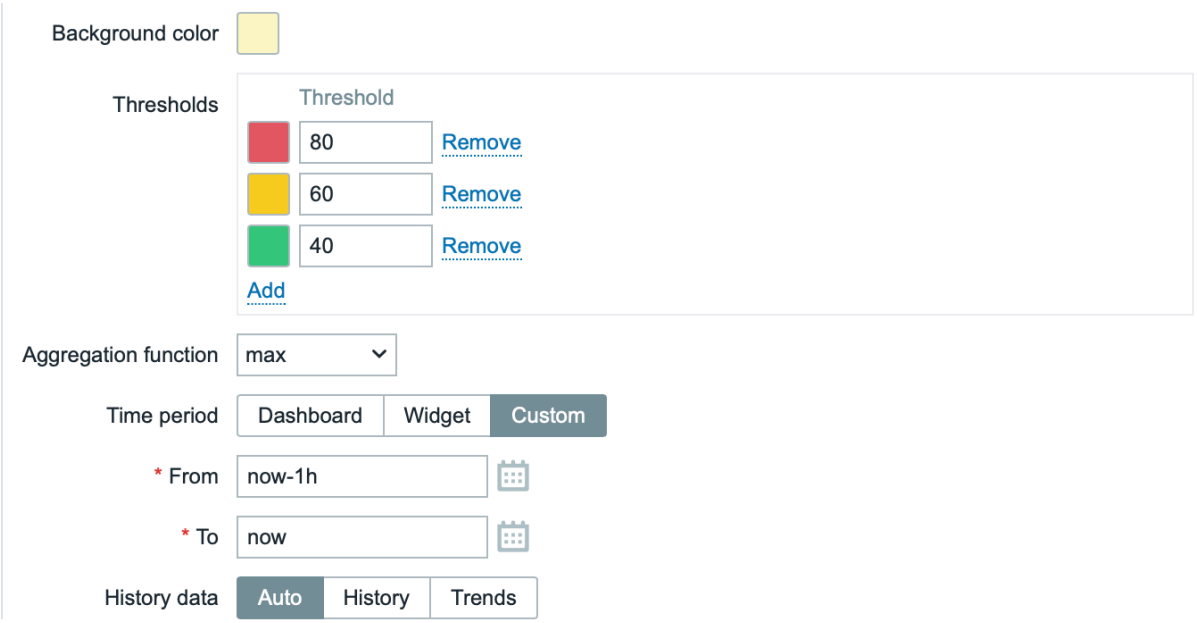
Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas sakļaujamajā sadaļā *Papildu konfigurācija*, un tikai tiem elementiem, kas ir atlasīti laukā *Rādīt* (skatīt iepriekš).

Ņemiet vērā, ka vairāki elementi nevar aizņemt vienu un to pašu vietu; ja tie ir novietoti vienā un tajā pašā vietā, tiks parādīts kļūdas ziņojums.

Vispārīgie logrīka parametri

Šie parametri nosaka fona krāsu (statisku vai dinamisku) visam logrīkam un agregācijas funkciju vērtību attēlošanai.

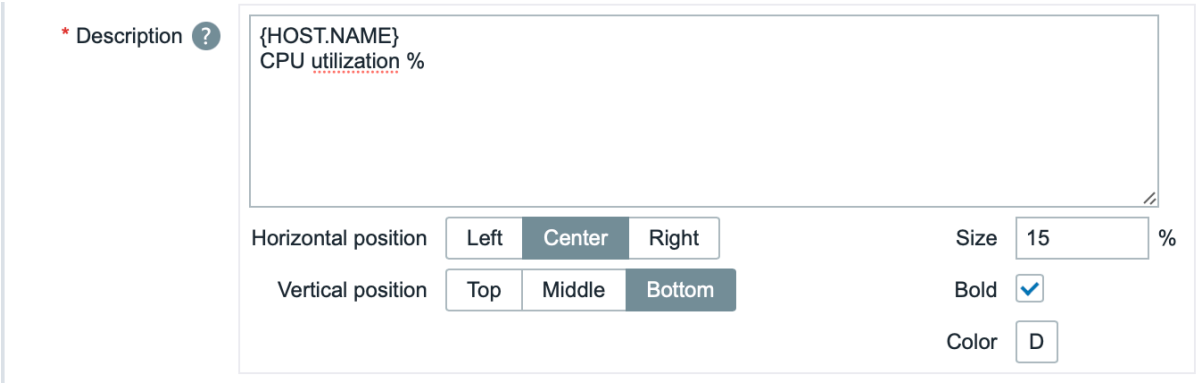


Fona krāsa	Atlasiet fona krāsu visam logrīkam no krāsu izvēlētāja. D apzīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu izvēlētājā noklikšķiniet uz pogas *Use default*.
Sliekšņi	Konfigurējiet dinamisko fona krāsu visam logrīkam. Noklikšķiniet uz *Add*, lai pievienotu slieksni, atlasiet fona krāsu no krāsu izvēlētāja un norādiet skaitlisku vērtību. Kad vienuma vērtība būs vienāda ar sliekšņa vērtību vai lielāka par to, fona krāsa mainīsies. Saglabājot saraksts tiks sakārtots augošā secībā. Ņemiet vērā, ka dinamiskā fona krāsa tiks attēlota pareizi tikai skaitliskiem vienumiem.
Agregācijas funkcija	Norādiet, kuru agregācijas funkciju izmantot atlasītajā *Laika periodā*: **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **count** - attēlot vērtību skaitu; **sum** - attēlot vērtību summu; **first** - attēlot pirmo vērtību; **last** - attēlot pēdējo vērtību. Ja ir atlasīts **not used**, tiek attēlota jaunākā vērtība (bez agregācijas). Agregācija ļauj attēlot agregētu vērtību izvēlētajam intervālam (5 minūtes, stunda, diena), nevis jaunāko vērtību. *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Funkcijai *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Laika periods	Laika periods, ko izmantot vērtību agregēšanai. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (iestatīts parametrā *Logrīks*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks attēlota pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu. Šis parametrs ir paslēpts, ja *Agregācijas funkcija* ir iestatīta uz ”not used”.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Widget”.
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Custom”.
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Custom”.

Vēstures dati	Ņemt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - ņemt vēstures datus; **Trends** - ņemt tendenču datus. Šis iestatījums attiecas tikai uz skaitliskiem datiem. Neskaitliskie dati vienmēr tiks ņemti no vēstures.

Apraksts

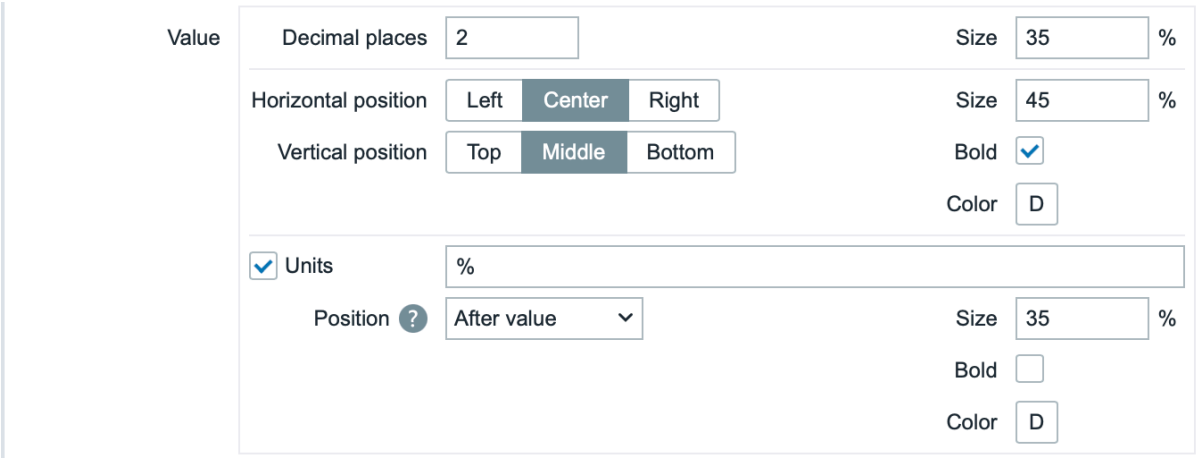
Šie parametri nosaka, kā jāattēlo vienuma apraksts.



Apraksts	Ievadiet vienuma aprakstu. Šis apraksts var aizstāt noklusējuma vienuma nosaukumu. Tiek atbalstīti vairākrindu apraksti. Iespējama teksta un atbalstīto makrosu kombinācija. {HOST.*}, {ITEM.*}, {INVENTORY.*} un lietotāja makrosi tiek atbalstīti.
Horizontālā pozīcija	Atlasiet vienuma apraksta horizontālo pozīciju — pa kreisi, pa labi vai centrā.
Vertikālā pozīcija	Atlasiet vienuma apraksta vertikālo pozīciju — augšā, apakšā vai vidū.
Izmērs	Ievadiet vienuma apraksta fonta izmēra augstumu (procentos attiecībā pret kopējo logrīka augstumu).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai vienuma aprakstu attēlotu treknrakstā.
Krāsa	Atlasiet vienuma apraksta krāsu krāsu atlasītājā. D apzīmē noklusējuma krāsu (atkarībā no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.

Vērtība

Šie parametri nosaka, kā jāattēlo vienuma vērtība.

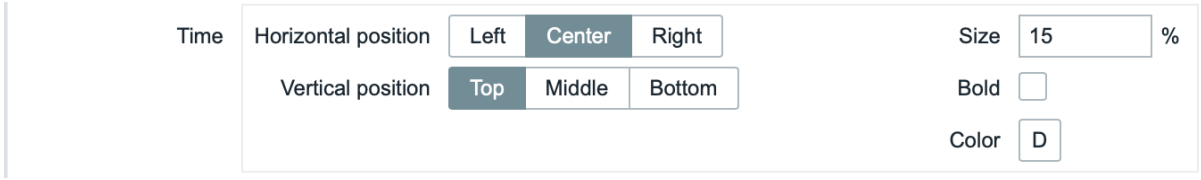


Decimālvietas	Izvēlieties, cik daudz decimālvietu tiks attēlots kopā ar vērtību. Šī vērtība ietekmēs tikai peldošā komata vienumus. Vienumiem, kuriem mērvienības ir iestatītas uz ”s”, ja šī vērtība ir lielāka par 0 (noklusējuma vērtība ir 2), logrīks noapaļo nozīmīgāko laika vienību un attēlo skaitlisku vērtību (piemēram, ”10.43m”), nevis pārveido vērtību par pilnu laika virkni. Ja iestatīts 0, sekundes tiek pārveidotas par cilvēkam lasāmu laika virkni (piemēram, ”4h 56m 30s”).

Izmērs	Ievadiet fonta izmēra augstumu decimālvietām (procentos attiecībā pret kopējo logrīka augstumu).
Horizontālā pozīcija	Izvēlieties vienuma vērtības horizontālo pozīciju — pa kreisi, pa labi vai centrā.
Vertikālā pozīcija	Izvēlieties vienuma vērtības vertikālo pozīciju — augšā, apakšā vai vidū.
Izmērs	Ievadiet fonta izmēra augstumu vienuma vērtībai (procentos attiecībā pret kopējo logrīka augstumu). Ņemiet vērā, ka vienuma vērtības izmēram ir prioritāte; citiem elementiem jāatvēl vieta vērtībai. Tomēr izmaiņu indikatoram, ja vērtība ir pārāk liela, tā tiks saīsināta, lai parādītu izmaiņu indikatoru.
Treknraksts	Atzīmējiet izvēles rūtiņu, lai attēlotu vienuma vērtību treknrakstā.
Krāsa	Izvēlieties vienuma vērtības krāsu no krāsu atlasītāja. D nozīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.
Mērvienības	
Mērvienības	Atzīmējiet izvēles rūtiņu, lai attēlotu mērvienības kopā ar vienuma vērtību. Ja ievadīsiet mērvienības nosaukumu, tas aizstās mērvienību no vienuma konfigurācijas.
Pozīcija	Izvēlieties vienuma mērvienības pozīciju — virs, zem, pirms vai pēc vērtības.
Izmērs	Ievadiet fonta izmēra augstumu vienuma mērvienībai (procentos attiecībā pret kopējo logrīka augstumu).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai attēlotu vienuma mērvienību treknrakstā.
Krāsa	Izvēlieties vienuma mērvienības krāsu no krāsu atlasītāja. D nozīmē noklusējuma krāsu (atkarīga no lietotāja saskarnes tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.

Laiks

Šie parametri nosaka, kā jāattēlo laiks (pulksteņa vērtība no vienuma vēstures).



Horizontālā pozīcija	Izvēlieties laika horizontālo pozīciju — pa kreisi, pa labi vai centrā.
Vertikālā pozīcija	Izvēlieties laika vertikālo pozīciju — augšā, apakšā vai vidū.
Izmērs	Ievadiet laika fonta augstumu (procentos attiecībā pret kopējo logrīka augstumu).
Treknraksts	Atzīmējiet izvēles rūtiņu, lai attēlotu laiku treknrakstā.
Krāsa	Izvēlieties laika krāsu no krāsu atlasītāja. D apzīmē noklusējuma krāsu (atkarībā no lietotāja saskarne tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.

Izmaiņu indikators

Šajā sadaļā var atlasīt izmaiņu indikatoru krāsu no krāsu atlasītāja.

Izmaiņu indikatori ir šādi:

- ↑ - vienuma vērtība palielinās (skaitliskiem vienumiem)
- ↓ - vienuma vērtība samazinās (skaitliskiem vienumiem)
- ↕ - vienuma vērtība ir mainījusies (teksta vienumiem un vienumiem ar vērtību kartēšanu)

Piezīme:

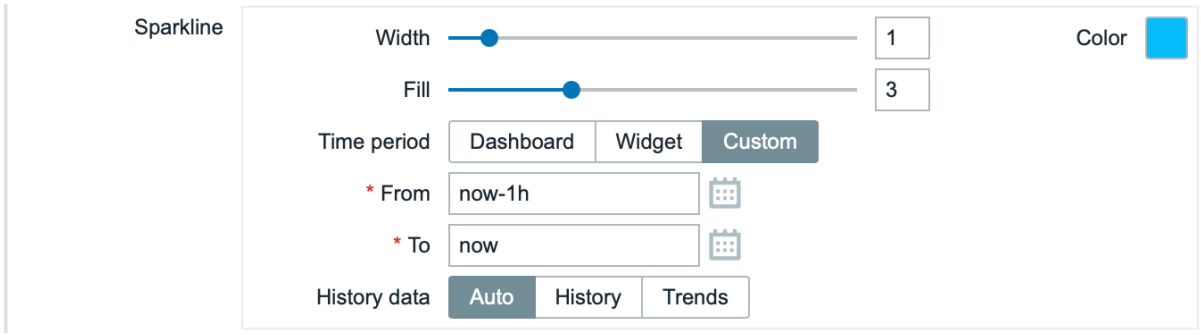
- Pieauguma (↑) un samazinājuma (↓) indikatori netiek rādīti, ja ir tikai viena vērtība.
- Vērtības vienmēr tiek salīdzinātas vienā un tajā pašā tvērumā; piemēram:
  - Jaunākā vērtība tiek salīdzināta ar iepriekšējo vērtību.
  - Mēneša vērtība tiek salīdzināta ar iepriekšējo mēnesi.
  - Agregācijām iepriekšējam periodam ir tāds pats ilgums kā atlasītajam periodam, un tas beidzas tieši pirms atlasītā perioda sākuma.

D apzīmē noklusējuma krāsu (atkarībā no lietotāja saskarne tēmas). Lai atgrieztos pie noklusējuma vērtības, krāsu atlasītājā noklikšķiniet uz pogas *Use default*.

Izmaiņu indikatora vertikālais izmērs ir vienāds ar vērtības izmēru (vērtības veselo daļu skaitliskiem vienumiem).

Sparkline

Šie parametri nosaka, kā jāattēlo sparkline diagramma.

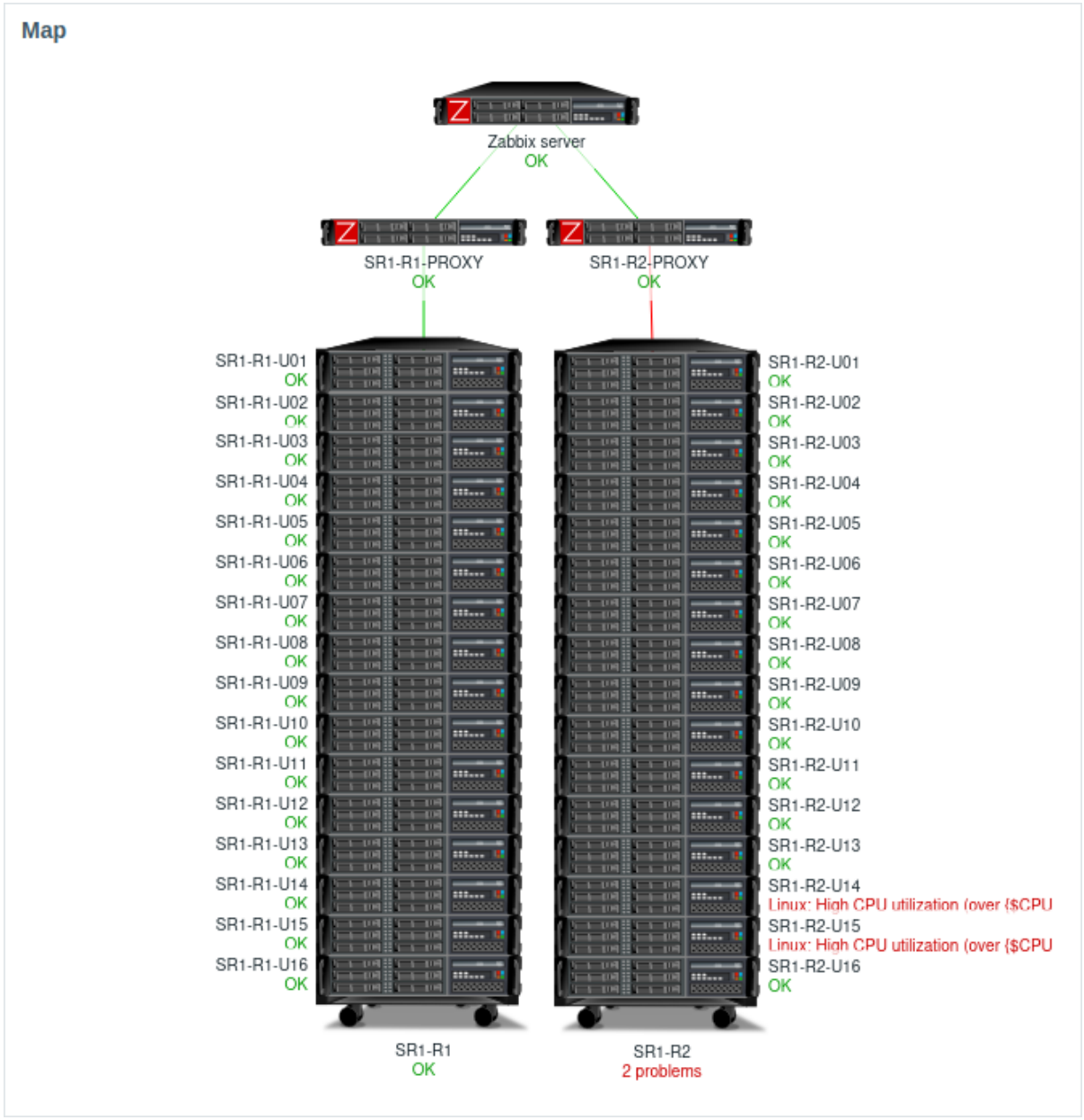


Width	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Color	Atlasiet līnijas un aizpildījuma krāsu.
Fill	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Time period	Norādiet laika periodu vērtībām, kas jāiekļauj sparkline diagrammā. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (norādīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgie logrīki joprojām var to izmantot kā laika perioda datu avotu. Šis parametrs nav pieejams, ja *Aggregation function* ir iestatīta uz ”not used”.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Widget”.
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz ”Custom”.
History data	Izmantot datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - izmantot vēstures datus; **Trends** - izmantot tendenču datus.

19 Karte

Pārskats

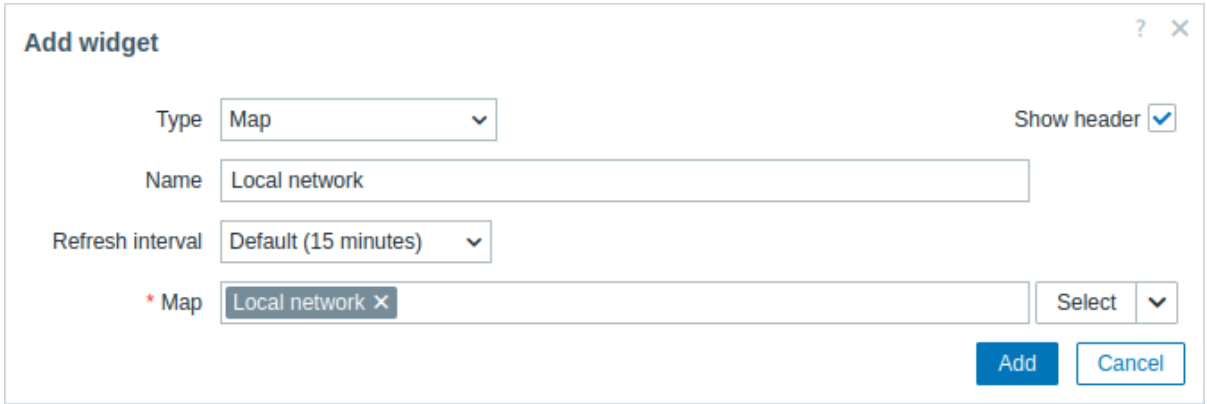
Logrīks *Karte* var attēlot tīkla karti vai līdzīgu vizualizāciju, sniedzot jums dinamisku jūsu tīkla pārskatu.



Varat to savienot pārī ar logrīku Kartes navigācijas koks, lai dinamiski attēlotu kokā atlasīto karti.

Konfigurācija

Lai konfigurētu, atlasiet *Map* kā tipu:



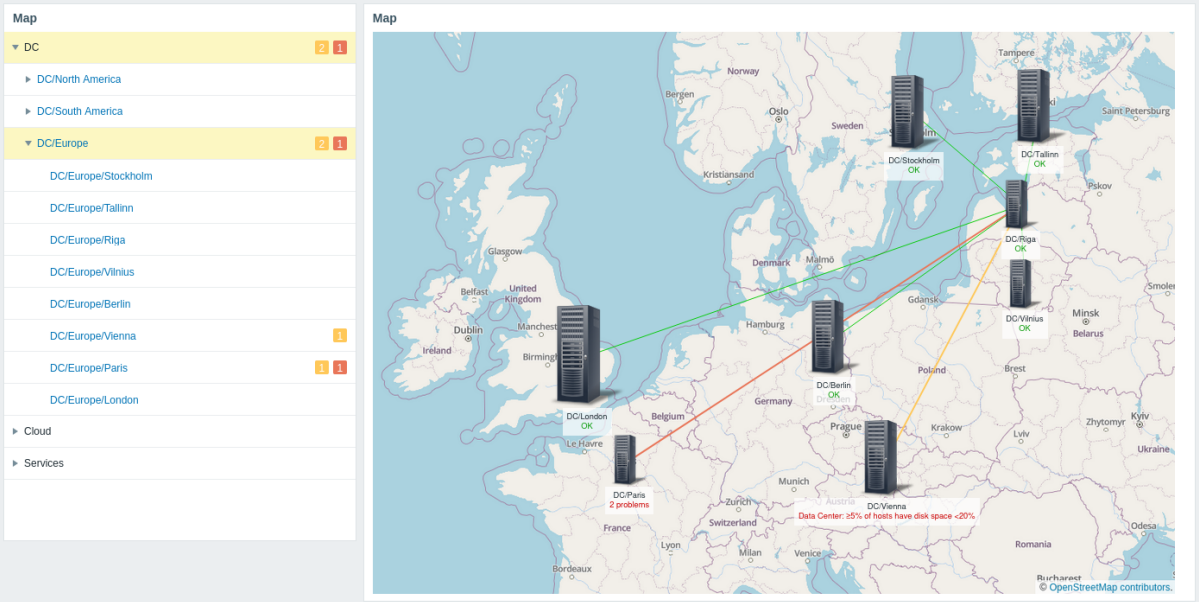
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Map	Iestatiet karti attēlošanai. Alternatīvi atlasiet saderīgu logrīku kā kartes datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt kartes vai logrīka nosaukumu, tiks piedāvāts atbilstošu karšu vai logrīku nolaižamais saraksts.

20 Kartes navigācijas koks

Pārskats

Logrīks *Kartes navigācijas koks* attēlo esošo karšu hierarhiju un problēmu skaitu katrai kartei un karšu grupai.

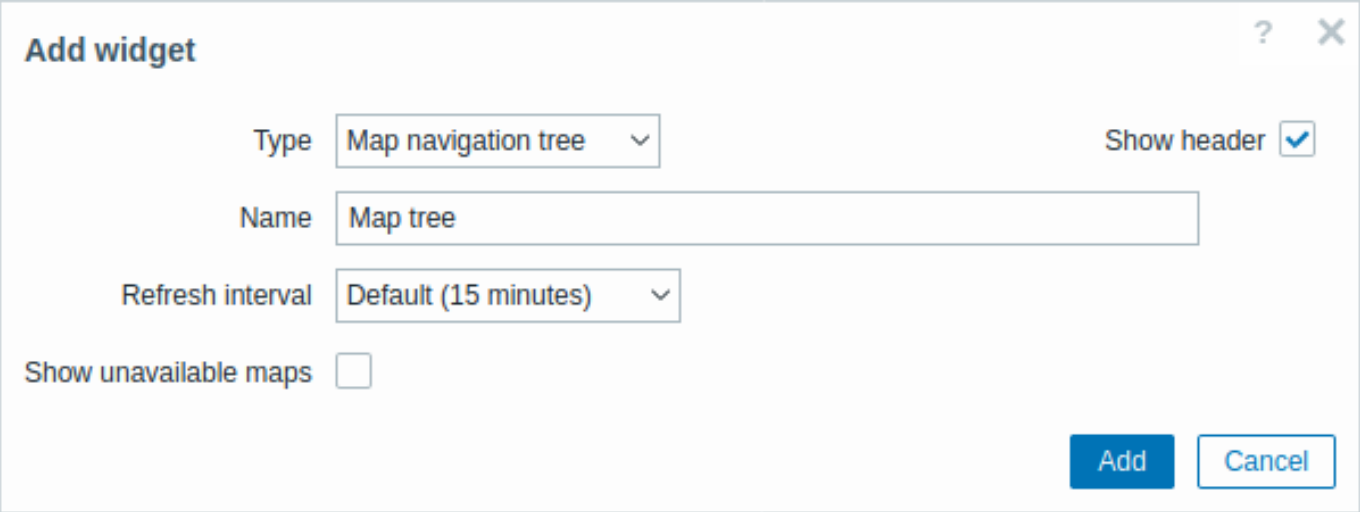


To var savienot pārī ar logrīku Karte, lai dinamiski attēlotu kokā atlasīto karti.

Hierarhijas augstākā līmeņa kartes problēmu skaits ietver visas apakškaršu problēmas, kā arī pašas kartes problēmas.

Konfigurācija

Lai konfigurētu, atlasiet *Kartes navigācijas koks* kā tipu:

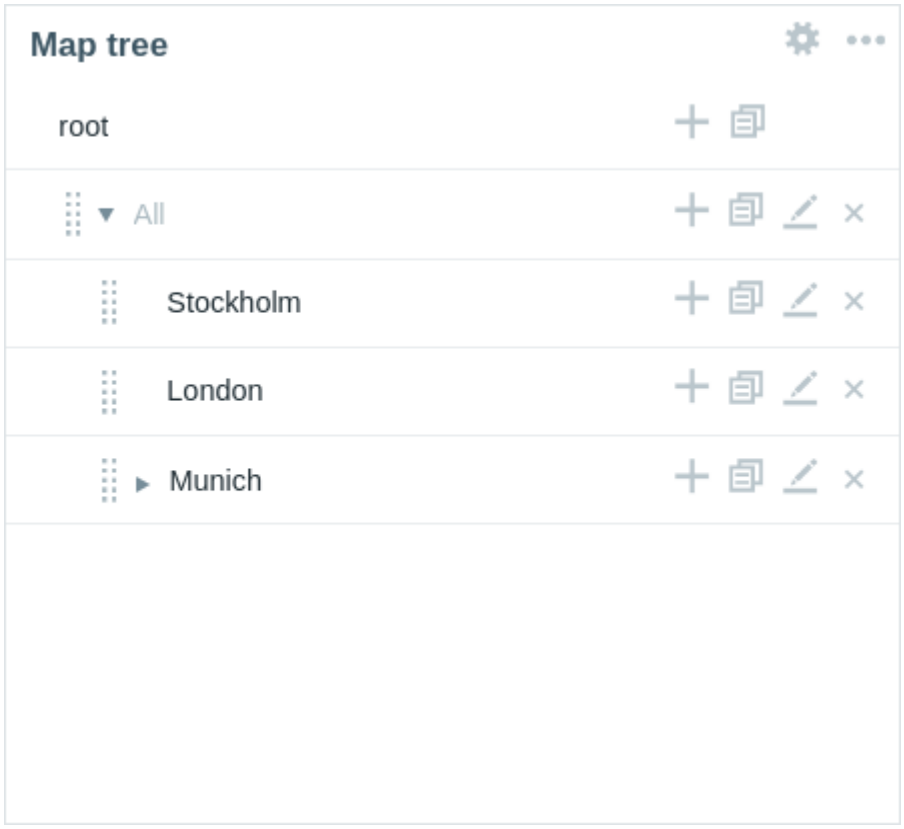


Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Rādīt nepieejamās kartes	Atzīmējiet šo izvēles rūtiņu, lai parādītu kartes, kurām lietotājam nav lasīšanas atļaujas. Nepieejamās kartes navigācijas kokā tiks attēlotas ar pelēkotu ikonu. Ņemiet vērā, ka, ja šī izvēles rūtiņa ir atzīmēta, pieejamās apakškartes tiek parādītas pat tad, ja vecākā līmeņa karte nav pieejama. Ja tā nav atzīmēta, pieejamās apakškartes nepieejamai vecākajai kartei vispār netiks parādītas. Problēmu skaits tiek aprēķināts, pamatojoties uz pieejamajām kartēm un pieejamajiem karšu elementiem.

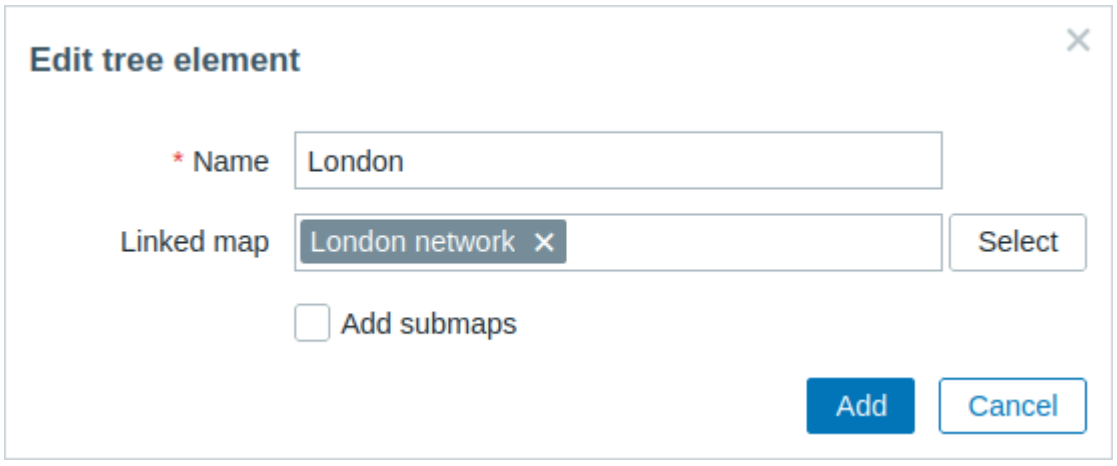
Navigācijas koka elementi tiek parādīti sarakstā. Jūs varat:

- vilkt elementu (ieskaitot tā bērnelementus) uz jaunu vietu sarakstā;
- izvērst vai sakļaut elementu, lai parādītu vai paslēptu tā bērnelementus;
- pievienot elementam bērnelementu (ar vai bez saistītās kartes);
- pievienot elementam vairākus bērnelementus (ar saistītām kartēm);
- rediģēt elementu;
- noņemt elementu (ieskaitot tā bērnelementus).



Elementa konfigurācija

Lai konfigurētu navigācijas koka elementu, pievienojiet jaunu elementu vai rediģējiet esošu elementu.



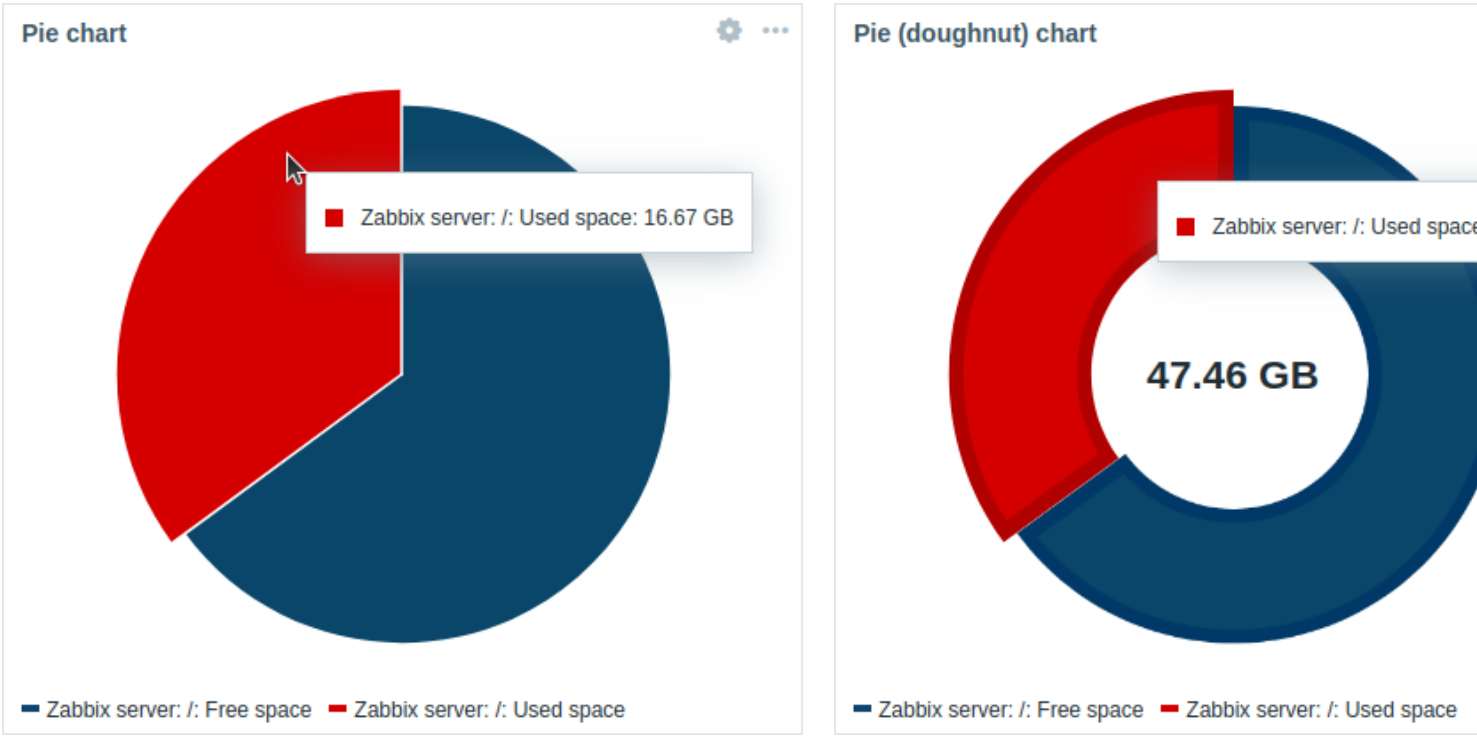
Ir pieejami šādi navigācijas koka elementa konfigurācijas parametri:

Name	Ievadiet navigācijas koka elementa nosaukumu.
Linked map	Atlasiet karti, ko saistīt ar navigācijas koka elementu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt kartes nosaukumu, tiks piedāvāts atbilstošu karšu nolaižamais saraksts.
Add submaps	Atzīmējiet šo izvēles rūtiņu, lai saistītās kartes apakškartes pievienotu kā navigācijas koka elementa bērnelementus.

21 Sektoru diagramma

Pārskats

Pīrāga diagrammas logrīks attēlo skaitliskus vienumu datus kā vektorgrafisku pīrāga vai gredzenveida diagrammu. Šis logrīks var palīdzēt vizualizēt, kā vienumi vai hosti veido kopējo datu kopu.

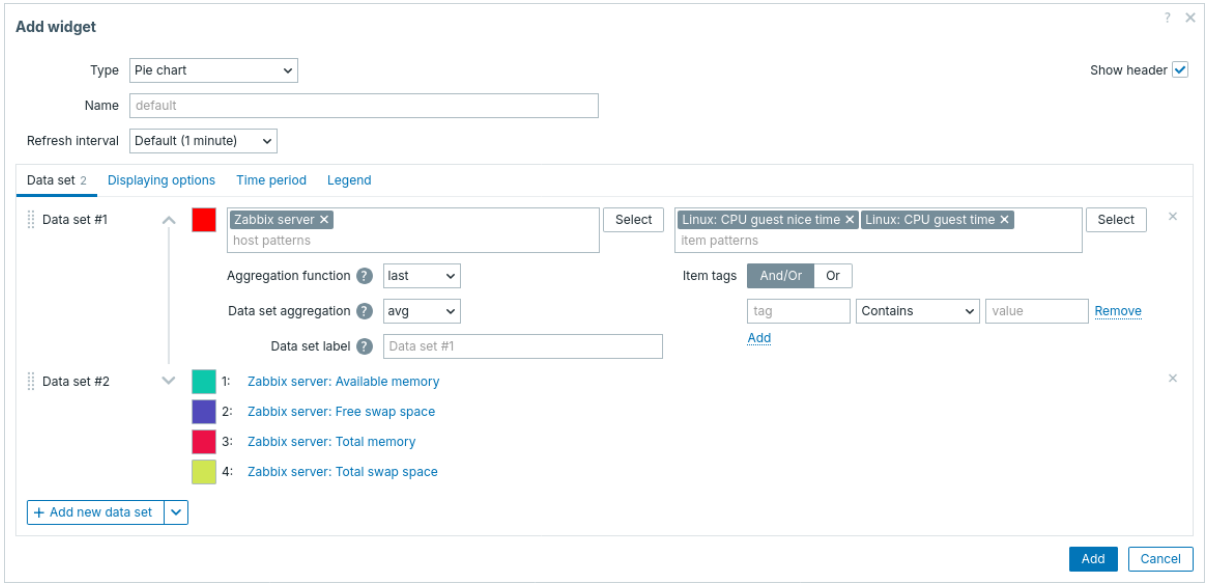


Novietojot peles kursoru virs sektora, iezīmētais sektors tiek izvērsts un tiek parādīta tā leģenda; noklikšķinot uz sektora, tas paliek izvērsts un tā leģenda redzama, līdz tas tiek aizvērts.

Pīrāga diagrammas logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Lejupielādēt attēlu*.

Konfigurācija

Lai konfigurētu, atlasiet *Pie chart* kā tipu:



Datu kopa

Cilne **Datu kopa** ļauj atlasīt datus sektoru diagrammai, pievienojot datu kopas. Var pievienot divu veidu datu kopas:

- *Vienumu šabloni* - tiek attēloti dati no atbilstošajiem vienumiem. Varat izvēlēties vienu pamatkrāsu vai atlasīt paletes rindu, lai katram atbilstošajam vienumam piešķirtu atšķirīgu krāsu.
- *Vienumu saraksts* - tiek attēloti dati no atlasītajiem vienumiem. Katra vienuma krāsu varat izvēlēties atsevišķi, izmantojot krāsu atlasītāju.

Pēc noklusējuma tiek pievienota *Vienumu šablonu* datu kopa.

Datu kopa	**Vienumu šablonu** datu kopai: Atlasiet vai ievadiet hosta un vienumu šablonus; sektoru diagrammā tiks attēloti to vienumu dati, kas atbilst ievadītajiem šabloniem; var tikt attēloti līdz 50 vienumiem. Atlasei var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item**, ja ir citi atbilstoši vienumi (piemēram, *item2*, *item3*). Hosta un vienumu šablonu norādīšana ir obligāta "Vienumu šablonu" datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksta** datu kopai: Atlasiet vienumus sektoru diagrammai, noklikšķinot uz pogas *Pievienot vienumu*. Nolaižamais saraksts *Tips* aiz vienuma nosaukuma ļauj atlasīt attēlošanas tipu katram vienumam: **Parasts** - vienuma vērtība sektoru diagrammā tiek attēlota proporcionāli (noklusējums); **Kopā** - vienuma vērtība aizņem visu sektoru diagrammu. Ņemiet vērā, ka katrā sektoru diagrammā var būt tikai viens "Kopā" vienums, un tas tiks novietots pirmais sektoru diagrammas leģendā. Ja vienumam ir iestatīts "Kopā", parametrs *Datu kopas agregācija* (skatiet zemāk) tiks atspējots un iestatīts uz "netiek izmantots". Varat arī atlasīt saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*. Logrīkiem ir piemērojamas tās pašas opcijas kā atsevišķiem vienumiem. Vienumu vai logrīku norādīšana ir obligāta "Vienumu saraksta" datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitlisku tipu vienumi. Konfigurējot logrīku veidnes panelī, parametrs hostu šablonu norādīšanai nav pieejams, un parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.

Agregācijas funkcija	Norādiet, kuru agregācijas funkciju izmantot katram datu kopas vienumam atlasītajā *Laika periodā*: **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **sum** - attēlot vērtību summu; **count** - attēlot vērtību skaitu; **first** - attēlot pirmo vērtību; **last** - attēlot pēdējo vērtību (noklusējums). Agregācija ļauj attēlot agregētu vērtību izvēlētajam intervālam (5 minūtes, stunda, diena). *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Parametram *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Datu kopas agregācija	Norādiet, kuru agregācijas funkciju izmantot visai datu kopai: **not used** - bez agregācijas, vienumi tiek attēloti atsevišķi (noklusējums); **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **sum** - attēlot vērtību summu; **count** - attēlot vērtību skaitu. Agregācija ļauj attēlot agregētu vērtību intervālam (5 minūtes, stunda, diena), kas atlasīts cilnē *Laika periods* vai tiek izmantots visam panelim.
Datu kopas etiķete	Norādiet pielāgotu etiķeti datu kopai. Etiķete tiek attēlota datu kopas konfigurācijā un sektoru diagrammas leģendā (agregētām datu kopām). Visas datu kopas tiek numurētas, ieskaitot tās, kurām ir norādīta *Datu kopas etiķete*. Ja etiķete nav norādīta, datu kopa tiks automātiski marķēta atbilstoši tās numuram (piemēram, ”Datu kopa #2”, ”Datu kopa #3” utt.). Datu kopu numerācija tiek pārrēķināta pēc datu kopu pārkārtošanas/vilkšanas. Pārāk garas datu kopu etiķetes tiks saīsinātas, lai ietilptu attēlošanas vietā (piemēram, ”Procesu skaits...”).
Vienumu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos vienumus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **And/Or** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Or nosacījuma; **Or** - pietiek, ja ir izpildīts viens nosacījums.

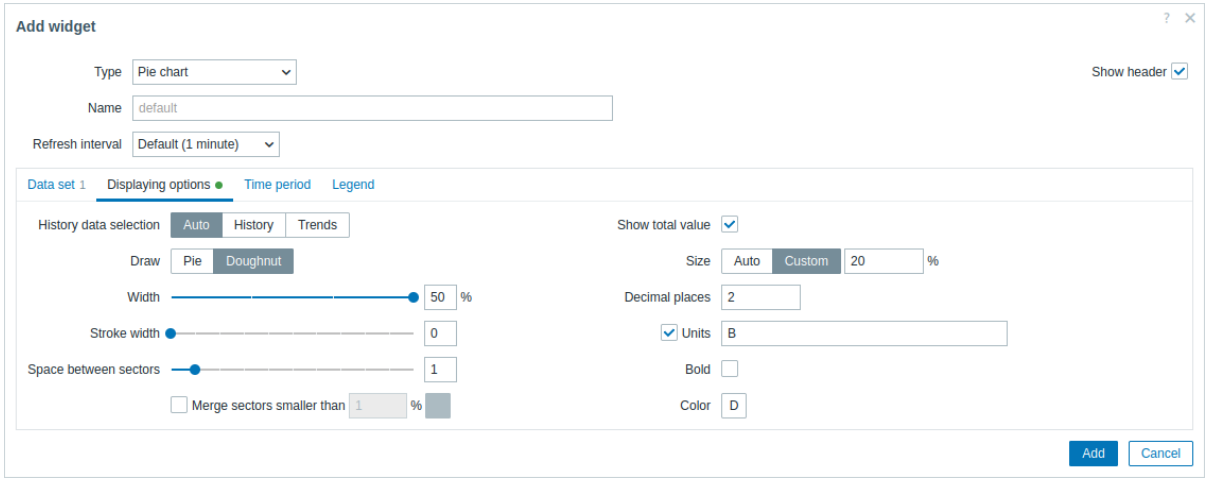
Datu kopas konfigurācijas informācija

Esošās datu kopas tiek parādītas sarakstā. Jūs varat pārkārtot, izvērst/sakļaut, mainīt krāsas un klonēt šīs datu kopas.

Plašāku informāciju skatiet par datu kopas konfigurācijas informāciju logrīkā *Graph*. Šī informācija attiecas arī uz *Pie chart* logrīku.

Attēlošanas opcijas

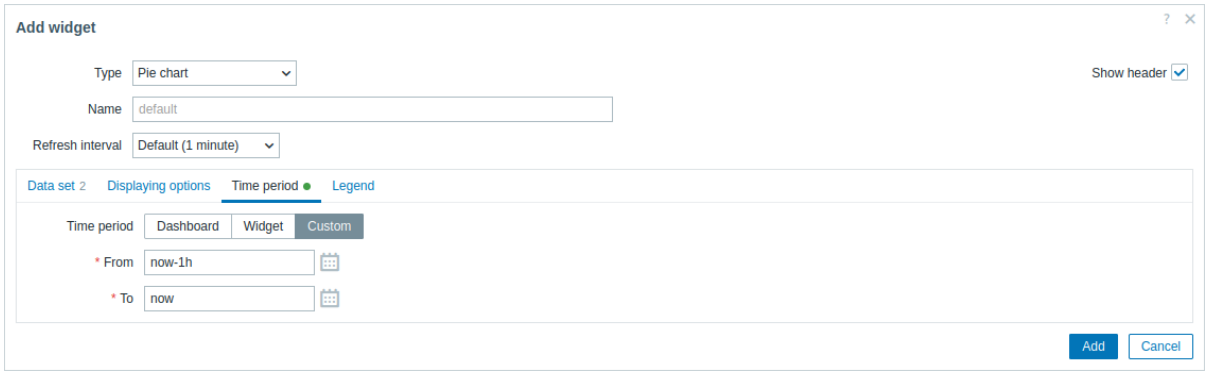
Cilne **Attēlošanas opcijas** ļauj definēt vēstures datu atlases un sektoru diagrammas vizualizācijas opcijas:



Vēstures datu atlase	Atlasiet datu avotu: **Auto** - dati tiek iegūti saskaņā ar klasisko algoritmu (noklusējums); **Vēsture** - dati no vēstures; **Tendences** - dati no tendencēm.
Zīmēt	Atlasiet sektoru diagrammas vizualizācijas stilu: **Sektoru diagramma** - pilna sektoru diagramma (sektori aizņem 100% no rādiusa); **Gredzenveida diagramma** - sektoru diagramma ar tukšu apli vidū (sektori izmanto līdz 50% no rādiusa).
Atstarpe starp sektoriem	Atlasiet atstarpes lielumu (vienībās no 0 līdz 10) starp sektoriem (noklusējums ir ”1”).
Apvienot sektorus, kas ir mazāki par N%	Atzīmējiet izvēles rūtiņu, lai apvienotu sektorus, kas ir mazāki par N%. Ja iespējots, atlasiet apvienoto sektoru krāsu un procentuālo slieksni (N) mazo sektoru apvienošanai.
Zīmēšanas stils: ***Gredzenveida diagramma***	
Platums	Atlasiet gredzenveida diagrammas platumu: 20, 30, 40 vai 50% (noklusējums) no rādiusa.
Kontūras platums	Atlasiet gredzenveida diagrammas sektora apmales platumu (0-10).
Rādīt kopējo vērtību	Atzīmējiet izvēles rūtiņu, lai rādītu kopējo vērtību gredzenveida diagrammas vidū.
Izmērs	Atlasiet kopējās vērtības izmēra opciju: **Auto** - teksta izmērs tiek automātiski pielāgots, lai tas būtu salasāms gredzenveida diagrammas vidū; **Pielāgots** - norādiet teksta izmēru kā procentu no kopējā logrīka augstuma.
Decimālvietas	Norādiet decimālvietu skaitu kopējai vērtībai (0-6).
Mērvienības	Norādiet kopējās vērtības mērvienības.
Treknraksts	Atzīmējiet izvēles rūtiņu, lai kopējo vērtību attēlotu treknrakstā.
Krāsa	Atlasiet kopējās vērtības krāsu.

Laika periods

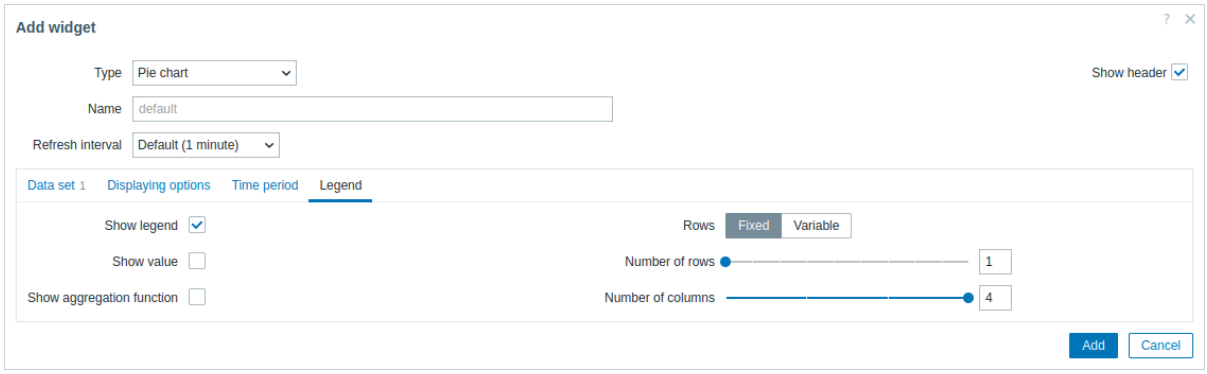
Cilne **Laika periods** ļauj iestatīt pielāgotu laika periodu sektoru diagrammas apkopošanas iestatījumiem:



Laika periods	Atlasiet datu avotu laika periodam: **Panelis** - izmantot paneļa laika perioda atlasītāju; **Logrīks** - izmantot saderīgu logrīku (norādīts parametrā *Logrīks*); **Pielāgots** - izmantot pielāgotu laika periodu, kas norādīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgi logrīki joprojām var to izmantot kā datu avotu laika periodam.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku (*Grafiks*, *Grafiks (klasiskais)*, *Grafika prototips*) kā datu avotu laika periodam. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Logrīks".
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` u.c.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` u.c.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Pielāgots".

Leģenda

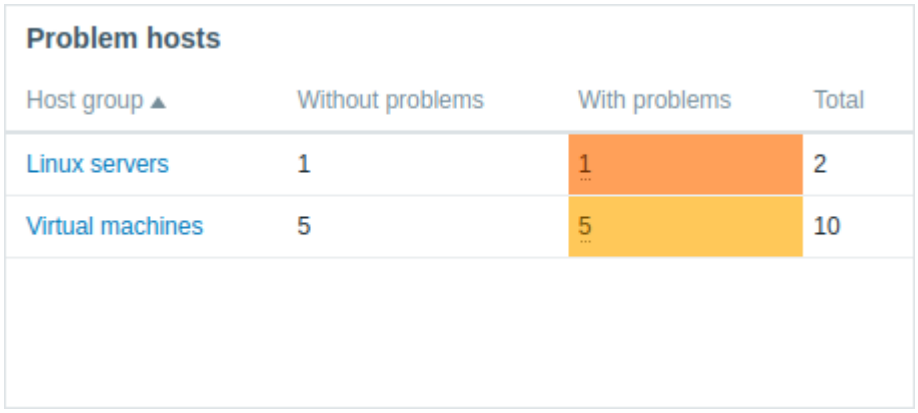
Cilne **Leģenda** ļauj pielāgot sektoru diagrammas leģendu:



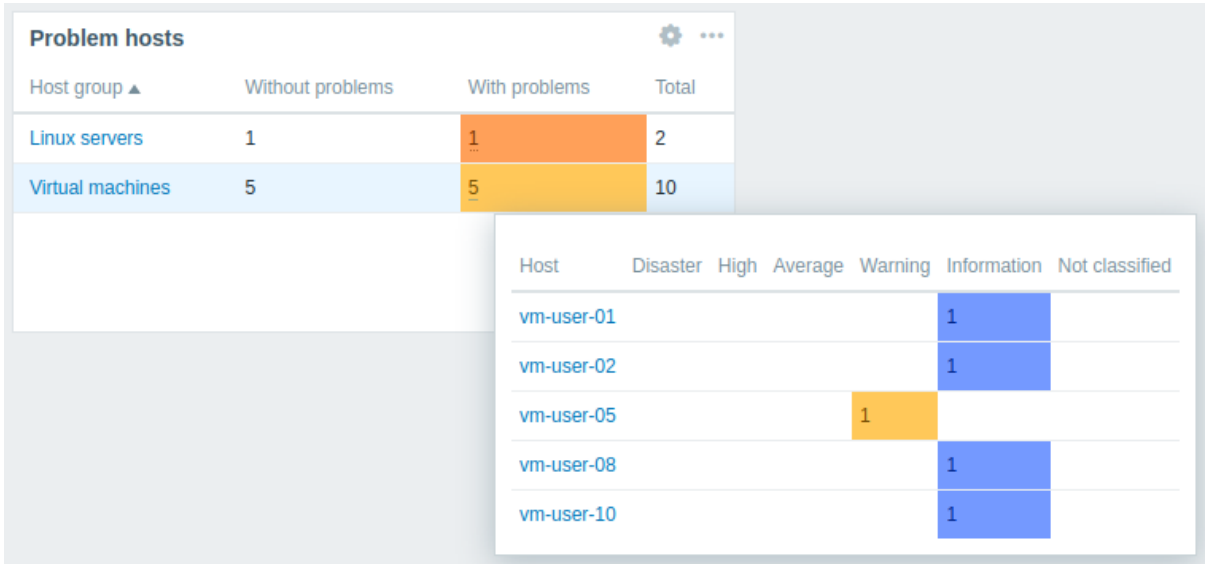
Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu leģendu sektoru diagrammā (pēc noklusējuma atzīmēta).
Rādīt vērtību	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu vienuma vērtību.
Rādīt agregācijas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu agregācijas funkciju.
Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** — attēloto rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** — attēloto rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.
Rindu skaits/ Maksimālais rindu skaits	Ja *Rindas* ir iestatīts uz "Fiksēts", iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja *Rindas* ir iestatīts uz "Mainīgs", iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja *Rādīt vērtību* nav atzīmēts.

Pārskats

Problēmu hostu logrīks parāda cēloņproblēmu skaitu katrā hostu grupā un attēlo augstāko problēmas būtiskuma līmeni katrā grupā. Tas ļauj filtrēt problēmas pēc dažādiem parametriem (hostu grupām, hostiem, problēmu nosaukumiem u.c.).

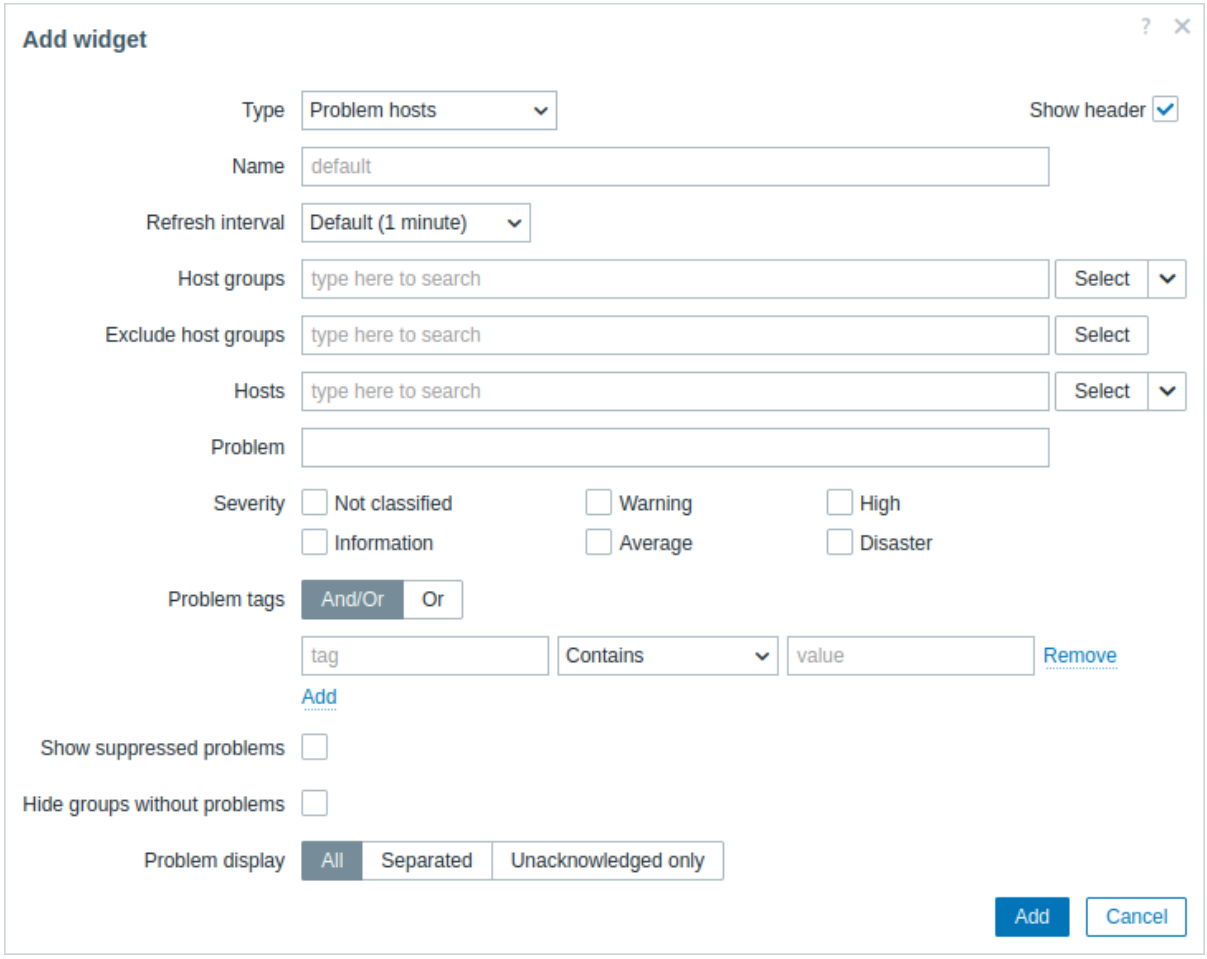


Novietojot kursoru virs problēmu skaita, tiek parādīts hostu ar problēmām saraksts:



Konfigurācija

Lai konfigurētu, kā tipu atlasiet *Problem hosts*:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

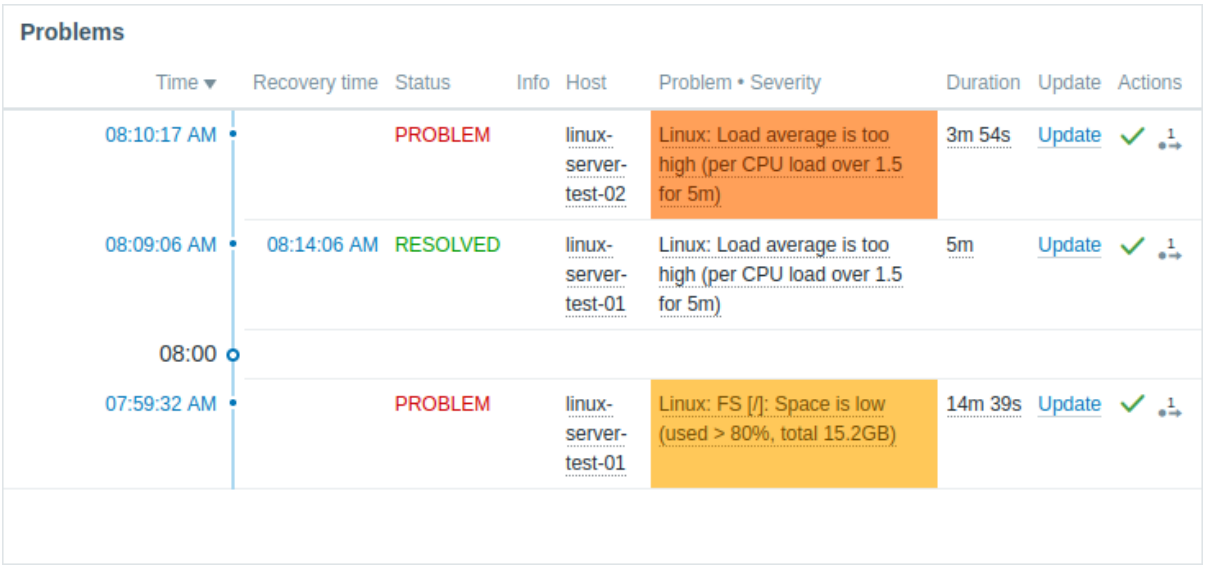
Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Logrīkā tiks attēloti hostu dati no šīm hostu grupām; ja netiek ievadīta neviena hostu grupa, tiks attēlotas visas hostu grupas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Izslēgt hostu grupas	Atlasiet hostu grupas, kuras paslēpt logrīkā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Hostu dati no šīm hostu grupām logrīkā netiks attēloti. Piemēram, hosti 001, 002, 003 var būt Grupā A, bet hosti 002, 003 arī Grupā B. Ja vienlaikus atlasām *rādīt* Grupu A un *izslēgt* Grupu B, panelī tiks attēloti tikai hosta 001 dati. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku vai paneli kā hostu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošo hostu nolaižamais saraksts. Ja netiek ievadīts neviens hosts, tiks attēloti visi hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Problēma	Varat ierobežot attēloto problemātisko hostu skaitu pēc problēmas nosaukuma. Ja šeit ievadīsiet virkni, tiks attēloti tikai tie hosti, kuru problēmu nosaukums satur ievadīto virkni. Makrosi netiek izvērsti.
Svarīgums	Atzīmējiet problēmu svarīguma līmeņus, lai filtrētu logrīkā attēlojamās problēmas. Ja nav atzīmēts neviens svarīguma līmenis, tiks attēlotas visas problēmas.

Problēmu birkas	Norādiet problēmu birkas, lai ierobežotu logrīkā attēloto problēmu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **And/Or** — visiem nosacījumiem jābūt izpildītiem, bet nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Or* nosacījuma; **Or** — pietiek, ja ir izpildīts viens nosacījums.
Rādīt apspiestās problēmas	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas dēļ.
Paslēpt grupas bez problēmām	Atzīmējiet opciju *Paslēpt grupas bez problēmām*, lai logrīkā paslēptu datus no hostu grupām bez problēmām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Problēmu attēlošana	Attēlot problēmu skaitu šādi: **Visas** — tiks attēlots pilns problēmu skaits; **Atdalīti** — neapstiprināto problēmu skaits tiks attēlots atsevišķi kā skaitlis no kopējā problēmu skaita; **Tikai neapstiprinātās** — tiks attēlots tikai neapstiprināto problēmu skaits.

23 Problēmas

Pārskats

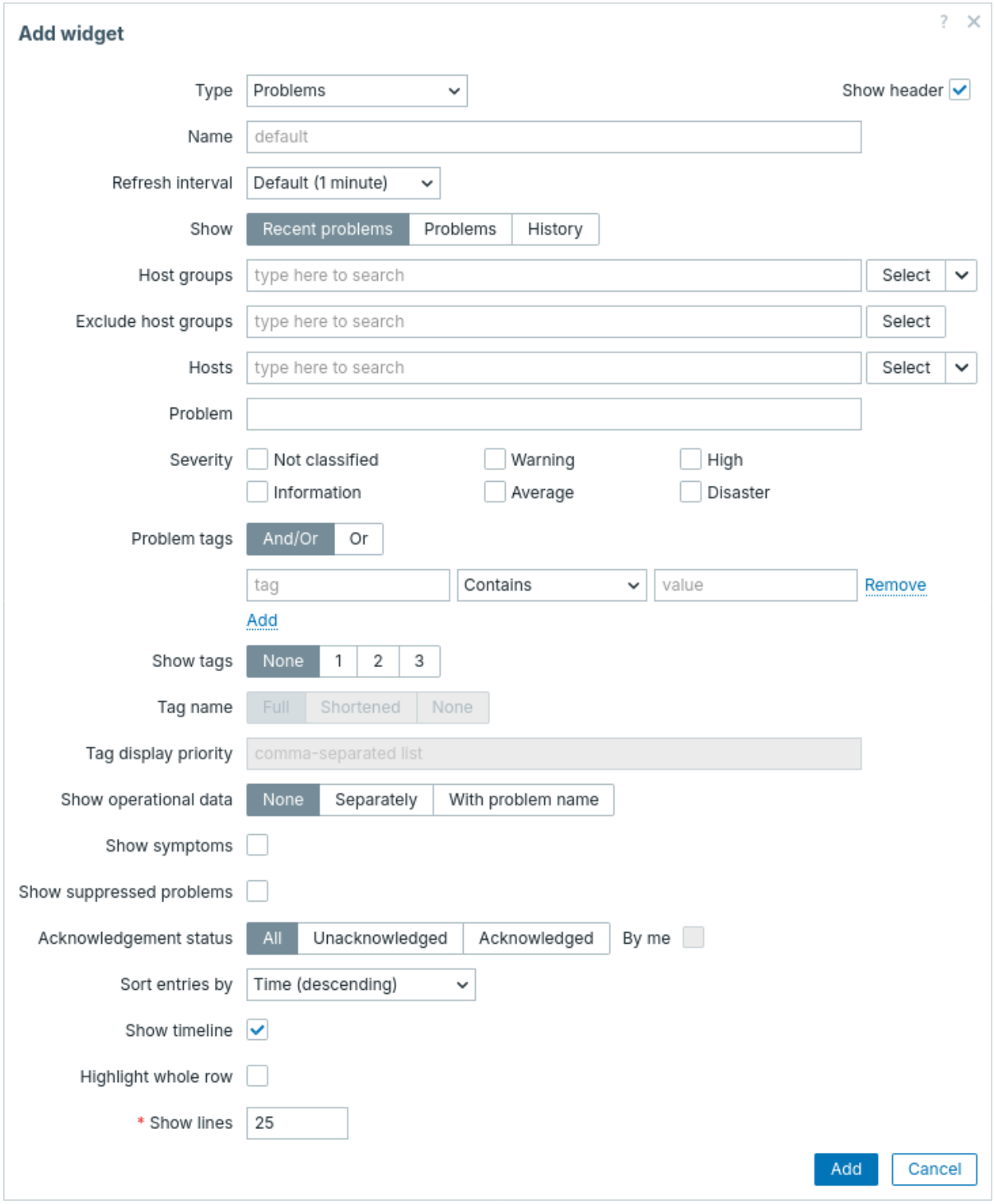
Logrīks *Problēmas* attēlo pašreizējās problēmas, kas filtrētas pēc dažādiem parametriem (hostu grupas, hosti, problēmu nosaukumi u.c.), sniedzot jums skaidru pārskatu par to, kam jāpievērš uzmanība.



Tas attēlo tos pašus datus kā *Uzraudzība* > *Problēmas* un var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Problems* kā tipu:



Varat ierobežot, cik daudz problēmu tiek attēlots logrīkā, dažādos veidos — pēc problēmas statusa, problēmas nosaukuma, svarīguma pakāpes, hostu grupas, hosta, notikuma birkas, apstiprinājuma statusa utt.

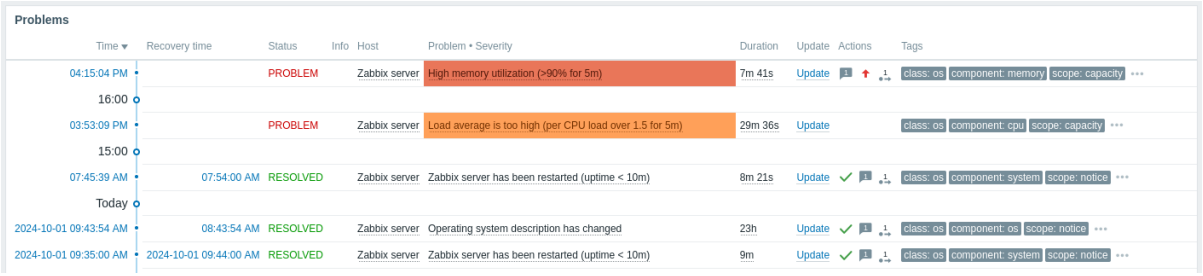
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Show	Filtrēt pēc problēmas statusa: **Recent problems** — tiek attēlotas neatrisinātas un nesen atrisinātas problēmas (noklusējums); **Problems** — tiek attēlotas neatrisinātas problēmas; **History** — tiek attēlota visu notikumu vēsture.
Host groups	Atlasiet hostu grupas, kuru problēmas attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Šo hostu grupu problēmas tiks attēlotas logrīkā; ja nav ievadīta neviena hostu grupa, tiks attēlotas visu hostu grupu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Exclude host groups	Atlasiet hostu grupas, kuru problēmas paslēpt logrīkā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Šo hostu grupu problēmas logrīkā netiks attēlotas. Piemēram, hosti 001, 002, 003 var būt Grupā A, bet hosti 002, 003 arī Grupā B. Ja vienlaikus atlasām *show* Grupai A un *exclude* Grupai B, logrīkā tiks attēlotas tikai hosta 001 problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosts	Atlasiet hostus, kuru problēmas attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku vai paneli kā hostu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Ja nav ievadīts neviens hosts, tiks attēlotas visu hostu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Problem	Varat ierobežot attēloto problēmu skaitu pēc to nosaukuma. Ja šeit ievadīsiet virkni, tiks attēlotas tikai tās problēmas, kuru nosaukums satur ievadīto virkni. Makrosi netiek izvērsti.
Severity	Atzīmējiet problēmu svarīguma pakāpes, lai filtrētu logrīkā attēlojamās problēmas. Ja nav atzīmēta neviena svarīguma pakāpe, tiks attēlotas visas problēmas.
Problem tags	Norādiet problēmu birkas, lai ierobežotu logrīkā attēloto problēmu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** — iekļaut norādītos birku nosaukumus; **Equals** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** — izslēgt norādītos birku nosaukumus; **Does not equal** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **And/Or** — visiem nosacījumiem jābūt izpildītiem, bet nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Or* nosacījuma; **Or** — pietiek, ja ir izpildīts viens nosacījums. Filtrējot, šeit norādītās birkas tiks attēlotas pirmās kopā ar problēmu, ja vien to nepārraksta saraksts *Tag display priority* (skatīt zemāk).
Show tags	Atlasiet attēlojamo birku skaitu: **None** — nav kolonnas *Tags*; **1** — kolonnā *Tags* ir viena birka; **2** — kolonnā *Tags* ir divas birkas; **3** — kolonnā *Tags* ir trīs birkas. Lai redzētu visas problēmas birkas, novietojiet peles kursoru virs trīs punktu ikonas.
Tag name	Atlasiet birkas nosaukuma attēlošanas režīmu: **Full** — birku nosaukumi un vērtības tiek attēloti pilnā apjomā; **Shortened** — birku nosaukumi tiek saīsināti līdz 3 simboliem, bet birku vērtības tiek attēlotas pilnā apjomā; **None** — tiek attēlotas tikai birku vērtības; bez nosaukumiem.
Tag display priority	Ievadiet birku attēlošanas prioritāti problēmai kā ar komatiem atdalītu birku sarakstu. Jāizmanto tikai birku nosaukumi, bez vērtībām. Piemērs: `Services,Applications,Application` Šajā sarakstā esošās birkas vienmēr tiks attēlotas pirmās, ignorējot dabisko kārtošanu alfabēta secībā.
Show operational data	Atlasiet operatīvo datu attēlošanas režīmu: **None** — operatīvie dati netiek attēloti; **Separately** — operatīvie dati tiek attēloti atsevišķā kolonnā; **With problem name** — pievienot operatīvos datus problēmas nosaukumam, izmantojot iekavas operatīvajiem datiem.
Show symptoms	Atzīmējiet izvēles rūtiņu, lai atsevišķā rindā attēlotu problēmas, kas klasificētas kā simptomi.

Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas vai atsevišķas problēmu apspiešanas dēļ.
Acknowledgement status	Filtrēt, lai attēlotu visas problēmas, tikai neapstiprinātās problēmas vai tikai apstiprinātās problēmas. Atzīmējiet papildu izvēles rūtiņu, lai izfiltrētu tās problēmas, kuras jebkad esat apstiprinājis jūs.
Sort entries by	Kārtot ierakstus pēc: **Time** (dilstoši vai augoši); **Severity** (dilstoši vai augoši); **Problem name** (dilstoši vai augoši); **Host** (dilstoši vai augoši). Ierakstu kārtošana pēc **Host** (dilstoši vai augoši) nav pieejama, konfigurējot logrīku veidnes panelī.
Show timeline	Atzīmējiet izvēles rūtiņu, lai attēlotu vizuālu laika skalu.
Highlight whole row	Atzīmējiet izvēles rūtiņu, lai izceltu visu rindu neatrisinātām problēmām. Izcelšanai tiek izmantota problēmas svarīguma pakāpes krāsa. *Highlight whole row* nav pieejams augsta kontrasta tēmās.
Show lines	Norādiet attēlojamo problēmu rindu skaitu.

Logrīka izmantošana

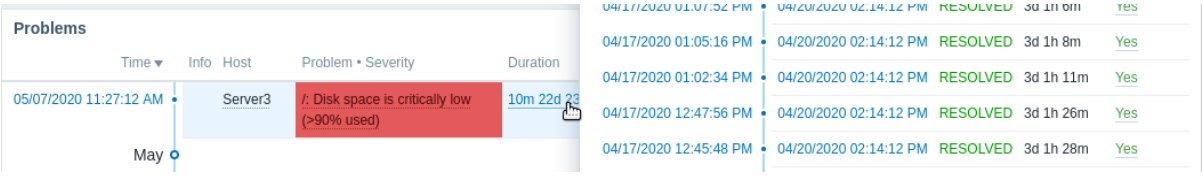


Problēmu logrīks piedāvā ātru piekļuvi papildu informācijai:

- Noklikšķiniet uz problēmas datuma (*Laiks* kolonnā) vai atkopšanas datuma (*Atkopšanas laiks* kolonnā), lai skatītu notikuma informāciju.
- Ja lauks *Info* nav tukšs, virziet kursoru virs parādītās ikonas, lai skatītu papildu informāciju.
- Noklikšķiniet uz hosta nosaukuma, lai atvērtu hosta izvēlni.
- Noklikšķiniet uz problēmas nosaukuma, lai atvērtu notikuma izvēlni.
- Virziet kursoru virs problēmas ilguma vai noklikšķiniet uz tā, lai skatītu problēmas notikuma uznirstošo logu.
- Nospiežot *Atjaunināt*, tiek atvērts logs Atjaunināt problēmu.
- Virziet kursoru virs pelēkās bultiņas ikonas kolonnā *Darbības* vai noklikšķiniet uz tās, lai skatītu izpildīto darbību sarakstu.

Problēmas notikuma uznirstošais logs

Problēmas notikuma uznirstošajā logā ir iekļauts šī trigera problēmu notikumu saraksts un, ja definēts, arī trigera apraksts un noklikšķināms URL.



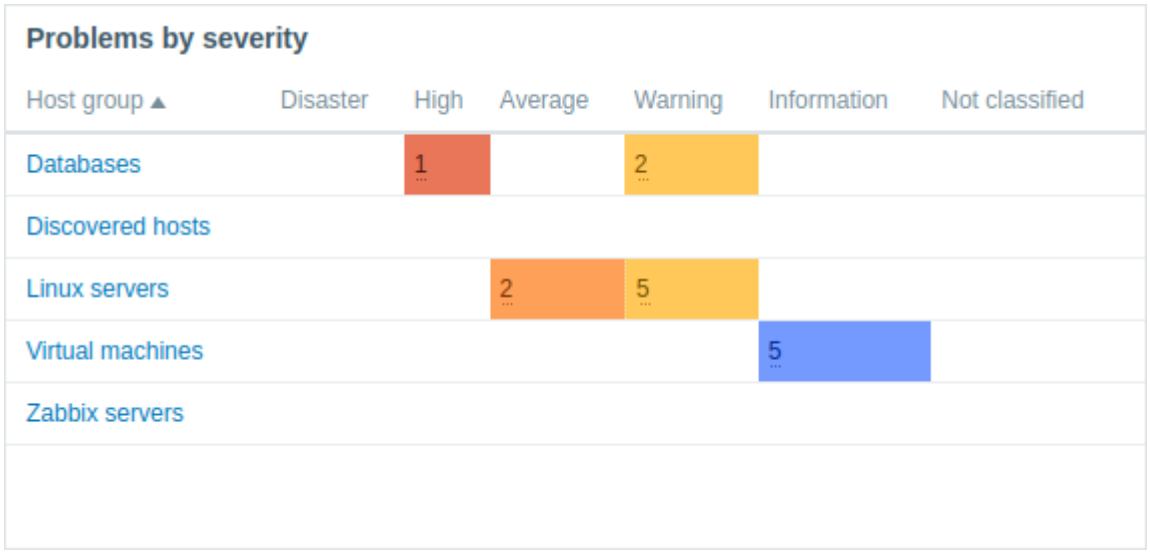
Lai atvērtu problēmas notikuma uznirstošo logu:

- Virziet peles kursoru virs problēmas ilguma logrīka *Problēmas* kolonnā *Ilgums*. Uznirstošais logs pazūd, tiklīdz novietojat peles kursoru prom no ilguma.
- Noklikšķiniet uz ilguma logrīka *Problēmas* kolonnā *Ilgums*. Uznirstošais logs pazūd tikai tad, ja vēlreiz noklikšķināt uz ilguma.

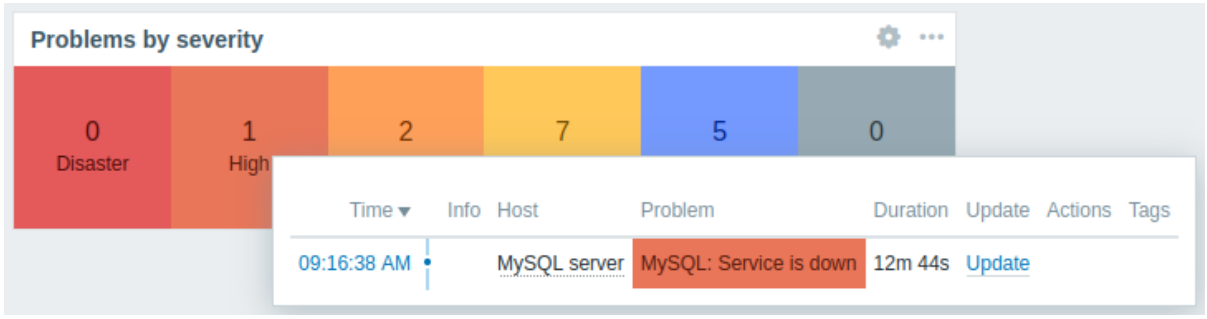
24 Problēmas pēc svarīguma pakāpes

Pārskats

Logrīks *Problēmas pēc nopietnības* parāda cēloņproblēmu skaitu, kas sagrupētas pēc nopietnības. Tas ļauj filtrēt problēmas pēc dažādiem parametriem (hostu grupas, hosti, problēmu nosaukumi utt.).

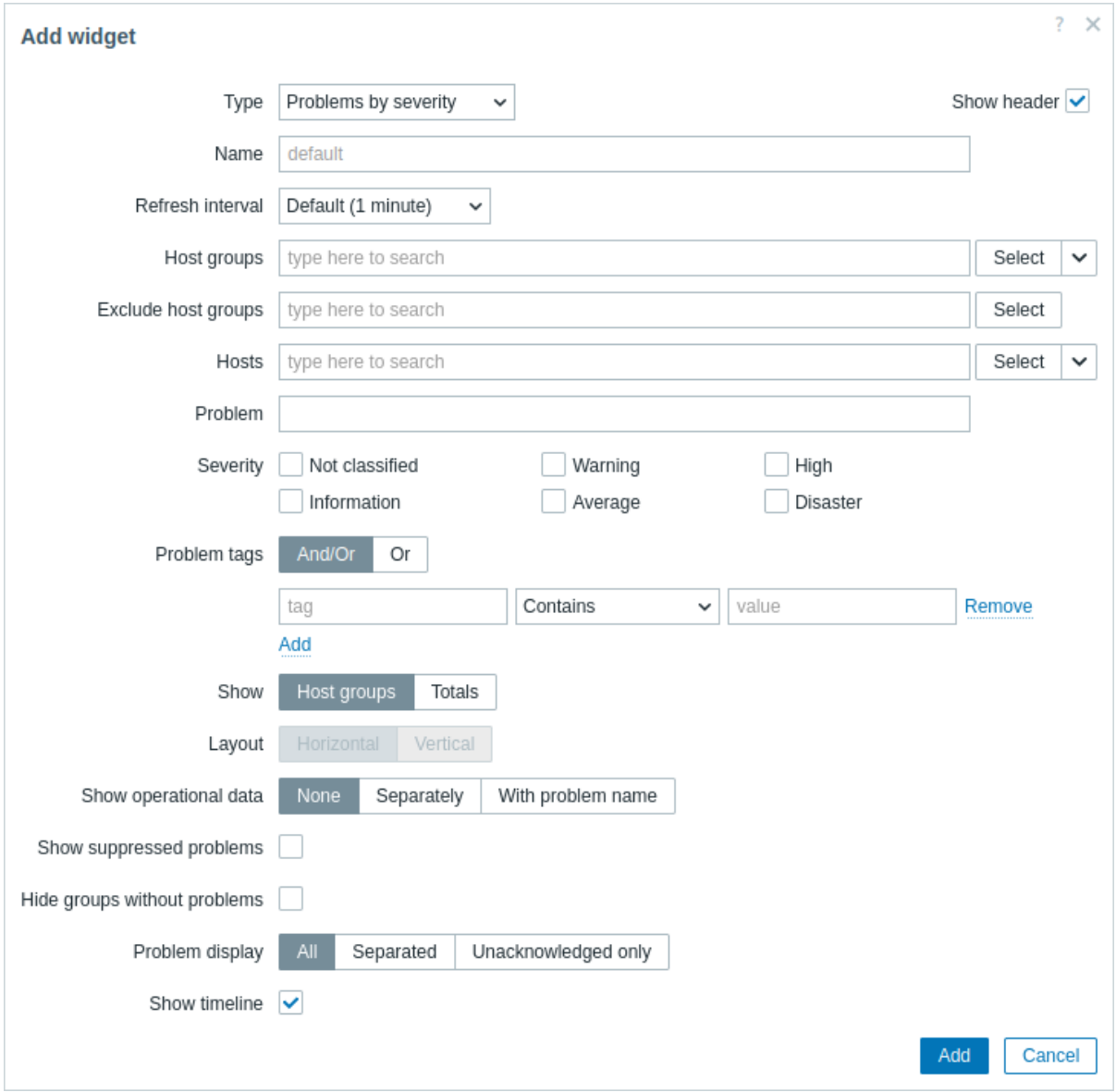


Novietojot kursoru virs problēmu skaita, tiek parādīts saistīto problēmu saraksts:



Konfigurācija

Lai konfigurētu, atlasiet *Problēmas pēc nopietnības* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

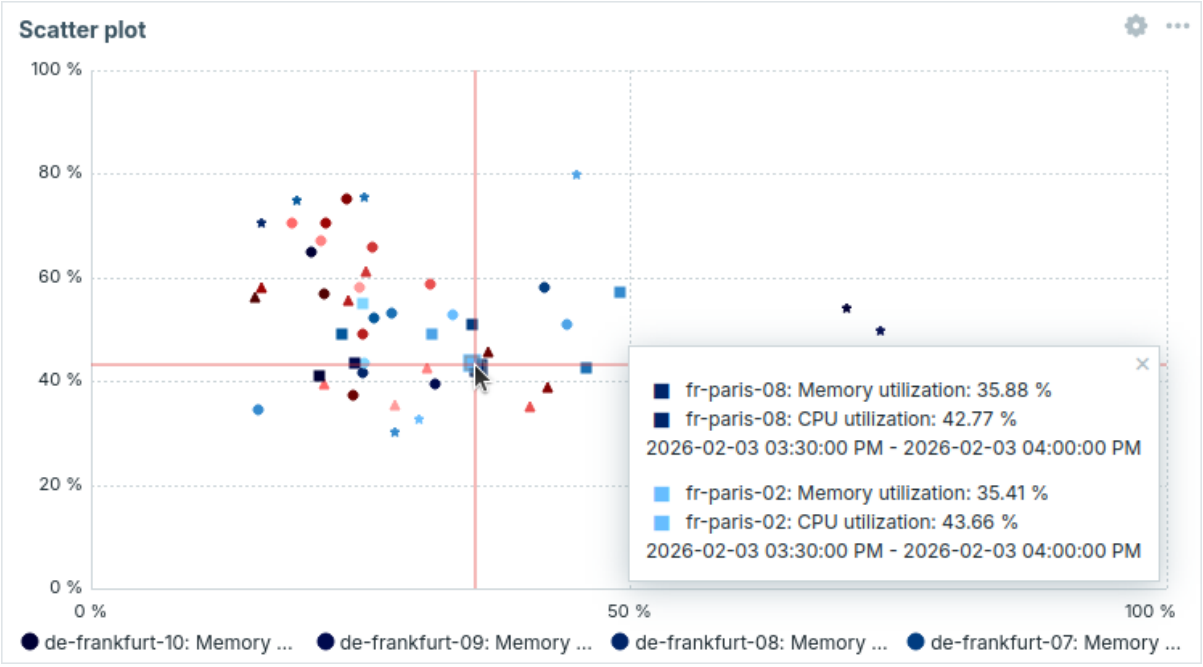
Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Logrīkā tiks attēloti šo hostu grupu hostu dati; ja netiek ievadīta neviena hostu grupa, tiks attēlotas visas hostu grupas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Izslēgt hostu grupas	Atlasiet hostu grupas, ko paslēpt logrīkā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Šo hostu grupu hostu dati logrīkā netiks attēloti. Piemēram, hosti 001, 002, 003 var būt Grupā A, bet hosti 002, 003 arī Grupā B. Ja vienlaikus atlasām *rādīt* Grupu A un *izslēgt* Grupu B, panelī tiks attēloti tikai hosta 001 dati. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku vai paneli kā hostu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Ja netiek ievadīts neviens hosts, tiks attēloti visi hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Problēma	Varat ierobežot attēloto problemātisko hostu skaitu pēc problēmas nosaukuma. Ja šeit ievadīsiet virkni, tiks attēloti tikai tie hosti, kuru problēmu nosaukums satur ievadīto virkni. Makrosi netiek izvērsti.
Nopietnība	Atzīmējiet problēmu nopietnības līmeņus, lai filtrētu logrīkā attēlojamās problēmas. Ja nav atzīmēts neviens nopietnības līmenis, tiks attēlotas visas problēmas.
Problēmu birkas	Norādiet problēmu birkas, lai ierobežotu logrīkā attēloto problēmu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **Un/Vai** - jāizpildās visiem nosacījumiem, bet nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** - pietiek, ja izpildās viens nosacījums.
Rādīt	Atlasiet rādīšanas opciju: **Hostu grupas** - attēlot problēmas pa hostu grupām; **Kopsummas** - attēlot visu atlasīto hostu grupu problēmu kopsummu krāsainos blokos atbilstoši problēmas nopietnībai. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī, un tiks attēlota tikai problēmu kopsumma.
Izkārtojums	Atlasiet izkārtojuma opciju: **Horizontāli** - krāsainie kopsummu bloki tiks attēloti horizontāli; **Vertikāli** - krāsainie kopsummu bloki tiks attēloti vertikāli. Šis parametrs ir pieejams, ja *Rādīt* ir iestatīts uz ”Kopsummas”.
Rādīt operatīvos datus	Attēlot operatīvos datus (skatiet *Operatīvie dati* aprakstu sadaļā *Uzraudzība* > *Problēmas*) kā: **Nav** - operatīvie dati netiks attēloti **Atsevišķi** - operatīvie dati tiks attēloti atsevišķā rindā **Kopā ar problēmas nosaukumu** - operatīvie dati tiks attēloti kopā ar problēmas nosaukumu.
Rādīt apspiestās problēmas	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas dēļ.
Paslēpt grupas bez problēmām	Atzīmējiet opciju *Paslēpt grupas bez problēmām*, lai logrīkā paslēptu datus no hostu grupām bez problēmām. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Problēmu attēlošana	Attēlot problēmu skaitu kā: **Visas** - tiks attēlots pilns problēmu skaits; **Atdalīti** - neapstiprināto problēmu skaits tiks attēlots atsevišķi kā daļa no kopējā problēmu skaita; **Tikai neapstiprinātās** - tiks attēlots tikai neapstiprināto problēmu skaits.
Rādīt laika skalu	Atzīmējiet izvēles rūtiņu, lai attēlotu vizuālu laika skalu.

25 Izkliedes diagramma

Pārskats

Izkliedes diagrammas logrīks attēlo saistību starp diviem rādītājiem, izvietojot atsevišķus datu punktus gar X un Y asi. Tas palīdz atklāt datu kopā modeļus, grupējumus, korelācijas un izņēmuma vērtības.



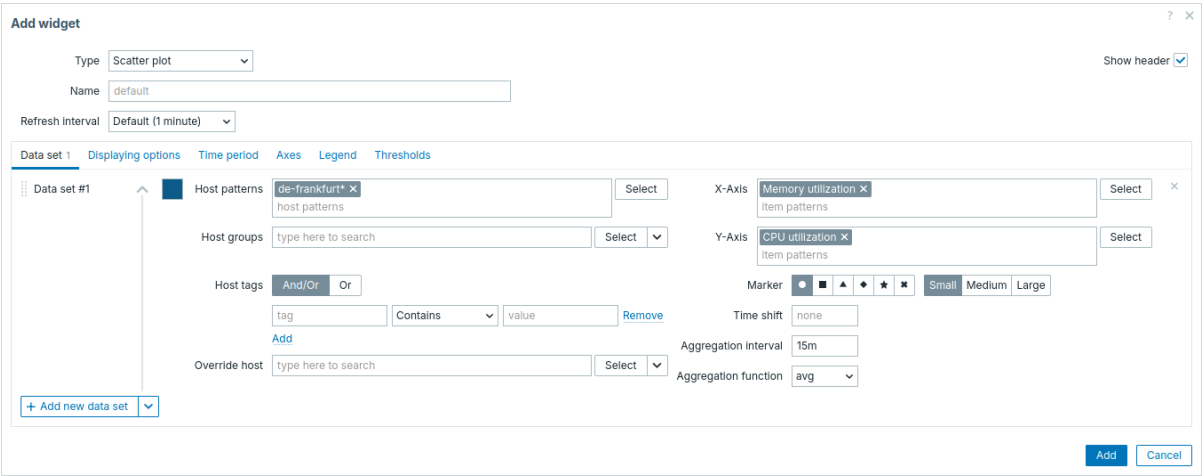
Novietojot kursoru virs datu punkta, tiek parādīts uznirstošs paskaidrojums ar šī datu punkta un citu punktu 6 px rādiusā esošo vienumu sarakstu; tie vispirms tiek kārtoti pēc izkliedes diagrammas X ass vērtībām augošā secībā un pēc tam pēc Y ass vērtībām dilstošā secībā.

Atlasot vienumu uznirstošajā paskaidrojumā, tā dati tiek pārraidīti uz citiem logrīkiem.

Izkliedes diagrammas logrīkā attēloto informāciju var lejupielādēt kā PNG attēlu, logrīka izvēlnē atlasot opciju *Lejupielādēt attēlu*.

Konfigurācija

Lai konfigurētu, atlasiet *Scatter plot* kā tipu:



Datu kopa

Cilne **Datu kopa** ļauj atlasīt datus izkliedes diagrammai, pievienojot datu kopas. Var pievienot divu veidu datu kopas:

- *Vienumu šabloni* - hosti ar atbilstošiem vienumiem tiek attēloti kā datu punkti. Varat izvēlēties vienu pamatkrāsu vai atlasīt paletes rindu, lai katram hostam ar atbilstošajiem vienumiem piešķirtu atšķirīgas krāsas.
- *Vienumu saraksts* - atlasītie vienumi tiek attēloti kā datu punkti, kur katrs punkts attēlo vienumu pāri (X un Y). Katras datu kopas krāsu varat izvēlēties atsevišķi no krāsu atlasītāja.

Datu punkti tiek izveidoti tikai tiem hostu/vienumu pāriem, kuriem abās asīs ir vismaz viena vērtība.

Pēc noklusējuma tiek pievienota *Vienumu šablonu* datu kopa.

Datu kopa	**Vienumu šablonu** datu kopai: Atlasiet vai ievadiet hostu un vienumu šablonus; vienumu dati, kas atbilst šiem šabloniem, tiks attēloti izkliedes diagrammā; var tikt attēloti līdz 100 vienumiem (50 uz katras ass kopumā 50 hostiem). Atlasei var izmantot aizstājējzīmju šablonus (piemēram, ∗ atgriezīs rezultātus, kas atbilst nullei vai vairāk rakstzīmēm). Lai norādītu aizstājējzīmes šablonu, ievadiet virkni manuāli un nospiediet *Enter*. Aizstājējzīmes simbols vienmēr tiek interpretēts, tāpēc nav iespējams atsevišķi pievienot, piemēram, vienumu ar nosaukumu *item** , ja ir citi atbilstoši vienumi (piemēram, *item2*, *item3*). Hostu un vienumu šablonu norādīšana ir obligāta ”Vienumu šablonu” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. **Vienumu saraksta** datu kopai: Atlasiet vienumus izkliedes diagrammai, noklikšķinot uz pogas *Pievienot vienumu*. Varat arī atlasīt saderīgus logrīkus kā vienumu datu avotu, noklikšķinot uz pogas *Pievienot logrīku*. Vienumu vai logrīku norādīšana ir obligāta ”Vienumu saraksta” datu kopām. Skatiet arī: Datu kopas konfigurācijas informācija. Ņemiet vērā, ka ir atļauti tikai skaitlisku tipu vienumi. Konfigurējot logrīku veidnes panelī, parametrs hostu šablonu norādīšanai nav pieejams, un parametrs vienumu saraksta norādīšanai ļauj atlasīt tikai veidnē konfigurētos vienumus.
Hostu grupas	Atlasiet hostu grupas. Alternatīvi varat atlasīt saderīgu logrīku kā hostu grupu datu avotu. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupu, tiek netieši atlasītas visas tajā ietvertās apakšgrupas. Logrīkā tiks attēloti hostu dati no šīm hostu grupām; ja netiek ievadīta neviena hostu grupa, tiks attēlotas visas hostu grupas. Šis parametrs ir pieejams tikai *Vienumu šablonu* datu kopām un nav pieejams, konfigurējot logrīku veidnes panelī.
Hostu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos hostus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **Un/Vai** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** - pietiek, ja ir izpildīts viens nosacījums. Šis parametrs ir pieejams tikai *Vienumu šablonu* datu kopām un nav pieejams, konfigurējot logrīku veidnes panelī.
Aizstāt hostu	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Marķieris	Izvēlieties marķiera tipu un izmēru hostu datu punktiem datu kopā.
Laika nobīde	Ja nepieciešams, norādiet laika nobīdi. Šajā laukā varat izmantot laika sufiksus. Ir atļautas negatīvas vērtības.

Agregācijas funkcija	Norādiet, kuru agregācijas funkciju izmantot katram vienumam datu kopā atlasītajā *Laika periodā*: **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **sum** - attēlot vērtību summu; **count** - attēlot vērtību skaitu (ja abās asīs pastāv vismaz viena vērtība); **first** - attēlot pirmo vērtību; **last** - attēlot pēdējo vērtību. Agregācija ļauj attēlot agregētu vērtību izvēlētajā intervālā (5 minūtes, stunda, diena), nevis visas vērtības. *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. *count* gadījumā neskaitliski dati tiks pārveidoti par skaitliskiem. Skatiet arī: Agregācija grafikos.
Agregācijas intervāls	Norādiet intervālu vērtību agregēšanai. Šajā laukā varat izmantot laika sufiksus. Skaitliska vērtība bez sufiksa tiks uzskatīta par sekundēm. Ņemiet vērā, ka, ja logrīks ir konfigurēts vēsturisko datu attēlošanai, pamatojoties uz trendiem (*Vēsturisko datu atlase* ir iestatīta uz *Trendi* vai *Auto*), ieteicams izmantot agregācijas intervālu, kas ir 1 stundas reizinājums (piemēram, 3600, 60m, 1h, 3h utt.). Trendi glabā pa stundām agregētas vērtības, tāpēc agregācijas intervāla izmantošana, kas nav 1 stundas reizinājums (piemēram, 100s, 7min, 15min, 90min utt.), var radīt grūti interpretējamus rezultātus. Agregācijas intervāla norādīšana ir obligāta. Agregācijas intervāls nosaka datu punkta laika intervālu un koordinātas; piemēram, ja X ass vienumiem ir 4 vērtības un Y ass vienumiem ir 1 vērtība, koordinātas atspoguļo 4 vērtību agregēšanu uz X ass un 1 vērtības agregēšanu uz Y ass norādītajā intervālā.

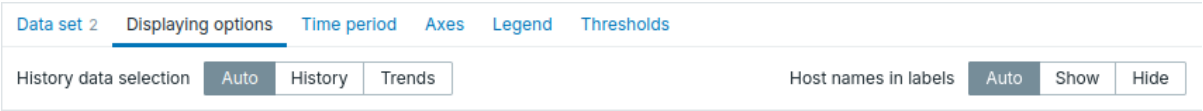
Datu kopas konfigurācijas informācija

Esošās datu kopas tiek parādītas sarakstā. Jūs varat pārkārtot, izvērst/sakļaut, mainīt krāsas un klonēt šīs datu kopas.

Lai iegūtu vairāk informācijas, skatiet datu kopas konfigurācijas informāciju logrīkā *Graph*. Šī informācija attiecas arī uz logrīku *Scatter plot*.

Attēlošanas opcijas

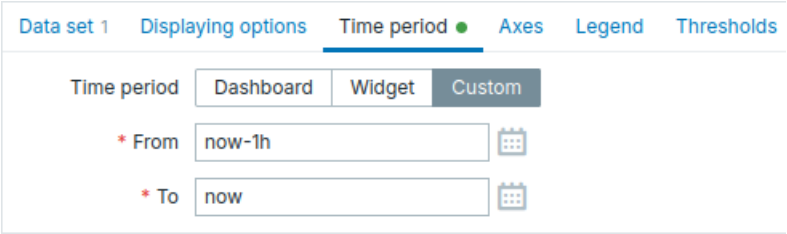
Cilne **Attēlošanas opcijas** ļauj definēt vēstures datu atlasi:



Vēstures datu atlase	Iestatiet izkliedes diagrammas datu avotu: **Auto** - dati tiek iegūti saskaņā ar klasiskās diagrammas algoritmu (noklusējums). **History** - dati no vēstures. **Trends** - dati no tendencēm.
Hostu nosaukumi etiķetēs	Izvēlieties, vai attēlot hostu nosaukumus diagrammas leģendā un uznirstošajā paskaidrojumā, kas parādās, novietojot kursoru virs diagrammas datiem: **Auto** - hostu nosaukumi tiek attēloti tikai tad, ja datu kopās ir vairāk nekā viens hosts (noklusējums). **Show** - hostu nosaukumi tiek attēloti. **Hide** - hostu nosaukumi tiek paslēpti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Laika periods

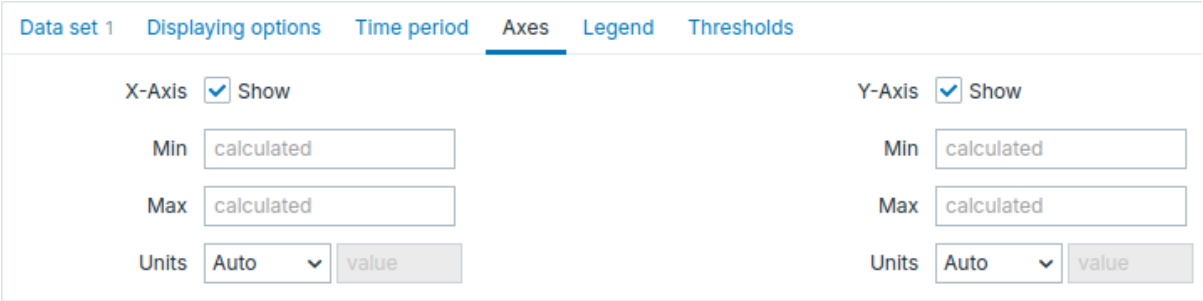
Cilne **Laika periods** ļauj iestatīt laika periodu, par kuru attēlot datus izkliedes diagrammā:



Laika periods	Atlasiet datu avotu laika periodam: **Panelis** — izmantot paneļa laika perioda atlasītāju; **Logrīks** — izmantot saderīgu logrīku (norādīts parametrā *Logrīks*); **Pielāgots** — izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks parādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika periods* konfigurācijas saderīgie logrīki joprojām var to izmantot kā datu avotu laika periodam.
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā datu avotu laika periodam. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Logrīks”.
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` u.c.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Pielāgots”.
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` u.c.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz ”Pielāgots”.

Asis

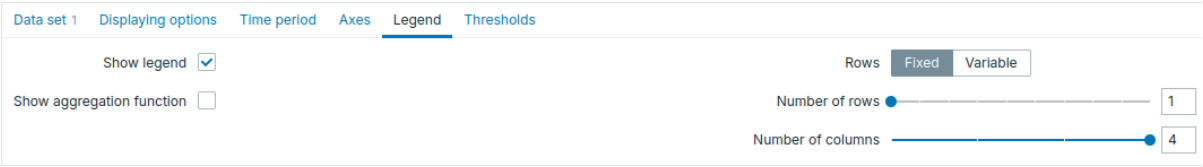
Cilne **Asis** ļauj pielāgot, kā tiek attēlotas asis:



X-Axis/Y-Axis	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu X asi/Y asi (pēc noklusējuma atzīmēta).
Min	Iestatiet attiecīgās ass minimālo vērtību.
Max	Iestatiet attiecīgās ass maksimālo vērtību.
Units	Izvēlieties mērvienību izkliedes diagrammas ass vērtībām no nolaižamā saraksta: **Auto** - ass vērtības tiek attēlotas, izmantojot pirmā vienuma mērvienību datu kopā; **Static** - ass vērtības tiek attēlotas, izmantojot *value* ievades laukā norādīto mērvienību; ja lauks tiek atstāts tukšs, tiek attēlotas tikai skaitliskās vērtības.

Leģenda

Cilne **Leģenda** ļauj pielāgot izkliedes diagrammas leģendu:

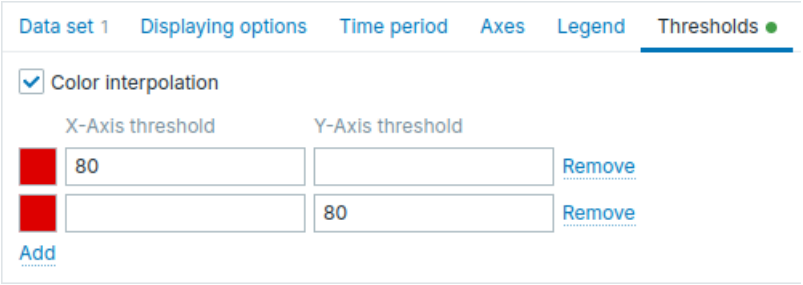


Rādīt leģendu	Noņemiet atzīmi no šīs izvēles rūtiņas, lai paslēptu leģendu izkliedes diagrammā (pēc noklusējuma atzīmēta).
Rādīt agregācijas funkciju	Atzīmējiet šo izvēles rūtiņu, lai leģendā rādītu agregācijas funkciju.

Rindas	Atlasiet leģendas rindu attēlošanas režīmu: **Fiksēts** — attēloto rindu skaitu nosaka parametra *Rindu skaits* vērtība; **Mainīgs** — attēloto rindu skaitu nosaka konfigurēto vienumu skaits, nepārsniedzot parametra *Maksimālais rindu skaits* vērtību.
Rindu skaits/ Maksimālais rindu skaits	Ja *Rindas* ir iestatīts uz ”Fiksēts”, iestatiet attēlojamo leģendas rindu skaitu (1-10). Ja *Rindas* ir iestatīts uz ”Mainīgs”, iestatiet maksimālo attēlojamo leģendas rindu skaitu (1-10).
Kolonnu skaits	Iestatiet attēlojamo leģendas kolonnu skaitu (1-4). Šis parametrs ir pieejams, ja *Rādīt min/vid/max* nav atzīmēts.

Sliekšņi

Cilne **Sliekšņi** ļauj konfigurēt sliekšņus, kas nosaka hostu marķieru krāsas:



Krāsu interpolācija	Atzīmējiet izvēles rūtiņu, lai iespējotu vienmērīgu pāreju starp sliekšņu krāsām hostu marķieriem. Šis parametrs tiek piemērots, ja X asij, Y asij vai abām asīm vienlaikus ir iestatīti divi vai vairāki sliekšņi.
X ass/Y ass slieksnis	Noklikšķiniet uz *Pievienot*, lai pievienotu slieksni, izvēlieties sliekšņa krāsu no krāsu atlasītāja un norādiet skaitlisku vērtību. Sliekšņus var piemērot vienai asij, abām asīm atsevišķi vai abām asīm vienlaikus. Piemēram, viens slieksnis ar X=80 un Y=80 iekrāsos marķierus, kad būs izpildīti abi nosacījumi. Saglabājot sliekšņu saraksts tiks sakārtots augošā secībā. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Tiek atbalstītas arī vērtību kartēšanas.

26 SLA atskaite

Pārskats

Logrīks *SLA report* attēlo SLA atskaites, palīdzot jums uzraudzīt pakalpojumu veiktspēju atbilstoši noteiktajiem mērķiem.

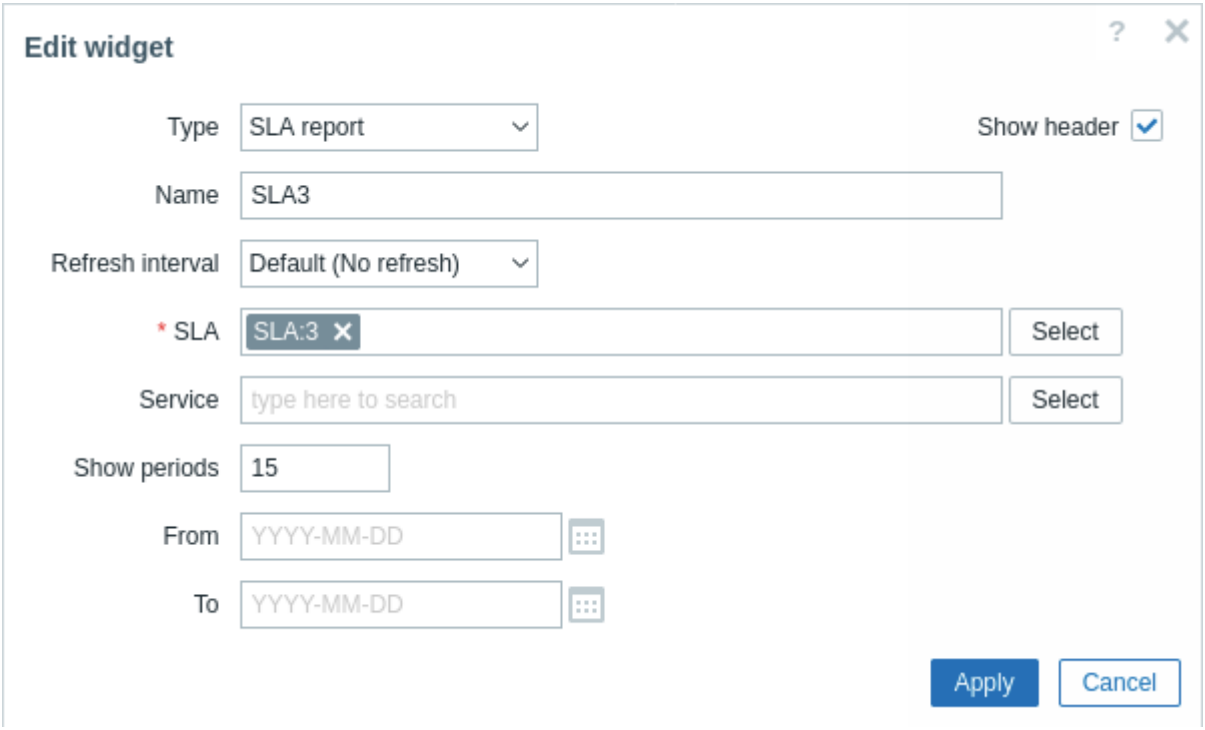
SLA report

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

Tas attēlo tos pašus datus kā *Pakalpojumi* > *SLA report*.

Konfigurācija

Lai konfigurētu, atlasiet *SLA report* kā tipu:



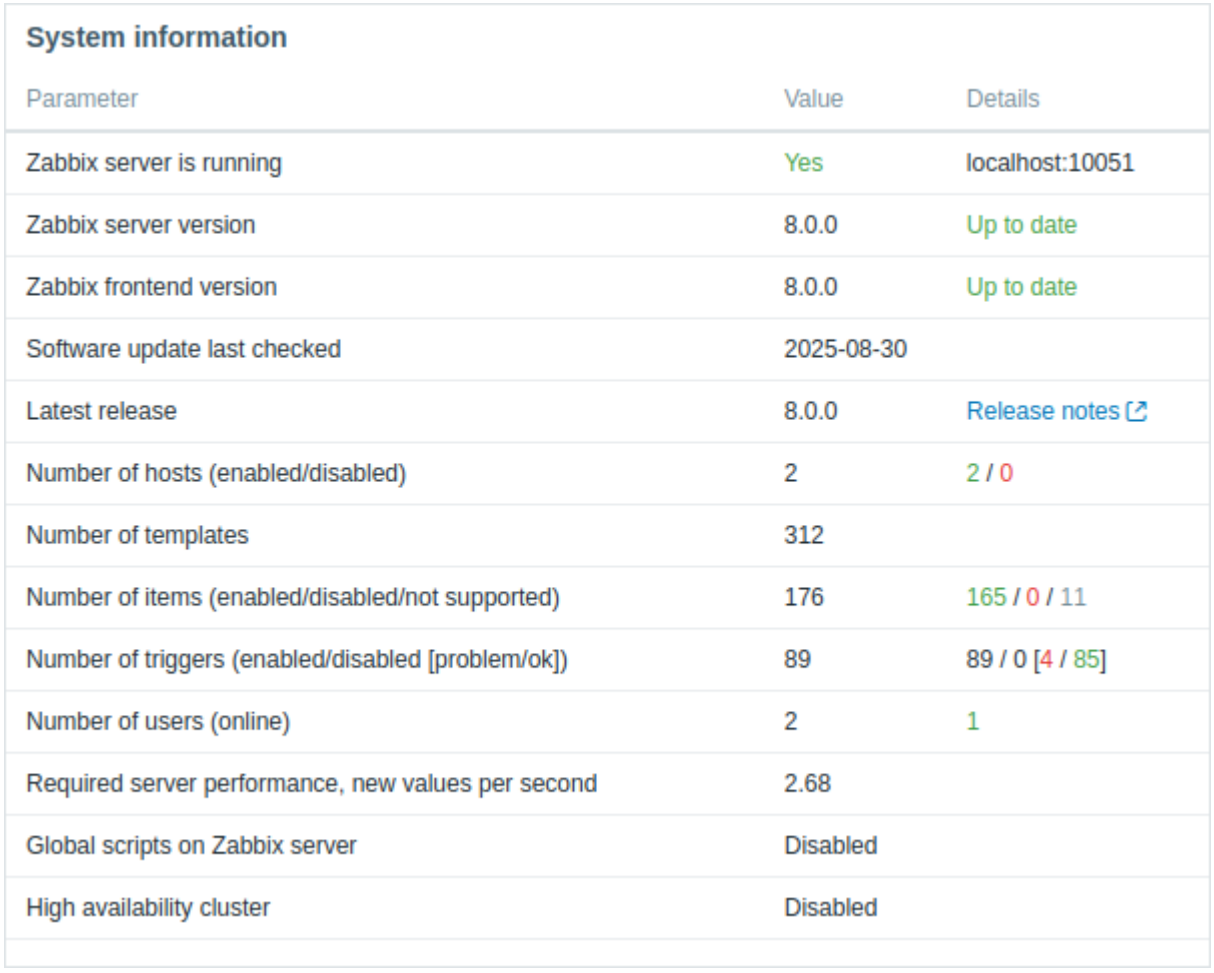
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

SLA	Atlasiet SLA atskaitei.
Service	Atlasiet pakalpojumu atskaitei.
Show periods	Iestatiet, cik periodi tiks attēloti logrīkā (pēc noklusējuma 20, maksimums 100).
From	Atlasiet atskaites sākuma datumu. Tiek atbalstīti relatīvie datumi: `now`, `now/d`, `now/w-1w` utt.; atbalstītie datuma modifikatori: `d`, `w`, `M`, `y`.
To	Atlasiet atskaites beigu datumu. Tiek atbalstīti relatīvie datumi: `now`, `now/d`, `now/w-1w` utt.; atbalstītie datuma modifikatori: `d`, `w`, `M`, `y`.

27 Sistēmas informācija

Pārskats

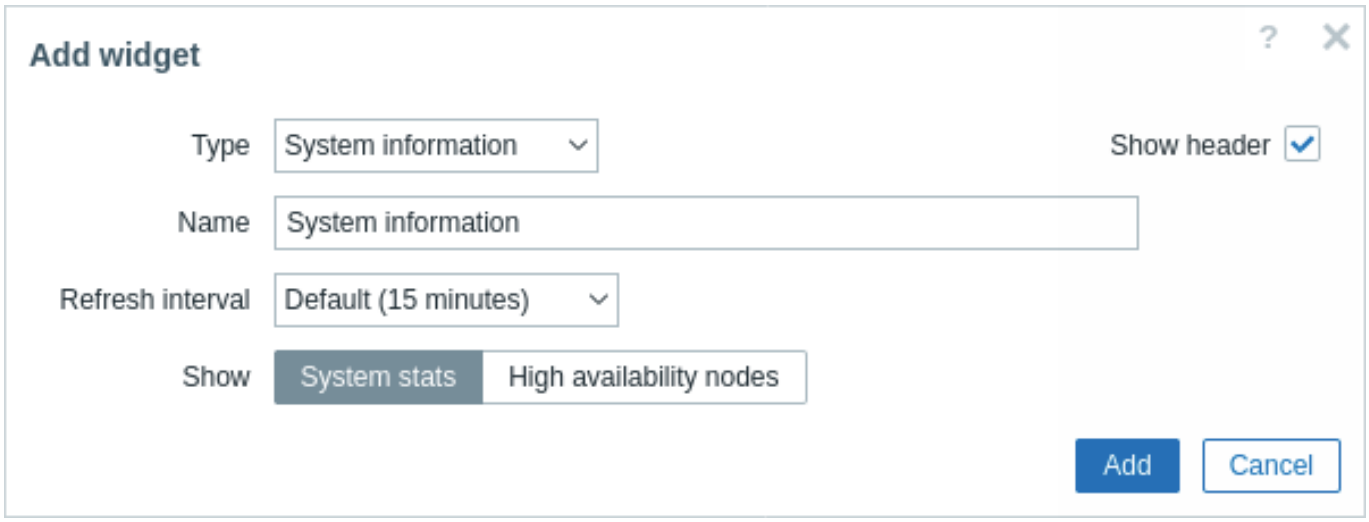
Logrīks *Sistēmas informācija* parāda galveno Zabbix servera un sistēmas datu kopsavilkumu vai detalizētu informāciju par augstas pieejamības mezgliem.



Tas parāda tos pašus datus kā *Atskaites* > *Sistēmas informācija*.

Konfigurācija

Lai konfigurētu, atlasiet *Sistēmas informācija* kā tipu:



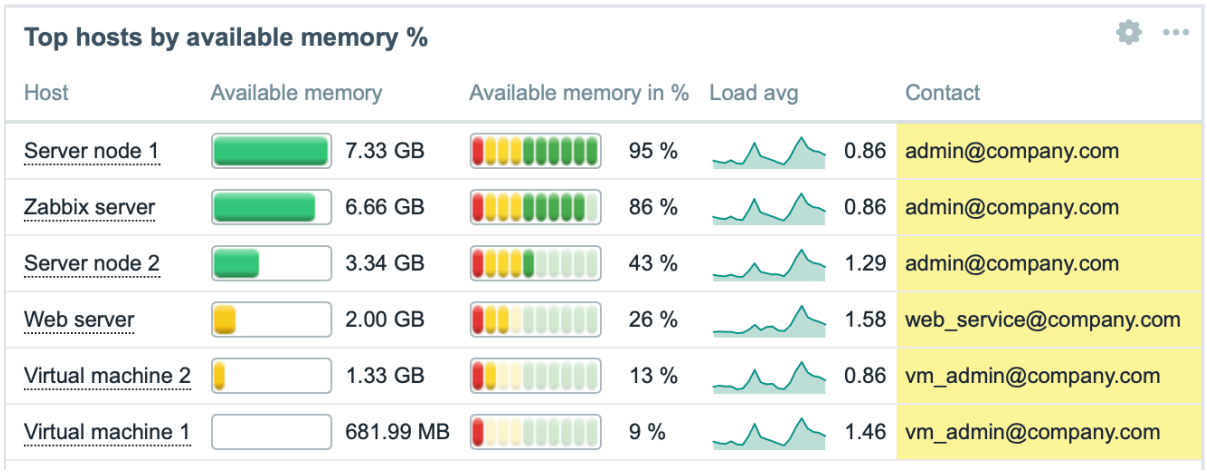
Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Rādīt	Atlasiet, ko attēlot: **Sistēmas statistika** - attēlot galveno Zabbix serveris un sistēmas datu kopsavilkumu; **Augstas pieejamības mezgli** - attēlot augstas pieejamības mezglu statusu (ja ir iespējota augstas pieejamības kopa).
Rādīt programmatūras atjauninājumu pārbaudes informāciju	Atzīmējiet izvēles rūtiņu, lai attēlotu Zabbix programmatūras atjauninājumu pārbaudes informāciju. Šī opcija ir pieejama tikai tad, ja Zabbix serveris konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude un laukā *Rādīt* ir atlasīta opcija ”Sistēmas statistika”.

28 Populārākie hosti

Pārskats

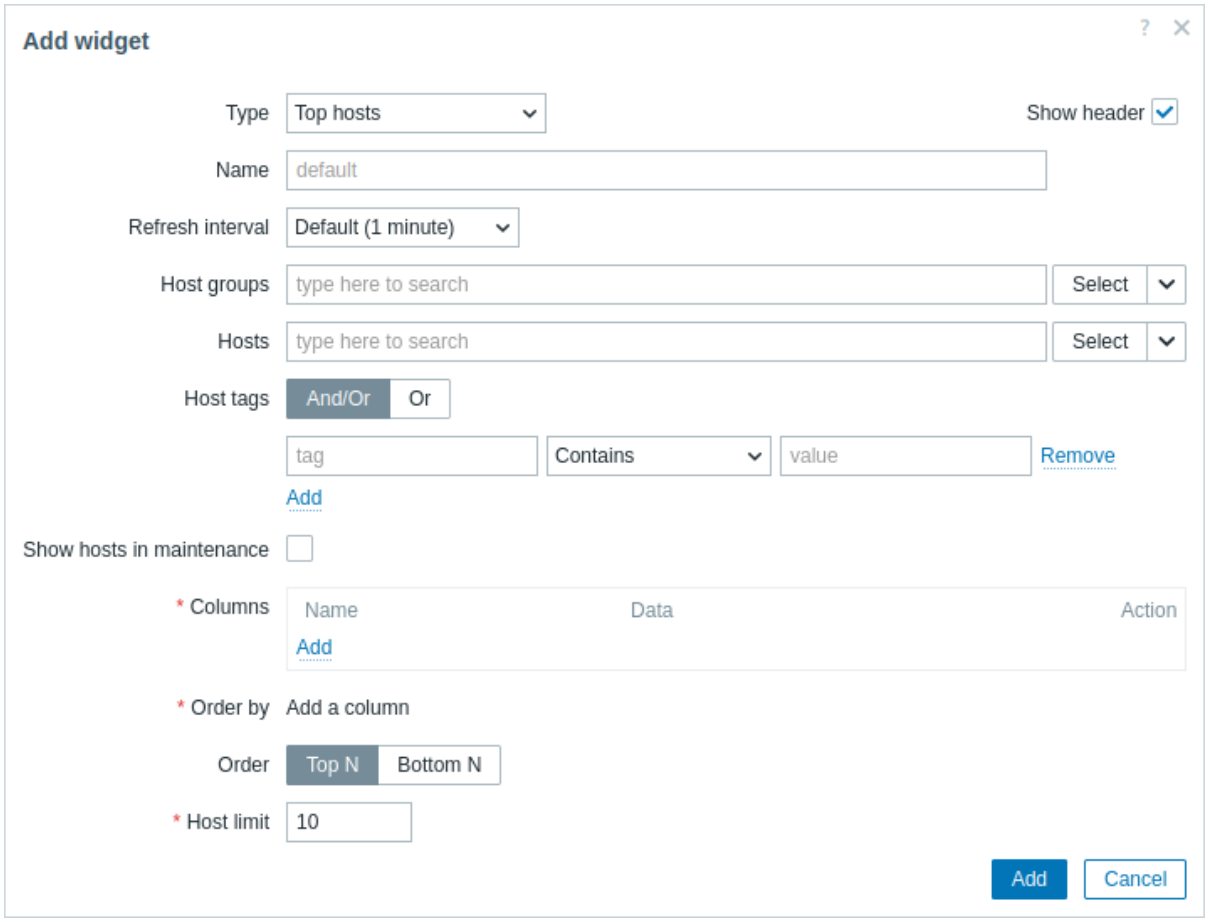
Logrīks *Top hosts* attēlo atlasīto vienumu augšējās/apakšējās vērtības no vairākiem hostiem.



Logrīks atbalsta līdz 1000 hostu vienlaicīgu attēlošanu.

Konfigurācija

Lai konfigurētu, atlasiet *Top hosts* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

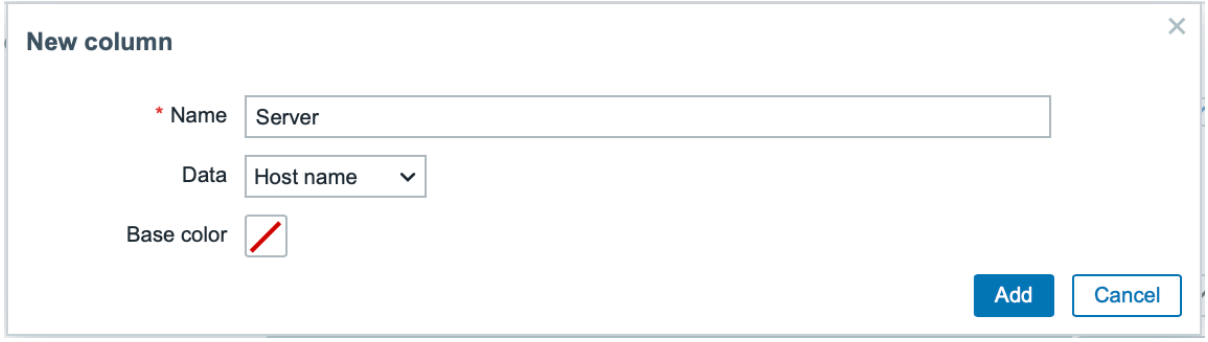
Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hostu birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto hostu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma definēšanai ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** - pietiek, ja ir izpildīts viens nosacījums. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Rādīt hostus uzturēšanā	Atzīmējiet šo izvēles rūtiņu, lai tiktu attēloti arī hosti uzturēšanā (šajā gadījumā blakus hosta nosaukumam tiks parādīta uzturēšanas ikona). Pēc noklusējuma nav atzīmēts.
Kolonnas	Pievienojiet attēlojamos datu kolonnas. Kolonnu secība nosaka to attēlojumu no kreisās uz labo pusi. Kolonnas var pārkārtot, velkot tās uz augšu vai uz leju aiz roktura pirms kolonnas nosaukuma.
Kārtot pēc	Norādiet kolonnu no definētā *Kolonnas* saraksta, ko izmantot *Top N* vai *Bottom N* kārtošanai.
Secība	Norādiet rindu kārtošanas secību: **Top N** - dilstošā secībā atbilstoši *Kārtot pēc* agregētajai vērtībai; **Bottom N** - augošā secībā atbilstoši *Kārtot pēc* agregētajai vērtībai.
Hostu limits	Attēlojamo hostu rindu skaits (1-1000). Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Kolonnu konfigurācija

Ir iespējams pievienot kolonnas ar trim datu tipiem: hosta nosaukums, vienuma vērtība vai teksts. Pieejamo kolonnas parametru saraksts ir atkarīgs no kolonnas datu tipa un vienuma vērtības tipam — no vērtības formāta.

Hosta nosaukuma kolonna

Hosta nosaukuma kolonna tiek izmantota, lai attēlotu hosta nosaukumu.



Atbalstītie parametri:

Name	Kolonnas nosaukums.
Data	Datu tips, ko attēlot kolonnā; atlasiet **Host name**.
Base color	Kolonnas fona krāsa.

Teksta kolonna

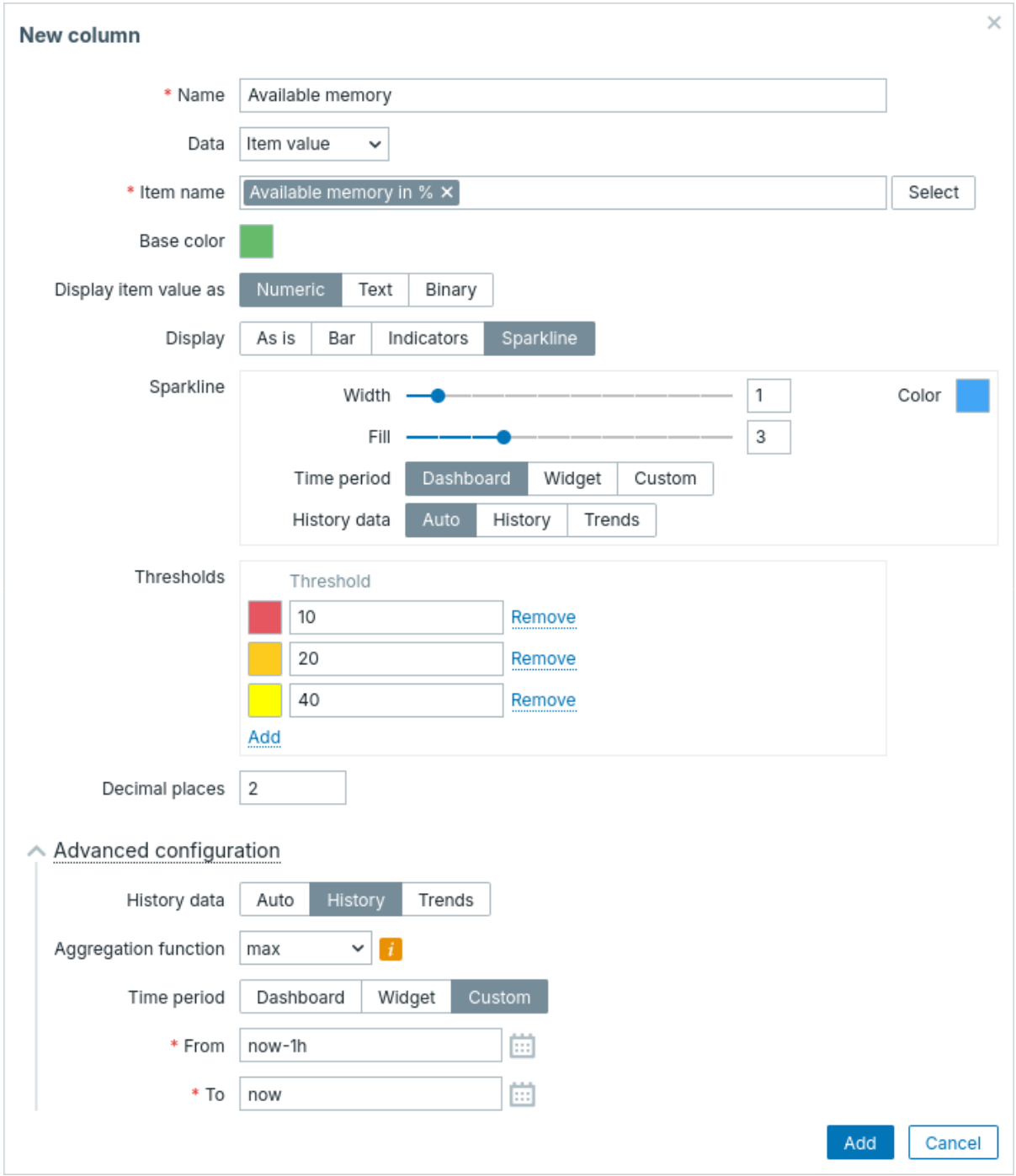
Teksta kolonna tiek izmantota, lai attēlotu jebkuru norādīto teksta virkni.

Atbalstītie parametri:

Name	Kolonnas nosaukums.
Data	Kolonnā attēlojamo datu tips, atlasiet **Text**.
Base color	Kolonnas fona krāsa.
Text	Ievadiet attēlojamo virkni. Var saturēt hosta un inventāra makrosus.

Vienuma vērtības kolonna

Vienuma vērtības kolonna tiek izmantota, lai attēlotu norādītā vienuma vērtību; tā atbalsta vairākus vērtību attēlošanas formātus un opcijas.



Atbalstītie parametri:

Name	Kolonnas nosaukums.
Data	Kolonnā attēlojamais datu tips; atlasiet **Item value**.
Item name	Atlasiet vienumu; tā nosaukums tiek izmantots, lai atrastu un attēlotu vienumus ar tādu pašu nosaukumu visos atlasītajos hostos. Ja hostam ir vairāki šādi vienumi, logrīks attēlo vienumu, kura atslēga alfabētiski ir pirmā (piemēram, *CPU utilization* ar atslēgu `proc.cpu.util`, nevis `system.cpu.util`). Konfigurējot logrīku veidnes panelī, var atlasīt tikai veidnē konfigurētos vienumus.
Base color	Kolonnas fona krāsa; aizpildījuma krāsa, ja *Item value* dati tiek attēloti kā josla/indikatori. Noklusējuma krāsu var aizstāt ar pielāgotu krāsu, ja vienuma vērtība pārsniedz kādu no norādītajiem sliekšņiem.
Display item value as	Vienuma vērtības attēlošanas formāts: **Numeric**, **Text** vai **Binary**. Atlasītā opcija nosaka, kuri papildu parametri būs pieejami. Skatiet katra formāta parametru sarakstu zemāk.
Advanced configuration	

History data	Ņemt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - ņemt vēstures datus; **Trends** - ņemt tendenču datus. Šis parametrs ir pieejams tikai skaitliskām vienuma vērtībām. Teksta un binārajām vērtībām dati vienmēr tiks ņemti no vēstures.
Aggregation function	Norādiet, kuru agregācijas funkciju izmantot atlasītajā *Time period*: **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **count** - attēlot vērtību skaitu; **sum** - attēlot vērtību summu; **first** - attēlot pirmo vērtību; **last** - attēlot pēdējo vērtību. Ja ir atlasīts **not used**, tiek attēlota jaunākā vērtība (bez agregācijas). Agregācija ļauj attēlot agregētu vērtību izvēlētajā intervālā (5 minūtes, stunda, diena), nevis jaunāko vērtību. Tikai skaitliskus datus var attēlot ar *min*, *max*, *avg* un *sum*. Parametram *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Time period	Norādiet laika periodu, ko izmantot vērtību agregēšanai. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (norādīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks attēlota pulksteņa ikona, kas, novietojot peles kursoru virs tās, norāda iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu. Šis parametrs nav pieejams, ja *Aggregation function* ir iestatīts uz "not used".
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Widget".
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".

Skaitliskas vienuma vērtības parametri:

Display	Nosakiet, kā vērtība jāattēlo: **As is** - kā parasts teksts; **Bar** - kā vienlaidus, ar krāsu aizpildīta josla; **Indicators** - kā segmentēta, ar krāsu aizpildīta josla; **Sparkline** - mini līniju diagramma.
Thresholds	Norādiet sliekšņu vērtības, pie kurām jāmainās fona/aizpildījuma krāsai. Saglabājot saraksts tiks sakārtots augošā secībā. Sparklines gadījumā sliekšņi tiek piemēroti tikai vienuma pēdējai vērtībai.
Decimal places	Norādiet, cik zīmes aiz komata tiks attēlotas vērtībai.
Bar/Indicators	
Min	Minimālā vērtība.
Max	Maksimālā vērtība.
Sparkline	
Width	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.
Color	Atlasiet līnijas un aizpildījuma krāsu.
Fill	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10.

Time period	Norādiet laika periodu vērtībām, kas jāiekļauj sparkline diagrammā. Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju **Widget** - izmantot saderīgu logrīku (norādīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks attēlota pulksteņa ikona, kas, novietojot peles kursoru virs tās, norāda iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Widget".
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
History data	Ņemt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **History** - ņemt vēstures datus; **Trends** - ņemt tendenču datus.

Teksta vienuma vērtības parametri:

Highlights	Norādiet regulārās izteiksmes, kurām atbilstot jāmainās fona/aizpildījuma krāsai.

Binārās vienuma vērtības parametri:

Show thumbnail	Norādiet, vai izveidot un attēlot sīktēlu attēlam, kas satur bināros datus, vai vērtības kolonnā attēlot hipersaiti *Show*, kas ved uz pilna izmēra attēlu.

29 Populārākie vienumi

Pārskats

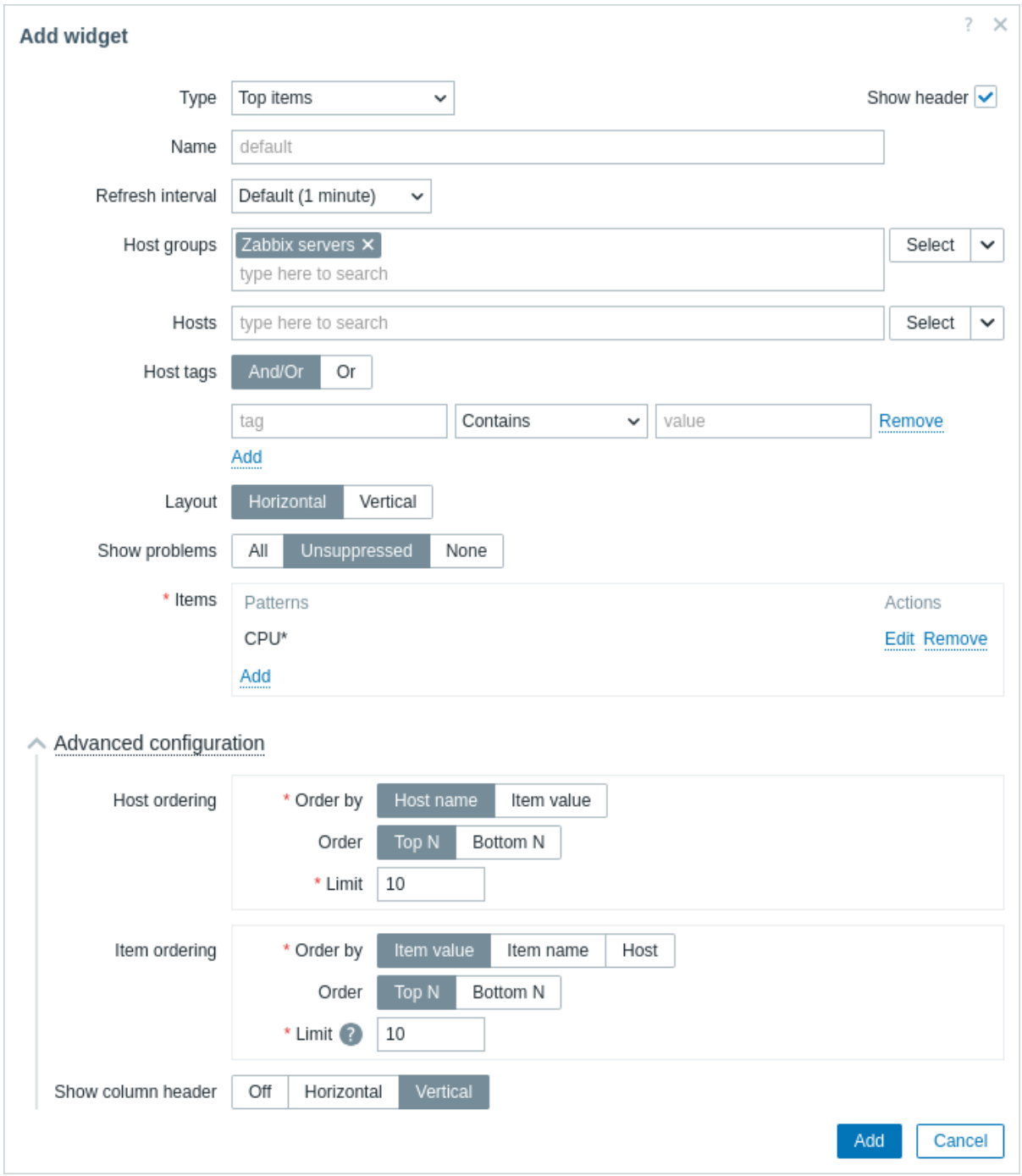
Logrīks *Top vienumi* parāda atlasīto vienumu augšējās/apakšējās vērtības, nodrošinot ātru to veiktspējas pārskatu.

Pēc noklusējuma logrīks parāda vērtības par pēdējām 24 stundām. Šo periodu var mainīt sadaļā *Administrēšana* > *Vispārīgi* > *GUI* (*Maksimālais vēstures attēlošanas periods*).

Noklikšķinot uz vienuma vērtības, tiek atvērta vienuma izvēlne.

Konfigurācija

Lai konfigurētu, atlasiet *Top items* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskās opcijas:

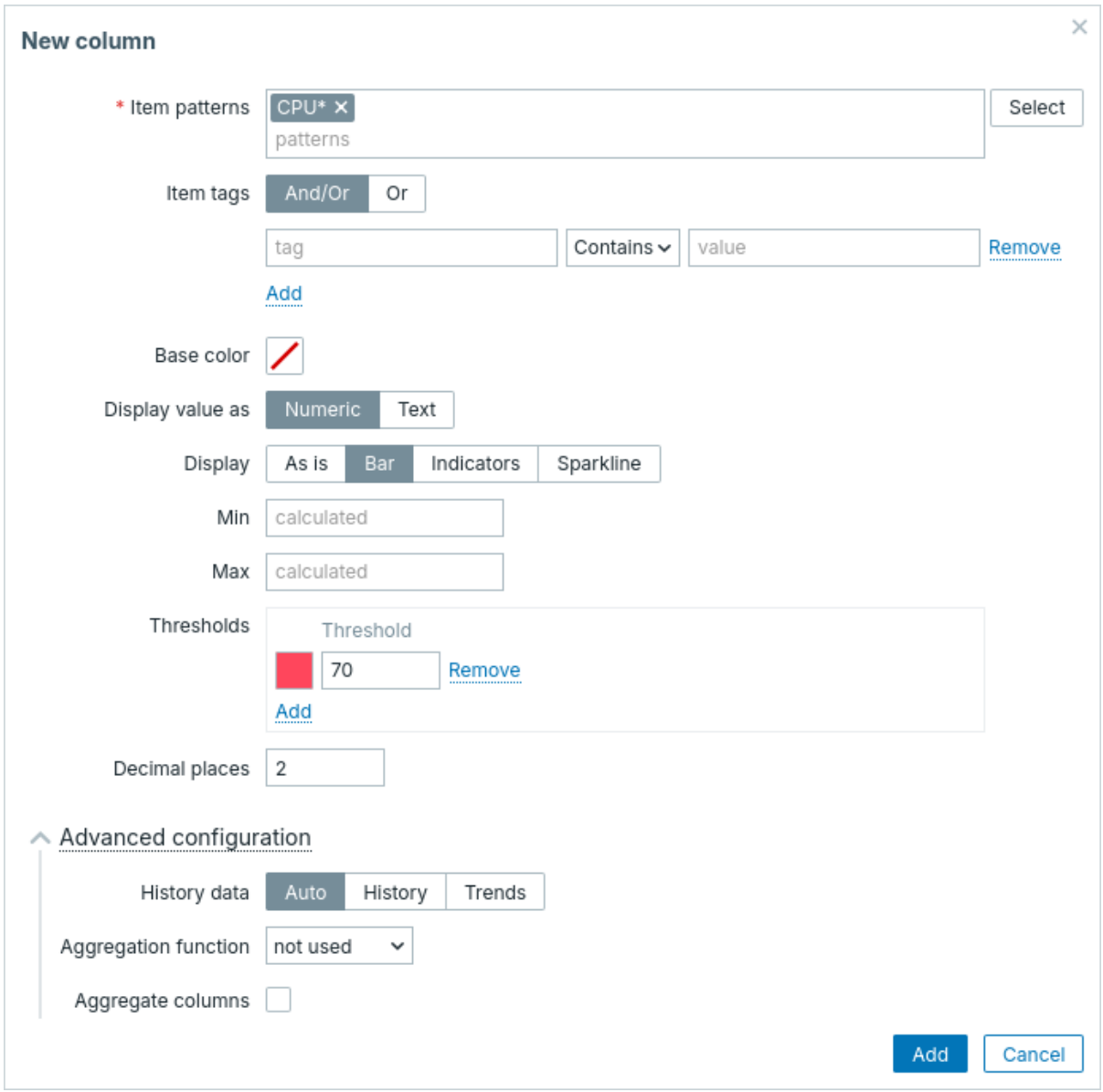
Hostu grupas	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Hostu birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto hostu datu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Or* nosacījuma; **Or** - pietiek, ja izpildās viens nosacījums.
Izkārtojums	Atlasiet izkārtojuma opciju: **Horizontal** - hostu nosaukumi tiks attēloti kreisajā pusē; **Vertical** - hostu nosaukumi tiks attēloti augšpusē.
Rādīt problēmas	Filtrējiet, kuras problēmas attēlot, pamatojoties uz to statusu (visas, neapspiestas, nevienu). Problēmas tiek norādītas ar krāsotām vienumu vērtībām, ignorējot vienumu kolonnas attēlošanas iestatījumus. Krāsas ir balstītas uz problēmas nopietnību, ko var pielāgot sadaļā *Administrēšana* > *Vispārīgi* > *Trigeru attēlošanas opcijas*. Problēmu nopietnības var mainīt, atjauninot problēmas.
Vienumi	Pievienojiet vienumu šablonus vai konkrētus vienumus attēlošanai (skatiet Kolonnu konfigurācija).
Paplašinātā konfigurācija	
Hostu kārtošana	Atlasiet hostu kolonnas/rindas kārtošanas opcijas.
Kārtot pēc	Kārtot hostus pēc: **Host name** - hosti tiks kārtoti pēc hosta nosaukuma; **Item value** - hosti tiks kārtoti pēc atlasīto vienumu vērtības.
Secība	Atlasiet, vai attēlot lielākās vai mazākās vērtības: **TopN** - augšējās N vērtības; **BottomN** - apakšējās N vērtības. N vērtība tiek atlasīta laukā *Limits*.
Limits	Ievadiet attēlojamo hostu limitu (diapazons 1-1000; noklusējums 10). Šī vērtība kļūs par N vērtību laukā *Secība*.
Vienumu kārtošana	Atlasiet vienumu kolonnas/rindas kārtošanas opcijas.
Kārtot pēc	Kārtot vienumus pēc: **Item value** - vienumi tiks kārtoti pēc vienuma vērtības; **Item name** - vienumi tiks kārtoti pēc vienuma nosaukuma; **Host** - vienumi tiks kārtoti pēc atlasītā hosta šablona.
Secība	Atlasiet, vai attēlot lielākās vai mazākās vērtības: **TopN** - augšējās N vērtības; **BottomN** - apakšējās N vērtības. N vērtība tiek atlasīta laukā *Limits*.
Limits	Ievadiet attēlojamo vienumu limitu (diapazons 1-1000; noklusējums 10). Šī vērtība kļūs par N vērtību laukā *Secība*.

Rādīt kolonnas galveni	Atlasiet kolonnas galvenes attēlošanas opcijas: **Off** - nerādīt kolonnas galveni; **Horizontal** - attēlot tekstu galvenē horizontāli; **Vertical** - attēlot tekstu galvenē vertikāli.

Kolonnas konfigurācija

Lai konfigurētu vienumu kolonnas, parametrā *Items* noklikšķiniet uz *Add*:



Kopīgie kolonnas parametri:

Item patterns	Norādiet vienu vai vairākus vienumu šablonus, izmantojot aizstājējzīmi. Alternatīvi atlasiet vienumus. Konfigurējot logrīku veidnes panelī, var atlasīt tikai veidnē konfigurētos vienumus.

Item tags	Norādiet birkas, lai ierobežotu logrīkā attēloto vienumu datu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **And/Or** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Or* nosacījuma; **Or** - pietiek, ja ir izpildīts viens nosacījums.
Base color	Atlasiet kolonnas fona krāsu vai aizpildījuma krāsu, ja *Display* ir iestatīts uz ”Bar” vai ”Indicators”. Ņemiet vērā, ka pamatkrāsu var aizstāt sliekšņa vai izcelšanas krāsas.
Display value as	Formāts vienuma vērtības attēlošanai: **Numeric** vai **Text**. Atlasītā opcija nosaka, kuri papildu parametri būs pieejami. Skatiet katra formāta parametru sarakstu tālāk.

Kolonnu parametri skaitliskām vērtībām:

Attēlojums	Definējiet, kā vērtība jāattēlo: **Kā ir** - kā parasts teksts; **Josla** - kā vienlaidus, ar krāsu aizpildīta josla; **Indikatori** - kā segmentēta, ar krāsu aizpildīta josla; **Dzirksteļlīnija** - mini līniju diagramma.
Min	Minimālā vērtība joslas/indikatoru attēlojumam.
Max	Maksimālā vērtība joslas/indikatoru attēlojumam.
Platums	Iestatiet diagrammas līnijas biezumu, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10. Šis parametrs ir paredzēts tikai dzirksteļlīnijas attēlojumam.
Aizpildījums	Iestatiet aizpildījuma krāsas caurspīdīguma līmeni, izmantojot slīdni vai manuāli ievadot vērtību diapazonā no 0 līdz 10. Šis parametrs ir paredzēts tikai dzirksteļlīnijas attēlojumam.
Krāsa	Atlasiet līnijas un aizpildījuma krāsu. Šis parametrs ir paredzēts tikai dzirksteļlīnijas attēlojumam.
Skatiet *Paplašinātā konfigurācija* lauku aprakstu, kas saistīti ar laika periodu un vēstures datu atlasi.	
Sliekšņi	Norādiet sliekšņa vērtības, pie kurām jāmainās fona/aizpildījuma krāsai. Saglabājot saraksts tiks sakārtots augošā secībā. Dzirksteļlīnijām sliekšņi tiek piemēroti tikai vienuma pēdējai vērtībai.
Zīmju skaits aiz komata	Norādiet, cik zīmes aiz komata tiks attēlotas kopā ar vērtību.
Paplašinātā konfigurācija	
Vēstures dati	Ņemt datus no vēstures vai tendencēm: **Auto** - automātiska atlase; **Vēsture** - ņemt vēstures datus; **Tendences** - ņemt tendenču datus. Šis parametrs ir pieejams tikai skaitliskiem vienumu vērtībām. Teksta vērtībām dati vienmēr tiks ņemti no vēstures.

Agregācijas funkcija	Norādiet, kuru agregācijas funkciju izmantot atlasītajā *Laika periodā*: **min** - attēlot mazāko vērtību; **max** - attēlot lielāko vērtību; **avg** - attēlot vidējo vērtību; **count** - attēlot vērtību skaitu; **sum** - attēlot vērtību summu; **first** - attēlot pirmo vērtību; **last** - attēlot pēdējo vērtību. Ja ir atlasīts **not used**, tiek attēlota jaunākā vērtība (bez agregācijas). Agregācija ļauj attēlot agregētu vērtību izvēlētajam intervālam (5 minūtes, stunda, diena), nevis jaunāko vērtību. *min*, *max*, *avg* un *sum* var attēlot tikai skaitliskus datus. Parametram *count* neskaitliskie dati tiks pārveidoti par skaitliskiem.
Laika periods	Norādiet laika periodu, ko izmantot vērtību agregēšanai. Atlasiet laika perioda datu avotu: **Panelis** - izmantot paneļa laika perioda atlasītāju; **Logrīks** - izmantot saderīgu logrīku (iestatīts parametrā *Logrīks*); **Pielāgots** - izmantot pielāgotu laika periodu, kas iestatīts parametros *No* un *Līdz*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks attēlota pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Laika perioda* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu. Šis parametrs nav pieejams, ja *Agregācijas funkcija* ir iestatīta uz "not used".
Logrīks	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Widget".
No	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Custom".
Līdz	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Laika periods* ir iestatīts uz "Custom".
Agregēt kolonnas	Ja atzīmēts, grupēt visus vienumus, kas atbilst kolonnas vienumu šabloniem, vienā apvienotā kolonnā/rindā. Atbilstošo vienumu vērtības tiks agregētas, izmantojot atlasīto *Kolonnu agregācijas funkciju*, un attēlotas zem *Apvienotās kolonnas nosaukuma*. Noklusējums: izslēgts (nav atzīmēts). Šis parametrs tiek rādīts tikai tad, ja *Attēlot vērtību kā* ir **Skaitliska**.
Kolonnu agregācijas funkcija	Izvēlieties, kā tiek apvienotas vērtības no grupētajiem vienumu šabloniem. Pieejamās opcijas: **min**, **max**, **avg**, **count**, **sum** (noklusējums). Šis parametrs tiek rādīts tikai tad, ja ir atzīmēts *Agregēt kolonnas*.
Apvienotās kolonnas nosaukums	Ievadiet agregētās kolonnas attēlojamo nosaukumu. Šis lauks ir obligāts, ja ir atzīmēts *Agregēt kolonnas*, un kļūst par kolonnas galveni apvienotajām vērtībām.

Kolonnu parametri teksta vērtībām:

Izcēlumi	Norādiet regulārās izteiksmes, kurām atbilstot jāmainās fona/aizpildījuma krāsai.

30 populārākie trigeri

Pārskats

Logrīks *Top trigeri* attēlo trigerus ar lielāko problēmu skaitu.

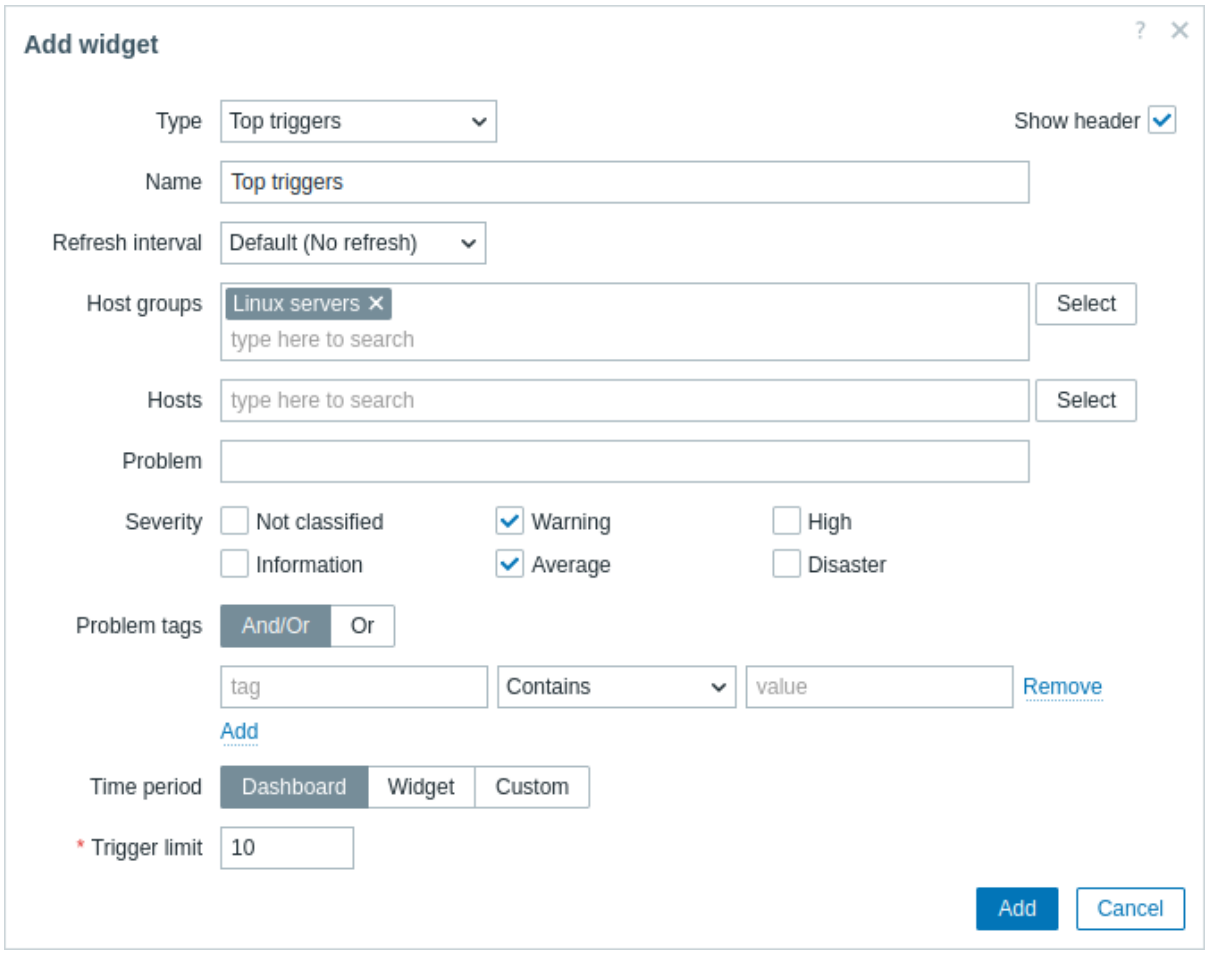
Top triggers

Host	Trigger	Severity	Number of problems
Zabbix server	Interface enp0s3: Link down	Average	2
Zabbix server	Load average is too high	Average	2
Zabbix server	Zabbix agent is not available	Average	2
Zabbix server	Zabbix server: More than 100 items having missing data for more than 10 minutes	Warning	2
Zabbix server	Zabbix server: Utilization of escalator processes is high	Average	2

Tas attēlo tos pašus datus kā *Atskaites > Top 100 trigeri* un var attēlot līdz 1000 ierakstiem.

Konfigurācija

Lai konfigurētu, atlasiet *Top triggers* kā logrīka tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas, kuru problēmas rādīt logrīkā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Šo hostu grupu problēmas tiks rādītas logrīkā; ja netiek ievadīta neviena hostu grupa, tiks rādītas visu hostu grupu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, kuru problēmas rādīt logrīkā. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošo hostu nolaižamais saraksts. Ja netiek ievadīts neviens hosts, tiks rādītas visu hostu problēmas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Problēma	Varat skatīt tikai noteiktām problēmām atbilstošos trigerus. Lai to izdarītu, ievadiet virkni, kas jāsalīdzina ar problēmas nosaukumu. Makrosi netiek izvērsti.

Svarīgums	Atzīmējiet trigeru svarīguma līmeņus, lai filtrētu logrīkā rādāmos trigerus. Ja nav atzīmēts neviens svarīguma līmenis, tiks rādīti visi trigeri.
Problēmu birkas	Norādiet problēmu birkas, kas jārāda logrīkā. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem, savukārt nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Or* nosacījuma; **Or** - pietiek, ja izpildās viens nosacījums.
Laika periods	Atlasiet laika perioda datu avotu: **Dashboard** - izmantot paneļa laika perioda atlasītāju; **Widget** - izmantot saderīgu logrīku (iestatīts parametrā *Widget*); **Custom** - izmantot pielāgotu laika periodu, kas iestatīts parametros *From* un *To*; ja tas ir iestatīts, logrīka augšējā labajā stūrī tiks rādīta pulksteņa ikona, kas, novietojot peles kursoru virs tās, norādīs iestatīto laiku. Ņemiet vērā, ka neatkarīgi no logrīka *Time period* konfigurācijas saderīgi logrīki joprojām var to izmantot kā laika perioda datu avotu.
Widget	Ievadiet vai atlasiet saderīgu logrīku kā laika perioda datu avotu. Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Widget".
From	Ievadiet vai atlasiet laika perioda sākumu. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
To	Ievadiet vai atlasiet laika perioda beigas. Tiek atbalstīta relatīvā laika sintakse (`now`, `now/d`, `now/w-1w` utt.). Šis parametrs ir pieejams, ja *Time period* ir iestatīts uz "Custom".
Trigeru limits	Iestatiet rādāmo trigeru skaitu. Iespējamais vērtību diapazons: 1-1000.

31 Trigeru pārskats

Pārskats

Logrīks *Trigeru pārskats* attēlo pašreizējos hostu trigeru stāvokļus kā krāsainu bloku tabulu. Tas nodrošina ātru vizuālu pārskatu par trigeru veselību un aktivitāti dažādos hostos.

Logrīks izmanto vairākus vizuālos elementus, lai attēlotu informāciju par trigeriem:

- PROBLEM trigeru bloku krāsas atbilst problēmu nopietnības līmeņiem.
- Mirgojoši bloki norāda uz nesenām trigeru stāvokļa izmaiņām (pēdējo 2 minūšu laikā).
- Pelēkas bultiņas norāda uz trigeriem ar atkarībām (virziet kursoru virs tām, lai skatītu detalizētu informāciju).
- Izvēles rūtiņas ikonas norāda uz apstiprinātām problēmām; lai šī ikona tiktu parādīta, visām trigerim aktīvajām vai atrisinātajām problēmām jābūt apstiprinātām.

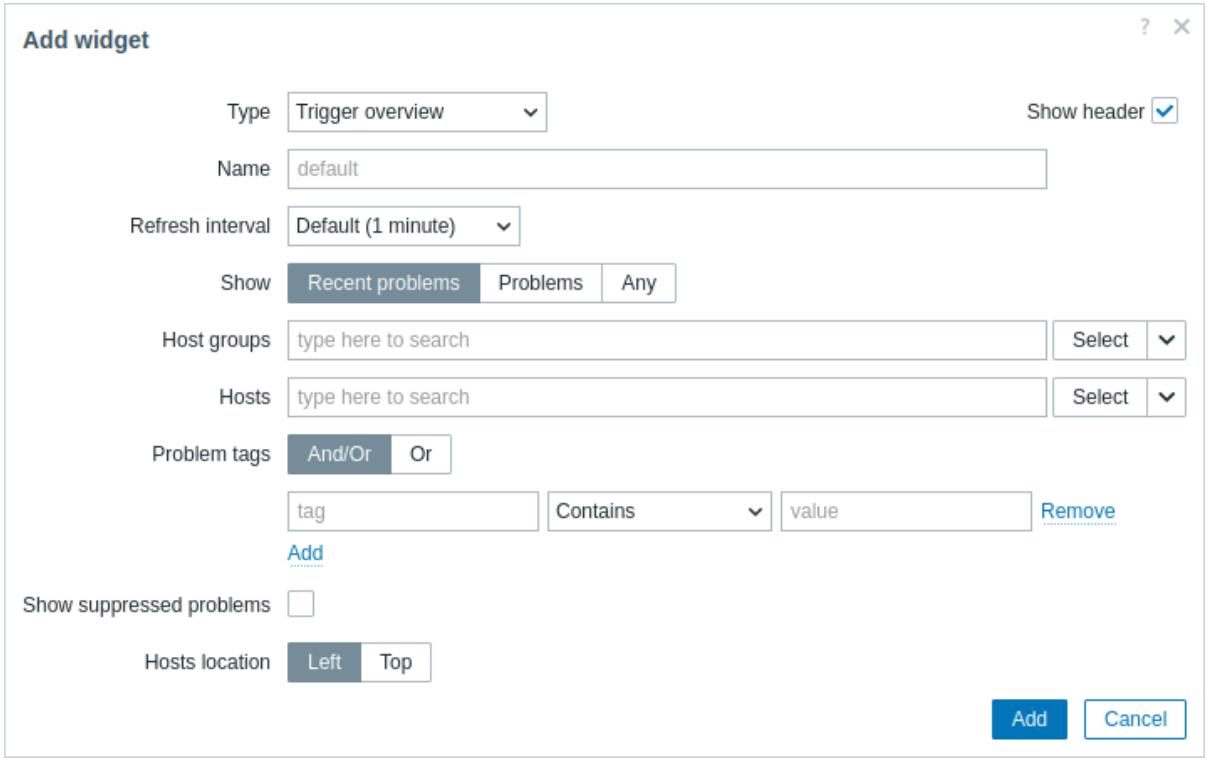
Note:
Trigeru nopietnības līmeņu krāsas var pielāgot sadaļā *Administration* > *General*. Trigeru nopietnības līmeņus var mainīt, atjauninot problēmas.

Noklikšķinot uz bloka, tiek piedāvātas saites uz ar trigeri saistītu informāciju (problēmas, trigeru konfigurācija u.c.).

Pēc noklusējuma tiek attēloti līdz 50 ierakstiem (to var konfigurēt sadaļā *Administration* > *General* > *GUI*, izmantojot iestatījumu *Max number of columns and rows in overview tables*). Ja ir vairāk trigeru, tiek parādīts ziņojums ar aicinājumu precizēt filtrēšanas kritērijus. Šis ierobežojums tiek piemērots pirms jebkādas papildu filtrēšanas (piemēram, pēc tagiem).

Konfigurācija

Lai konfigurētu, atlasiet *Trigeru pārskats* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

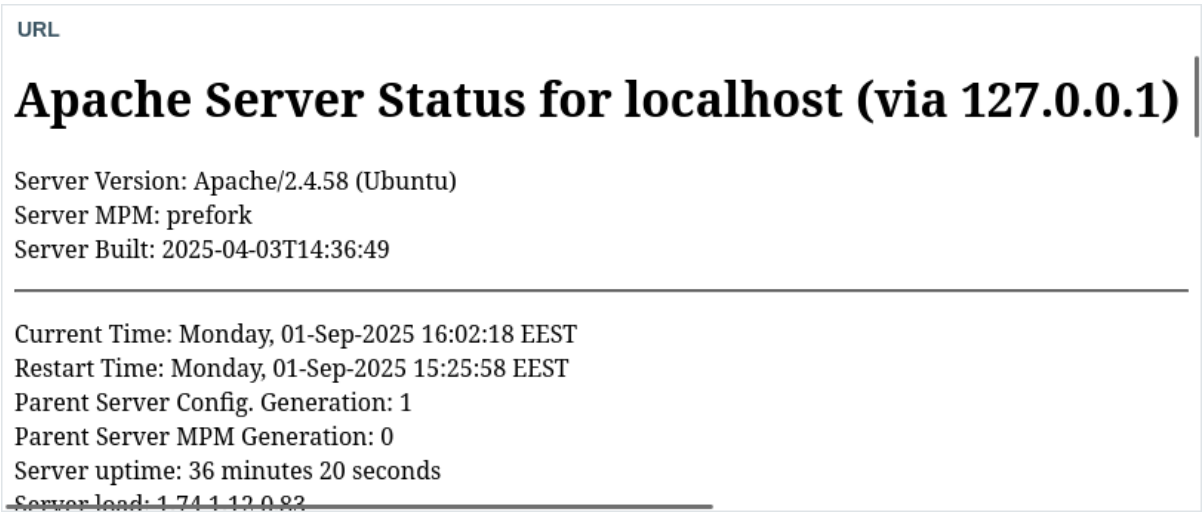
Rādīt	Filtrēt trigerus pēc trigeru stāvokļa: **Nesenās problēmas** - *(noklusējums)* rādīt trigerus, kas nesen bija vai joprojām ir PROBLEM stāvoklī (atrisinātas un neatrisinātas); **Problēmas** - rādīt trigerus, kas ir PROBLEM stāvoklī (neatrisinātas); **Jebkuri** - rādīt visus trigerus.
Hostu grupas	Atlasiet hostu grupas. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošu grupu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko pabeigšanu, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošu hostu nolaižamais saraksts. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Problēmu birkas	Norādiet birkas, lai filtrētu logrīkā attēlotos trigerus. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. **Piezīme:** Ja parametrs *Rādīt* ir iestatīts uz 'Jebkuri', tiks attēloti visi trigeri pat tad, ja ir norādītas birkas. Tomēr, kamēr nesenās trigeru stāvokļa izmaiņas (attēlotas kā mirgojoši bloki) tiks atjauninātas visiem trigeriem, trigeru stāvokļa detaļas (problēmas nopietnības krāsa un tas, vai problēma ir apstiprināta) tiks atjauninātas tikai tiem trigeriem, kas atbilst norādītajām birkām. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** - pietiek, ja izpildās viens nosacījums.
Rādīt apspiestās problēmas	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas dēļ.
Izkārtojums	Atlasiet izkārtojuma opciju: **Horizontāls** - hostu nosaukumi tiks attēloti kreisajā pusē; **Vertikāls** - hostu nosaukumi tiks attēloti augšpusē.

32 URL

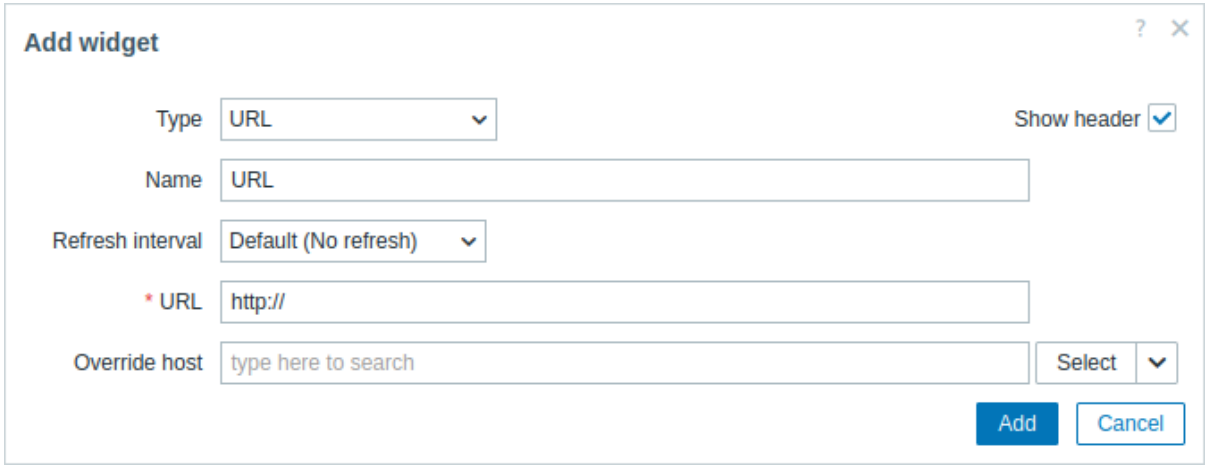
Pārskats

URL logdaļa attēlo saturu, kas iegūts no URL.



Konfigurācija

Lai konfigurētu, atlasiet *URL* kā tipu:



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

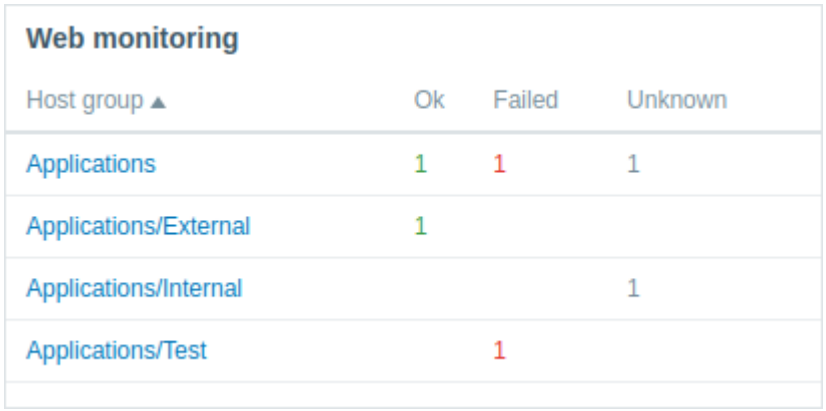
URL	Ievadiet attēlojamo URL (līdz 2048 rakstzīmēm). Ārējiem URL jāsākas ar `http://` vai `https://`. Iekšējie URL atbalsta relatīvos ceļus (piemēram, `zabbix.php?action=report.status`). Tiek atbalstīti {HOST.*} makrosi.
Override host	Atlasiet saderīgu logrīku vai paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Attention:
Pārlūkprogrammas var neielādēt logrīkā konfigurētu HTTP lapu, ja Zabbix lietotāja saskarnei piekļūst, izmantojot HTTPS. Tāpat, iespējams, būs jāmaina X-Frame-Options HTTP response header, lai attēlotu saturu ārpus Zabbix lietotāja saskarnes.

33 Tīmekļa uzraudzība

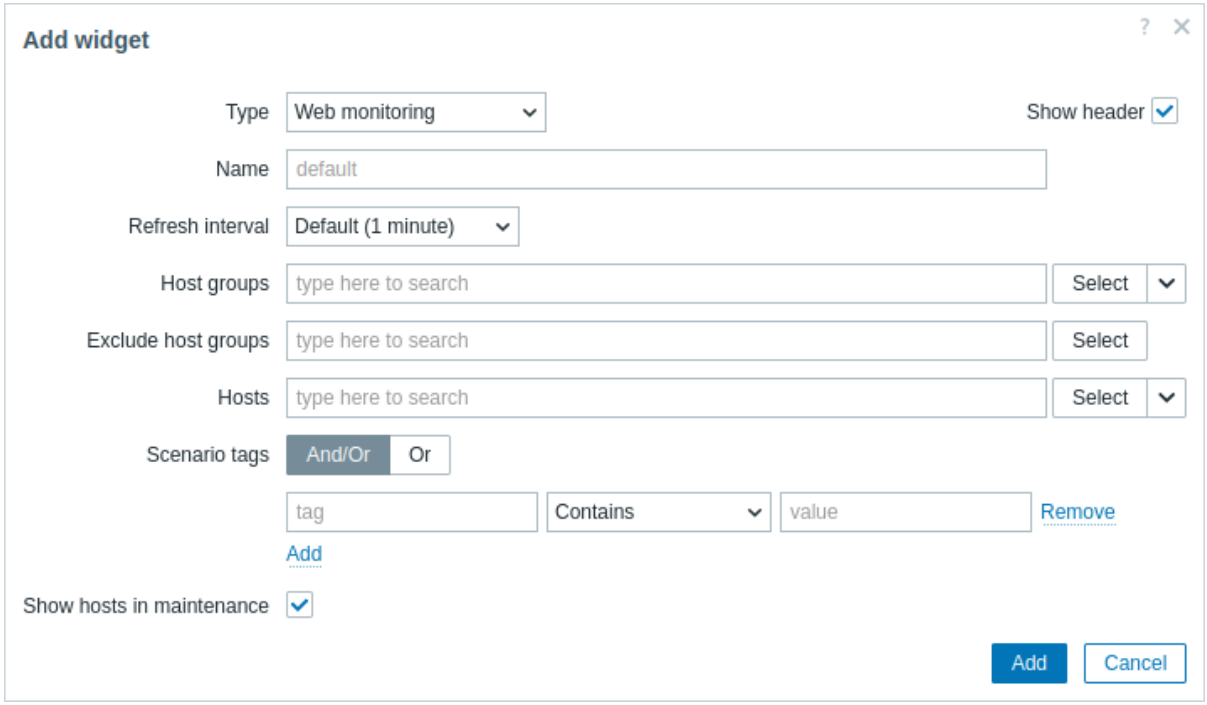
Pārskats

Logrīks *Tīmekļa uzraudzība* attēlo aktīvo tīmekļa scenāriju statusu.



Noklikšķinot uz hostu grupas nosaukuma, tiek atvērta sadaļa *Uzraudzība* > *Hosti* > *Tīmekļa scenāriji*, kur var skatīt scenārija detalizētu informāciju.

Konfigurācija



Papildus parametriem, kas ir kopīgi visiem logrīkiem, varat iestatīt šādas specifiskas opcijas:

Hostu grupas	Atlasiet hostu grupas, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku kā datu avotu hostu grupām. Šis lauks atbalsta automātisko aizpildi, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Šo hostu grupu hostu dati tiks attēloti logrīkā; ja netiek ievadīta neviena hostu grupa, tiks attēlotas visas hostu grupas. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Izslēgt hostu grupas	Atlasiet hostu grupas, ko paslēpt logrīkā. Šis lauks atbalsta automātisko aizpildi, tāpēc, sākot rakstīt grupas nosaukumu, tiks piedāvāts atbilstošo grupu nolaižamais saraksts. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Šo hostu grupu hostu dati logrīkā netiks attēloti. Piemēram, hosti 001, 002, 003 var būt Grupā A, bet hosti 002, 003 arī Grupā B. Ja vienlaikus atlasām *rādīt* Grupu A un *izslēgt* Grupu B, panelī tiks attēloti tikai hosta 001 dati. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.
Hosti	Atlasiet hostus, ko attēlot logrīkā. Alternatīvi atlasiet saderīgu logrīku vai paneli kā datu avotu hostiem. Šis lauks atbalsta automātisko aizpildi, tāpēc, sākot rakstīt hosta nosaukumu, tiks piedāvāts atbilstošo hostu nolaižamais saraksts. Ja netiek ievadīts neviens hosts, tiks attēloti visi hosti. Šis parametrs nav pieejams, konfigurējot logrīku veidnes panelī.

Scenāriju birkas	Norādiet birkas, lai ierobežotu logrīkā attēloto tīmekļa scenāriju skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - jāizpildās visiem nosacījumiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc *Vai* nosacījuma; **Vai** - pietiek, ja izpildās viens nosacījums.
Rādīt hostus uzturēšanā	Iekļaut statistikā hostus, kas atrodas uzturēšanā. Šis parametrs ir apzīmēts kā *Rādīt datus uzturēšanā*, konfigurējot logrīku veidnes panelī.

2 Uzraudzība

Pārskats

Izvēlne Monitoring ir paredzēta datu attēlošanai. Neatkarīgi no tā, kādas informācijas apkopošanai, vizualizēšanai un apstrādei ir konfigurēts Zabbix, tā tiks attēlota dažādās izvēlnes Monitoring sadaļās.

Skata režīma pogas

Tālāk norādītās pogas, kas atrodas augšējā labajā stūrī, ir kopīgas katrai sadaļai:

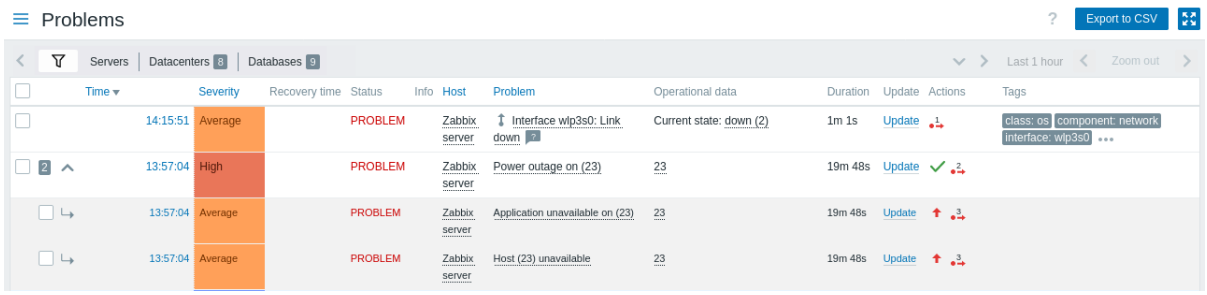
	Parādīt lapu kioska režīmā. Šajā režīmā tiek parādīts tikai lapas saturs. Lai izietu no kioska režīma, pārvietojiet peles kursoru, līdz parādās izejas poga , un noklikšķiniet uz tās. Jūs tiksiet atgriezts parastajā režīmā.

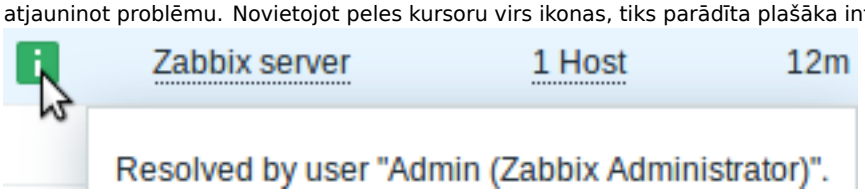
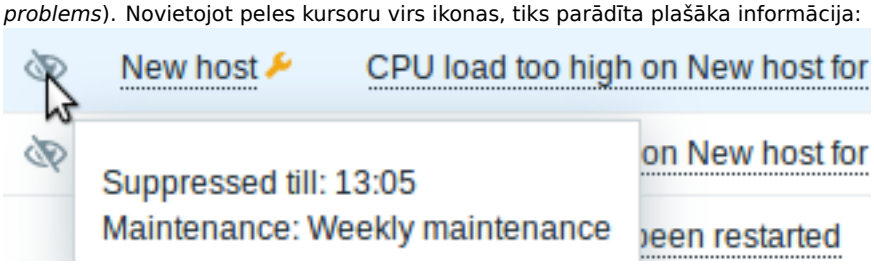
1 Problēmas

Pārskats

Sadaļā *Monitoring* > *Problems* jūs varat redzēt, kādas problēmas jums pašlaik ir. Problēmas ir tie trigeri, kas atrodas stāvoklī "Problem".

Pēc noklusējuma visas jaunās problēmas tiek klasificētas kā cēloņa problēmas. Ir iespējams noteiktas problēmas manuāli pārklasificēt par cēloņa problēmas simptomu problēmām. Plašāku informāciju skatiet sadaļā cēloņa un simptoma notikumi.



Kolonna	Apraksts
Checkbox	Tiek parādītas izvēles rūtiņas problēmu atlasei. Ikonām blakus izvēles rūtiņām ir šāda nozīme: - simptoma notikumu skaits cēloņa problēmai; - izvērst, lai parādītu simptoma notikumus; - sakļaut, lai paslēptu simptoma notikumus; - šis ir simptoma notikums.
Time	Tiek parādīts problēmas sākuma laiks.
Severity	Tiek parādīta problēmas nopietnība. Problēmas nopietnība sākotnēji ir balstīta uz pamatā esošā problēmas trigera nopietnību, taču pēc notikuma iestāšanās to var atjaunināt, izmantojot *Update problem* ekrānu. Problēmas nopietnības krāsa tiek izmantota kā šūnas fona krāsa problēmas laikā.
Recovery time	Tiek parādīts problēmas atrisināšanas laiks.
Status	Tiek parādīts problēmas statuss: **Problem** - neatrisināta problēma **Resolved** - nesen atrisināta problēma. Nesen atrisinātās problēmas var paslēpt, izmantojot filtru. Jaunas un nesen atrisinātas problēmas mirgo 2 minūtes. Atrisinātās problēmas kopumā tiek rādītas 5 minūtes. Abas šīs vērtības ir konfigurējamas sadaļā *Administration* > *General* > *Trigger displaying options*.
Info	Zaļa informācijas ikona tiek parādīta, ja problēma ir aizvērta ar globālo korelāciju vai manuāli, atjauninot problēmu. Novietojot peles kursoru virs ikonas, tiks parādīta plašāka informācija:  Šī ikona tiek parādīta, ja tiek rādīta apspiesta problēma (skatiet filtra opciju *Show suppressed problems*). Novietojot peles kursoru virs ikonas, tiks parādīta plašāka informācija: 
Host	Tiek parādīts problēmas hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne.
Problem	Tiek parādīts problēmas nosaukums. Problēmas nosaukums ir balstīts uz pamatā esošā problēmas trigera nosaukumu. Makrosi trigera nosaukumā tiek atrisināti problēmas rašanās brīdī, un atrisinātās vērtības vairs netiek atjauninātas. *Ņemiet vērā*, ka problēmas nosaukumam ir iespējams pievienot operatīvos datus, kas parāda dažas jaunākās vienumu vērtības. Noklikšķinot uz problēmas nosaukuma, tiek atvērta notikuma izvēlne. Turot kursoru virs ikonas aiz problēmas nosaukuma, tiks parādīts trigera apraksts (tām problēmām, kurām tas ir).
Operational data	Tiek parādīti operatīvie dati, kas satur jaunākās vienumu vērtības. Operatīvie dati var būt teksta un vienuma vērtību makrosu kombinācija, ja tas ir konfigurēts trigera līmenī. Ja trigera līmenī operatīvie dati nav konfigurēti, tiek parādītas visu izteiksmē iekļauto vienumu jaunākās vērtības. Šī kolonna tiek parādīta tikai tad, ja filtrā opcijai *Show operational data* ir atlasīts *Separately*.
Duration	Tiek parādīts problēmas ilgums. Skatiet arī: Negatīvs problēmas ilgums
Update	Noklikšķiniet uz saites *Update*, lai pārietu uz problēmas atjaunināšanas ekrānu, kur problēmai var veikt dažādas darbības, tostarp pievienot komentārus un apstiprināt problēmu.

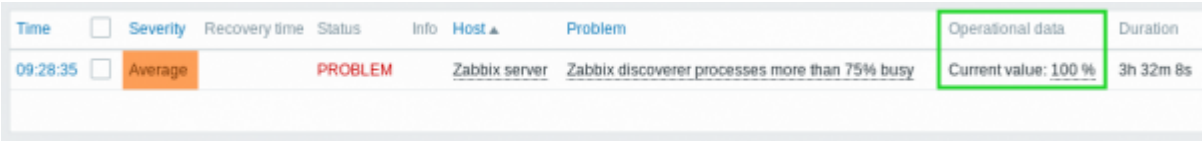
Kolonna	Apraksts
Actions	Tiek parādīta ar problēmu saistīto darbību vēsture, izmantojot simboliskas ikonas: - problēma ir apstiprināta. Šī ikona vienmēr tiek parādīta pirmā. - ir pievienoti komentāri. Tiek parādīts arī komentāru skaits. - problēmas nopietnība ir palielināta (piemēram, Information > Warning). - problēmas nopietnība ir samazināta (piemēram, Warning > Information). - problēmas nopietnība ir mainīta, bet atgriezta sākotnējā līmenī (piemēram, Warning > Information > Warning). - problēma ir apspiesta. - problēma vairs nav apspiesta. - darbības ir veiktas. Tiek parādīts arī darbību skaits. - darbības ir veiktas, vismaz viena ir izpildes procesā. Tiek parādīts arī darbību skaits. - darbības ir veiktas, vismaz viena ir neizdevusies. Tiek parādīts arī darbību skaits. Novietojot peles kursoru virs ikonām, tiek parādīti uznirstošie logi ar detalizētu informāciju par darbību. Skatiet detalizētas informācijas skatīšanu, lai uzzinātu vairāk par ikonām, kas tiek izmantotas darbību uznirstošajā logā.
Tags	Tiek parādīti tagi (ja tādi ir). Papildus var tikt parādīti arī tagi no ārējas biļešu sistēmas (skatiet opciju *Process tags*, konfigurējot webhooks).

Problēmu operatīvie dati

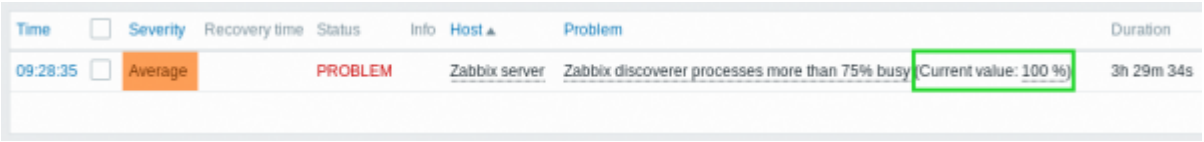
Ir iespējams attēlot pašreizējo problēmu operatīvos datus, t.i., jaunākās vienumu vērtības pretstatā vienumu vērtībām problēmas rašanās brīdī.

Operatīvo datu attēlošanu var konfigurēt filtrā *Monitoring* > *Problems* vai attiecīgā paneļa logrīka konfigurācijā, izvēloties vienu no trim opcijām:

- *None* - operatīvie dati netiek attēloti
- *Separately* - operatīvie dati tiek attēloti atsevišķā kolonnā



- *With problem name* - operatīvie dati tiek pievienoti problēmas nosaukumam iekavās. Operatīvie dati tiek pievienoti problēmas nosaukumam tikai tad, ja trigera konfigurācijā lauks *Operational data* nav tukšs.



Operatīvo datu saturu var konfigurēt katram trigerim laukā *Operational data*. Šis lauks pieņem patvaļīgu virkni ar makrosiem, no kuriem vissvarīgākais ir makross `{ITEM.LASTVALUE<1-9>}`.

`{ITEM.LASTVALUE<1-9>}` šajā laukā vienmēr tiks aizvietots ar jaunākajām vienumu vērtībām trigera izteiksmē. `{ITEM.VALUE<1-9>}` šajā laukā tiks aizvietots ar vienumu vērtībām trigera stāvokļa maiņas brīdī (t.i., pārejot uz `**Problem**`, pārejot uz `**OK**`, lietotājam manuāli aizverot vai aizverot ar korelāciju).

Ņemiet vērā, ka problēmas manuāla aizvēršana nerada jaunu vērtību, tāpēc `{ITEM.LASTVALUE<1-9>}` vai `{ITEM.VALUE<1-9>}` aizvietotā vērtība joprojām rādīs vērtību no problēmas rašanās brīža.

`{ITEM.LASTVALUE<1-9>}` vai `{ITEM.VALUE<1-9>}` tiks aizvietots ar *UNKNOWN*, ja jaunākā vēstures vērtība ir savākta agrāk nekā pirms *Max history display period* laika (skatiet *Administration > General*).

Negatīvs problēmas ilgums

Dažās bieži sastopamās situācijās patiešām ir iespējams negatīvs problēmas ilgums, t. i., kad problēmas atrisināšanas laiks ir agrāks par problēmas izveides laiku, piemēram:

- Ja kāds hosts tiek uzraudzīts ar starpniekserveri un notiek tīkla kļūda, kā rezultātā kādu laiku no starpniekservera netiek saņemti dati, serveris aktivizēs nodata(/host/key) trigeri. Kad savienojums tiek atjaunots, serveris no starpniekservera saņems vienuma datus ar laiku no pagātnes. Tad nodata(/host/key) problēma tiks atrisināta, un tai būs negatīvs problēmas ilgums;
- Ja vienuma dati, kas atrisina problēmas notikumu, tiek nosūtīti ar Zabbix sender un satur laikspiedolu, kas ir agrāks par problēmas izveides laiku, tiks parādīts arī negatīvs problēmas ilgums.

Note:
Negatīvs problēmas ilgums nekādā veidā neietekmē konkrēta trigera SLA calculation vai Availability report; tas ne samazina, ne palielina problēmas laiku.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Masveida atjaunināšana* - atjaunināt atlasītās problēmas, pārejot uz problēmu atjaunināšanas ekrānu

Lai izmantotu šo opciju, atzīmējiet izvēles rūtiņas pie attiecīgajām problēmām, pēc tam noklikšķiniet uz pogas *Masveida atjaunināšana*.

Pogas

Poga labajā pusē piedāvā šādu opciju:

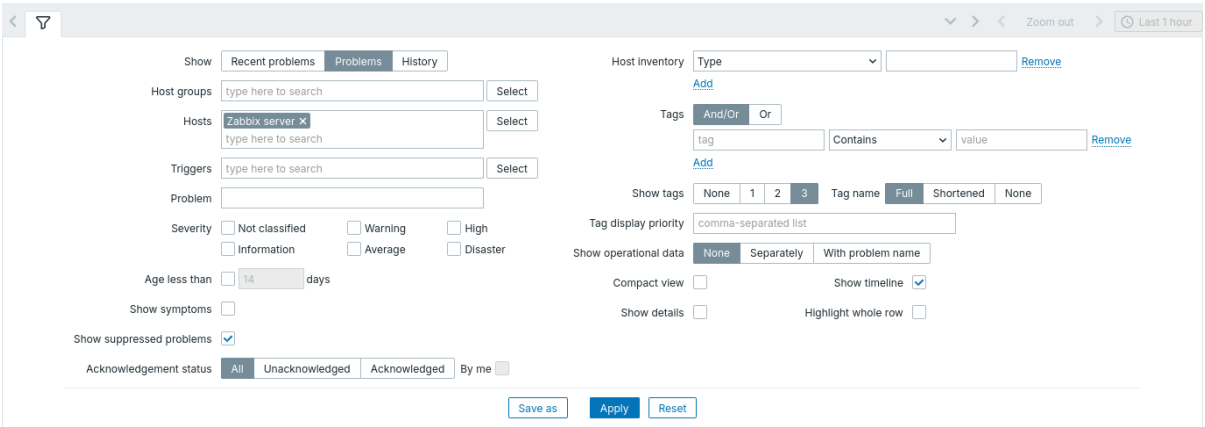
Export to CSV	Eksportēt saturu no visām lapām uz CSV failu.

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tās problēmas, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrosu vērtībām.

Filtrs atrodas virs tabulas. Iecienītos filtra iestatījumus var saglabāt kā cilnes un pēc tam ātri atvērt, noklikšķinot uz cilnēm virs filtra.



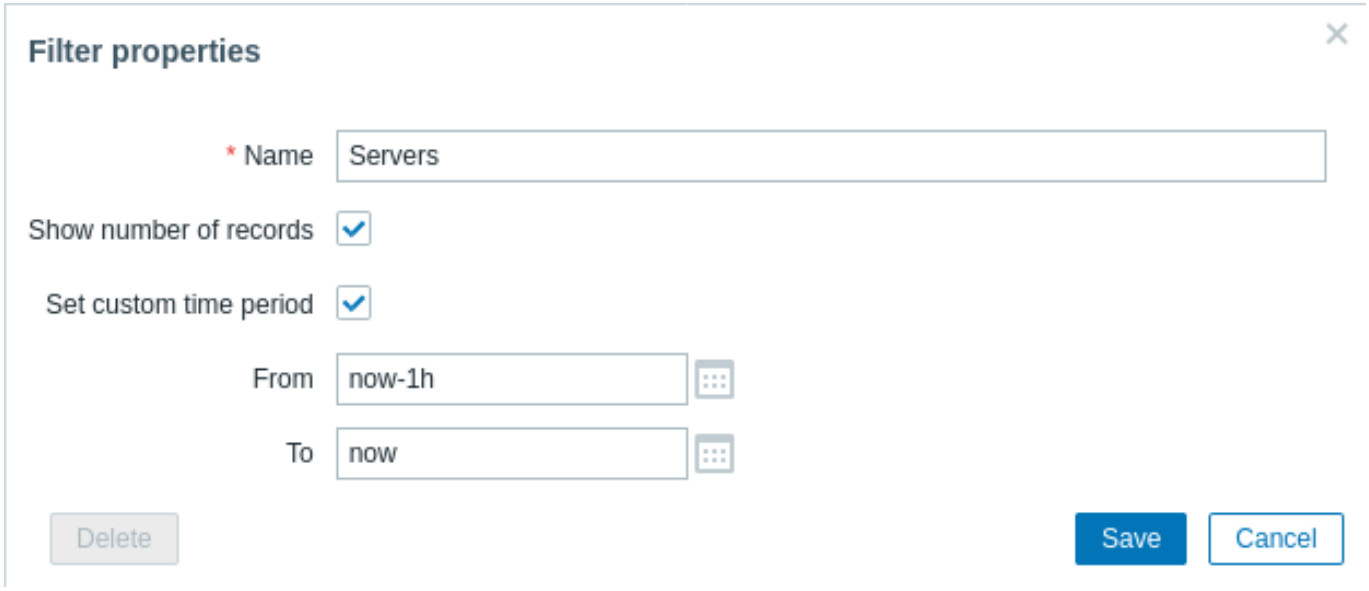
Parametrs	Apraksts
Show	Filtrēt pēc problēmas statusa: **Recent problems** - tiek attēlotas neatrisinātas un nesen atrisinātas problēmas (noklusējums) **Problems** - tiek attēlotas neatrisinātas problēmas **History** - tiek attēlota visu notikumu vēsture
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas.
Hosts	Filtrēt pēc viena vai vairākiem hostiem.
Triggers	Filtrēt pēc viena vai vairākiem trigeriem.
Problem	Filtrēt pēc problēmas nosaukuma.

Parametrs	Apraksts
Severity	Filtrēt pēc trigeru (problēmu) svarīguma pakāpes.
Age less than	Filtrēt pēc tā, cik sena ir problēma.
Show symptoms	Atzīmējiet izvēles rūtiņu, lai atsevišķā rindā attēlotu problēmas, kas klasificētas kā simptomi.
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas vai atsevišķas problēmu apspiešanas dēļ.
Acknowledgement status	Filtrs, lai attēlotu visas problēmas, tikai neapstiprinātās problēmas vai tikai apstiprinātās problēmas. Atzīmējiet papildu izvēles rūtiņu, lai izfiltrētu tās problēmas, kuras jebkad esat apstiprinājis jūs.
Host inventory	Filtrēt pēc inventarizācijas tipa un vērtības.
Tags	Filtrēt pēc notikuma taga nosaukuma un vērtības. Ir iespējams gan iekļaut, gan izslēgt noteiktus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos tagu nosaukumus **Equals** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Contains** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Does not exist** - izslēgt norādītos tagu nosaukumus **Does not equal** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Does not contain** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina tipi: **And/Or** - jāizpildās visiem nosacījumiem; nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma **Or** - pietiek, ja izpildās viens nosacījums Filtrējot, šeit norādītie tagi tiks attēloti pirmie kopā ar problēmu, ja vien to nepārraksta saraksts *Tag display priority* (skatīt zemāk).
Show tags	Atlasiet attēlojamo tagu skaitu: **None** - sadaļā *Monitoring* > *Problems* nav kolonnas *Tags* **1** - kolonnā *Tags* ir viens tags **2** - kolonnā *Tags* ir divi tagi **3** - kolonnā *Tags* ir trīs tagi Lai redzētu visus problēmas tagus, novietojiet peles kursoru virs trīs punktu ikonas.
Tag name	Atlasiet taga nosaukuma attēlošanas režīmu: **Full** - tagu nosaukumi un vērtības tiek attēloti pilnā apjomā **Shortened** - tagu nosaukumi tiek saīsināti līdz 3 simboliem; tagu vērtības tiek attēlotas pilnā apjomā **None** - tiek attēlotas tikai tagu vērtības; nosaukumi netiek rādīti
Tag display priority	Ievadiet problēmas tagu attēlošanas prioritāti kā ar komatiem atdalītu tagu sarakstu (piemēram: `customer,scope,component`). Jāizmanto tikai tagu nosaukumi, bez vērtībām. Šajā sarakstā esošie tagi vienmēr tiks attēloti pirmie, ignorējot dabisko kārtošanu alfabētiskā secībā.
Show operational data	Atlasiet operatīvo datu attēlošanas režīmu: **None** - operatīvie dati netiek attēloti **Separately** - operatīvie dati tiek attēloti atsevišķā kolonnā **With problem name** - pievienot operatīvos datus problēmas nosaukumam, izmantojot iekavas operatīvajiem datiem
Compact view	Atzīmējiet izvēles rūtiņu, lai iespējotu kompakto skatu.
Show details	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmu pamatā esošās trigeru izteiksmes. Atspējots, ja ir atzīmēta izvēles rūtiņa *Compact view*.
Show timeline	Atzīmējiet izvēles rūtiņu, lai attēlotu vizuālo laika skalu un grupēšanu. Atspējots, ja ir atzīmēta izvēles rūtiņa *Compact view*.
Highlight whole row	Atzīmējiet izvēles rūtiņu, lai izceltu visu rindu neatrisinātām problēmām. Izcelšanai tiek izmantota problēmas svarīguma pakāpes krāsa. *Highlight whole row* nav pieejams augsta kontrasta tēmās.

Cilnes iecienītākajiem filtriem

Bieži izmantotas filtra parametru kopas var saglabāt cilnēs.

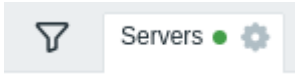
Lai saglabātu jaunu filtra parametru kopu, atveriet galveno cilni un konfigurējiet filtra iestatījumus, pēc tam nospiediet pogu *Save as*. Jaunā uznirstošajā logā norādiet *Filter properties*.



Parametrs	Apraksts
Name	Filtra nosaukums, kas tiks parādīts ciļņu sarakstā.
Show number of records	Atzīmējiet, ja vēlaties, lai blakus cilnes nosaukumam tiktu parādīts problēmu skaits.
Override time period selector	Atzīmējiet, lai šai filtra kopai iestatītu konkrētu noklusējuma laika periodu. Ja tas ir iestatīts, šīs cilnes laika periodu varēsiet mainīt tikai, atjauninot filtra iestatījumus. Cilnēm bez pielāgota laika perioda laika diapazonu var mainīt, nospiežot laika atlasītāja pogu augšējā labajā stūrī (pogas nosaukums ir atkarīgs no izvēlētā laika intervāla: This week, Last 30 minutes, Yesterday u.c.). Šī opcija ir pieejama tikai filtriem sadaļā *Monitoring* > *Problems*.
From/To	Laika perioda sākums un beigas absolūtā (`Y-m-d H:i:s`) vai relatīvā laika sintaksē (`now-1d`). Pieejams, ja ir atzīmēta opcija *Set custom time period*.

Pēc saglabāšanas filtrs tiek izveidots kā nosaukta filtra cilne un nekavējoties aktivizēts.

Lai rediģētu esoša filtra filtra īpašības, nospiediet zobrata simbolu blakus aktīvās cilnes nosaukumam.



Piezīmes:

- Lai paslēptu filtra apgabalu, noklikšķiniet uz pašreizējās cilnes nosaukuma. Vēlreiz noklikšķiniet uz aktīvās cilnes nosaukuma, lai atkal atvērtu filtra apgabalu.
- Tiek atbalstīta navigācija ar tastatūru: izmantojiet bultiņas, lai pārslēgtos starp cilnēm, nospiediet *Enter*, lai atvērtu.
- Lai pārslēgtos starp saglabātajiem filtriem, var izmantot kreiso/labo pogu virs filtra. Savukārt poga ar lejupvērstu bultiņu atver nolaižamo izvēlni ar visiem saglabātajiem filtriem, un jūs varat noklikšķināt uz vajadzīgā.
- Filtru cilnes var pārkārtot, velkot un nometot.
- Ja saglabāta filtra iestatījumi ir mainīti (bet nav saglabāti), aiz filtra nosaukuma tiek parādīts zaļš punkts. Lai atjauninātu filtru atbilstoši jaunajiem iestatījumiem, noklikšķiniet uz pogas *Update*, kas tiek parādīta pogas *Save as* vietā.
- Pašreizējie filtra iestatījumi tiek saglabāti lietotāja profilā. Kad lietotājs atkal atver lapu, filtra iestatījumi paliek tādi paši.

Note:
Lai koplietotu filtrus, nokopējiet aktīvā filtra URL un nosūtiet to citiem. Pēc šī URL atvēršanas citi lietotāji varēs saglabāt šo parametru kopu kā pastāvīgu filtru savā Zabbix kontā.
 Skatiet arī: Lapas parametri.

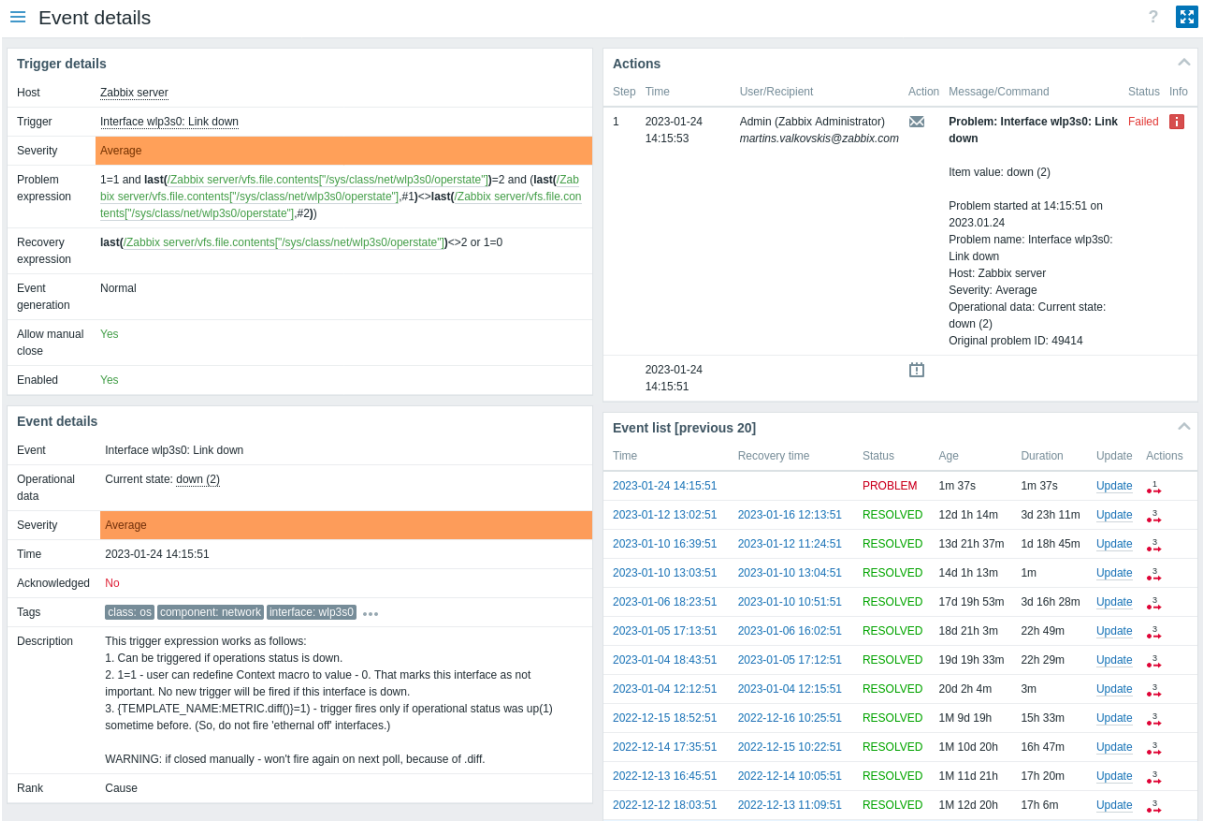
Filtra pogas

Apply	Piemērot norādītos filtrēšanas kritērijus (nesaglabājot).
Reset	Atiestatīt pašreizējo filtru un atgriezties pie pašreizējās cilnes saglabātajiem parametriem. Galvenajā cilnē tas notīrīs filtru.

Save as	Saglabāt pašreizējos filtra parametrus jaunā cilnē. Pieejams tikai galvenajā cilnē.
Update	Aizstāt cilnes parametrus ar pašlaik norādītajiem parametriem. Nav pieejams galvenajā cilnē.

Detalizētas informācijas skatīšana

Problēmas sākuma un atkopšanas laiki sadaļā *Monitoring* > *Problems* ir saites. Noklikšķinot uz tām, tiek atvērta detalizētāka informācija par notikumu.



Ņemiet vērā, ka problēmas nopietnības līmenis trigerim un problēmas notikumam var atšķirties — ja tas ir atjaunināts problēmas notikumam, izmantojot *Update problem* ekrānu.

Darbību sarakstā aktivitātes tipa apzīmēšanai tiek izmantotas šādas ikonas:

- - ģenerēts problēmas notikums.
- - ziņojums ir nosūtīts.
- - problēmas notikums ir apstiprināts.
- - problēmas notikums nav apstiprināts.
- - ir pievienots komentārs.
- - problēmas nopietnības līmenis ir paaugstināts (piemēram, Information > Warning).
- - problēmas nopietnības līmenis ir pazemināts (piemēram, Warning > Information).
- - problēmas nopietnības līmenis ir mainīts, bet atgriezts sākotnējā līmenī (piemēram, Warning > Information > Warning).
- - ir izpildīta attālā komanda.
- - problēmas notikums ir atkopies.
- - problēma ir aizvērta manuāli.
- - problēma ir apspiesta.

- - problēma vairs nav apspiesta.
- - problēma ir pārveidota par simptomu problēmu.
- - problēma ir pārveidota par cēloņa problēmu.

1 Cēloņu un simptomu problēmas

Pārskats

Pēc noklusējuma visas jaunās problēmas tiek klasificētas kā cēloņa problēmas. Ir iespējams manuāli pārklasificēt noteiktas problēmas kā cēloņa problēmas simptomu problēmas.

Piemēram, elektroapgādes pārtraukums var būt patiesais pamatcēlonis, kāpēc kāds hosts nav sasniedzams vai kāds pakalpojums nedarbojas. Šajā gadījumā problēmas "hosts nav sasniedzams" un "pakalpojums nedarbojas" ir jāklasificē kā "elektroapgādes pārtraukuma" — cēloņa problēmas — simptomu problēmas.

Cēloņa un simptoma hierarhija atbalsta tikai divus līmeņus. Problēmai, kas jau ir simptoms, nevar piešķirt "pakārtotas" simptomu problēmas; visas problēmas, kas piešķirtas kā simptomi simptomu problēmai, kļūs par tās pašas cēloņa problēmas simptomiem.

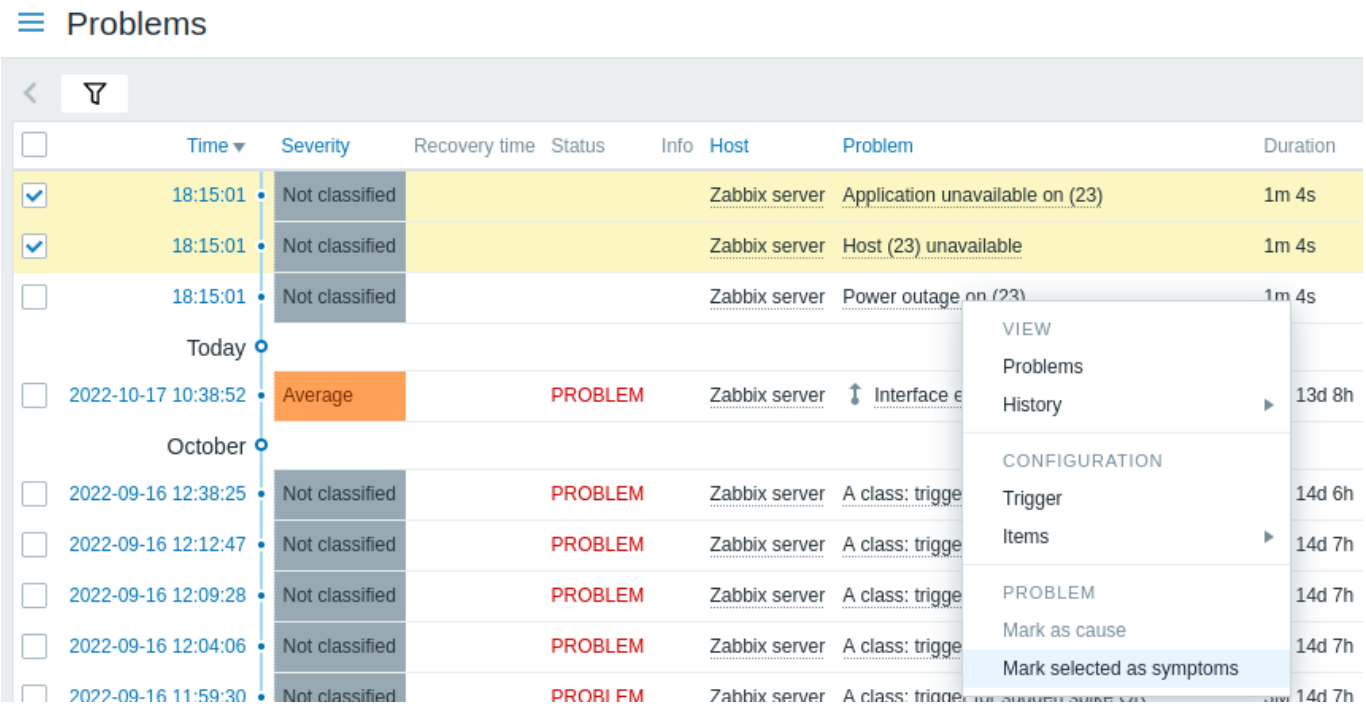
Tikai cēloņa problēmas tiek ieskaitītas kopējā problēmu skaitā kartēs, paneļa logrīkos, piemēram, *Problēmas pēc svarīguma* vai *Problēmu hosti*, utt. Tomēr problēmu ranžēšana neietekmē pakalpojumus.

Simptoma problēmu var saistīt tikai ar vienu cēloņa problēmu. Simptomu problēmas netiek automātiski atrisinātas, ja cēloņa problēma tiek atrisināta vai aizvērta.

Konfigurācija

Lai pārklasificētu problēmu par simptomu problēmu, vispirms atlasiet to problēmu sarakstā. Var atlasīt vienu vai vairākas problēmas.

Pēc tam pārejiet uz cēloņa problēmu un tās konteksta izvēlnē noklikšķiniet uz opcijas *Atzīmēt atlasīto kā simptomus*.



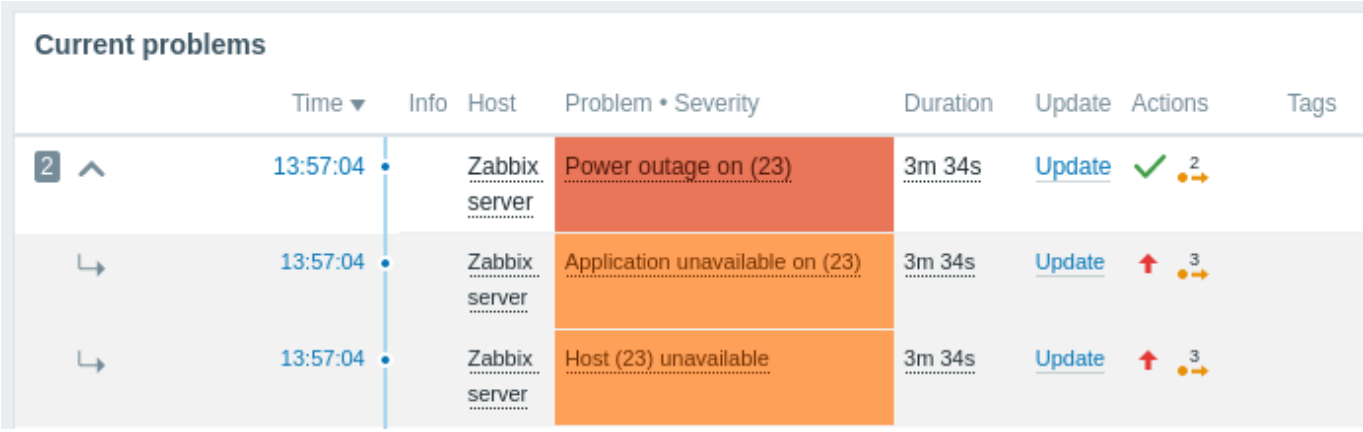
Pēc tam serveris atjaunināsatlasītās problēmas, padarot tās par cēloņa problēmas simptomu problēmām.

Kamēr problēmas statuss tiek atjaunināts, tas tiek attēlots vienā no diviem veidiem:

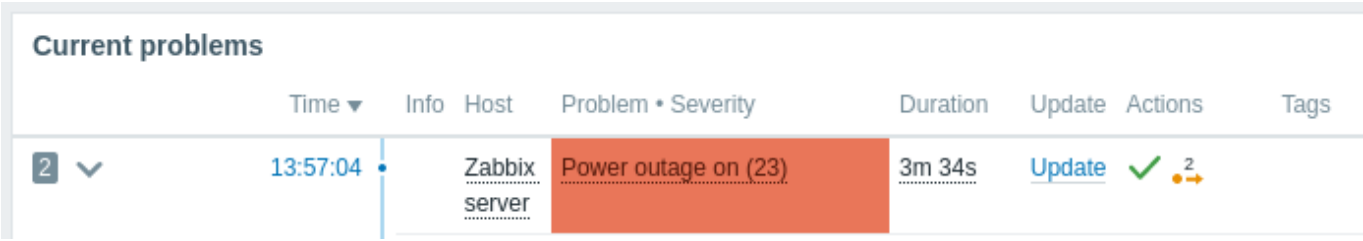
- Kolonnā Status tiek rādīts mirgojošs statuss "UPDATING";
- Kolonnā Info tiek rādīta mirgojoša ikona vai (tas ir spēkā, ja filtrā ir atlasīts tikai *Problems* un tādēļ kolonna Status netiek rādīta).

Attēlošana

Simptomu problēmas tiek attēlotas zem cēloņa problēmas un attiecīgi atzīmētas sadaļā *Monitoring -> Problems* (un paneļa logrīkā *Problems*) — ar ikonu, mazāku fontu un atšķirīgu fonu.



Sakļautajā skatā ir redzama tikai cēloņa problēma; par simptomu problēmu esamību norāda skaitlis rindas sākumā un ikona skata izvēršanai.



Ir iespējams arī papildus attēlot simptomu problēmas parastā fontā un atsevišķā rindā. Lai to izdarītu, filtra iestatījumos vai logrīka konfigurācijā atlasiet *Show symptoms*.

Atgriešana uz cēloņa problēmu

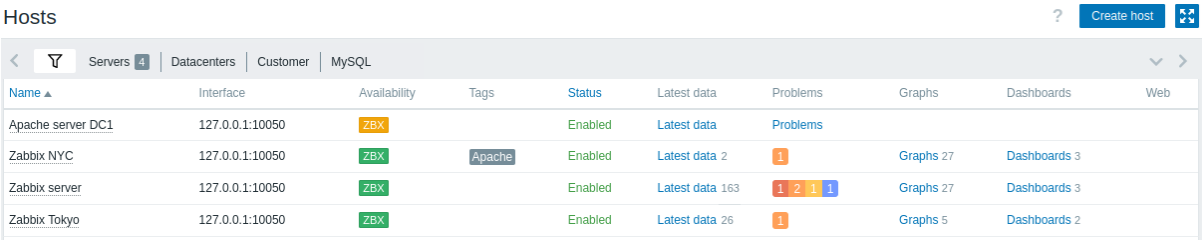
Simptoma problēmu var atgriezt atpakaļ uz cēloņa problēmu. Lai to izdarītu, veiciet vienu no šīm darbībām:

- noklikšķiniet uz opcijas *Atzīmēt kā cēloni* simptoma problēmas konteksta izvēlnē;
- atzīmējiet opciju *Pārveidot par cēloni* problēmas atjaunināšanas ekrānā un noklikšķiniet uz *Atjaunināt* (šī opcija darbosies arī tad, ja būs atlasītas vairākas problēmas).

2 Hosti

Pārskats

Sadaļā *Uzraudzība* → *Hosti* tiek parādīts pilns uzraudzīto hostu saraksts ar detalizētu informāciju par hosta saskarni, pieejamību, tagiem, pašreizējām problēmām, statusu (iespējots/atspējots) un saitēm, lai ērti pārietu uz hosta jaunākajiem datiem, problēmu vēsturi, grafikiem, paneļiem un tīmekļa scenārijiem.



Column	Description
Name	Redzamais hosta nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta hosta izvēlne. Oranža uzgriežņu atslēgas ikona aiz nosaukuma norāda, ka šim hostam ir apkopes režīms. Noklikšķiniet uz kolonnas virsraksta, lai kārtotu hostus pēc nosaukuma augošā vai dilstošā secībā.
Interface	Tiek parādīta hosta galvenā saskarne.

Column	Description
Availability	Tiek parādīta hosta pieejamība katrai konfigurētajai saskarnei. Pieejamības ikonas attēlo hosta saskarnes pašreizējo statusu Zabbix serverī. Tāpēc, ja lietotāja saskarnē atspējojat hostu, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas. Līdzīgi, ja hostu iespējojat, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas un aptaujājis hostu. Pieejamības ikonas attēlo tikai tos saskarņu tipus (Agent, SNMP, IPMI, JMX), kas ir konfigurēti. Novietojot kursoru virs ikonas, tiek parādīts uznirstošais logs ar visu tā paša tipa saskarņu sarakstu, detalizētu informāciju, statusu un kļūdām. Agent saskarnei uznirstošajā logā tiek parādītas saskarnes (pasīvās) un aktīvās pārbaudes. Ja hostam ir tikai aktīvās pārbaudes, Agent saskarnes ikona tiek parādīta pat tad, ja hostam nav konfigurēta Agent saskarne. Kolonna ir tukša hostiem bez saskarnēm. Viena hosta saskarnes statusu nosaka savienojums starp iespējotu vienumu, kas izmanto šo saskarni, un hostu. Statuss var būt: **Pieejams** - savienojums ar hostu bija veiksmīgs; **Nav pieejams** - savienojums ar hostu nebija veiksmīgs (taimauts, ugunsmūra problēmas u.c.); **Nezināms** - savienojums ar hostu nav mēģināts vai rezultāts nav zināms. Papildu informāciju par to, kā Zabbix serveris nosaka saskarnes statusu, skatiet Nezināms saskarnes statuss un Nesasniedzama/nepieejama hosta saskarnes iestatījumi. Visu viena tipa hosta saskarņu (Agent, SNMP, IPMI, JMX) statusu nosaka tās saskarnes, kuras izmanto vismaz viens iespējots vienums. Statusu norāda ikonas krāsa: **Zaļa** - visas saskarnes ir pieejamas; **Dzeltena** - vismaz viena saskarne nav pieejama, un vismaz viena ir pieejama vai nezināma; **Sarkana** - neviena saskarne nav pieejama; **Pelēka** - vismaz vienas saskarnes statuss ir nezināms, bet neviena nav nepieejama. **Aktīvo pārbaužu pieejamība.** Ja hostā ir iespējota vismaz viena aktīvā pārbaude, aktīvās pārbaudes arī ietekmē kopējo Agent saskarnes pieejamību, kā aprakstīts iepriekš. Lai noteiktu aktīvo pārbaužu pieejamību, aģenta aktīvo pārbaužu pavedienā tiek sūtīti heartbeat ziņojumi. Heartbeat ziņojumu biežumu kontrolē parametrs `HeartbeatFrequency` Zabbix aģenta vai aģenta 2 konfigurācijā (noklusējums 60 sekundes, diapazons 0-3600). Aktīvās pārbaudes tiek uzskatītas par nepieejamām, ja aktīvās pārbaudes heartbeat ir vecāks par 2 x HeartbeatFrequency sekundēm. **Piezīme:** Zabbix aģenti, kas ir vecāki par versiju 6.2.x, nesūta aktīvo pārbaužu heartbeat ziņojumus, tāpēc to hostu pieejamība paliek nezināma.
Tags	Hosta un visu piesaistīto veidņu tagi, ar neatrisinātām makro vērtībām.
Status	Hosta statuss - *Iespējots* vai *Atspējots*. Noklikšķiniet uz kolonnas virsraksta, lai kārtotu hostus pēc statusa augošā vai dilstošā secībā.
Latest data	Noklikšķinot uz saites, tiks atvērta lapa *Uzraudzība* **-** *Jaunākie dati* ar visiem jaunākajiem datiem, kas savākti no hosta. Vienumu skaits ar jaunākajiem datiem tiek parādīts pelēkā krāsā.
Problems	Atvērto hosta problēmu skaits, sakārtots pēc smaguma pakāpes. Kvadrāta krāsa norāda problēmas smaguma pakāpi. Skaitlis kvadrātā nozīmē problēmu skaitu attiecīgajai smaguma pakāpei. Noklikšķinot uz ikonas, tiks atvērta lapa *Uzraudzība* **-** *Problēmas* pašreizējam hostam. Ja hostam nav problēmu, saite uz šī hosta sadaļu *Problēmas* tiek parādīta kā teksts. Izmantojiet filtru, lai izvēlētos, vai jāiekļauj apspiestās problēmas (pēc noklusējuma nav iekļautas).
Graphs	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie grafiki. Grafiku skaits tiek parādīts pelēkā krāsā. Ja hostam nav grafiku, saite netiek parādīta.
Dashboards	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie paneļi. Paneļu skaits tiek parādīts pelēkā krāsā. Ja hostam nav paneļu, saite netiek parādīta.
Web	Noklikšķinot uz saites, tiks parādīti hostam konfigurētie tīmekļa scenāriji. Tīmekļa scenāriju skaits tiek parādīts pelēkā krāsā. Ja hostam nav tīmekļa scenāriju, saite netiek parādīta.

Pogas

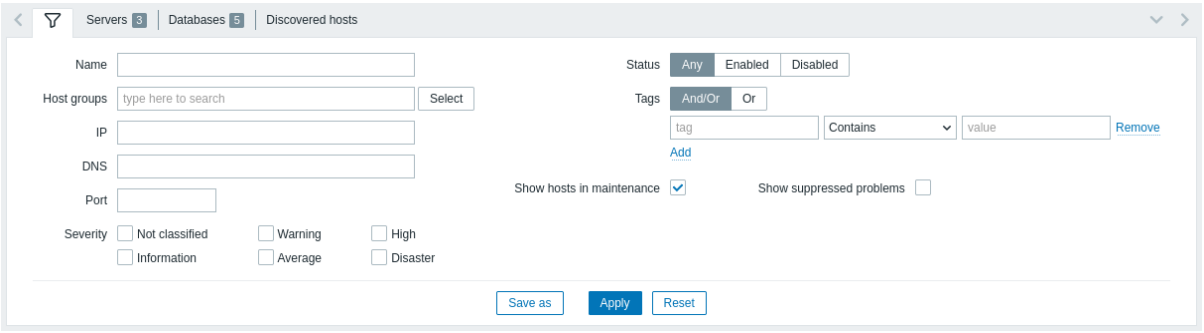
Izveidot hostu ļauj izveidot jaunu hostu. Šī poga ir pieejama tikai Admin un Super Admin lietotājiem.

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas Uzraudzības lapā.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tos hostus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makro vērtībām.

Filtrs atrodas virs tabulas. Ir iespējams filtrēt hostus pēc nosaukuma, hostu grupas, IP vai DNS, saskarnes porta, tagiem, problēmas smaguma pakāpes, statusa (iespējots/atspējots/jebkurš); varat arī izvēlēties, vai attēlot apspiestās problēmas un hostus, kuri pašlaik atrodas uzturēšanā.



Parametrs	Apraksts
Name	Filtrēt pēc redzamā hosta nosaukuma.
Host groups	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas.
IP	Filtrēt pēc IP adreses.
DNS	Filtrēt pēc DNS nosaukuma.
Port	Filtrēt pēc porta numura.
Severity	Filtrēt pēc problēmas smaguma pakāpes. Pēc noklusējuma tiek attēlotas visu smaguma pakāpju problēmas. Problēmas tiek attēlotas, ja tās nav apspiestas.
Status	Filtrēt pēc hosta statusa.
Tags	Filtrēt pēc hosta taga nosaukuma un vērtības. Hostus var filtrēt pēc hosta līmeņa tagiem, kā arī pēc tagiem no visām piesaistītajām veidnēm, ieskaitot ligzdotās veidnes. Ir iespējams gan iekļaut, gan izslēgt noteiktus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** - iekļaut norādītos tagu nosaukumus **Equals** - iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Contains** - iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Does not exist** - izslēgt norādītos tagu nosaukumus **Does not equal** - izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi) **Does not contain** - izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina veidi: **And/Or** - jāizpildās visiem nosacījumiem; nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma **Or** - pietiek, ja izpildās viens nosacījums
Show hosts in maintenance	Atzīmējiet izvēles rūtiņu, lai attēlotu hostus, kas atrodas uzturēšanā (pēc noklusējuma tiek attēloti).
Show suppressed problems	Atzīmējiet izvēles rūtiņu, lai attēlotu problēmas, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas vai atsevišķas problem suppression dēļ.

Filtra saglabāšana

Iecienītos filtra iestatījumus var saglabāt kā cilnes un pēc tam tiem ātri piekļūt, noklikšķinot uz attiecīgās cilnes virs filtra.

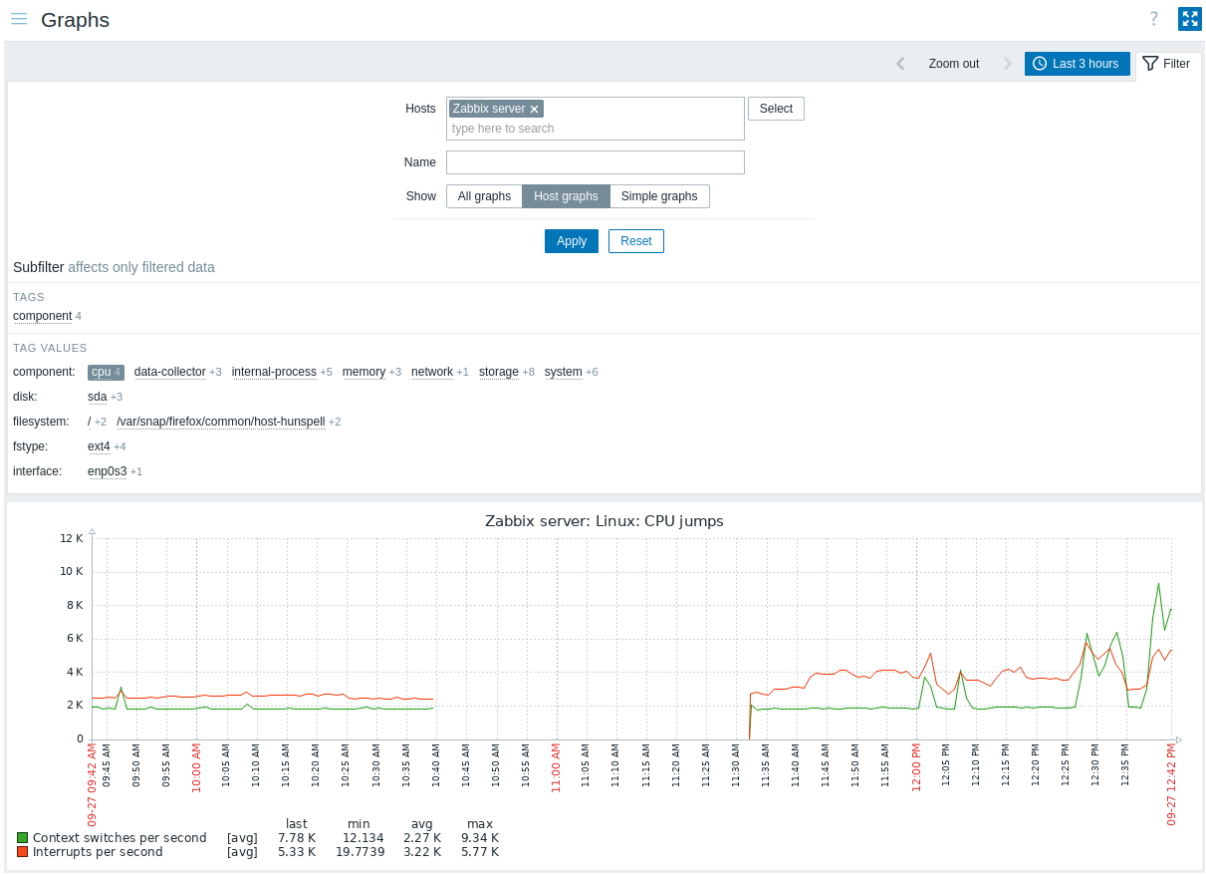
Skatiet vairāk informācijas par filtru saglabāšanu.

1 Grafiki

Pārskats

Hostu grafikiem var piekļūt no *Uzraudzība* → *Hosti*, noklikšķinot uz Grafiki attiecīgajam hostam.

Var attēlot jebkuru pielāgotu grafiku, kas ir konfigurēts hostam, kā arī jebkuru vienkāršo grafiku.



Grafiki tiek kārtoti pēc:

- grafika nosaukuma (pielāgotie grafiki)
- vienuma nosaukuma (vienkāršie grafiki)

Grafiki atspējotiem hostiem arī ir pieejami.

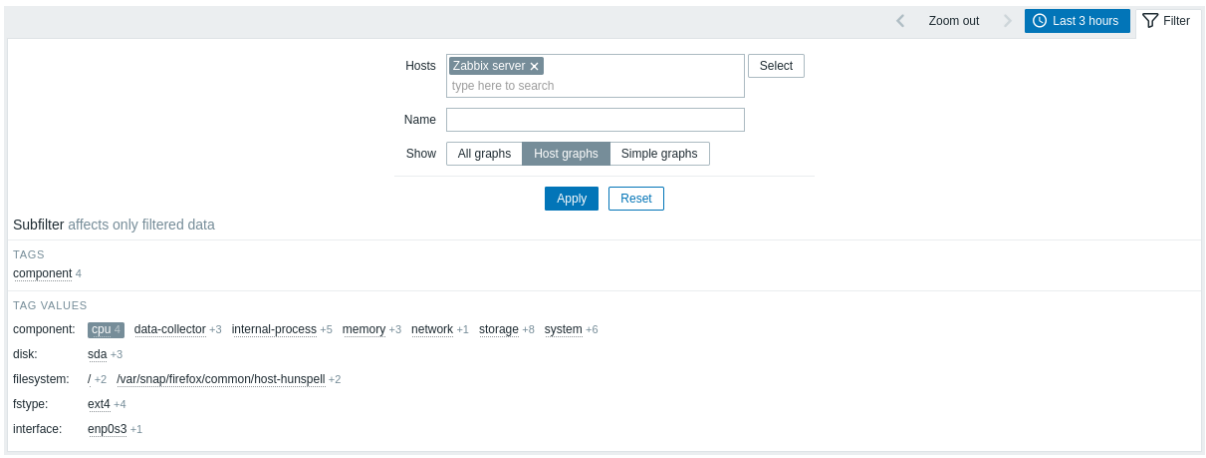
Laika perioda atlasītājs

Pievērsiet uzmanību laika perioda atlasītājam virs diagrammas. Tas ļauj atlasīt bieži nepieciešamus periodus ar vienu peles klikšķi.

Skatiet arī: *Laika perioda atlasītājs*

Filtra izmantošana

Lai skatītu konkrētu grafiku, atlasiet to filtrā. Filtrs ļauj norādīt hostu, grafika nosaukumu un opciju *Rādīt* (visi/hosta grafiki/vienkāršie grafiki).



Ja filtrā nav atlasīts neviens hosts, netiek parādīts neviens grafiks.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt filtrēšanu no galvenā filtra.

Tas satur klikšķināmas saites ātrai piekļuvei saistītajiem grafikiem. Grafiki ir saistīti pēc kopīgas entītijas — birkas nosaukuma vai vērtības. Noklikšķinot uz birkas nosaukuma/vērtības, tas tiek izcelts ar pelēku fonu, un grafiki tiek nekavējoties filtrēti (nav nepieciešams galvenajā filtrā noklikšķināt uz *Apply*). Noklikšķinot uz cita birkas nosaukuma/vērtības, tas tiek pievienots filtrētajiem rezultātiem. Atkārtoti noklikšķinot uz birkas nosaukuma/vērtības, filtrēšana tiek noņemta.

Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kuru apjoms ir ierobežots līdz 1000 ierakstiem. Ja vēlaties apakšfiltrā redzēt vairāk ierakstu, jums jāpalielina parametra *Limit for search and filter results* vērtība (sadaļā *Administration -> General -> GUI*).

Atšķirībā no galvenā filtra, apakšfiltrs tiek atjaunināts kopā ar katru tabulas atsvaidzināšanas pieprasījumu, lai vienmēr iegūtu aktuālu informāciju par pieejamajām filtrēšanas opcijām un to skaitītāju vērtībām.

Horizontāli attēloto entītiju skaits ir ierobežots līdz 100. Ja to ir vairāk, beigās tiek parādīta trīs punktu ikona; tā nav klikšķināma. Vertikālie saraksti (piemēram, birkas ar to vērtībām) ir ierobežoti līdz 20 ierakstiem. Ja to ir vairāk, tiek parādīta trīs punktu ikona; tā nav klikšķināma.

Skaitlis blakus katrai klikšķināmajai entītijai norāda grafiku skaitu, kas tai ir galvenā filtra rezultātos.

Kad viena entītija ir atlasīta, skaitļi pie citām pieejamajām entītijām tiek parādīti ar pluszīmi, norādot, cik grafiku var pievienot pašreizējai atlasei.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

2 Hostu paneļi

Pārskats

Hostu paneļi izskatās līdzīgi globālajiem paneļiem; tomēr hostu paneļiem nav īpašnieka, un tie attēlo datus tikai par atlasīto hostu.

Skatot hostu paneļus, varat pārslēgties starp konfigurētajiem paneļiem, noklikšķinot uz:

- paneļu cilnēm;
- bultu pogām zem galvenes;
- bultas pogas zem galvenes, kas parādīs pilnu pieejamo hostu paneļu sarakstu.

Lai pārslēgtos uz sadaļu *Monitoring* → *Hosts*, noklikšķiniet uz navigācijas saites *All hosts* zem galvenes augšējā kreisajā stūrī.

Konfigurācija

Hostu paneļi tiek konfigurēti veidnes līmenī. Kad veidne ir piesaistīta hostam, šim hostam tiek ģenerēti hostu paneļi. Ņemiet vērā, ka hostu paneļus *nevar* konfigurēt sadaļā *Paneļi*, kas ir paredzēta globālajiem paneļiem.

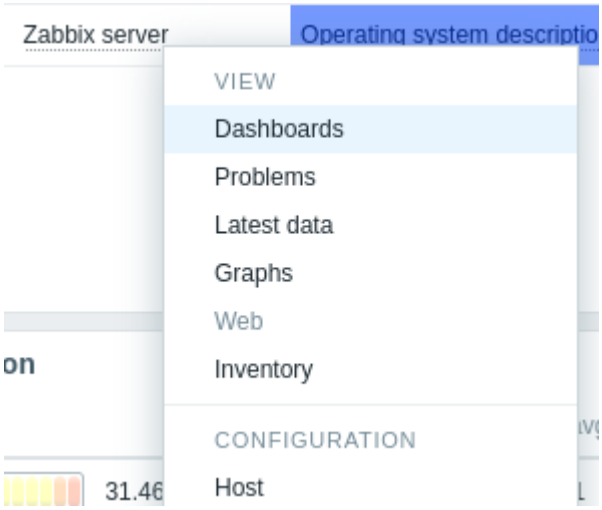
Hostu paneļu logrīkus arī var konfigurēt tikai veidnes līmenī, izņemot atsvaidzināšanas intervāla maiņu. Turklāt hostu paneļu logrīkus var kopēt tikai uz citiem hostu paneļiem tās pašas veidnes ietvaros. Ņemiet vērā, ka logrīkus no globālajiem paneļiem *nevar* kopēt uz hostu paneļiem.

Note:
Pirms Zabbix 5.2 hostu paneļi bija hostu ekrāni. Importējot vecāku veidni, kas satur ekrānus, ekrānu imports tiks ignorēts.

Piekļuve

Hostu paneļiem var piekļūt:

- pēc hosta nosaukuma meklēšanas globālajā meklēšanā (meklēšanas rezultātos noklikšķiniet uz saites *Dashboards*);
- pēc noklikšķināšanas uz hosta nosaukuma sadaļā *Inventārs* → *Hosti* (hosta pārskatā noklikšķiniet uz saites *Dashboards*);
- no hosta izvēlnes, noklikšķinot uz *Dashboards*.



Ņemiet vērā, ka hostu paneļiem *nevar* piekļūt tieši sadaļā *Dashboards*, jo tā ir paredzēta globālajiem paneļiem.

3 Tīmekļa scenāriji

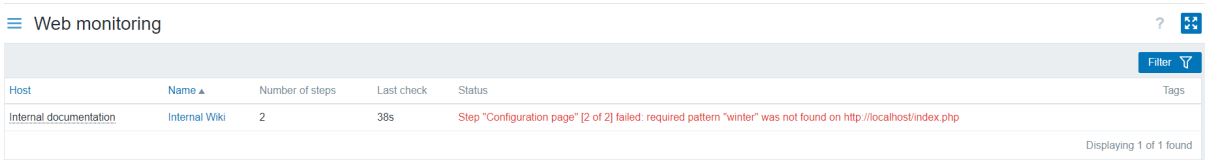
Pārskats

Hosta tīmekļa scenārija informācijai var piekļūt no *Uzraudzība* → *Hosti*, attiecīgajam hostam noklikšķinot uz Web.

Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne. Atspējotu hostu dati ir pieejami arī. Atspējota hosta nosaukums tiek rādīts sarkanā krāsā.

Tīmekļa scenārija statuss var būt:

- OK (zaļš) — visas darbības ir izpildītas sekmīgi, un Zabbix ir savācis datus par scenāriju.
- Kļūda (sarkans) — viena vai vairākas darbības nav izdevušās (skatiet kļūdas ziņojumu, lai iegūtu norādes, kuri parametri varētu būt jālabo).
- Nezināms (pelēks) — darbību statuss vēl nav pieejams vai dati nav savākti.



Maksimālais vienā lapā parādīto scenāriju skaits ir atkarīgs no lietotāja profila iestatījuma *Rindu skaits lapā* setting.

Pēc noklusējuma tiek rādītas tikai tās vērtības, kas ietilpst pēdējās 24 stundās. Šis ierobežojums ir ieviests, lai uzlabotu sākotnējo ielādes laiku lielām jaunāko datu lapām. Jūs varat pagarināt šo laika periodu, mainot parametra *Maksimālais vēstures attēlošanas periods* vērtību izvēlnes sadaļā *Administrēšana → Vispārīgi → GUI*.

Scenārija nosaukums ir saite uz detalizētāku statistiku par to:

Filtra izmantošana

Lapa parāda visu atlasītā hosta tīmekļa scenāriju sarakstu. Lai skatītu cita hosta vai hostu grupas tīmekļa scenārijus, neatgriežoties uz lapu *Monitoring → Hosts*, atlasiet šo hostu vai grupu filtrā. Jūs varat arī filtrēt scenārijus pēc tagiem.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

3 Jaunākie dati

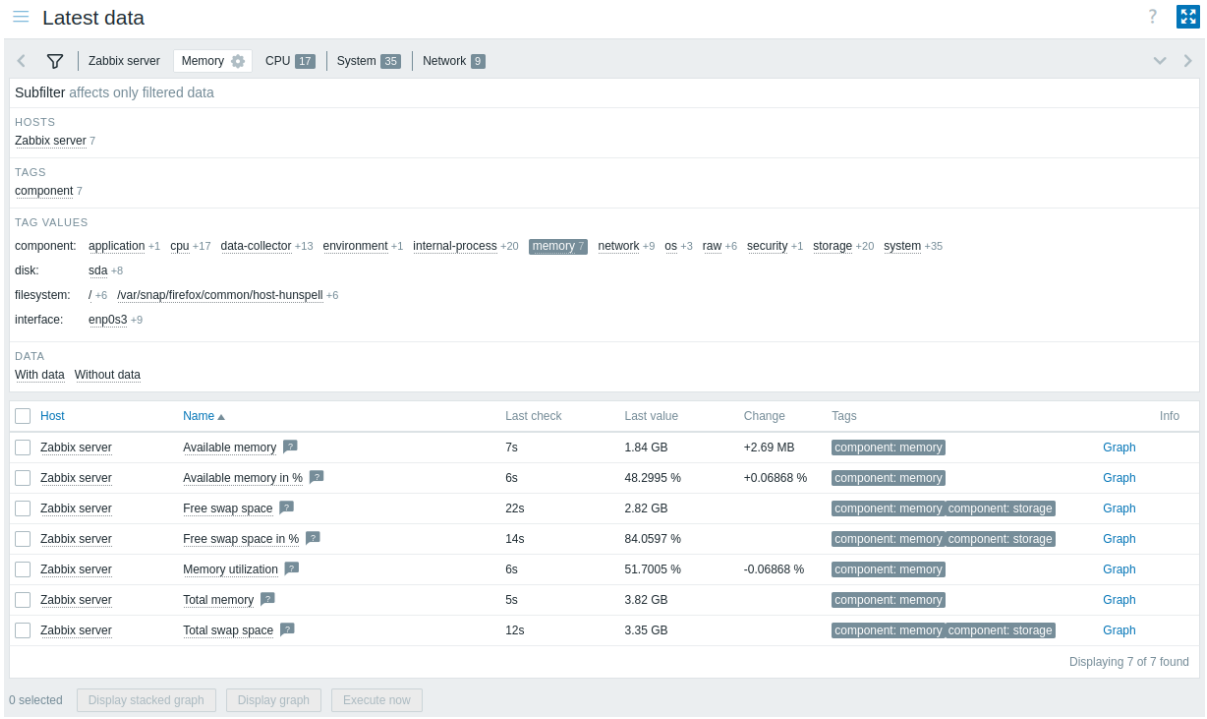
Pārskats

Sadaļā *Uzraudzība* → *Jaunākie dati* tiek attēlotas jaunākās vērtības, kas iegūtas no vienumiem.

Šajā sadaļā ir šādi elementi:

- Filtrs
- Apakšfiltrs
- Vienumu saraksts

Note:
Apakšfiltrs un vienumu saraksts tiek attēlots tikai tad, ja filtrs ir iestatīts un ir rezultāti, ko attēlot.



Kolonna	Apraksts
Hosts	Hosta nosaukums, kuram pieder vienums. Noklikšķinot uz nosaukuma, tiek atvērta hosta konteksta izvēlne. Ja hosts atrodas uzturēšanā, aiz hosta nosaukuma tiek attēlota oranža uzgriežņu atslēgas ikona. Ja hosts ir atspējots, hosta nosaukums tiek attēlots sarkanā krāsā. Ņemiet vērā, ka atspējotu hostu dati (tostarp grafiki un vienumu vērtību saraksti) ir pieejami sadaļā *Jaunākie dati*.
Nosaukums	Vienuma nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta vienuma izvēlne. Jautājuma zīmes ikona tiek attēlota blakus vienuma nosaukumam visiem vienumiem, kuriem ir apraksts. Novietojiet kursoru virs ikonas, lai parādītu uznirstošu paskaidrojumu ar vienuma aprakstu.
Pēdējā pārbaude	Laiks kopš pēdējās vienuma pārbaudes.
Pēdējā vērtība	Jaunākā vienuma vērtība. Vērtības tiek attēlotas, piemērojot mērvienību konvertēšanu un vērtību kartēšanu. Novietojiet kursoru virs vērtības, lai parādītu neapstrādātos datus (saīsinātus līdz 8192 rakstzīmēm). Binārās vērtības tiek attēlotas kā vietturis (*binary value*), nevis faktiskā vērtība. Pēc noklusējuma tiek attēlotas tikai tās vērtības, kas saņemtas pēdējo 24 stundu laikā. Šis ierobežojums uzlabo lielu jaunāko datu lapu sākotnējo ielādes laiku; lai to palielinātu, atjauniniet parametra *Max history display period* vērtību sadaļā *Administrēšana* → *Vispārīgi* → *GUI*.
Izmaiņa	Atšķirība starp iepriekšējo vērtību un jaunāko vērtību. Vienumiem ar atjaunināšanas biežumu 1 diena vai vairāk izmaiņas apjoms nekad netiks attēlots (ar noklusējuma iestatījumu). Šajā gadījumā pēdējā vērtība vispār netiks attēlota, ja tā saņemta pirms vairāk nekā 24 stundām.

Kolonna	Apraksts
Birkas	Ar vienumu saistītās birkas. Birkas vienumu sarakstā ir noklikšķināmas. Noklikšķinot uz birkas, tā tiek iespējota apakšfiltrā, liekot vienumu sarakstam attēlot tikai tos vienumus, kas satur šo birku (un jebkuras citas birkas, kas iepriekš atlasītas apakšfiltrā). Ņemiet vērā, ka pēc tam, kad vienumi ir filtrēti šādā veidā, birkas vienumu sarakstā vairs nav noklikšķināmas. Turpmāka modificēšana, pamatojoties uz birkām (piemēram, lai noņemtu birkas vai norādītu citus filtrus), jāveic apakšfiltrā.
Grafiks/Vēsture	Saite uz vienuma vērtību vienkāršo grafiku/vēsturi.
Informācija	Papildu informācija par vienumu. Ja vienumam ir kļūdas (piemēram, tas ir kļuvis neatbalstīts), tiek attēlota informācijas ikona. Novietojiet kursoru virs ikonas, lai skatītu detalizētu informāciju.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Masveida darbības

Pogas zem saraksta piedāvā masveida darbības ar vienu vai vairākiem atlasītiem vienumiem:

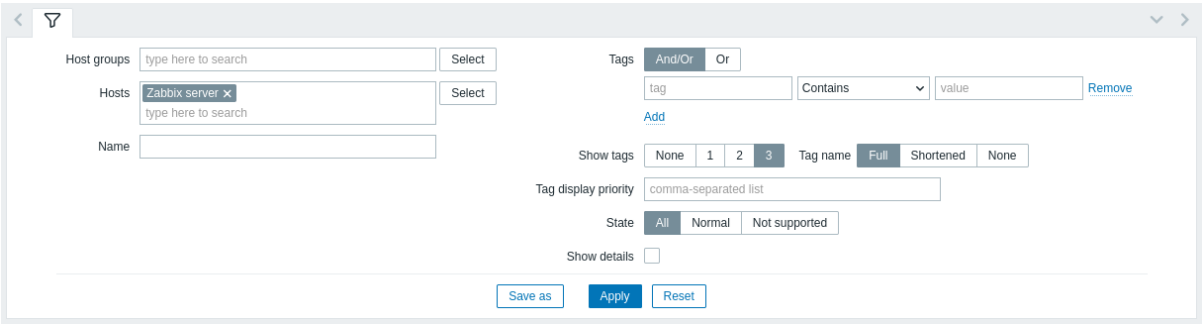
- *Rādīt slāņoto grafiku* - parādīt slāņotu ad-hoc grafiku.
- *Rādīt grafiku* - parādīt vienkāršu ad-hoc grafiku.
- *Izpildīt tagad* - nekavējoties izpildīt pārbaudi, lai iegūtu jaunas vienuma vērtības. Atbalstīts tikai **pasīvajām** pārbaudēm (skatiet sīkāku informāciju). Šī opcija ir pieejama tikai hostiem ar lasīšanas-rakstīšanas piekļuvi. Piekļuve šai opcijai hostiem ar tikai lasīšanas atļaujām ir atkarīga no lietotāja lomas opcijas *Invoke "Execute now" on read-only hosts*.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumiem un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos vienumus, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makro vērtībām.

Filtra ikona atrodas virs vienumu saraksta un apakšfiltra. Noklikšķiniet uz tās, lai izvērstu filtru.



Filtrs ļauj sašaurināt vienumu sarakstu pēc hostu grupas, hosta, vienuma nosaukuma, taga, stāvokļa un citiem iestatījumiem. Norādot vecāku hostu grupu filtrā, tiek netieši atlasītas arī visas tajā ietvertās ligzdotās hostu grupas. Sīkāku informāciju par filtrēšanu pēc tagiem skatiet sadaļā *Monitoring* → *Problems*.

Filtra opcija *Show details* ļauj paplašināt vienumiem parādīto informāciju. Atzīmējiet to, lai parādītu tādu detalizētu informāciju kā vienuma atsvaidzināšanas intervāls, vēstures un tendenču iestatījumi, vienuma tips un vienuma kļūdas (fine/unsupported).

Filtra saglabāšana

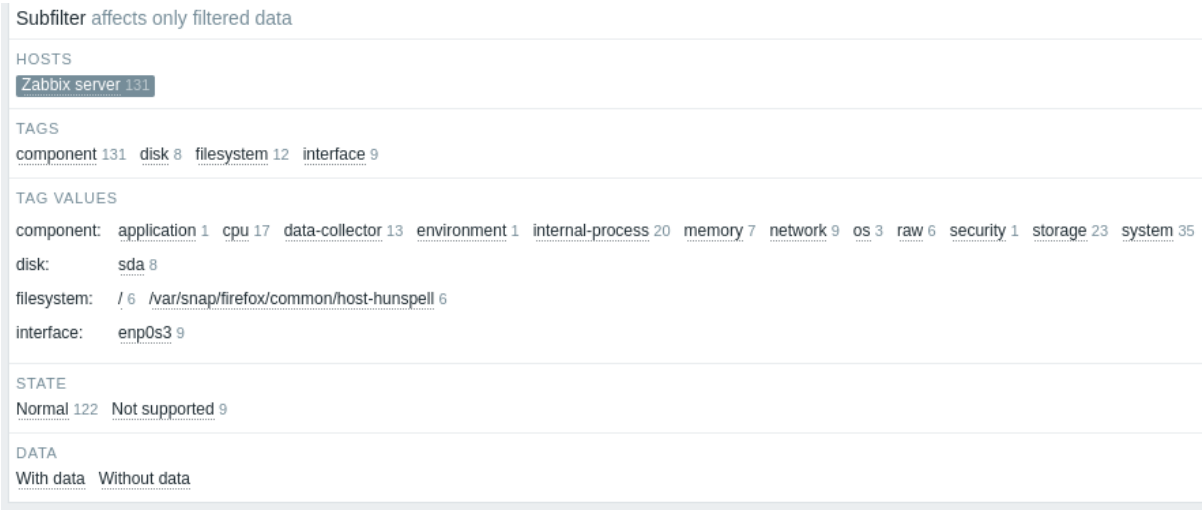
Iecienītā filtra iestatījumus var saglabāt kā cilnes un pēc tam tiem ātri piekļūt, noklikšķinot uz attiecīgās cilnes virs filtra.

Skatiet vairāk informācijas par filtru saglabāšanu.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt filtrēšanu no galvenā filtra.

Tas satur klikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumi ir saistīti pēc kopīgas entītijas — hosts, birkas nosaukuma vai vērtības, vienuma stāvokļa vai datu statusa. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (nav nepieciešams galvenajā filtrā noklikšķināt uz *Apply*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Atkārtoti noklikšķinot uz entītijas, filtrēšana tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kuru apjoms ir ierobežots līdz 1000 ierakstiem. Ja ir 20 hosti un katram ir 100 vienumi (tātad kopā 2000 ierakstu), apakšfiltrā būs redzama tikai puse no hostiem. Ja vēlaties apakšfiltrā redzēt vairāk ierakstu, jums jāpalielina parametra *Limit for search and filter results* vērtība (sadaļā *Administration -> General -> GUI*).

Atšķirībā no galvenā filtra, apakšfiltrs tiek atjaunināts ar katru tabulas atsvaidzināšanas pieprasījumu, lai vienmēr būtu pieejama aktuāla informācija par pieejamajām filtrēšanas opcijām un to skaitītāju vērtībām.

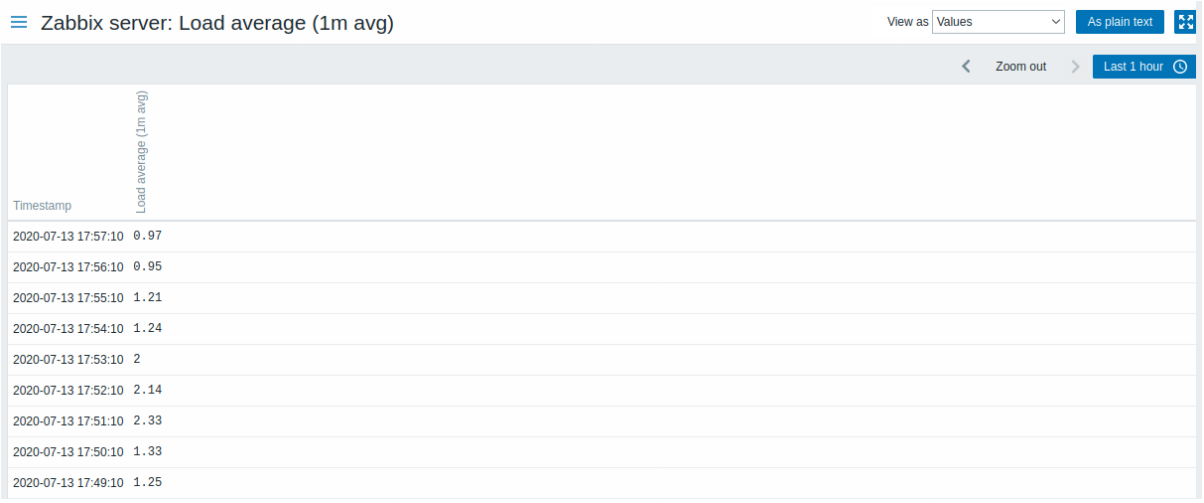
Katrai entītiju grupai (hosti, birkas, birku vērtības utt.) tiek parādītas līdz 10 entītiju rindām. Ja entītiju ir vairāk, šo sarakstu var izvērst, lai parādītu ne vairāk kā 1000 ierakstus (vērtība SUBFILTER_VALUES_PER_GROUP sadaļā frontend definitions), noklikšķinot uz trīs punktu ••• ikonas saraksta beigās. Sadaļai *Tag values* sarakstu var izvērst, lai parādītu ne vairāk kā 200 birku nosaukumus ar tiem atbilstošajām vērtībām. Ņemiet vērā, ka pēc pilnīgas izvēršanas sarakstu vairs nevar sakļaut.

Skaitlis blakus katrai klikšķināmajai entītijai norāda tajā grupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Noklikšķinot uz entītijas, skaitļi pie citām pieejamajām entītijām tiek parādīti ar pluszīmi, norādot, cik vienumu var tikt pievienoti pašreizējai atlasei. Entītijas bez vienumiem netiek rādītas, ja vien tās iepriekš nav atlasītas apakšfiltrā.

Grafiki un vēsture

Kolonna *Graph/History* vienumu sarakstā piedāvā šādas saites:

- **History** - visiem teksta vienumiem, ved uz sarakstiem (*Values/500 latest values*), kuros tiek parādīta iepriekšējo vienuma vērtību vēsture.
- **Graph** - visiem skaitliskajiem vienumiem, ved uz vienkāršu grafiku. Ņemiet vērā, ka, kad grafiks tiek parādīts, nolaižamā izvēlne augšējā labajā stūrī piedāvā arī iespēju pārslēgties uz *Values/500 latest values*.



Šajā sarakstā parādītās vērtības ir neapstrādātas, tas ir, tām netiek piemērota nekāda pēcapstrāde.

Note:
Kopējo parādīto vērtību skaitu nosaka parametra *Limit for search and filter results* vērtība, kas iestatīta sadaļā *Administration → General → GUI*.

4 Kartes

Pārskats

Sadaļā *Monitoring* → *Maps* varat konfigurēt, pārvaldīt un skatīt tīkla kartes.

Atverot šo sadaļu, jūs redzēsiet vai nu pēdējo karti, kurai piekļuvāt, vai arī visu to karšu sarakstu, kurām jums ir piekļuve.

Visas kartes var būt publiskas vai privātas. Publiskās kartes ir pieejamas visiem lietotājiem, savukārt privātās kartes ir pieejamas tikai to īpašniekam un lietotājiem, ar kuriem karte ir koplietota.

Karšu saraksts

Attēlotie dati:

Kolonna	Apraksts
Nosaukums	Kartes nosaukums. Noklikšķiniet uz nosaukuma, lai skatītu karti.
Platums	Tiek attēlots kartes platums.
Augstums	Tiek attēlots kartes augstums.
Darbības	Ir pieejamas divas darbības: **Properties** - iestatīt kartes vispārīgās īpašības **Edit** - piekļūt režģim, lai pievienotu kartes elementus

Lai konfigurētu jaunu karti, augšējā labajā stūrī noklikšķiniet uz pogas *Create map*. Lai importētu karti no YAML, XML vai JSON faila, augšējā labajā stūrī noklikšķiniet uz pogas *Import*. Lietotājs, kurš importē karti, tiks iestatīts kā tās īpašnieks.

Divas pogas zem saraksta piedāvā dažas masveida rediģēšanas iespējas:

- *Export* - eksportēt kartes uz YAML, XML vai JSON failu
- *Delete* - dzēst kartes

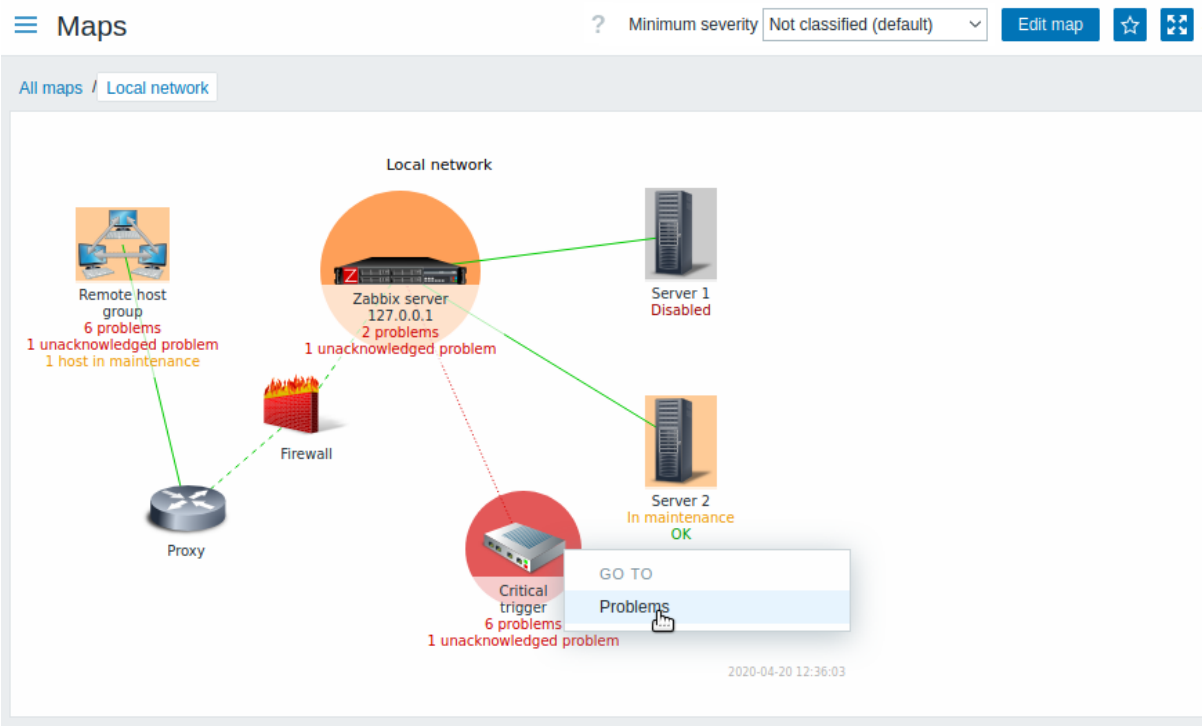
Lai izmantotu šīs iespējas, atzīmējiet izvēles rūtiņas pie attiecīgajām kartēm, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās kartes, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makrosiem.

Karšu skatīšana

Lai skatītu karti, noklikšķiniet uz tās nosaukuma visu karšu sarakstā.



Varat izmantot nolaižamo izvēlni kartes virsrakstjoslā, lai atlasītu zemāko attēlojamo problēmu trigeru nopietnības līmeni. Nopietnības līmenis, kas atzīmēts kā *noklusējuma*, ir līmenis, kas iestatīts kartes konfigurācijā. Ja karte satur apakškarti, pārejot uz apakškarti, tiks saglabāts augstāka līmeņa kartes nopietnības līmenis (izņemot gadījumu, ja tas ir *Neklasificēts*; šajā gadījumā tas netiks nodots apakškartei).

Ikonu izcelšana

Ja kartes elements ir problēmas stāvoklī, tas tiek izcelts ar apaļu apli. Apļa aizpildījuma krāsa atbilst problēmas smaguma pakāpes krāsai. Kopā ar elementu tiks attēlotas tikai tās problēmas, kuru smaguma līmenis ir vienāds ar atlasīto vai augstāks par to. Ja visas problēmas ir apstiprinātas, ap apli tiek attēlota bieza zaļa apmale.

Papildus:

- hosts uzturēšanā tiek izcelts ar oranžu, aizpildītu kvadrātu. Ņemiet vērā, ka uzturēšanas izcelšanai ir prioritāte pār problēmas smaguma pakāpes izcelšanu, ja kartes elements ir hosts.
- atspējots (neuzraudzīts) hosts tiek izcelts ar pelēku, aizpildītu kvadrātu.

Izcelšana tiek attēlota, ja kartes konfigurācijā ir atzīmēta izvēles rūtiņa *Ikonu izcelšana*.

Neseno izmaiņu marķieri

Uz iekšu vērsti sarkani trijstūri ap elementu norāda uz nesenu trigera stāvokļa maiņu — tādu, kas notikusi pēdējo 30 minūšu laikā. Šie trijstūri tiek rādīti, ja kartes konfigurācijā ir atzīmēta izvēles rūtiņa *Mark elements on trigger status change*.

Saites

Noklikšķinot uz kartes elementa, tiek atvērta izvēlne ar dažām pieejamām saitēm. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne.

Pogas

Pogas labajā pusē piedāvā šādas iespējas:

Edit map	Pāriet uz kartes satura rediģēšanu.
☆	Pievienot karti izlases logrīkam sadaļā Paneļi.
★	Karte atrodas izlases logrīkā sadaļā Paneļi. Noklikšķiniet, lai noņemtu karti no izlases logrīka.

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitorings.

Lasāms kopsavilkums kartēs

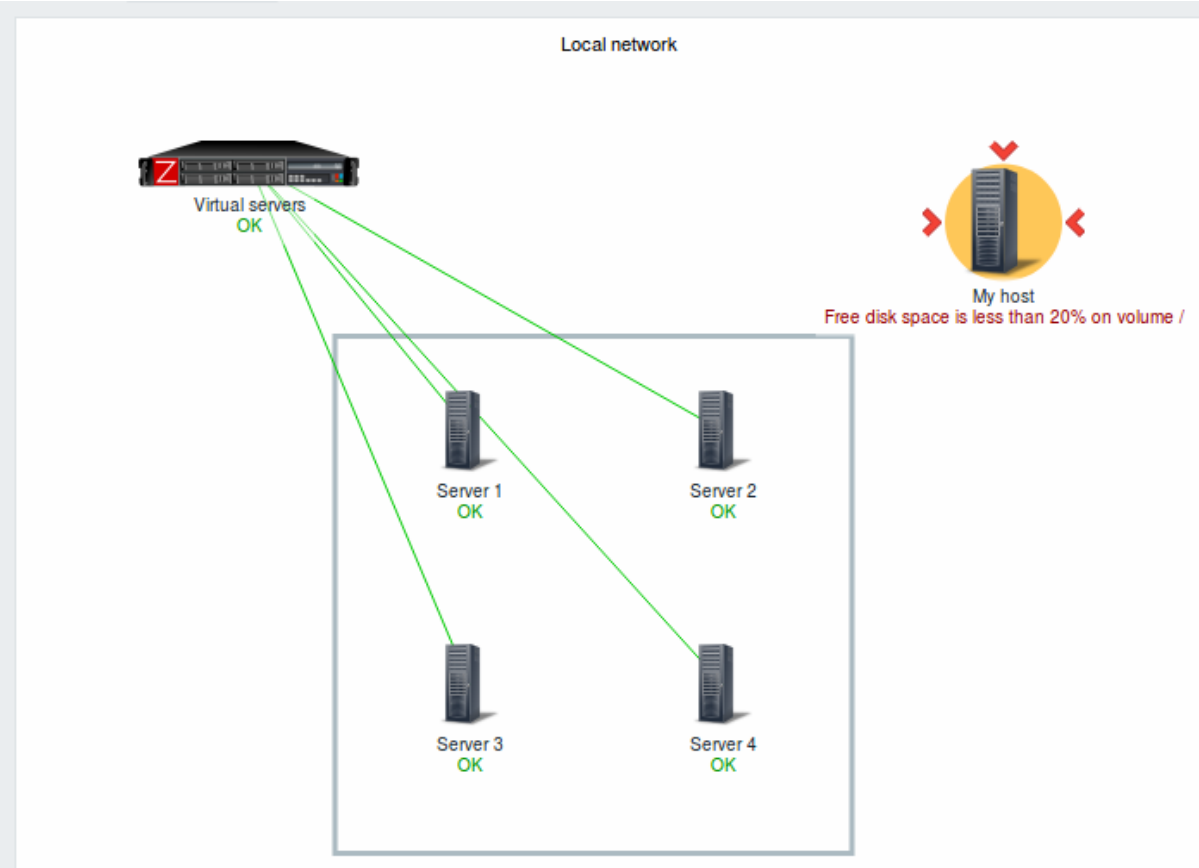
Ir pieejams slēpts "aria-label" rekvizīts, kas ļauj kartes informāciju nolasīt ar ekrāna lasītāju. Ir pieejams gan vispārīgs kartes apraksts, gan atsevišķu elementu apraksts šādā formātā:

- kartes aprakstam: `<Kartes nosaukums>, <* no * vienumiem problēmas stāvoklī>, <* problēmas kopā>.`
- viena elementa ar vienu problēmu aprakstam: `<Elementa tips>, Statuss <Elementa statuss>, <Elementa nosaukums>, <Problēmas apraksts>.`
- viena elementa ar vairākām problēmām aprakstam: `<Elementa tips>, Statuss <Elementa statuss>, <Elementa nosaukums>, <* problēmas>.`
- viena elementa bez problēmām aprakstam: `<Elementa tips>, Statuss <Elementa statuss>, <Elementa nosaukums>.`

Piemēram, ir pieejams šāds apraksts:

```
'Lokālais tīkls, 1 no 6 elementiem problēmas stāvoklī, 1 problēma kopā. Hosts, Statuss problēma, Mans host
```

šādai kartei:



Atsaukšanās uz tīkla karti

Uz tīkla kartēm var atsaukties, izmantojot GET parametrus `sysmapid` un `mapname`. Piemēram,

```
http://zabbix/zabbix/zabbix.php?action=map.view&mapname=Local%20network
```

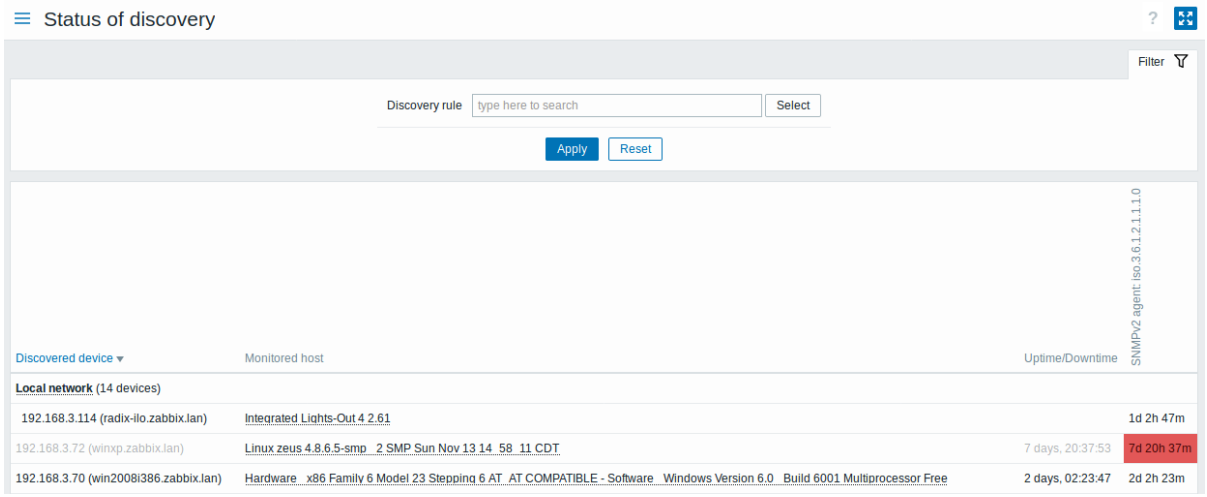
atvērs karti ar šādu nosaukumu (Local network).

Ja ir norādīti gan `sysmapid` (kartes ID), gan `mapname` (kartes nosaukums), parametram `mapname` ir augstāka prioritāte.

5 Atklāšana

Pārskats

Sadaļā *Uzraudzība* → *Atklāšana* tiek parādīti tīkla atklāšanas rezultāti. Atklātās ierīces ir sakārtotas pēc atklāšanas kārtulas.



Parādītie dati:

Kolonna	Apraksts
Atklātā ierīce	Atklātās ierīces ir uzskaitītas, grupētas pēc atklāšanas kārtulas. Noklikšķinot uz atklāšanas kārtulas, tiek atvērta kārtulas izvēlne, kurā ir saite uz atklāšanas kārtulas konfigurācijas formu.
Uzraudzītais hosts	Ja ierīce jau tiek uzraudzīta, šajā kolonnā tiks norādīts hosta nosaukums. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne.
Darbspējas laiks/Nedarbspējas laiks	Šajā kolonnā tiek parādīts laiks, cik ilgi ierīce ir atklāta vai zaudēta pēc iepriekšējās atklāšanas.
Atklāšanas pārbaude	Tiek parādīts katras atklātās ierīces atsevišķā pakalpojuma (atklāšanas pārbaudes) stāvoklis. Sarkana šūna norāda, ka pakalpojums nav pieejams. Pakalpojuma darbspējas vai nedarbspējas laiks ir iekļauts šūnā. Šī kolonna tiek parādīta tikai tad, ja pakalpojums ir atrasts vismaz vienā atklātajā ierīcē.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos atklāšanas noteikumus, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Ja filtrā nekas nav atlasīts, tiek parādīti visi iespējotie atklāšanas noteikumi. Lai parādīšanai atlasītu konkrētu atklāšanas noteikumu, sāciet ievadīt tā nosaukumu filtrā. Visi atbilstošie iespējotie atklāšanas noteikumi tiks uzskaitīti atlasei. Var atlasīt vairāk nekā vienu atklāšanas noteikumu.

3 Pakalpojumi

Pārskats

Izvēlne Pakalpojumi ir paredzēta Zabbix pakalpojumu uzraudzības funkcijām.

1 Pakalpojumi

Pārskats

Šajā sadaļā varat redzēt augsta līmeņa statusu visiem pakalpojumiem, kas ir konfigurēti Zabbix, pamatojoties uz jūsu infrastruktūru.

Pakalpojums var būt hierarhija, kas sastāv no vairākiem citu pakalpojumu līmeņiem, ko sauc par ”bērn” pakalpojumiem, kuri ietekmē kopējo pakalpojuma statusu (skatiet arī pakalpojumu uzraudzības funkcionalitātes pārskatu.)

Galvenās pakalpojuma stāvokļa kategorijas ir *OK* vai *Problēma*, kur *Problēmas* stāvoklis tiek attēlots ar atbilstošo problēmas būtiskuma nosaukumu un krāsu.

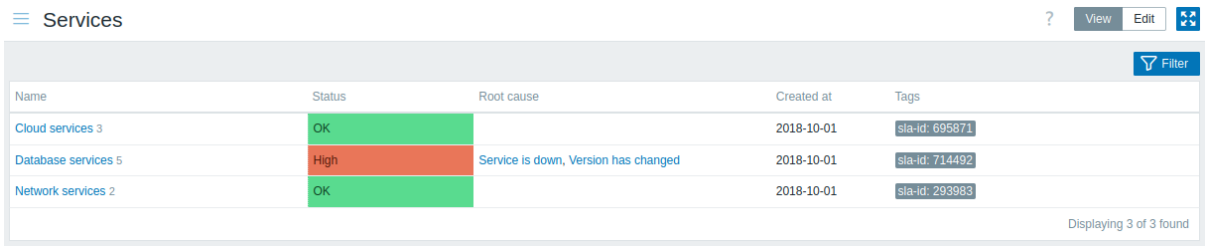
Skata režīms ļauj uzraudzīt pakalpojumus ar to statusu un citu informāciju, taču šajā sadaļā varat arī konfigurēt pakalpojumu hierarhiju (pievienot/rediģēt pakalpojumus, bērn pakalpojumus), pārslēdzoties uz rediģēšanas režīmu.

Lai pārslēgtos no skata režīma uz rediģēšanas režīmu (un atpakaļ), noklikšķiniet uz attiecīgās pogas augšējā labajā stūrī:

- View - skatīt pakalpojumus
- Edit - pievienot/rediģēt pakalpojumus un bērn pakalpojumus

Ņemiet vērā, ka piekļuve rediģēšanai ir atkarīga no lietotāja lomas iestatījumiem.

Pakalpojumu skatīšana



Tiek parādīts esošo pakalpojumu saraksts.

Parādītie dati:

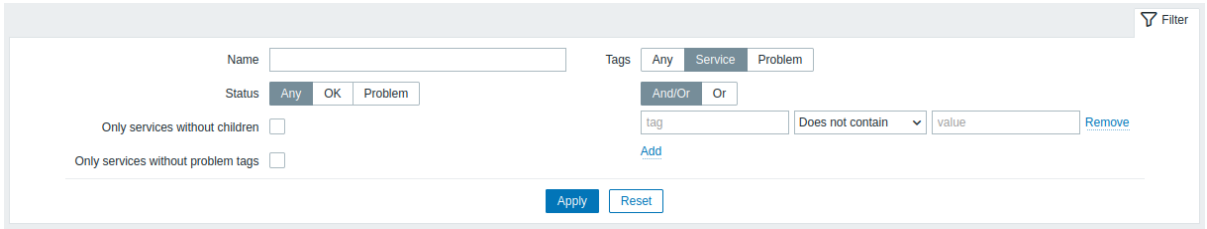
Parametrs	Apraksts
Nosaukums	Pakalpojuma nosaukums. Pakalpojuma nosaukums ir saite uz pakalpojuma informāciju. Skaitlis aiz nosaukuma norāda, cik apakšpakalpojumu ir pakalpojumam.
Statuss	Pakalpojuma stāvoklis: **OK** - nav problēmu **<problēmas krāsa un smaguma pakāpe>** - norāda problēmu un tās smaguma pakāpi. Ja ir vairākas problēmas, tiek parādīta kritiskākās problēmas krāsa un smaguma pakāpe.
Pamatcēlonis	Tiek uzskaitītas pamatā esošās problēmas, kas tieši vai netieši ietekmē pakalpojuma stāvokli. Tās pašas problēmas tiek uzskaitītas, kā tās atgriež {SERVICE.ROOTCAUSE} makross. Noklikšķiniet uz problēmas nosaukuma, lai skatītu sīkāku informāciju par to sadaļā *Uzraudzība* → *Problēmas*. Problēmas, kas neietekmē pakalpojuma stāvokli, sarakstā netiek iekļautas.
Izveidots	Tiek parādīts laiks, kad pakalpojums tika izveidots.
Birkas	Tiek parādītas pakalpojuma birkas. Birkas tiek izmantotas, lai identificētu pakalpojumu pakalpojumu darbībās un SLA.

Pogas

Skata režīma pogas, kas ir kopīgas visām sadaļām, ir aprakstītas lapā Monitoring.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tos pakalpojumus, kas jūs interesē.



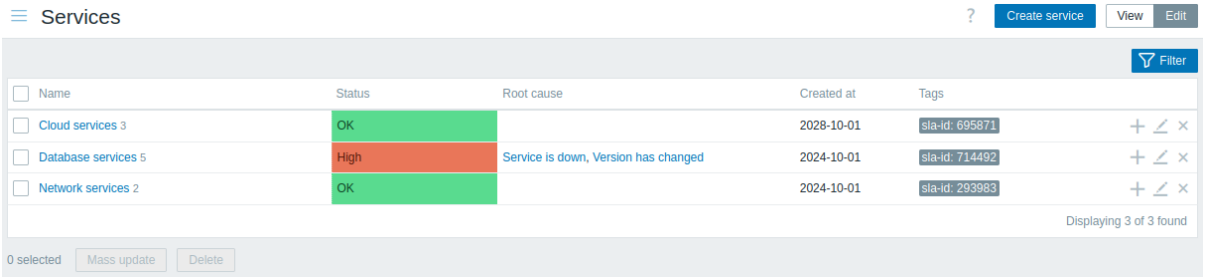
Parametrs	Apraksts
Name	Filtrēt pēc pakalpojuma nosaukuma.
Status	Filtrēt pēc pakalpojuma stāvokļa.
Only services without children	Atzīmējiet izvēles rūtiņu, lai attēlotu tikai pakalpojumus bez apakšpakalpojumiem. Šis parametrs ir pieejams tikai pakalpojuma rediģēšanas režīmā.
Only services without problem tags	Atzīmējiet izvēles rūtiņu, lai attēlotu tikai pakalpojumus bez problēmu tagiem. Šis parametrs ir pieejams tikai pakalpojuma rediģēšanas režīmā.

Parametrs	Apraksts
Tags	Filtrēt pēc pakalpojuma taga nosaukuma un vērtības vai pakalpojuma problēmas taga nosaukuma un vērtības (pakalpojuma rediģēšanas režīmā). Ir iespējams gan iekļaut, gan izslēgt noteiktus tagus un tagu vērtības. Var iestatīt vairākus nosacījumus. Taga nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Exists** — iekļaut norādītos tagu nosaukumus; **Equals** — iekļaut norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Contains** — iekļaut norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** — izslēgt norādītos tagu nosaukumus; **Does not equal** — izslēgt norādītos tagu nosaukumus un vērtības (reģistrjutīgi); **Does not contain** — izslēgt norādītos tagu nosaukumus, kuru tagu vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** — visiem nosacījumiem jābūt izpildītiem; nosacījumi ar vienādu taga nosaukumu tiks grupēti pēc Or nosacījuma; **Or** — pietiek, ja ir izpildīts viens nosacījums.

Pakalpojumu rediģēšana

Noklikšķiniet uz pogas *Edit*, lai piekļūtu rediģēšanas režīmam. Kad ir ieslēgts rediģēšanas režīms, saraksts tiek papildināts ar izvēles rūtiņām pirms ierakstiem, kā arī ar šīm papildu opcijām:

- - pievienot šim pakalpojumam bērnpakalpojumu
- - rediģēt šo pakalpojumu
- - dzēst šo pakalpojumu

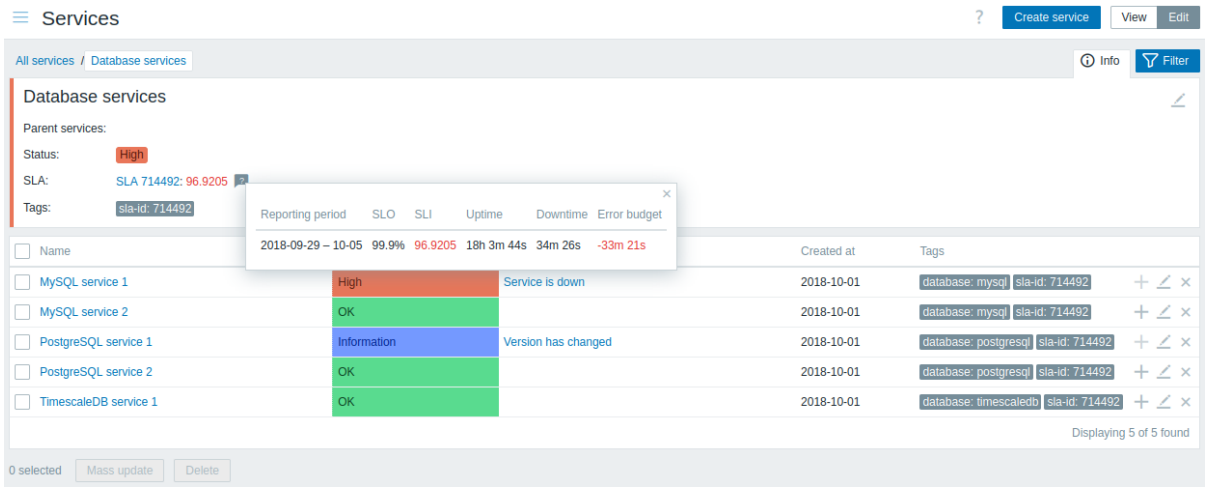


Lai konfigurētu jaunu pakalpojumu, noklikšķiniet uz pogas *Create service* augšējā labajā stūrī.

Pakalpojuma informācija

Lai piekļūtu pakalpojuma informācijai, noklikšķiniet uz pakalpojuma nosaukuma. Lai atgrieztos visu pakalpojumu sarakstā, noklikšķiniet uz *Visi pakalpojumi*.

Pakalpojuma informācija ietver informācijas lodziņu un bērnpakalpojumu sarakstu.



Lai piekļūtu informācijas lodziņam, noklikšķiniet uz cilnes *Info*. Informācijas lodziņā ir šādi ieraksti:

- Vecākpakalpojumu nosaukumi (ja tādi ir)
- Šī pakalpojuma pašreizējais statuss
- Šī pakalpojuma pašreizējais(-ie) SLA formātā `SLA nosaukums:pakalpojuma līmeņa indikators`. Arī "SLA nosaukums" ir saite uz šī pakalpojuma SLA atskaiti. Novietojot peles kursoru informācijas lodziņā blakus pakalpojuma līmeņa indikatoram (SLI), tiek parādīts uznirstošs informācijas saraksts ar SLI detalizētu informāciju. Pakalpojuma līmeņa indikators parāda pašreizējo pakalpojuma līmeni procentos.
- Pakalpojuma birkas

Informācijas lodziņā ir arī saite uz pakalpojuma konfigurāciju.

Lai izmantotu bērnpakalpojumu filtru, noklikšķiniet uz cilnes *Filter*.

Atrodoties rediģēšanas režīmā, bērnpakalpojumu saraksts tiek papildināts ar papildu rediģēšanas iespējām:

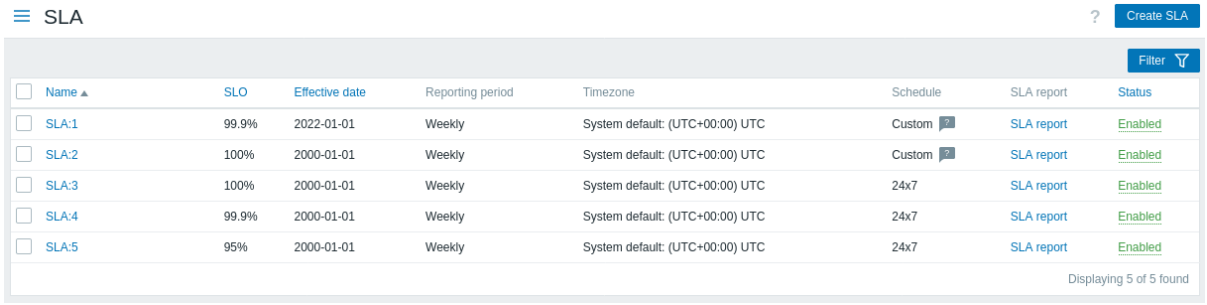
- - pievienot šim pakalpojumam bērnpakalpojumu
- - rediģēt šo pakalpojumu
- - dzēst šo pakalpojumu

2 SLA

Pārskats

Šajā sadaļā ir iespējams skatīt un konfigurēt SLA.

SLA



Tiek parādīts konfigurēto SLA saraksts. *Ņemiet vērā*, ka tiks parādīti tikai tie SLA, kas saistīti ar lietotājam pieejamiem pakalpojumiem (tikai lasīšanas režīmā, ja vien lietotāja lomai nav iespējota *Manage SLA*).

Parādītie dati:

Parametrs	Apraksts
Name	Tiek parādīts SLA nosaukums. Nosaukums ir saite uz SLA konfigurāciju.
SLO	Tiek parādīts pakalpojuma līmeņa mērķis (SLO).
Effective date	Tiek parādīts SLA aprēķina sākuma datums.

Parametrs	Apraksts
Reporting period	Tiek parādīts SLA pārskatā izmantotais periods — *diena*, *nedēļa*, *mēnesis*, *ceturksnis* vai *gads*.
Time zone	Tiek parādīta SLA laika josla.
Schedule	Tiek parādīts SLA grafiks — 24x7 vai pielāgots.
SLA report	Noklikšķiniet uz saites, lai skatītu šī SLA pārskatu.
Status	Tiek parādīts SLA statuss — iespējots vai atspējots.

3 SLA atskaite

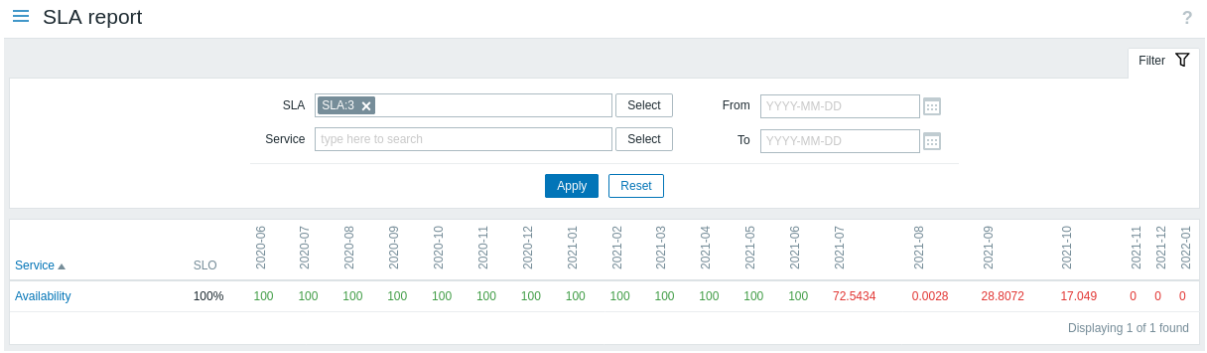
Pārskats

Šajā sadaļā var skatīt SLA atskaites, pamatojoties uz filtrā atlasītajiem kritērijiem.

SLA atskaites var attēlot arī kā informācijas paneļa logrīku.

Pārskats

Filtrs ļauj atlasīt pārskatu pēc SLA nosaukuma, kā arī pakalpojuma nosaukuma. Ir iespējams arī ierobežot attēloto periodu.

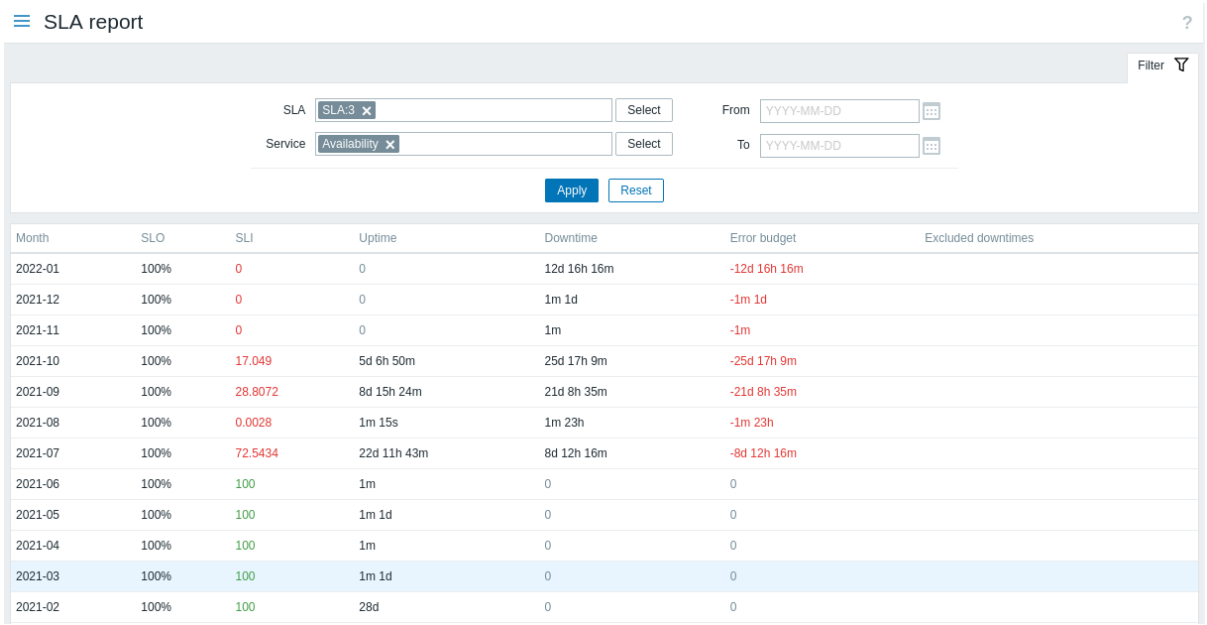


Katrā kolonnā (periodā) tiek attēlots šī perioda SLI. SLI, kas neatbilst iestatītajam SLO, ir izcelti sarkanā krāsā.

Pārskatā tiek attēloti 20 periodi. Var attēlot ne vairāk kā 100 periodus, ja ir norādīts gan *No* datums, gan *Līdz* datums.

Pārskata detaļas

Ja pārskatā noklikšķināt uz pakalpojuma nosaukuma, varat piekļūt citam pārskatam, kurā tiek attēlots detalizētāks skats.



Ņemiet vērā, ka negatīvs problēmas ilgums neietekmē SLA aprēķinu vai pārskatu veidošanu.

4 Inventārs

Pārskats

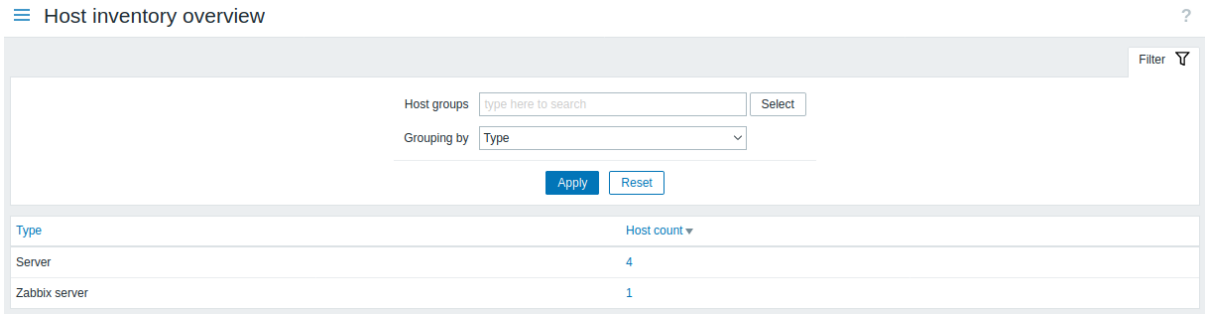
Izvēlnē Inventārs ir sadaļas, kas nodrošina hostu inventāra datu pārskatu pēc izvēlēta parametra, kā arī iespēju skatīt hosta inventāra detalizētu informāciju.

1 Pārskats

Pārskats

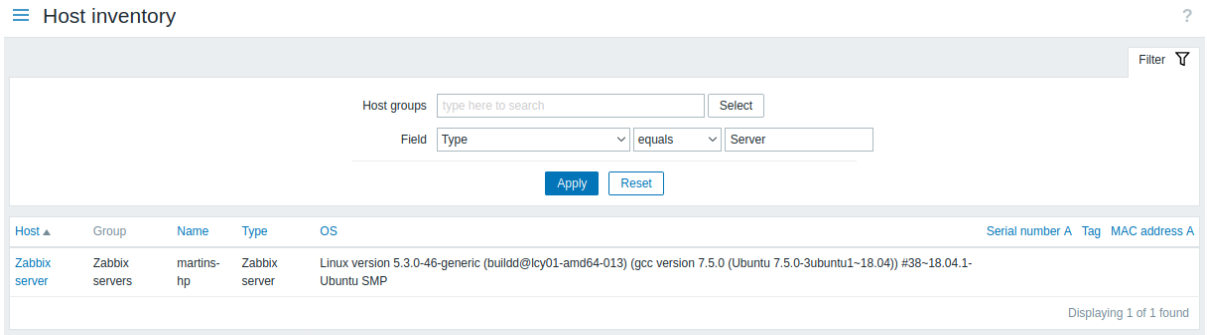
Sadaļa *Inventārs* → *Pārskats* nodrošina iespējas iegūt hostu inventāra datu pārskatu.

Lai tiktu parādīts pārskats, izvēlieties hostu grupas (vai nevienu) un inventāra lauku, pēc kura attēlot datus. Tiks parādīts hostu skaits, kas atbilst katram izvēlētā lauka ierakstam.



Pārskata pilnīgums ir atkarīgs no tā, cik daudz inventāra informācijas tiek uzturēts hostiem.

Skaitļi kolonnā *Hostu skaits* ir saites; tās ļauj atfiltrēt šos hostus tabulā *Hostu inventāri*.

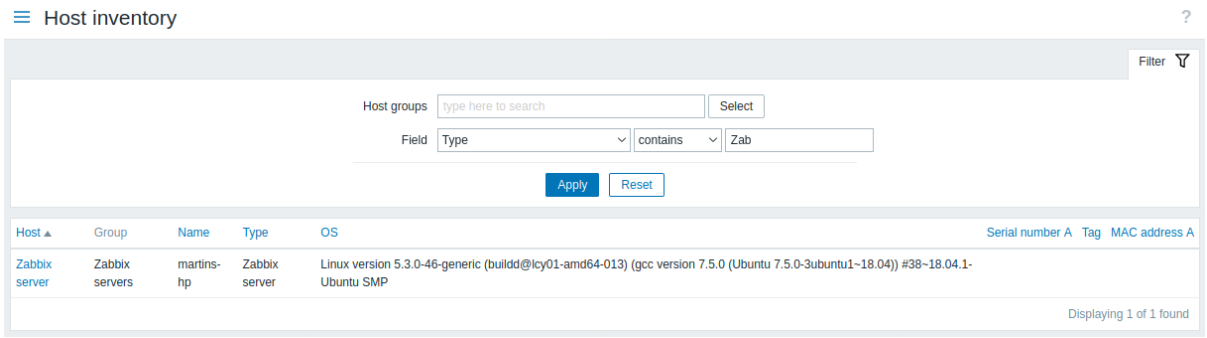


2 Hosti

Pārskats

Sadaļā *Inventārs* → *Hosti* tiek attēloti hostu inventāra dati.

Varat filtrēt hostus pēc hostu grupas(-ām) un pēc jebkura inventāra lauka, lai attēlotu tikai tos hostus, kas jūs interesē.

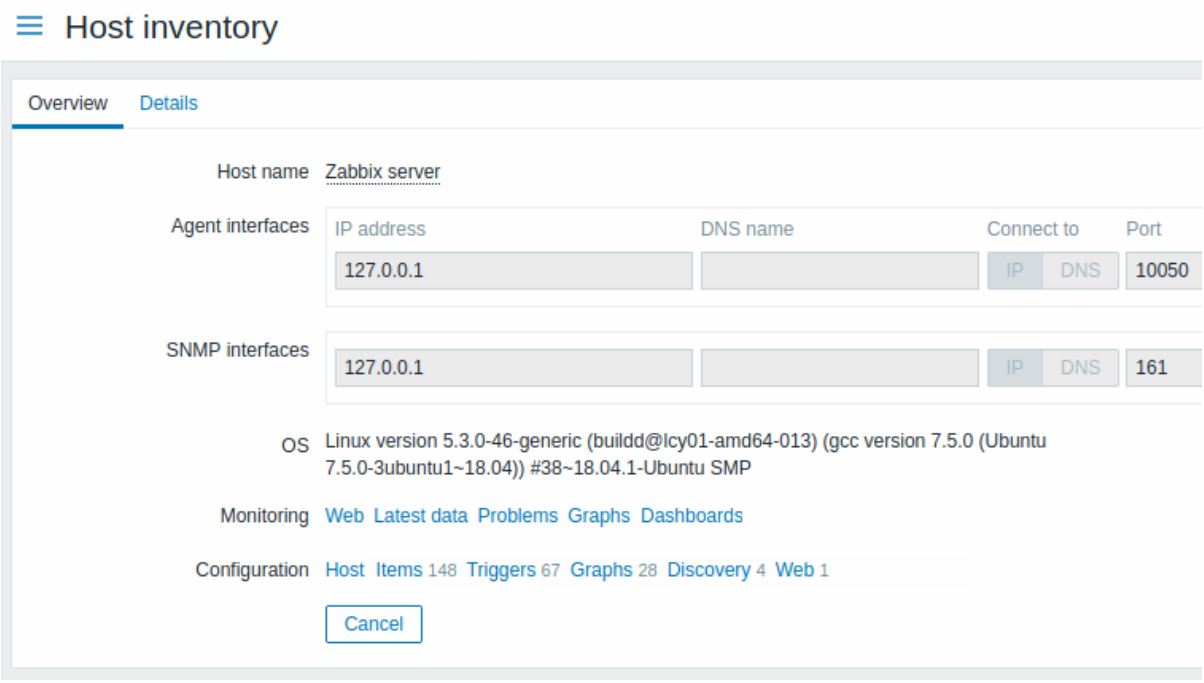


Lai attēlotu visu hostu inventāru, nospiediet pogu ”Atiestatīt”.

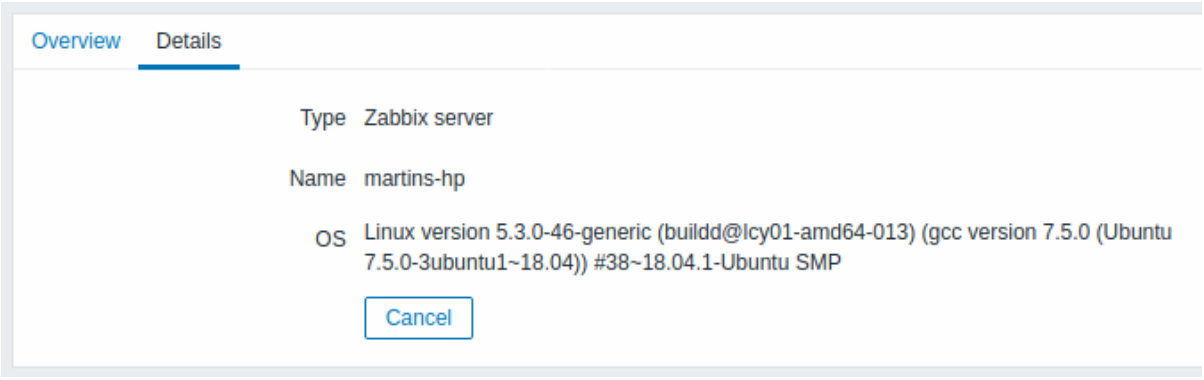
Lai gan tabulā tiek attēloti tikai daži galvenie inventāra lauki, varat apskatīt arī visu šim hostam pieejamo inventāra informāciju. Lai to izdarītu, noklikšķiniet uz hosta nosaukuma pirmajā kolonnā.

Inventāra informācija

Cilnē **Pārskats** ir pieejama vispārīga informācija par hostu, jaunākie uzraudzības dati un hosta konfigurācijas opcijas:



Cilnē **Detaļas** ir pieejama visa hostam pieejamā inventāra informācija:



Inventāra datu pilnīgums ir atkarīgs no tā, cik daudz inventāra informācijas tiek uzturēts hostam. Ja informācija netiek uzturēta, cilne *Detaļas* ir atspējota.

5 Pārskati

Pārskats

Izvēlnē Pārskati ir pieejamas vairākas sadaļas, kurās ir dažādi iepriekš definēti un lietotāja pielāgojami pārskati, kas paredzēti tādu parametru kā sistēmas informācija, trigeri un savāktie dati pārskata attēlošanai.

1 Sistēmas informācija

Pārskats

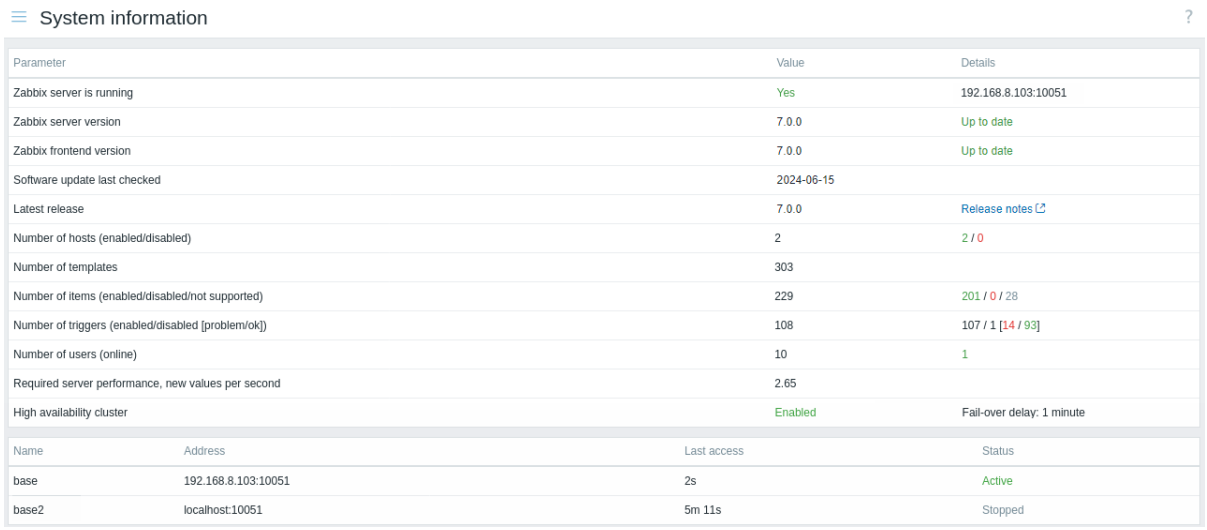
Sadaļā *Pārskati* → *Sistēmas informācija* tiek parādīts galveno Zabbix servera un sistēmas datu kopsavilkums. Sistēmas dati tiek vākti, izmantojot iekšējos vienumus.

Ņemiet vērā, ka augstas pieejamības iestatījumā ir iespējams novirzīt sistēmas informācijas avotu (servera instanci). Lai to izdarītu, rediģējiet failu *zabbix.conf.php* — noņemiet komentāru un iestatiet $ZBX_SERVER vai abus $ZBX_SERVER un $ZBX_SERVER_PORT uz serveri, kas nav tas, kurš tiek rādīts kā aktīvs. Ņemiet vērā, ka, iestatot tikai $ZBX_SERVER, tiks izmantota noklusējuma vērtība (10051) parametram $ZBX_SERVER_PORT.

Ja ir iespējots augstas pieejamības iestatījums, zem sistēmas statistikas tiek parādīts atsevišķs bloks ar informāciju par augstas pieejamības mezgliem. Šis bloks ir redzams tikai Zabbix *Super Admin* lietotājiem.

Sistēmas informācija ir pieejama arī kā paneļa logrīks.

Sistēmas statistika



Attēlotie dati:

Parametrs	Vērtība	Detalizēta informācija
Zabbix serveris darbojas	Zabbix servera statuss: **Jā** - serveris darbojas **Nē** - serveris nedarbojas *Piezīme:* Lai attēlotu pārējo informāciju, tīmekļa lietotāja saskarnei ir nepieciešams, lai serveris darbotos un serverī būtu palaists vismaz viens trapper process (parametrs StartTrappers failā zabbix_server.conf > 0).	Zabbix servera atrašanās vieta un ports.
Zabbix servera versija	Tiek attēlots pašreizējās servera versijas numurs. *Piezīme:* Tas tiek attēlots tikai tad, ja Zabbix serveris darbojas.	Tiek attēlots servera versijas statuss: **Aktuāla** - tiek izmantota jaunākā versija; **Pieejams jauns atjauninājums** - ir pieejama jaunāka versija; **Novecojusi** - šīs versijas pilna atbalsta periods ir beidzies. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek attēlots, ja pēdējā programmatūras atjauninājumu pārbaude tika veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.
Zabbix lietotāja saskarnes versija	Tiek attēlots Zabbix lietotāja saskarnes versijas numurs.	Tiek attēlots Zabbix lietotāja saskarnes versijas statuss: **Aktuāla** - tiek izmantota jaunākā versija; **Pieejams jauns atjauninājums** - ir pieejama jaunāka versija; **Novecojusi** - šīs versijas pilna atbalsta periods ir beidzies. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek attēlots, ja pēdējā programmatūras atjauninājumu pārbaude tika veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.
Programmatūras atjauninājumu pēdējā pārbaude	Tiek attēlots pēdējās Zabbix programmatūras atjauninājumu pārbaudes datums. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude.	

Parametrs	Vērtība	Detalizēta informācija
Jaunākais laidiens	Tiek attēlots jaunāka laidiena numurs (ja pieejams) pašreizējai Zabbix versijai. Šī informācija ir pieejama tikai tad, ja Zabbix servera konfigurācijā ir iespējota programmatūras atjauninājumu pārbaude. Nekas netiek attēlots, ja pēdējā programmatūras atjauninājumu pārbaude tika veikta pirms vairāk nekā nedēļas vai nav datu par pašreizējo versiju.	Tiek attēlota saite uz jaunākā pieejamā Zabbix laidiena laidiena piezīmēm.
Hostu skaits	Tiek attēlots kopējais konfigurēto hostu skaits.	Uzraudzīto/neuzraudzīto hostu skaits.
Veidņu skaits	Tiek attēlots kopējais veidņu skaits.	
Vienumu skaits	Tiek attēlots kopējais vienumu skaits.	Uzraudzīto/atspējoto/neatbalstīto hosta līmeņa vienumu skaits. Vienumi atspējotos hostos tiek skaitīti kā atspējoti.
Trigeru skaits	Tiek attēlots kopējais trigeru skaits.	Iespējoto/atspējoto hosta līmeņa trigeru skaits; iespējoto trigeru sadalījums atbilstoši stāvokļiem ”Problēma”/”Labi”. Trigeri, kas norādīti stāvoklī ”Labi”, ietver trigerus ar statusu ”Nezināms”. Trigeri, kas ir atkarīgi no atspējotiem vienumiem vai piešķirti atspējotiem hostiem, tiek skaitīti kā atspējoti.
Lietotāju skaits	Tiek attēlots kopējais konfigurēto lietotāju skaits.	Tiešsaistē esošo lietotāju skaits.
Nepieciešamā servera veiktspēja, jaunas vērtības sekundē	Tiek attēlots paredzamais jauno vērtību skaits, ko Zabbix serveris apstrādā sekundē.	*Nepieciešamā servera veiktspēja* ir aptuvens novērtējums un var būt noderīgs kā vadlīnija. Lai iegūtu precīzus apstrādāto vērtību skaitļus, izmantojiet `zabbix[wcache,values,all]` iekšējo vienumu. Aprēķinā tiek iekļauti iespējotie vienumi no uzraudzītajiem hostiem. Žurnāla vienumi tiek skaitīti kā viena vērtība katrā vienuma atjaunināšanas intervālā. Tiek skaitītas regulārā intervāla vērtības; elastīgā un plānošanas intervāla vērtības netiek skaitītas. Aprēķins netiek koriģēts ”nodata” uzturēšanas perioda laikā. Trapper vienumi netiek skaitīti.
Globālie skripti Zabbix serverī	Šajā laukā tiks attēlots **Atspējots**, ja globālie skripti Zabbix serverī ir atspējoti, servera konfigurācijā iestatot EnableGlobalScripts=0.	
Augstas pieejamības klasteris	Zabbix servera augstas pieejamības klastera statuss: **Atspējots** - atsevišķs serveris **Iespējots** - pastāv vismaz viens augstas pieejamības mezgls	Ja iespējots, tiek attēlota pārslēgšanās aizture.

Sistēmas informācija attēlos arī kļūdas ziņojumu šādos gadījumos:

- Izmantotajai datubāzei nav nepieciešamās rakstzīmju kopas vai kārtošanas secības (UTF-8).
- Datubāzes versija ir zemāka vai augstāka par atbalstīto diapazonu (pieejams tikai lietotājiem ar *Super admin role* tipu).
- Housekeeping priekš TimescaleDB ir nepareizi konfigurēts (vēstures vai tendenču tabulās ir saspiesti gabali, bet opcijas *Override item history period* vai *Override item trend period* ir atspējotas).

Augstas pieejamības mezgli

Ja ir iespējota augstas pieejamības kopa, tad tiek parādīts vēl viens datu bloks ar katra augstas pieejamības mezgla statusu.

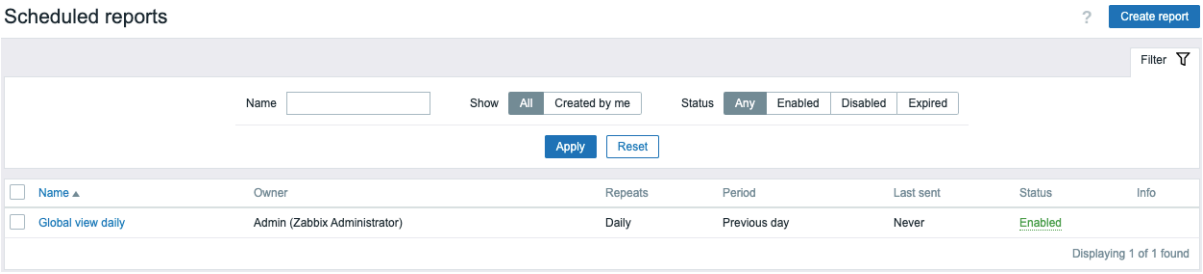
Parādītie dati:

Column	Description
Name	Mezgla nosaukums, kā definēts servera konfigurācijā.
Address	Mezgla IP adrese un ports.
Last access	Mezgla pēdējās piekļuves laiks. Novietojot kursoru virs šūnas, tiek parādīts pēdējās piekļuves laikspiedols pilnajā formātā.
Status	Mezgla statuss. Tabulas rindas tiek kārtotas pēc šiem statusiem šādā prioritātes secībā: **Active** - mezgls darbojas un ir pieejams **Unavailable** - mezgls nav bijis redzams ilgāk par pārslēgšanās aizturi (iespējams, vēlēsieties noskaidrot iemeslu) **Stopped** - mezgls ir apturēts vai to neizdevās palaist (iespējams, vēlēsieties to palaist vai dzēst) **Standby** - mezgls darbojas un gaida

2 Plānotie pārskati

Pārskats

Sadaļā *Pārskati* → *Plānotie pārskati* lietotāji ar pietiekamām atļaujām var konfigurēt informācijas paneļu PDF versiju plānoto ģenerēšanu, kas tiks nosūtītas pa e-pastu norādītajiem saņēmējiem.



Atvēršanas ekrānā tiek parādīta informācija par plānotajiem pārskatiem, ko var filtrēt ērtākai navigācijai — skatiet tālāk esošo sadaļu Filtra izmantošana.

Parādītie dati:

Kolonna	Apraksts
Nosaukums	Pārskata nosaukums. Noklikšķinot uz tā, tiek atvērta pārskata konfigurācijas forma.
Īpašnieks	Lietotājs, kurš izveidoja pārskatu.
Atkārtojas	Pārskata ģenerēšanas biežums (katru dienu/katru nedēļu/katru mēnesi/katru gadu).
Periods	Laika periods, par kuru tiek sagatavots pārskats (iepriekšējā diena, nedēļa, mēnesis vai gads).
Pēdējo reizi nosūtīts	Datums un laiks, kad tika nosūtīts jaunākais pārskats.
Statuss	Pašreizējais pārskata statuss (iespējots/atspējots/beidzies). Lietotāji ar pietiekamām atļaujām var mainīt statusu, noklikšķinot uz tā — no ”Iespējots” uz ”Atspējots” (un atpakaļ); no ”Beidzies” uz ”Atspējots” (un atpakaļ). Lietotājiem ar nepietiekamām tiesībām statuss nav noklikšķināms.

Kolonna	Apraksts
Informācija	Parāda informatīvas ikonas: Sarkana ikona norāda, ka pārskata ģenerēšana nav izdevusies; novietojot kursoru virs tās, tiks parādīts uznirstošs paskaidrojums ar informāciju par kļūdu. Dzeltena ikona norāda, ka pārskats tika ģenerēts, bet nosūtīšana dažiem (vai visiem) saņēmējiem nav izdevusies, vai arī pārskata derīguma termiņš ir beidzies; novietojot kursoru virs tās, tiks parādīts uznirstošs paskaidrojums ar papildu informāciju.

Filtra izmantošana

Jūs varat izmantot filtru, lai sašaurinātu atskaišu sarakstu. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātiem makrosiem.

Ir pieejamas šādas filtrēšanas opcijas:

- *Nosaukums* - ir atļauta daļēja nosaukuma atbilstība
- *Rādīt* - pašreizējā lietotāja izveidotās vai visas atskaites
- *Statuss* - izvēlieties starp ”Jebkurš” (rādīt visas atskaites), ”Iespējots”, ”Atspējots” vai ”Beidzies”

Filtrs atrodas zem sadaļas *Plānotās atskaites* nosaukuma. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filtrs* augšējā labajā stūrī.

Masveida atjaunināšana

Dažkārt var būt nepieciešams vienlaikus dzēst vai mainīt vairāku atskaišu statusu. Tā vietā, lai atvērtu katru atsevišķo atskaiti rediģēšanai, šim nolūkam varat izmantot masveida atjaunināšanas funkciju.

Lai masveidā atjauninātu dažas atskaites, rīkojieties šādi:

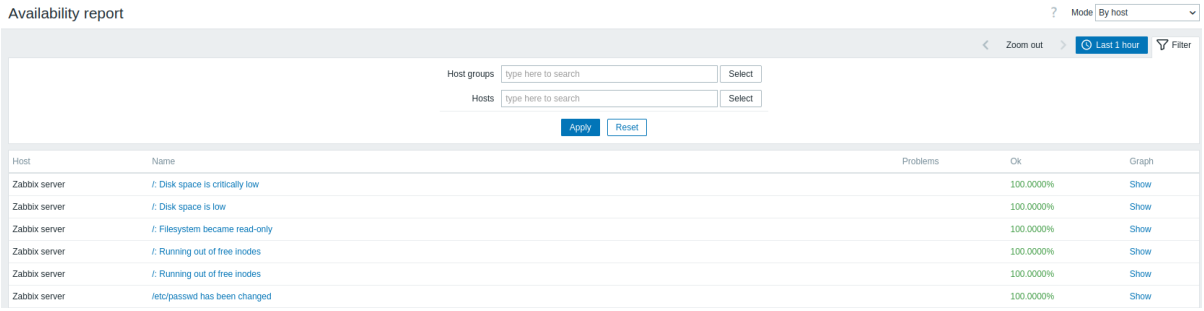
- Sarakstā atzīmējiet to atskaišu izvēles rūtiņas, kuras vēlaties atjaunināt
- Noklikšķiniet uz vajadzīgās pogas zem saraksta, lai veiktu izmaiņas (*Iespējot*, *Atspējot* vai *Dzēst*)

3 Pieejamības atskaite

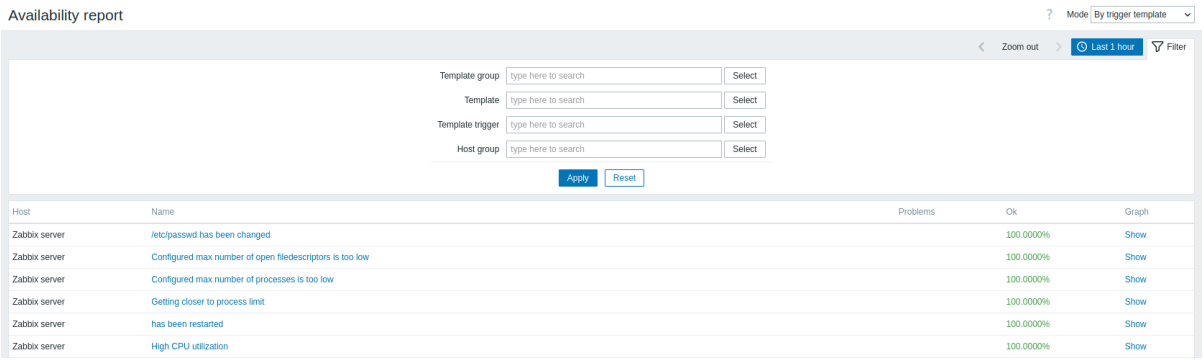
Pārskats

Sadaļā *Pārskati* > *Pieejamības pārskats* varat redzēt, kādu laika daļu katrs trigeris ir bijis problēmas/OK stāvoklī.

Katram stāvoklim tiek attēlots laika procentuālais īpatsvars, kas ļauj viegli noteikt dažādu jūsu sistēmas elementu pieejamību.



Nolaižamajā sarakstā augšējā labajā stūrī varat izvēlēties atlases režīmu — vai attēlot trigerus pēc hostiem vai pēc trigeriem, kas pieder veidnei.



Trigera nosaukums ir saite uz šī trigera jaunākajiem notikumiem.

Filtra izmantošana

Filtrs var palīdzēt sašaurināt parādīto hostu un/vai trigeru skaitu. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makrosiem.

Filtrs atrodas zem sadaļas *Pieejamības atskaite* nosaukuma. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filtrs* labajā pusē.

Filtrēšana pēc trigera veidnes

Režīmā *Pēc trigera veidnes* rezultātus var filtrēt pēc viena vai vairākiem tālāk norādītajiem parametriem.

Parametrs	Apraksts
Veidņu grupa	Filtrēt hostus pēc trigeriem, kas ir mantoti no veidnēm, kuras pieder atlasītajai veidņu grupai. Norādot vecāku veidņu grupu, tiek netieši atlasītas arī visas tajā ietvertās ligzdotās veidņu grupas.
Veidne	Filtrēt hostus pēc trigeriem, kas ir mantoti no atlasītās veidnes, ieskaitot ligzdotās veidnes. Ja ligzdotai veidnei ir savi trigeri, šie trigeri netiks parādīti.
Veidnes trigeris	Filtrēt hostus pēc atlasītā trigera. Citi filtrēto hostu trigeri netiks parādīti.
Hostu grupa	Filtrēt hostus, kas pieder atlasītajai hostu grupai.

Filtrēšana pēc hosta

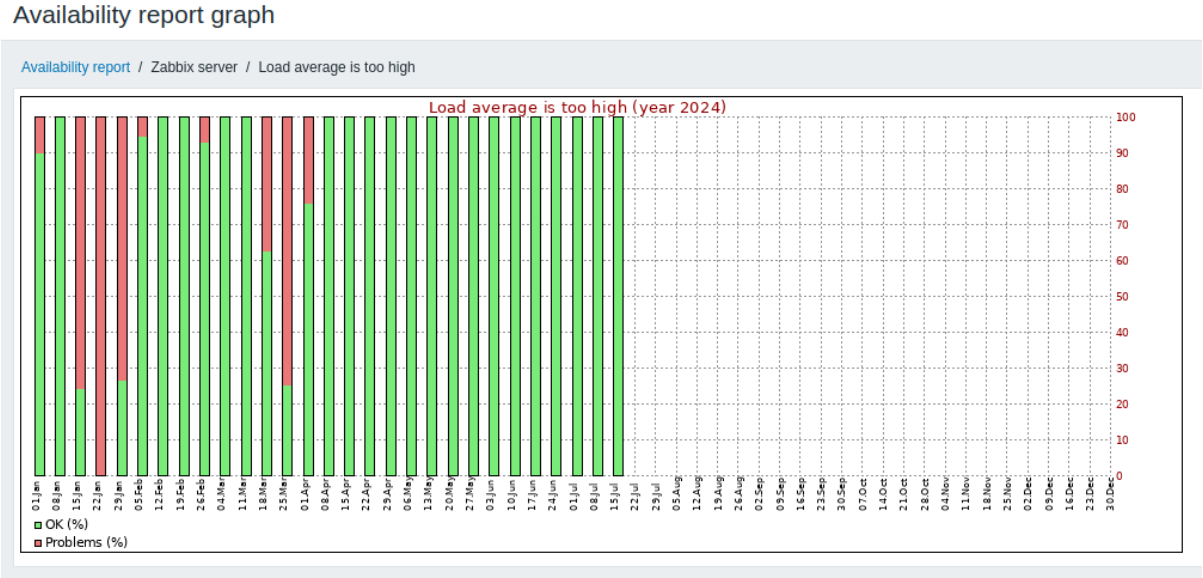
Režīmā *Pēc hosta* rezultātus var filtrēt pēc hosta vai hostu grupas. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās ligzdotās hostu grupas.

Laika perioda atlasītājs

Laika perioda atlasītājs ļauj ar vienu klikšķi atlasīt bieži izmantotus laika periodus. Atlasītāju var izvērst un sakļaut, noklikšķinot uz cilnes *Laika periods* blakus filtram.

Noklikšķinot uz *Rādīt* kolonnā *Grafiks*, pieejamības informācija tiek attēlota joslu diagrammā, kur katra josla attēlo kādu no pašreizējā gada aizvadītajām nedēļām.

Availability report graph



Joslas zaļā krāsa apzīmē OK laiku, bet sarkanā — problēmas laiku.

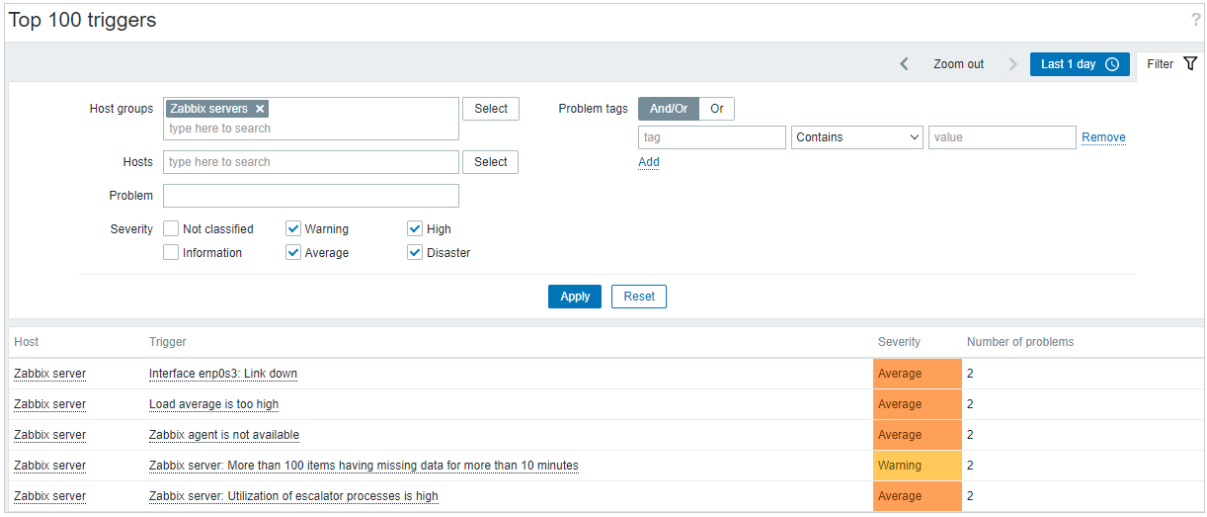
Uzturēšanas periodu ietekme

Uzturēšana automātiski neizslēdz laiku no pieejamības pārskata. Tikai uzturēšana, kas konfigurēta kā *Datu vākšana nenotiek*, aptur aptaujāšanu (tāpēc problēmas netiek ģenerētas) un tādējādi atstāj pieejamības procentu nemainīgu ietekmētajiem trigeriem.

4 Top 100 trigeri

Pārskats

Sadaļā *Pārskati* → *Top 100 trigeri* varat redzēt trigerus ar lielāko atlasītajā periodā konstatēto problēmu skaitu.



Gan hosta, gan trigeru kolonnas ieraksti ir saites, kas piedāvā dažas noderīgas iespējas:

- hostam — noklikšķinot uz hosta nosaukuma, tiek atvērta hosta izvēlne
- trigerim — noklikšķinot uz trigera nosaukuma, tiek atvērtas saites uz jaunākajiem notikumiem, vienkāršo grafiku katram trigera vienumam, kā arī paša trigera un katra trigera vienuma konfigurācijas veidlapām

Filtra izmantošana

Varat izmantot filtru, lai attēlotu trigerus pēc hostu grupas, hosta, problēmas nosaukuma, tagiem vai trigera svarīguma. Norādot vecāku hostu grupu, tiek netieši atlasītas arī visas ligzdotās hostu grupas. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Filtrs atrodas zem sadaļas *Top 100 trigeri* nosaukuma. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filtrs* labajā pusē.

Laika perioda atlasītājs

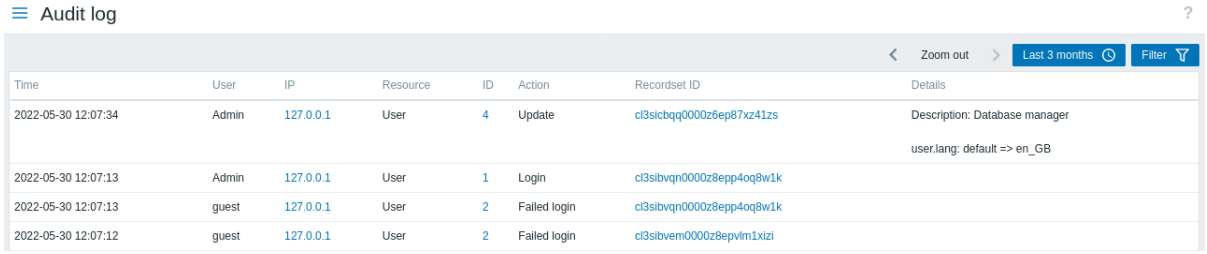
Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamos periodus. Plašāku informāciju skatiet sadaļā Laika perioda un hostu atlasītāji.

5 Audita žurnāls

Pārskats

Sadaļā *Pārskati* → *Audita žurnāls* var apskatīt lietotāju un sistēmas aktivitāšu ierakstus.

Note:
Lai audita ieraksti tiktu vākti un attēloti, sadaļā *Administrēšana* → *Audita žurnāls* ir jāatzīmē izvēles rūtiņa *Iespējot audita reģistrēšanu*. Ja šis iestatījums nav iespējots, aktivitāšu vēsture netiks ierakstīta datubāzē un netiks rādīta audita žurnālā.



Audita žurnālā tiek attēloti šādi dati:

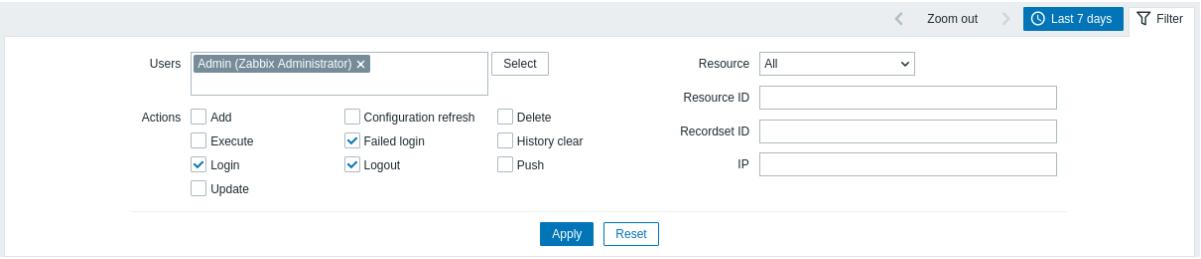
Column	Description
Time	Audita ieraksta laikspiedols.
User	Lietotājs, kurš veica aktivitāti.
IP	IP adrese, no kuras aktivitāte tika uzsākta. Noklikšķinot uz hipersaites, audita žurnāla ieraksti tiks filtrēti pēc šīs IP adreses.
Resource	Ietekmētā resursa tips (*API token*, *Darbība*, *Autentifikācija*, *Automātiskā reģistrācija* u.c.).
ID	Ietekmētā resursa ID. Noklikšķinot uz hipersaites, audita žurnāla ieraksti tiks filtrēti pēc šī resursa ID.

Column	Description
Action	Aktivitātes tips (*Pievienot*, *Konfigurācijas atsvaidzināšana*, *Dzēst*, *Izpildīt*, *Neveiksmīga pieteikšanās*, *Vēstures tīrīšana*, *Pieteikšanās*, *Atteikšanās*, *Push*, *Atjaunināt*).
Recordset ID	Kopīgs ID visiem audita žurnāla ierakstiem, kas izveidoti vienas un tās pašas operācijas rezultātā. Piemēram, sasaistot veidni ar hostu, katrai mantotajai veidnes entītijai (vienums, trigeris u.c.) tiek izveidots atsevišķs audita žurnāla ieraksts — visiem šiem ierakstiem būs vienāds *Recordset ID*. Noklikšķinot uz hipersaites, audita žurnāla ieraksti tiks filtrēti pēc šī *Recordset ID*.
Details	Resursa apraksts un detalizēta informācija par veikto aktivitāti. Ja ierakstā ir vairāk nekā divas rindas, tiks parādīta papildu saite *Details*. Noklikšķiniet uz šīs saites, lai skatītu pilnu izmaiņu sarakstu.

Note:
Ja trapper vienums vai HTTP aģenta vienums (ar iespējotu trapping) ir saņēmis datus, ieraksts audita žurnālā tiks pievienots tikai tad, ja dati tika nosūtīti, izmantojot API metodi `history.push`, nevis utilītu Zabbix sender.

Filtra izmantošana

Filtrs atrodas zem *Audita žurnāla* joslas. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filtrs* augšējā labajā stūrī.



Varat izmantot filtru, lai sašaurinātu ierakstus pēc lietotāja, ietekmētā resursa, resursa ID, veiktās darbības (*Recordset ID*) un IP adreses. Atkarībā no resursa filtrā var atlasīt vienu vai vairākas konkrētas darbības.

Lai nodrošinātu labāku meklēšanas veiktspēju, visi dati tiek meklēti ar neatrisinātām makro vērtībām.

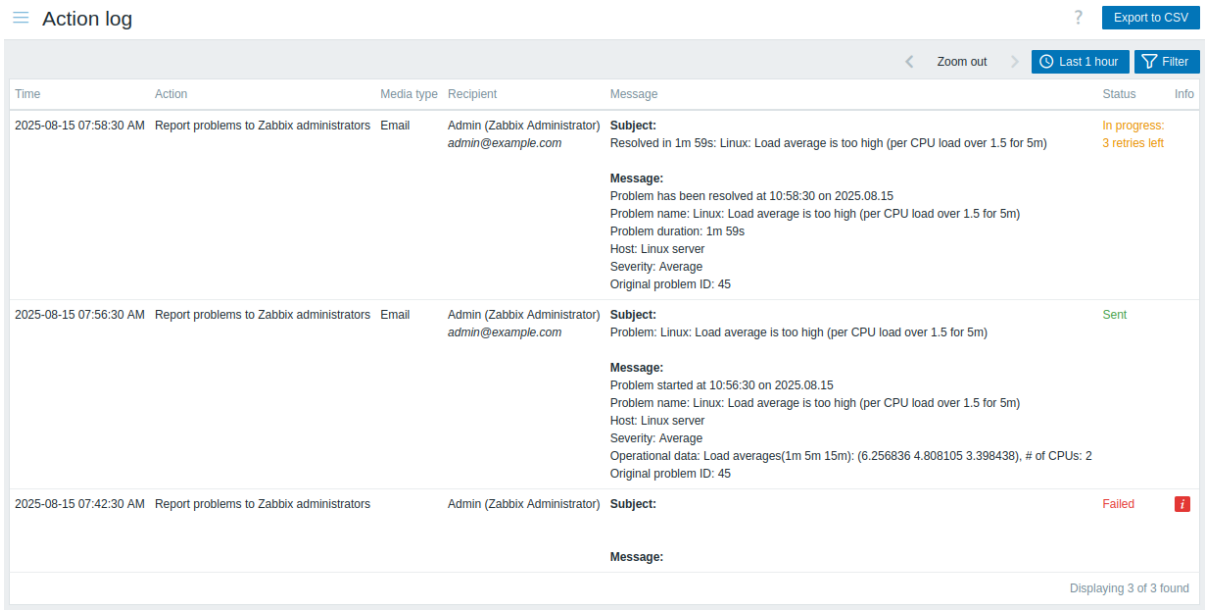
Laika perioda atlasītājs

Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi atlasīt bieži nepieciešamus periodus. Papildinformāciju skatiet sadaļā Laika perioda un hostu atlasītāji.

6 Darbību žurnāls

Pārskats

Sadaļā *Pārskati* → *Darbību žurnāls* varat skatīt detalizētu informāciju par operācijām (paziņojumiem, attālinātajām komandām), kas izpildītas darbības ietvaros.



Attēlotie dati:

Kolonna	Apraksts
Laiks	Operācijas laikspiedols.
Darbība	Darbības nosaukums, kas izraisīja operācijas.
Medija tips	Medija tips (piemēram, Email, Jira u.c.), kas izmantots paziņojuma nosūtīšanai. Operācijām, kurās tika izpildītas attālinātās komandas, šī kolonna būs tukša.
Saņēmējs	Informācija par paziņojuma saņēmēju — lietotājvārds, vārds un uzvārds (iekavās), kā arī papildu informācija atkarībā no medija tipa (e-pasts, lietotājvārds u.c.). Operācijām, kurās tika izpildītas attālinātās komandas, šī kolonna būs tukša.
Ziņojums	Ziņojuma saturs, attālinātā komanda vai globālā skripta nosaukums. Attālinātā komanda no mērķa hosta tiek atdalīta ar kolu: `<host>:<command>`. Piemēram, ja attālinātā komanda tika izpildīta uz Zabbix servera, informācija būs šādā formātā: `Zabbix server:<command>`.
Statuss	Operācijas statuss: *Procesā* - notiek paziņojuma nosūtīšanas operācija (tiek attēlots arī atlikušais reižu skaits, cik vēl serveris mēģinās nosūtīt paziņojumu) *Nosūtīts* - paziņojums ir nosūtīts *Izpildīts* - attālinātā komanda ir izpildīta *Neizdevās* - operācija nav pabeigta
Informācija	Kļūdas informācija (ja tāda ir) par operācijas izpildi.

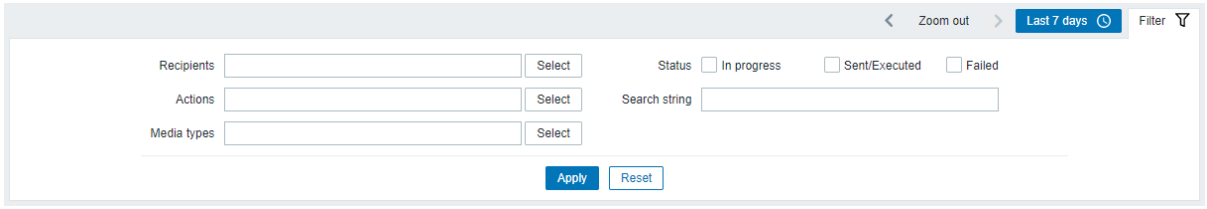
Pogas

Poga lapas augšējā labajā stūrī piedāvā šādu opciju:

Export to CSV	Eksportēt darbību žurnāla ierakstus no pašreizējās lapas uz CSV failu. Ja ir lietots filtrs, tiks eksportēti tikai filtrētie ieraksti. Eksportētajā CSV failā kolonnas ”Recipient” un ”Message” ir sadalītas vairākās kolonnās — ”Recipient’s Zabbix username”, ”Recipient’s name”, ”Recipient’s surname”, ”Recipient”, kā arī ”Subject”, ”Message”, ”Command”.

Filtra izmantošana

Filtrs atrodas zem *Darbību žurnāla* joslas. To var atvērt un sakļaut, noklikšķinot uz cilnes *Filtrs* lapas augšējā labajā stūrī.



Varat izmantot filtru, lai sašaurinātu ierakstus pēc paziņojumu saņēmējiem, darbībām, mediju tipiem, statusa vai pēc ziņojuma/attālās komandas satura (*Meklēšanas virkne*). Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātiem makrosiem.

Laika perioda atlasītājs

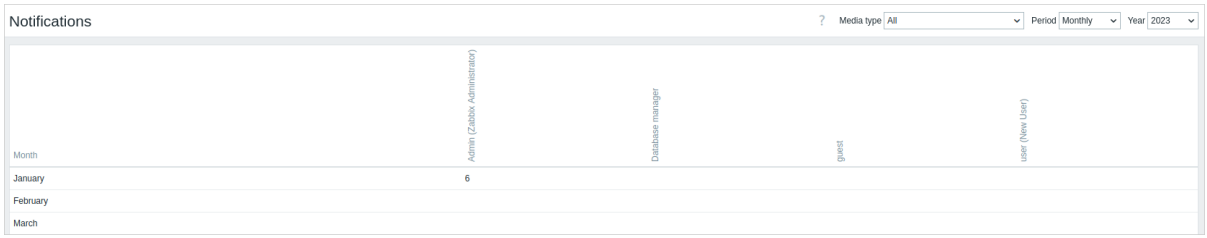
Laika perioda atlasītājs blakus filtram ļauj ar vienu peles klikšķi izvēlēties bieži nepieciešamus periodus. Plašāku informāciju skatiet Laika perioda un hostu atlasītāji.

7 Paziņojumi

Pārskats

Sadaļā *Pārskati* → *Paziņojumi* tiek attēlota atskaite par katram lietotājam nosūtīto paziņojumu skaitu.

Izmantojot nolaižamās izvēlnes augšējā labajā stūrī, varat izvēlēties medija veidu (vai visus), periodu (dati par katru dienu/nedēļu/mēnesi/gadu) un gadu nosūtītajiem paziņojumiem.



Katrā kolonnā tiek attēlotas kopsummas par vienu sistēmas lietotāju.

6 Datu vākšana

Pārskats

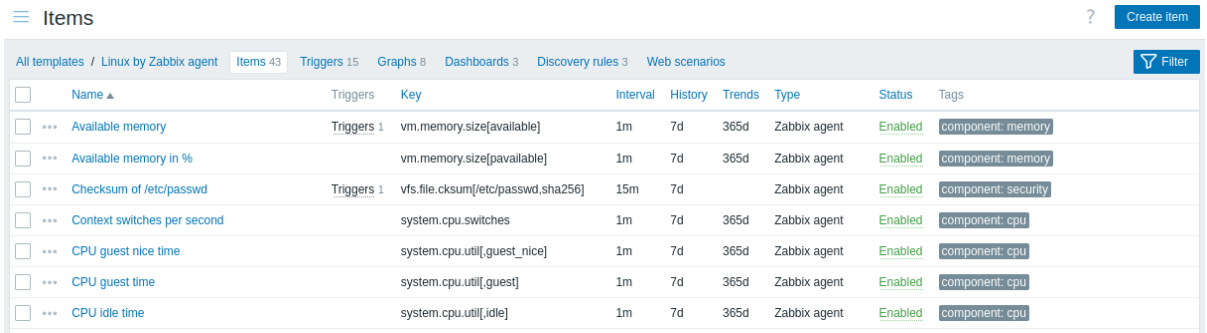
Šajā izvēlnē ir sadaļas, kas saistītas ar datu vākšanas konfigurēšanu.

1 Vienumi

Pārskats

Veidnes vienumu sarakstam var piekļūt sadaļā *Datu ievākšana* → *Veidnes*, noklikšķinot uz *Vienumi* attiecīgajai veidnei.

Tiek parādīts esošo vienumu saraksts.



Parādītie dati:

Kolonna	Apraksts
Vienuma izvēlne	Noklikšķiniet uz trīs punktu ikonas, lai atvērtu vienuma izvēlni.
Veidne	Veidne, kurai pieder vienums. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma. Šī kolonna tiek rādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai nav atlasīta neviena veidne.

Kolonna	Apraksts
Nosaukums	Vienuma nosaukums tiek parādīts kā zila saite uz vienuma informāciju. Noklikšķinot uz vienuma nosaukuma saites, tiek atvērta vienuma konfigurācijas forma. Ja vienums ir mantots no citas veidnes, pirms vienuma nosaukuma tiek parādīts veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts vienumu saraksts šīs veidnes līmenī.
Trigeri	Novietojot peles kursoru virs Trigeri, tiks parādīts informācijas lodziņš ar trigeriem, kas saistīti ar vienumu. Trigeru skaits tiek parādīts pelēkā krāsā.
Atslēga	Tiek parādīta vienuma atslēga.
Intervāls	Tiek parādīts pārbaudes biežums.
Vēsture	Tiek parādīts, cik dienas tiks glabāta vienuma datu vēsture.
Tendences	Tiek parādīts, cik dienas tiks glabāta vienuma tendenču vēsture.
Tips	Tiek parādīts vienuma tips (Zabbix aģents, SNMP aģents, vienkāršā pārbaude utt.).
Statuss	Tiek parādīts vienuma statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt — no Iespējots uz Atspējots (un atpakaļ).
Birkas	Tiek parādītas vienuma birkas. Var tikt parādītas līdz trim birkām (nosaukums:vērtība pāri). Ja birku ir vairāk, tiek parādīta saite "...", kas ļauj redzēt visas birkas, novietojot virs tās peles kursoru.

Lai konfigurētu jaunu vienumu, noklikšķiniet uz pogas *Izveidot vienumu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

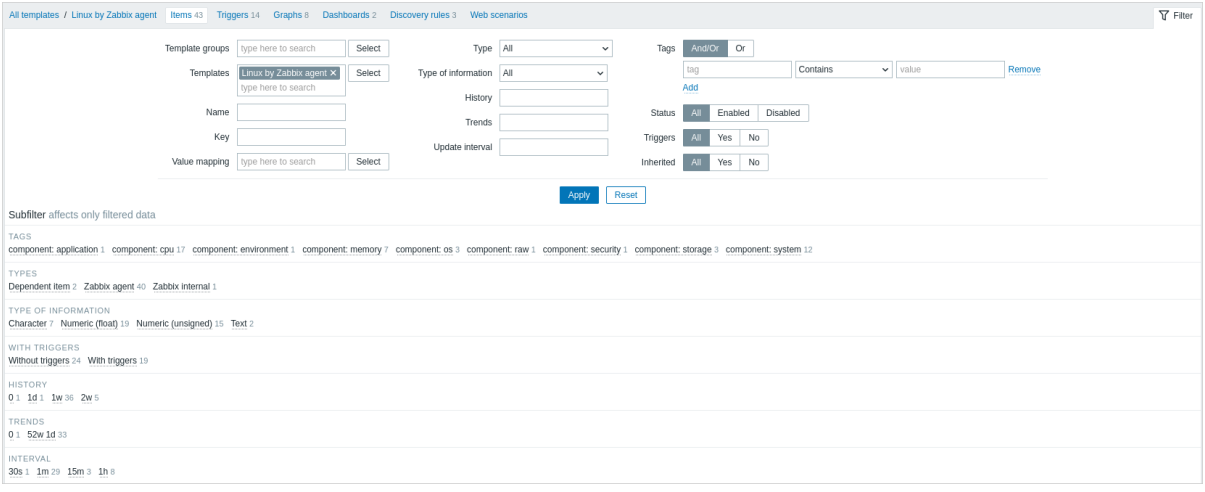
- *Iespējot* - mainīt vienuma statusu uz *Iespējots*.
- *Atspējot* - mainīt vienuma statusu uz *Atspējots*.
- *Kopēt* - kopēt vienumus uz citiem hostiem vai veidnēm.
- *Masveida atjaunināšana* - atjaunināt vairākus rekvizītus vairākiem vienumiem vienlaikus.
- *Dzēst* - dzēst vienumus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Vienumu sarakstā var būt ļoti daudz vienumu. Izmantojot filtru, varat atsijāt daļu no tiem, lai ātri atrastu meklētos vienumus. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makrokomandām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



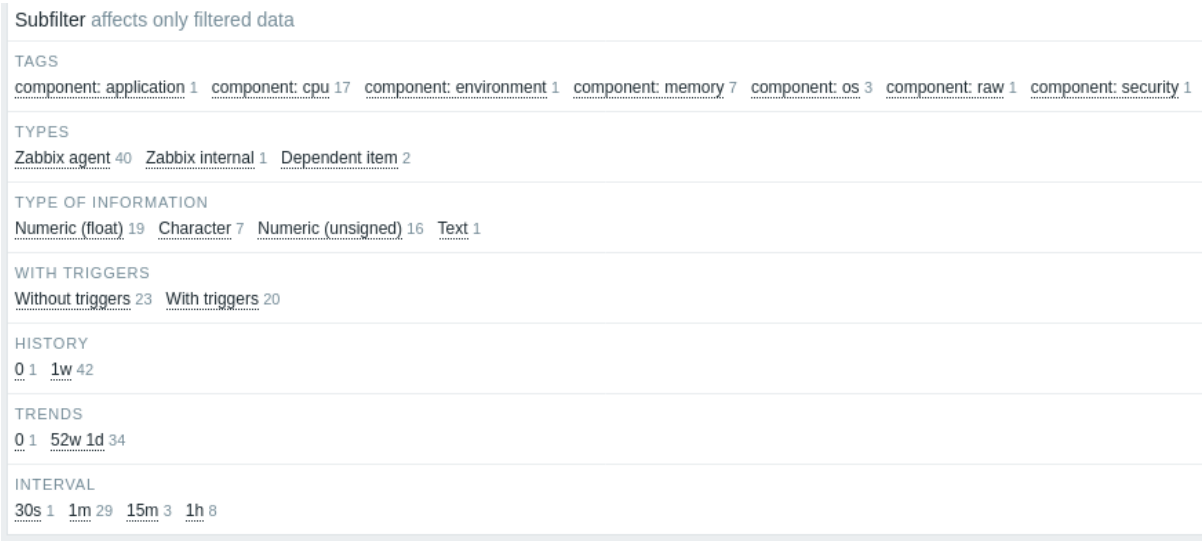
Parametrs	Apraksts
Veidņu grupas	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecākgrupas veidņu grupu, tiek netieši atlasītas arī visas ligzdotās grupas.
Veidnes	Filtrēt pēc vienas vai vairākām veidnēm.
Nosaukums	Filtrēt pēc vienuma nosaukuma.
Atslēga	Filtrēt pēc vienuma atslēgas.
Vērtību kartēšana	Filtrēt pēc izmantotās vērtību kartes. Šis parametrs netiek rādīts, ja opcija *Veidnes* ir tukša.

Parametrs	Apraksts
Tips	Filtrēt pēc vienuma tipa (Zabbix aģents, SNMP aģents u.c.).
Informācijas tips	Filtrēt pēc informācijas tipa (bez zīmes skaitlisks, peldošā komata skaitlis u.c.).
Vēsture	Filtrēt pēc tā, cik ilgi tiek glabāta vienuma vēsture.
Tendences	Filtrēt pēc tā, cik ilgi tiek glabātas vienuma tendences.
Atjaunināšanas intervāls	Filtrēt pēc vienuma atjaunināšanas intervāla.
Birkas	Norādiet birkas, lai ierobežotu parādīto vienumu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Neeksistē** — izslēgt norādītos birku nosaukumus **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina tipi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai **Vai** — pietiek, ja ir izpildīts viens nosacījums
Statuss	Filtrēt pēc vienuma statusa — *Iespējots* vai *Atspējots*.
Trigeri	Filtrēt vienumus ar trigeriem (vai bez tiem).
Mantots	Filtrēt vienumus, kas mantoti (vai nav mantoti) no piesaistītajām veidnēm.

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt filtrēšanu no galvenā filtra.

Tas satur klikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumi ir saistīti pēc kopīgas entītijas — birkas, vienuma tipa, vienuma atjaunināšanas intervāla utt. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (nav nepieciešams galvenajā filtrā noklikšķināt uz *Pielietot*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Atkārtoti noklikšķinot uz entītijas, filtrēšana tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kuru apjoms ir ierobežots līdz 1000 ierakstiem. Ja vēlaties apakšfiltrā redzēt vairāk ierakstu, jums jāpalielina parametra *Limit for search and filter results* vērtība (sadaļā *Administration -> General -> GUI*).

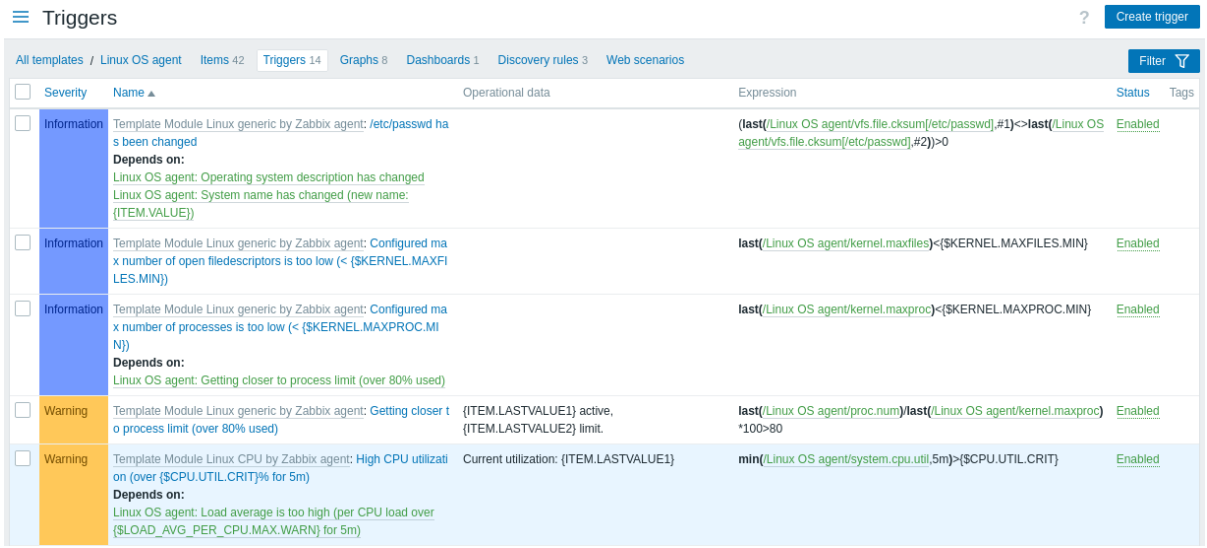
Atšķirībā no galvenā filtra, apakšfiltrs tiek atjaunināts ar katru tabulas atsvaidzināšanas pieprasījumu, lai vienmēr būtu pieejama aktuāla informācija par pieejamajām filtrēšanas opcijām un to skaitītāju vērtībām.

Horizontāli attēloto entītiju skaits ir ierobežots līdz 100. Ja to ir vairāk, beigās tiek parādīta trīs punktu ikona; tā nav klikšķināma.

Skaitlis blakus katrai klikšķināmajai entītijai norāda tajā grupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Noklikšķinot uz entītijas, skaitļi pie citām pieejamajām entītijām tiek parādīti ar pluszīmi, norādot, cik vienumu var pievienot pašreizējai atlasei. Entītijas bez vienumiem netiek rādītas, ja vien tās iepriekš nav atlasītas apakšfiltrā.

Pārskats

Veidnes trigeru sarakstam var piekļūt sadaļā *Datu ievākšana* → *Veidnes*, noklikšķinot uz *Trigeri* attiecīgajai veidnei.



Attēlotie dati:

Kolonna	Apraksts
Svarīgums	Trigera svarīgums tiek attēlots gan ar nosaukumu, gan šūnas fona krāsu.
Veidne	Veidne, kurai pieder trigeris. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma. Šī kolonna tiek rādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai nav atlasīta neviena veidne.
Nosaukums	Trigera nosaukums tiek attēlots kā zila saite uz trigera detalizēto informāciju. Noklikšķinot uz trigera nosaukuma saites, tiek atvērta trigera konfigurācijas forma. Ja trigeris ir mantots no citas veidnes, pirms trigera nosaukuma tiek attēlots veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts trigeru saraksts šīs veidnes līmenī.
Darbības dati	Trigera darbības datu definīcija, kas satur patvaļīgas virknes un makrosus, kuri tiks dinamiski atrisināti sadaļā *Uzraudzība* → *Problēmas*.
Izteiksme	Tiek attēlota trigera izteiksme. Izteiksmes veidnes-vienuma daļa tiek attēlota kā saite, kas ved uz vienuma konfigurācijas formu.
Statuss	Tiek attēlots trigera statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt — no Iespējots uz Atspējots (un atpakaļ).
Birkas	Ja trigeris satur birkas, šajā kolonnā tiek attēlots birkas nosaukums un vērtība.

Lai konfigurētu jaunu trigeri, noklikšķiniet uz pogas *Izveidot trigeri* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

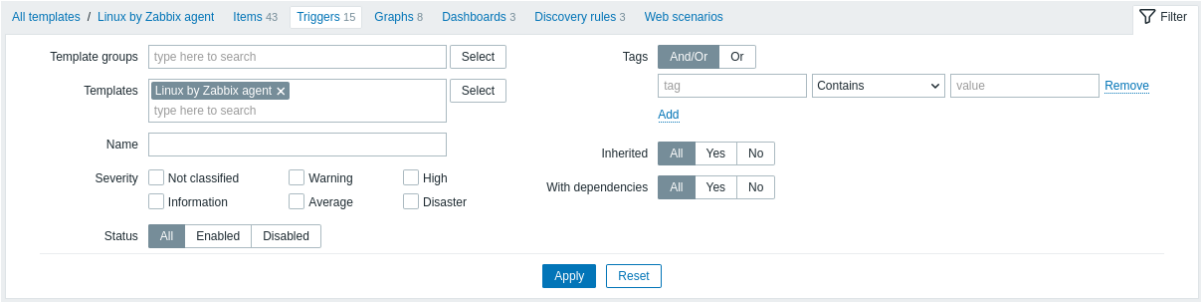
- *Enable* — mainīt trigera statusu uz *Enabled*
- *Disable* — mainīt trigera statusu uz *Disabled*
- *Copy* — kopēt trigerus uz citiem hostiem vai veidnēm
- *Mass update* — vienlaikus atjaunināt vairākas īpašības vairākiem trigeriem
- *Delete* — dzēst trigerus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem trigeriem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tos trigerus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



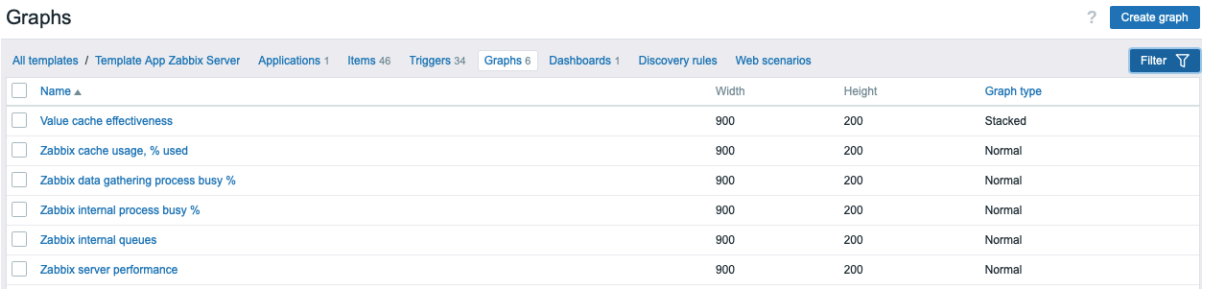
Parametrs	Apraksts
Veidņu grupas	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecākgrupas veidņu grupu, tiek netieši atlasītas arī visas ligzdotās grupas.
Veidnes	Filtrēt pēc vienas vai vairākām veidnēm. Ja iepriekš augstāk jau ir atlasītas veidņu grupas, veidņu atlase ir ierobežota ar šīm grupām.
Nosaukums	Filtrēt pēc trigera nosaukuma.
Svarīgums	Atlasiet, lai filtrētu pēc viena vai vairākiem trigeru svarīguma līmeņiem.
Statuss	Filtrēt pēc trigera aktivizācijas statusa (Iespējots/Atspējots).
Birkas	Filtrēt pēc trigera birkas nosaukuma un vērtības. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Neeksistē** - izslēgt norādītos birku nosaukumus **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - visiem nosacījumiem jābūt izpildītiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Vai nosacījuma **Vai** - pietiek, ja ir izpildīts viens nosacījums Makrokomandas un makrokomandu funkcijas tiek atbalstītas birkas nosaukuma un birkas vērtības laukos.
Mantots	Filtrēt trigerus, kas mantoti (vai nav mantoti) no piesaistītajām veidnēm.
Ar atkarībām	Filtrēt trigerus ar (vai bez) atkarībām.

3 Grafiki

Pārskats

Pielāgoto grafiku sarakstam veidnei var piekļūt sadaļā *Datu ievākšana* → *Veidnes*, noklikšķinot uz *Grafiki* attiecīgajai veidnei.

Tiek parādīts esošo grafiku saraksts.



Parādītie dati:

Kolonna	Apraksts
Veidne	Veidne, kurai pieder grafiks. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma. Šī kolonna tiek rādīta tikai tad, ja filtrā ir atlasītas vairākas veidnes vai nav atlasīta neviena veidne.
Nosaukums	Pielāgotā grafika nosaukums, kas tiek parādīts kā zila saite uz grafika detalizēto informāciju. Noklikšķinot uz grafika nosaukuma saites, tiek atvērta grafika konfigurācijas forma. Ja grafiks ir mantots no citas veidnes, pirms grafika nosaukuma tiek parādīts veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts grafiku saraksts šīs veidnes līmenī.
Platums	Tiek parādīts grafika platums.
Augstums	Tiek parādīts grafika augstums.
Grafika tips	Tiek parādīts grafika tips — *Parasts*, *Sakrauts*, *Sektoru* vai *Izcelts*.

Lai konfigurētu jaunu grafiku, noklikšķiniet uz pogas *Izveidot grafiku* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Kopēt* - kopēt grafikus uz citiem hostiem vai veidnēm
- *Dzēst* - dzēst grafikus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafikiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

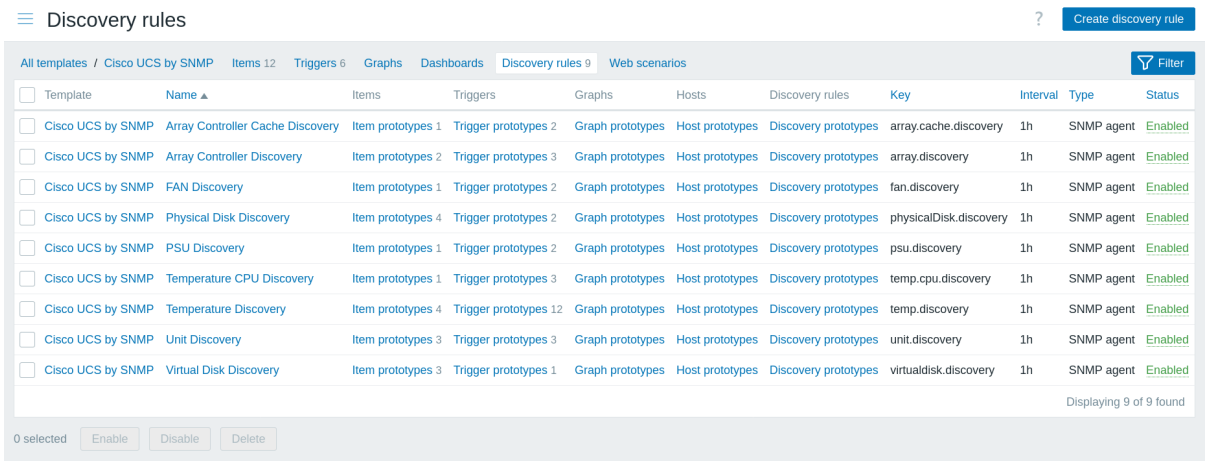
Jūs varat filtrēt grafikus pēc veidņu grupas un veidnes. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

4 Atklāšanas kārtulas

Pārskats

Zema līmeņa atklāšanas noteikumu sarakstam veidnei var piekļūt sadaļā *Datu ievākšana* → *Veidnes*, noklikšķinot uz *Atklāšana* attiecīgajai veidnei.

Tiek parādīts esošo zema līmeņa atklāšanas noteikumu saraksts. Ir arī iespējams skatīt visus atklāšanas noteikumus neatkarīgi no veidnes vai visus konkrētas veidņu grupas atklāšanas noteikumus, mainot filtra iestatījumus.



Parādītie dati:

Kolonna	Apraksts
Veidne	Veidne, kurai pieder atklāšanas noteikums. Noklikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma.

Kolonna	Apraksts
Nosaukums	Noteikuma nosaukums, kas tiek parādīts kā zila saite. Noklikšķinot uz noteikuma nosaukuma, tiek atvērta zema līmeņa atklāšanas noteikuma konfigurācijas forma. Ja atklāšanas noteikums ir mantots no citas veidnes, pirms noteikuma nosaukuma tiek parādīts veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts atklāšanas noteikumu saraksts šīs veidnes līmenī.
Vienumi	Tiek parādīta saite uz vienumu prototipu sarakstu. Esošo vienumu prototipu skaits tiek parādīts pelēkā krāsā.
Trigeri	Tiek parādīta saite uz trigeru prototipu sarakstu. Esošo trigeru prototipu skaits tiek parādīts pelēkā krāsā.
Grafiki	Tiek parādīta saite uz grafiku prototipu sarakstu. Esošo grafiku prototipu skaits tiek parādīts pelēkā krāsā.
Hosti	Tiek parādīta saite uz hostu prototipu sarakstu. Esošo hostu prototipu skaits tiek parādīts pelēkā krāsā.
Atslēga	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Intervāls	Tiek parādīts atklāšanas izpildes biežums.
Tips	Tiek parādīts atklāšanai izmantotā vienuma tips (Zabbix aģents, SNMP aģents u.c.).
Statuss	Tiek parādīts atklāšanas noteikuma statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt — no Iespējots uz Atspējots (un atpakaļ).

Lai konfigurētu jaunu zema līmeņa atklāšanas noteikumu, noklikšķiniet uz pogas *Izveidot atklāšanas noteikumu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

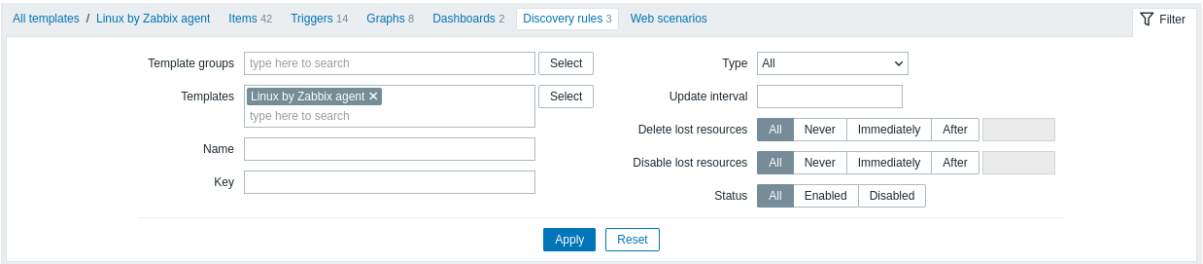
- *Iespējot* — mainīt zema līmeņa atklāšanas kārtulas statusu uz *Iespējots*
- *Atspējot* — mainīt zema līmeņa atklāšanas kārtulas statusu uz *Atspējots*
- *Dzēst* — dzēst zema līmeņa atklāšanas kārtulas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām atklāšanas kārtulām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tās atklāšanas kārtulas, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus, piemēram, veidni, atklāšanas kārtulas nosaukumu, vienuma atslēgu, vienuma tipu utt.



Parametrs	Apraksts
Veidņu grupas	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecākgrupas veidņu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas.
Veidnes	Filtrēt pēc vienas vai vairākām veidnēm.
Nosaukums	Filtrēt pēc atklāšanas kārtulas nosaukuma.
Atslēga	Filtrēt pēc atklāšanas vienuma atslēgas.
Tips	Filtrēt pēc atklāšanas vienuma tipa.
Atjaunināšanas intervāls	Filtrēt pēc atjaunināšanas intervāla. Nav pieejams Zabbix trapper un atkarīgajiem vienumiem.
Zaudēto resursu dzēšana	Filtrēt pēc perioda *Zaudēto resursu dzēšana*.
Zaudēto resursu atspējošana	Filtrēt pēc perioda *Zaudēto resursu atspējošana*.

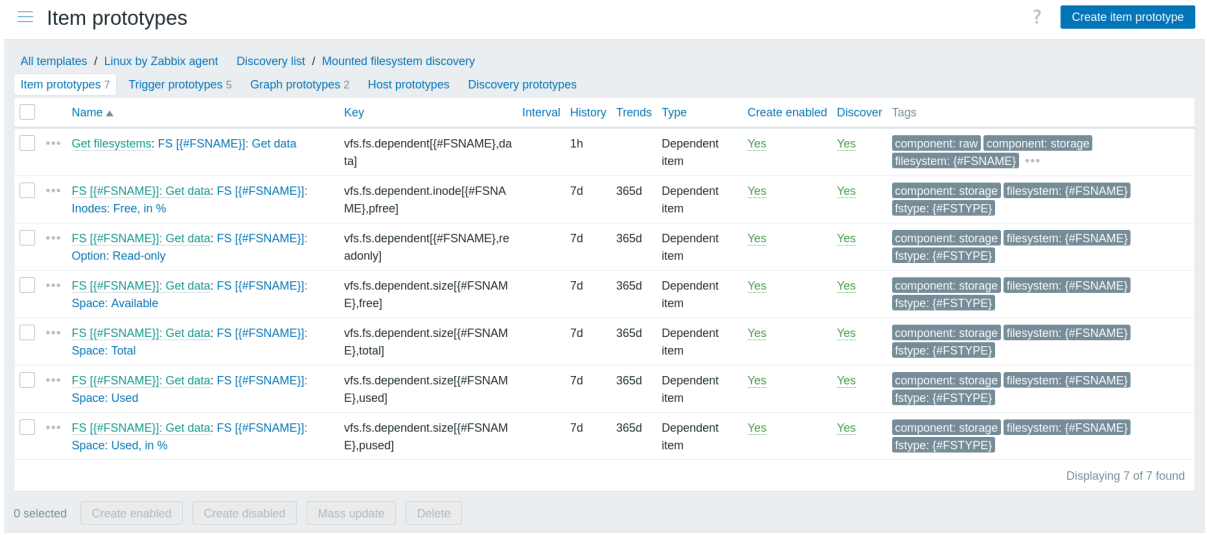
Parametrs	Apraksts
Statuss	Filtrēt pēc atklāšanas kārtulas statusa (Visi/Iespējots/Atspējots).

1 Vienumu prototipi

Pārskats

Šajā sadaļā tiek attēloti veidnē konfigurētie zema līmeņa atklāšanas kārtulas vienumu prototipi.

Ja veidne ir piesaistīta hostam, vienumu prototipi kļūs par pamatu reālu hosta vienumu izveidei zema līmeņa atklāšanas laikā.



Attēlotie dati:

Column	Description
Name	Vienuma prototipa nosaukums tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta vienuma prototipa konfigurācijas forma. Ja vienuma prototips pieder piesaistītai veidnei, pirms vienuma prototipa nosaukuma kā pelēka saite tiek attēlots veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts vienumu prototipu saraksts piesaistītās veidnes līmenī. Ja vienuma prototips ir tipa ”Dependent item”, pirms vienuma prototipa nosaukuma zaļā krāsā tiek attēlots galvenā vienuma nosaukums (tāpat kā vienumu prototipu gadījumā hosta līmenī).
Key	Tiek attēlota vienuma prototipa atslēga.
Interval	Tiek attēlots pārbaudes biežums.
History	Tiek attēlots, cik dienas saglabāt vienuma datu vēsturi.
Trends	Tiek attēlots, cik dienas saglabāt vienuma tendenču vēsturi.
Type	Tiek attēlots vienuma prototipa tips (Zabbix aģents, SNMP aģents, vienkāršā pārbaude u.c.).
Create enabled	Izveidot vienumu, pamatojoties uz šo prototipu, kā: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp ’Yes’ un ’No’, noklikšķinot uz tiem.
Discover	Atklāt vienumu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp ’Yes’ un ’No’, noklikšķinot uz tiem.
Tags	Tiek attēlotas vienuma prototipa birkas.

Lai konfigurētu jaunu vienuma prototipu, noklikšķiniet uz pogas *Create item prototype* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotus* - izveidot šos vienumus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos vienumus kā *Atspējotus*
- *Masveida atjaunināšana* - masveidā atjaunināt šos vienumu prototipus
- *Dzēst* - dzēst šos vienumu prototipus

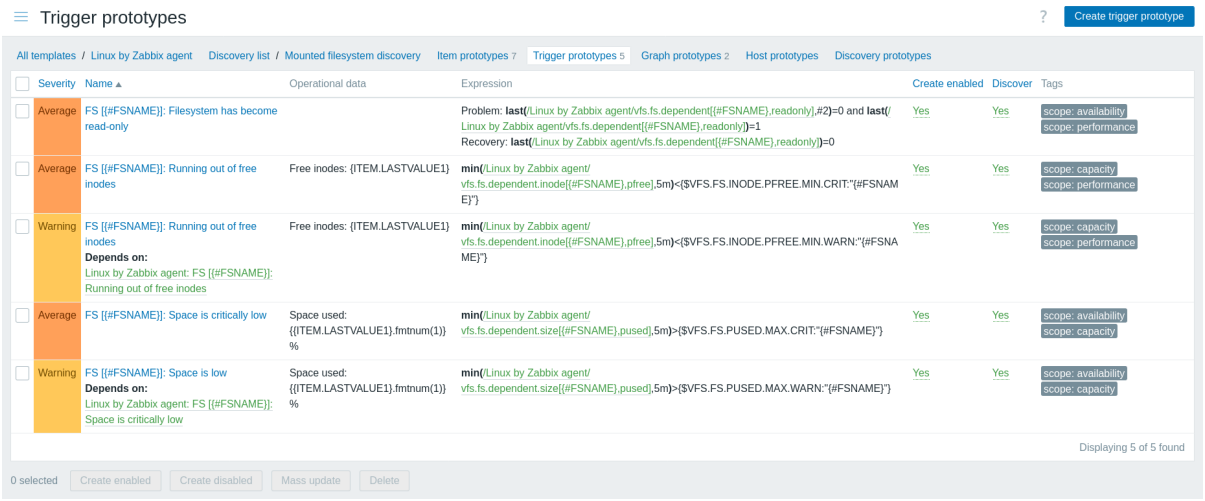
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

2 Trigeru prototipi

Pārskats

Šajā sadaļā tiek parādīti veidnē konfigurētie zema līmeņa atklāšanas kārtulas trigeru prototipi.

Ja veidne ir piesaistīta hostam, trigeru prototipi kļūs par pamatu reālu hosta trigeru izveidei zema līmeņa atklāšanas laikā.



Parādītie dati:

Column	Description
Severity	Trigera smaguma pakāpe tiek parādīta gan ar nosaukumu, gan ar šūnas fona krāsu.
Name	Trigera prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta trigera prototipa konfigurācijas forma. Ja trigera prototips pieder piesaistītai veidnei, pirms trigera nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts trigeru prototipu saraksts piesaistītās veidnes līmenī.
Expression	Tiek parādīta trigera izteiksme. Izteiksmes daļa template-item tiek parādīta kā saite, kas ved uz vienuma konfigurācijas formu.
Operational data	Tiek parādīts trigera operacionālo datu formāts, kas satur patvaļīgas virknes un makrosus, kuri dinamiski tiks atrisināti sadaļā *Monitoring* → *Problems*.
Create enabled	Izveidot trigeri, pamatojoties uz šo prototipu, kā: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Discover	Atklāt trigeri, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Tags	Tiek parādīti trigera prototipa tagi.

Lai konfigurētu jaunu trigera prototipu, noklikšķiniet uz pogas *Create trigger prototype* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotus* — izveidot šos trigerus kā *Iespējotus*
- *Izveidot atspējotus* — izveidot šos trigerus kā *Atspējotus*
- *Masveida atjaunināšana* — masveidā atjaunināt šos trigeru prototipus
- *Dzēst* — dzēst šos trigeru prototipus

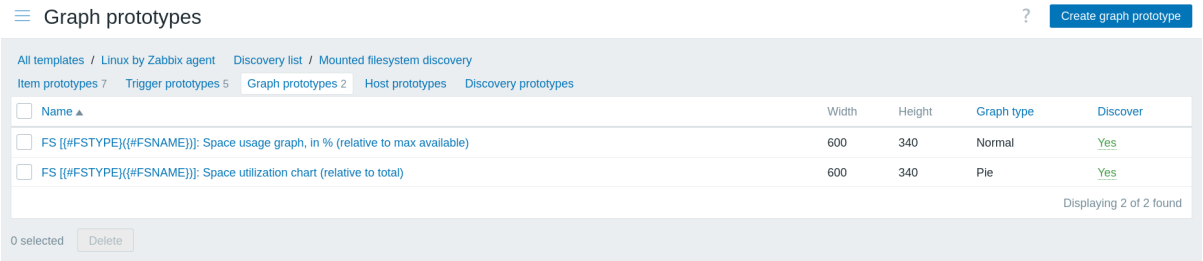
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem trigeru prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

3 Grafiku prototipi

Pārskats

Šajā sadaļā tiek attēloti veidnē konfigurētie zema līmeņa atklāšanas kārtulas grafiku prototipi.

Ja veidne ir piesaistīta hostam, grafiku prototipi kļūs par pamatu reālu hosta grafiku izveidei zema līmeņa atklāšanas laikā.



Attēlotie dati:

Kolonna	Apraksts
Name	Grafika prototipa nosaukums, kas tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta grafika prototipa konfigurācijas forma. Ja grafika prototips pieder piesaistītai veidnei, pirms grafika nosaukuma kā pelēka saite tiek attēlots veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts grafiku prototipu saraksts piesaistītās veidnes līmenī.
Width	Tiek attēlots grafika prototipa platums.
Height	Tiek attēlots grafika prototipa augstums.
Type	Tiek attēlots grafika prototipa tips — *Normal*, *Stacked*, *Pie* vai *Exploded*.
Discover	Atklāt grafiku, pamatojoties uz šo prototipu: **Yes** — atklāt **No** — neatklāt. Varat pārslēgties starp “Yes” un “No”, noklikšķinot uz tiem.

Lai konfigurētu jaunu grafika prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Create graph prototype*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Dzēst* - dzēst šos grafiku prototipus

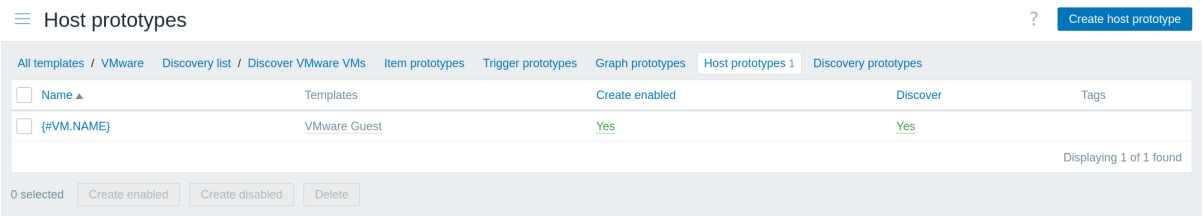
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafiku prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

4 Hostu prototipi

Pārskats

Šajā sadaļā tiek attēloti veidnē konfigurētie zema līmeņa atklāšanas kārtulas hostu prototipi.

Ja veidne ir piesaistīta hostam, hostu prototipi kļūs par pamatu reālu hostu izveidei zema līmeņa atklāšanas laikā.



Attēlotie dati:

Column	Description
Name	Hosta prototipa nosaukums, kas tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta hosta prototipa konfigurācijas forma. Ja hosta prototips pieder piesaistītai veidnei, pirms hosta nosaukuma kā pelēka saite tiek attēlots veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts hosta prototipu saraksts piesaistītās veidnes līmenī.
Templates	Tiek attēlotas hosta prototipa veidnes.

Column	Description
Create enabled	Izveidot hostu, pamatojoties uz šo prototipu, šādā statusā: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Discover	Atklāt hostu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Tags	Tiek attēlotas hosta prototipa birkas.

Lai konfigurētu jaunu hosta prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Create host prototype*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotus* - izveidot šos hostus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos hostus kā *Atspējotus*
- *Dzēst* - dzēst šos hostu prototipus

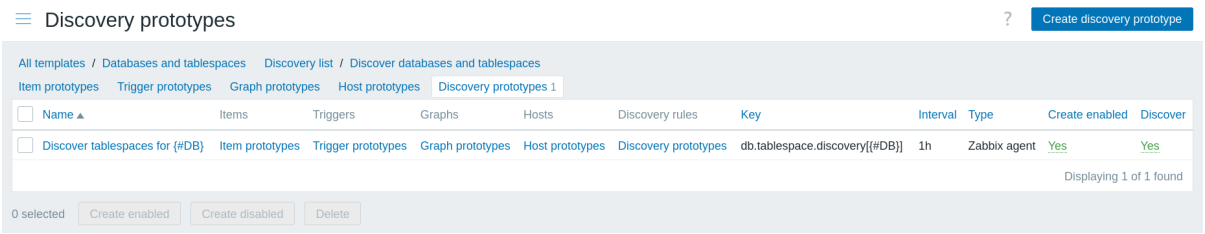
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem hostu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

5 Atklāšanas prototipi

Pārskats

Šajā sadaļā tiek attēloti veidnē konfigurētie zema līmeņa atklāšanas kārtulas atklāšanas prototipi.

Ja veidne ir piesaistīta hostam, atklāšanas prototipi kļūs par pamatu, lai izveidotu vecākās atklāšanas kārtulas reālos atklāšanas prototipus.



Attēlotie dati:

Column	Description
Name	Atklāšanas prototipa nosaukums tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta atklāšanas prototipa konfigurācijas forma. Ja atklāšanas prototips pieder piesaistītai veidnei, pirms atklāšanas prototipa nosaukuma kā pelēka saite tiek attēlots veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts atklāšanas prototipu saraksts piesaistītās veidnes līmenī.
Items	Tiek attēlota saite uz vienumu prototipu sarakstu. Esošo vienumu prototipu skaits tiek attēlots pelēkā krāsā.
Triggers	Tiek attēlota saite uz trigeru prototipu sarakstu. Esošo trigeru prototipu skaits tiek attēlots pelēkā krāsā.
Graphs	Tiek attēlota saite uz grafiku prototipu sarakstu. Esošo grafiku prototipu skaits tiek attēlots pelēkā krāsā.
Hosts	Tiek attēlota saite uz hostu prototipu sarakstu. Esošo hostu prototipu skaits tiek attēlots pelēkā krāsā.
Discovery rules	Tiek attēlota saite uz atklāšanas prototipu sarakstu. Esošo atklāšanas prototipu skaits tiek attēlots pelēkā krāsā.
Key	Tiek attēlota atklāšanai izmantotā vienuma atslēga.
Interval	Tiek attēlots atklāšanas izpildes biežums.
Type	Tiek attēlots atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u.c.).
Create enabled	Tiek attēlots atklāšanas prototipa izveides statuss — izveide iespējota (*Yes*) vai izveide atspējota (*No*). Noklikšķinot uz statusa, to var mainīt — no Yes uz No (un atpakaļ).

Column	Description
Discover	Tiek attēlots atklāšanas prototipa atklāšanas statuss — atklāt (*Yes*) vai neatklāt (*No*). Noklikšķinot uz statusa, to var mainīt — no Yes uz No (un atpakaļ).

Lai konfigurētu jaunu atklāšanas prototipu, augšējā labajā stūrī noklikšķiniet uz pogas *Create discovery prototype*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotus* - izveidot šos atklāšanas prototipus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos atklāšanas prototipus kā *Atspējotus*
- *Dzēst* - dzēst šos atklāšanas prototipus

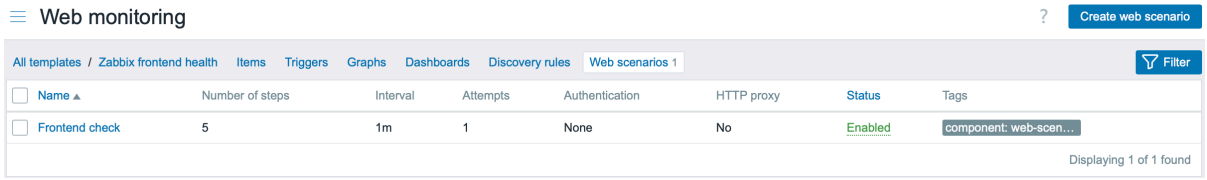
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

5 Tīmekļa scenāriji

Pārskats

Veidnes tīmekļa scenāriju sarakstam var piekļūt sadaļā *Datu ievākšana → Veidnes*, noklikšķinot uz *Tīmeklis* attiecīgajai veidnei.

Tiek parādīts esošo tīmekļa scenāriju saraksts.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Tīmekļa scenārija nosaukums. Noklikšķinot uz tīmekļa scenārija nosaukuma, tiek atvērta tīmekļa scenārija konfigurācijas forma. Ja tīmekļa scenārijs ir mantots no citas veidnes, pirms tīmekļa scenārija nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts tīmekļa scenāriju saraksts šīs veidnes līmenī.
Soļu skaits	Scenārijā iekļauto soļu skaits.
Atjaunināšanas intervāls	Cik bieži scenārijs tiek izpildīts.
Mēģinājumi	Cik mēģinājumu tiek veikts, izpildot tīmekļa scenārija soļus.
Autentifikācija	Tiek parādīta autentifikācijas metode — Basic, NTLM, Kerberos, Digest vai None.
HTTP starpniekserveris	Parāda HTTP starpniekserveri vai 'Nē', ja tas netiek izmantots.
Statuss	Tiek parādīts tīmekļa scenārija statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Birkas	Tiek parādītas tīmekļa scenārija birkas. Var tikt parādītas līdz trim birkām (nosaukums:vērtība pāri). Ja birku ir vairāk, tiek parādīta saite "...", kas, novietojot virs tās peles kursoru, ļauj apskatīt visas birkas.

Lai konfigurētu jaunu tīmekļa scenāriju, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot tīmekļa scenāriju*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

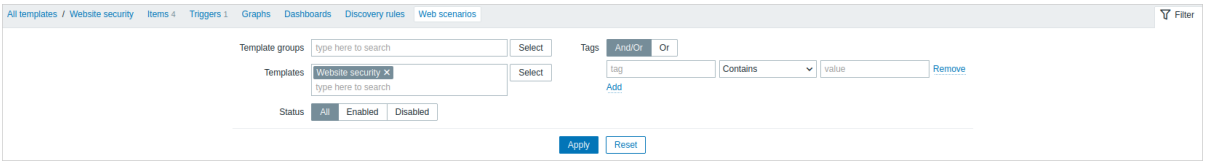
- *Iespējot* - mainīt scenārija statusu uz *Iespējots*
- *Atspējot* - mainīt scenārija statusu uz *Atspējots*
- *Dzēst* - dzēst tīmekļa scenārijus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem tīmekļa scenārijiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos scenārijus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs tīmekļa scenāriju saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā scenārijus var filtrēt pēc veidņu grupas, veidnes, statusa un birkām.

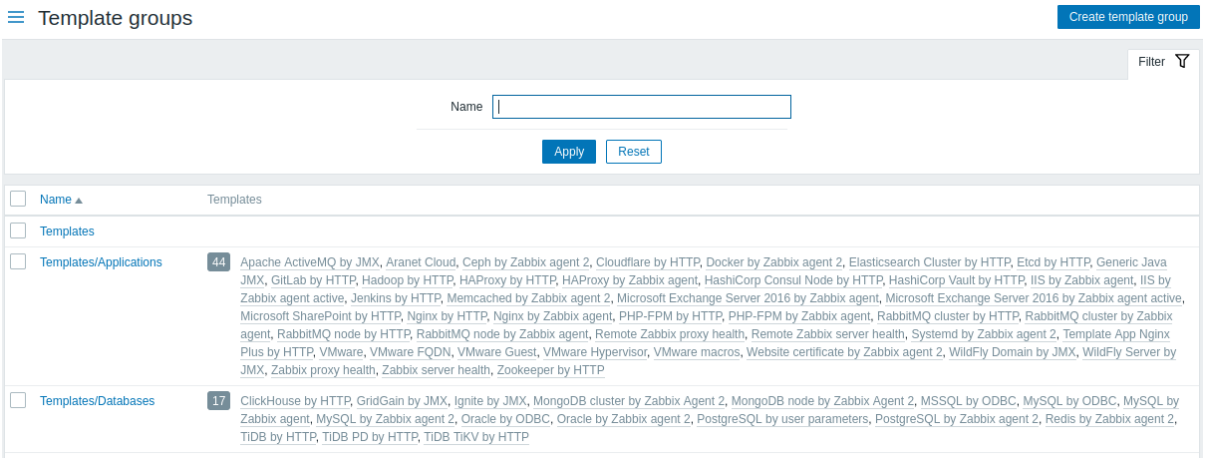


1 Veidņu grupas

Pārskats

Sadaļā *Datu ievākšana* → *Veidņu grupas* lietotāji var konfigurēt un uzturēt veidņu grupas.

Tiek parādīts esošo veidņu grupu saraksts ar to detalizēto informāciju. Jūs varat meklēt un filtrēt veidņu grupas pēc nosaukuma.



Parādītie dati:

Column	Description
Name	Veidņu grupas nosaukums. Noklikšķinot uz grupas nosaukuma, tiek atvērta grupas konfigurācijas forma.
Templates	Veidņu skaits grupā (parādīts pelēkā krāsā), kam seko grupas dalībnieku saraksts. Noklikšķinot uz veidnes nosaukuma, tiks atvērta veidnes konfigurācijas forma. Noklikšķinot uz skaita, tiek atvērts šīs grupas veidņu saraksts.

Masveida rediģēšanas opcijas

Lai vienlaikus dzēstu vairākas veidņu grupas, atzīmējiet izvēles rūtiņas pie attiecīgajām grupām un pēc tam noklikšķiniet uz pogas Dzēst zem saraksta.

Filtra izmantošana

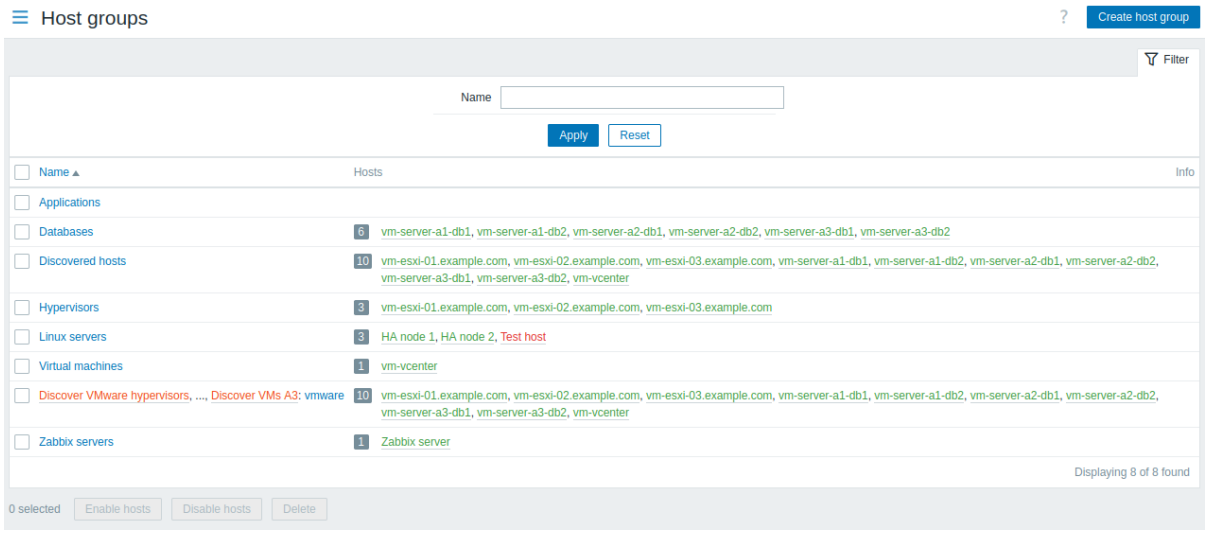
Varat izmantot filtru, lai parādītu tikai tās veidņu grupas, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

2 Hostu grupas

Pārskats

Sadaļā *Datu ievākšana* → *Hostu grupas* lietotāji var konfigurēt un uzturēt hostu grupas.

Tiek parādīts esošo hostu grupu saraksts ar detalizētu informāciju. Jūs varat meklēt un filtrēt hostu grupas pēc nosaukuma.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Hostu grupas nosaukums. Noklikšķinot uz grupas nosaukuma, tiek atvērta grupas konfigurācijas forma. Atklātās hostu grupas tiek parādītas ar zema līmeņa atklāšanas kārtulas nosaukumu kā prefiksu. Noklikšķinot uz LLD kārtulas nosaukuma, tiek atvērta hosta prototipa konfigurācijas forma. Ņemiet vērā, ka atklātās hostu grupas tiek dzēstas, kad tiek dzēstas visas LLD kārtulas, kas tās atklāja.
Hosti	Hostu skaits grupā (parādīts pelēkā krāsā), kam seko grupas dalībnieku saraksts. Noklikšķinot uz hosta nosaukuma, tiks atvērta hosta konfigurācijas forma. Noklikšķinot uz skaita, visā hostu sarakstā tiks atlasīti tie hosti, kas pieder šai grupai.
Informācija	Tiek parādīta kļūdas informācija (ja tāda ir) par hostu grupu.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Iespējot hostus* - mainīt visu grupas hostu statusu uz ”Uzraudzīts”
- *Atspējot hostus* - mainīt visu grupas hostu statusu uz ”Neuzraudzīts”
- *Dzēst* - dzēst hostu grupas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām hostu grupām, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

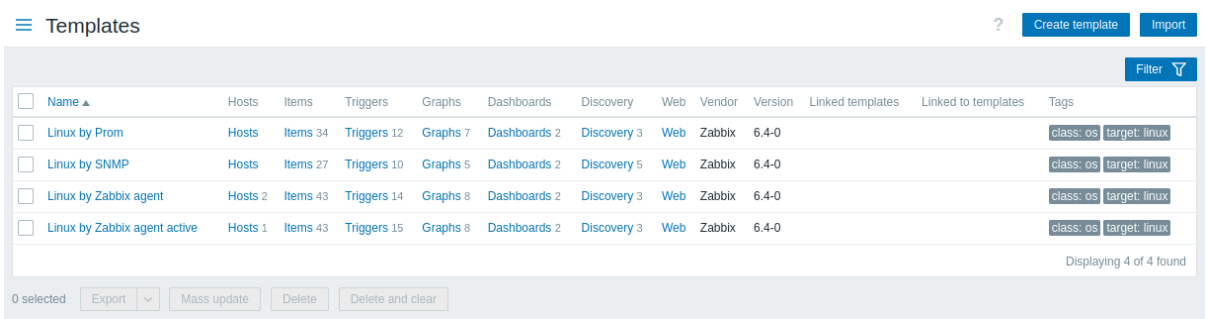
Varat izmantot filtru, lai parādītu tikai tās hostu grupas, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makrokomandām.

3 Veidnes

Pārskats

Sadaļā *Datu vākšana* → *Veidnes* lietotāji var konfigurēt un uzturēt veidnes.

Tiek parādīts esošo veidņu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Veidnes nosaukums. Klikšķinot uz veidnes nosaukuma, tiek atvērta veidnes konfigurācijas forma.
Hosti	Rediģējamo hostu skaits, ar kuriem veidne ir saistīta; tikai lasāmi hosti netiek iekļauti. Klikšķinot uz *Hosti*, tiks atvērts hostu saraksts, filtrējot tikai tos hostus, kas ir saistīti ar veidni.
Entītijas (Vienumi, Trigeri, Grafiki, Paneļi, Atklāšana, Tīmeklis)	Attiecīgo entītiju skaits veidnē (parādīts pelēkā krāsā). Klikšķinot uz entītijas nosaukuma, visā šīs entītijas sarakstā tiks atlasītas tās, kas pieder veidnei.
Saistītās veidnes	Veidnes, kas ir saistītas ar veidni.
Saistīta ar veidnēm	Veidnes, ar kurām veidne ir saistīta.
Piegādātājs, *Versija*	Veidnes piegādātājs un versija; tiek parādīti, ja veidnes konfigurācija satur šādu informāciju, un tikai standarta veidnēm, importētām veidnēm vai veidnēm, kas modificētas, izmantojot Template API. Standarta veidnēm versija tiek parādīta šādi: Zabbix galvenā versija, atdalītājs (”-”), revīzijas numurs (palielinās ar katru jaunu veidnes versiju un tiek atiestatīts ar katru jaunu Zabbix galveno versiju). Piemēram, 7.0-0, 7.0-3, 8.0-0, 8.0-3.
Birkas	Veidnes birkas, ar neatrisinātām makro vērtībām.

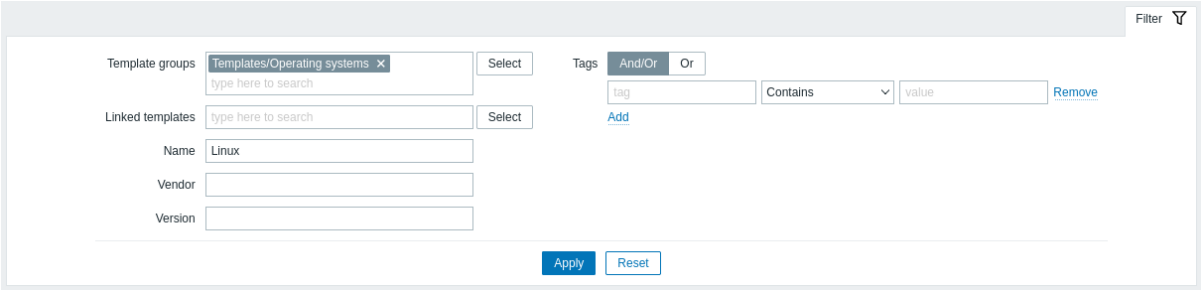
Lai konfigurētu jaunu veidni, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot veidni*.

Lai importētu veidni no YAML, XML vai JSON faila, augšējā labajā stūrī noklikšķiniet uz pogas *Importēt*.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tās veidnes, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Saite *Filter* ir pieejama zem pogām *Create template* un *Import*. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat atlasīt veidnes pēc veidņu grupas, saistītajām veidnēm, nosaukuma un birkām.



Parametrs	Apraksts
Template groups	Filtrēt pēc vienas vai vairākām veidņu grupām. Norādot vecākgrupas veidņu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas.
Linked templates	Filtrēt pēc tieši saistītām veidnēm.
Name	Filtrēt pēc veidnes nosaukuma.
Vendor	Filtrēt pēc veidnes piegādātāja.
Version	Filtrēt pēc veidnes versijas.

Parametrs	Apraksts
Tags	Filtrēt pēc veidnes birkas nosaukuma un vērtības. Filtrēšana ir iespējama tikai pēc veidnes līmeņa birkām (nevis mantotajām). Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami vairāki operatori: **Exists** - iekļaut norādītos birku nosaukumus; **Equals** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Contains** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Does not exist** - izslēgt norādītos birku nosaukumus; **Does not equal** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Does not contain** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **And/Or** - jāizpildās visiem nosacījumiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Or; **Or** - pietiek, ja izpildās viens nosacījums.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Eksportēt* — eksportēt veidni YAML, XML vai JSON failā;
- *Masveida atjaunināšana* — atjaunināt vairākus rekvizītus vairākām veidnēm vienlaikus;
- *Dzēst* — dzēst veidni, atstājot ar to saistītās entītijas (vienumus, trigerus utt.) hostos;
- *Dzēst un notīrīt* — dzēst veidni un tās saistītās entītijas no hostiem.

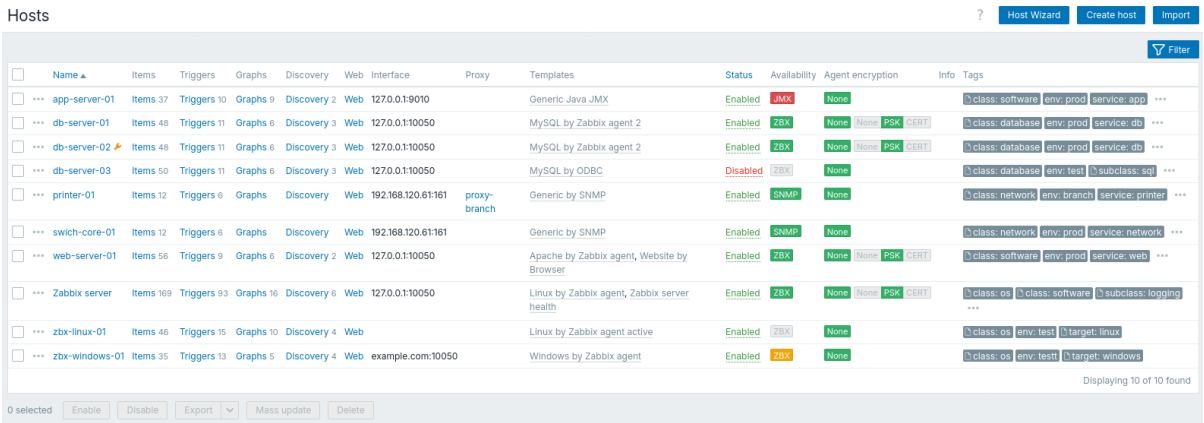
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām veidnēm un pēc tam noklikšķiniet uz vajadzīgās pogas.

4 Hosti

Pārskats

Sadaļā *Datu vākšana* → *Hosti* lietotāji var konfigurēt un uzturēt hostus.

Tiek parādīts esošo hostu saraksts ar to detalizēto informāciju.



Parādītie dati:

Kolonna	Apraksts
Hosta izvēlne	Noklikšķiniet uz trīs punktu ikonas, lai atvērtu hosta izvēlni.
Nosaukums	Hosta nosaukums. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma. Oranža uzgriežņu atslēgas ikona aiz hosta nosaukuma norāda, ka šim hostam ir iestatīta uzturēšana. Uzturēšanas informācija tiek parādīta, novietojot peles kursoru uz ikonas.

Kolonna	Apraksts
Entītijas (Vienumi, Trigeri, Grafiki, Atklāšana, Tīmeklis)	Noklikšķinot uz entītijas nosaukuma, tiks parādīti hosta vienumi, trigeri utt. Attiecīgo entītiju skaits tiek parādīts pelēkā krāsā.
Saskarne	Tiek parādīta hosta galvenā saskarne.
Starpniekserveris	Šajā kolonnā tiek parādīti piešķirtie starpniekserveri: <Proxy name> - hosts tiek uzraudzīts ar atsevišķu starpniekserveri (pat ja starpniekserveris ir starpniekserveru grupas daļa); <Proxy group name: proxy name> - hosts tiek uzraudzīts ar starpniekserveru grupu, un Zabbix serveris ir piešķīris starpniekserveri hosta uzraudzībai; <Proxy group name> - hosts tiek uzraudzīts ar starpniekserveru grupu bez starpniekserveriem vai arī, ja Zabbix serveris nav piešķīris starpniekserveri hosta uzraudzībai; Nothing - hosts netiek uzraudzīts ne ar starpniekserveri, ne ar starpniekserveru grupu. Šī kolonna tiek parādīta tikai tad, ja filtra opcija *Monitored by* ir iestatīta uz 'Any', 'Proxy' vai 'Proxy group'.
Veidnes	Tiek parādītas ar hostu saistītās veidnes. Ja saistītajā veidnē ir iekļautas citas veidnes, tās tiek parādītas iekavās, atdalītas ar komatu. Noklikšķinot uz veidnes nosaukuma, tiks atvērta tās konfigurācijas forma.
Statuss	Tiek parādīts hosta statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt manuāli. Atklātie hosti, kas ir zaudēti, tiek atzīmēti ar informācijas ikonu. Uznirstošajā paskaidrojumā ir sniegta detalizēta informācija par to statusu.
Pieejamība	Tiek parādīta hosta pieejamība katrai konfigurētajai saskarnei. Pieejamības ikonas attēlo hosta saskarnes pašreizējo statusu Zabbix serverī. Tāpēc, ja lietotāja saskarnē atspējojat hostu, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas. Līdzīgi, ja iespējojat hostu, tā pieejamība tiks atjaunināta pēc tam, kad Zabbix serveris būs sinhronizējis konfigurācijas izmaiņas un aptaujājis hostu. Pieejamības ikonas attēlo tikai tos saskarņu tipus (Aģents, SNMP, IPMI, JMX), kas ir konfigurēti. Novietojot kursoru virs ikonas, tiek parādīts uznirstošs logs ar visu tā paša tipa saskarņu sarakstu, detalizētu informāciju, statusu un kļūdām. Aģenta saskarnei uznirstošajā logā tiek parādītas saskarnes (pasīvās) un aktīvās pārbaudes. Ja hostam ir tikai aktīvās pārbaudes, Aģenta saskarnes ikona tiek parādīta pat tad, ja hostam nav konfigurēta aģenta saskarne. Kolonna ir tukša hostiem bez saskarnēm. Vienas hosta saskarnes statusu nosaka savienojums starp iespējotu vienumu, kas izmanto šo saskarni, un hostu. Statuss var būt: **Pieejams** - savienojums ar hostu bija veiksmīgs; **Nav pieejams** - savienojums ar hostu bija neveiksmīgs (taimauts, ugunsmūra problēmas utt.); **Nezināms** - savienojums ar hostu nav mēģināts vai rezultāts nav zināms. Papildu informāciju par to, kā Zabbix serveris nosaka saskarnes statusu, skatiet Nezināms saskarnes statuss un Nesasniedzama/nepieejama hosta saskarnes iestatījumi. Visu viena tipa hosta saskarņu (Aģents, SNMP, IPMI, JMX) statusu nosaka tās saskarnes, kuras izmanto vismaz viens iespējots vienums. Statusu norāda ikonas krāsa: **Zaļa** - visas saskarnes ir pieejamas; **Dzeltena** - vismaz viena saskarne nav pieejama, un vismaz viena ir pieejama vai nezināma; **Sarkana** - visas saskarnes nav pieejamas; **Pelēka** - vismaz viena saskarne ir nezināma, bet neviena nav nepieejama. **Aktīvo pārbaužu pieejamība.** Ja hostā ir iespējota vismaz viena aktīvā pārbaude, aktīvās pārbaudes arī ietekmē kopējo Aģenta saskarnes pieejamību, kā aprakstīts iepriekš. Lai noteiktu aktīvo pārbaužu pieejamību, aģenta aktīvo pārbaužu pavedienā tiek sūtīti sirdspukstu ziņojumi. Sirdspukstu ziņojumu biežumu kontrolē parametrs `HeartbeatFrequency` Zabbix aģenta vai aģenta 2 konfigurācijā (noklusējums 60 sekundes, diapazons 0-3600). Aktīvās pārbaudes tiek uzskatītas par nepieejamām, ja aktīvās pārbaudes sirdspuksts ir vecāks par 2 x HeartbeatFrequency sekundēm. **Piezīme:** Zabbix aģenti, kas ir vecāki par versiju 6.2.x, nesūta aktīvo pārbaužu sirdspukstus, tāpēc to hostu pieejamība paliek nezināma.

Kolonna	Apraksts
Aģenta šifrēšana	Tiek parādīts savienojumu uz hostu un no hosta šifrēšanas statuss: **Nav** - nav šifrēšanas; **PSK** - tiek izmantota iepriekš koplietota atslēga; **Cert** - tiek izmantots sertifikāts.
Informācija	Tiek parādīta kļūdu informācija (ja tāda ir) par hostu.
Birkas	Hosta birkas, kurām makrosi nav atrisināti.

Lai konfigurētu jaunu hostu, augšējā labajā stūrī noklikšķiniet uz pogas *Host Wizard* vai *Create host*. Lai importētu hostu no YAML, XML vai JSON faila, augšējā labajā stūrī noklikšķiniet uz pogas *Import*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

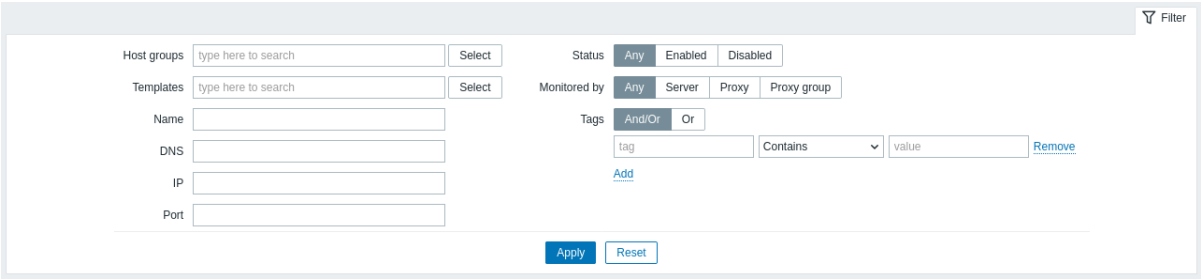
- *Iespējot* - mainīt hosta statusu uz *Uzraudzīts*;
- *Atspējot* - mainīt hosta statusu uz *Neuzraudzīts*;
- *Eksportēt* - eksportēt hostus YAML, XML vai JSON failā;
- *Masveida atjaunināšana* - atjaunināt vairākus rekvizītus vairākiem hostiem vienlaikus;
- *Dzēst* - dzēst hostus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem hostiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos hostus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makro vērtībām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



Parametrs	Apraksts
Hostu grupas	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas.
Veidnes	Filtrēt pēc piesaistītajām veidnēm.
Nosaukums	Filtrēt pēc redzamā hosta nosaukuma.
DNS	Filtrēt pēc DNS nosaukuma.
IP	Filtrēt pēc IP adreses.
Ports	Filtrēt pēc porta numura.
Statuss	Filtrēt pēc hosta statusa.
Uzrauga	Filtrēt hostus, kurus uzrauga Zabbix serveris, starpniekserveris vai starpniekserveru grupa.
Starpniekserveri	Filtrēt hostus, kurus uzrauga šeit norādītie starpniekserveri. Šis lauks ir pieejams tikai tad, ja laukā *Uzrauga* ir atlasīts ”Starpniekserveris”.
Starpniekserveru grupas	Filtrēt hostus, kurus uzrauga šeit norādītās starpniekserveru grupas. Šis lauks ir pieejams tikai tad, ja laukā *Uzrauga* ir atlasīta ”Starpniekserveru grupa”.

Parametrs	Apraksts
Birkas	Filtrēt pēc hosta birkas nosaukuma un vērtības. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma pārbaudei ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus; **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** - izslēgt norādītos birku nosaukumus; **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **Un/Vai** - jāizpildās visiem nosacījumiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** - pietiek, ja izpildās viens nosacījums.

Nezināms saskarnes statuss

Zabbix serveris attēlo statusu ”Nezināms” hosta saskarnei (aģents, SNMP, IPMI, JMX) šādos gadījumos:

- Hosts ir atspējots.
- Hostam ir iestatīta uzraudzība ar starpniekserveri, citu starpniekserveri vai serveri, ja iepriekš to uzraudzīja starpniekserveris.
- Hostu uzrauga starpniekserveris, kas šķiet bezsaistē (no starpniekservera nav saņemti atjauninājumi maksimālā sirdspukstu intervāla laikā — 1 stunda).
- Visi hosta vienumi ar šo saskarnes tipu ir atspējoti.
- Šim saskarnes tipam nav konfigurēts neviens aptaujātājs (piemēram, servera konfigurācijas parametrs `StartAgentPollers` vai `StartSNMPPollers` ir iestatīts uz 0).

Saskarnes pieejamība tiek iestatīta uz ”Nezināms” pēc Zabbix servera konfigurācijas kešatmiņas sinhronizācijas.

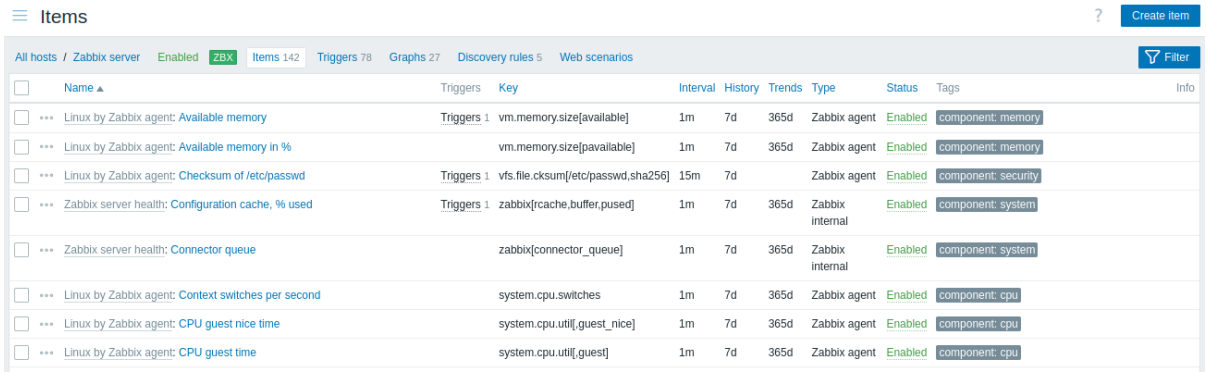
Saskarņu pieejamība (Pieejama/Nav pieejama) hostiem, kurus uzrauga starpniekserveri, tiek atjaunota pēc starpniekservera konfigurācijas kešatmiņas sinhronizācijas.

1 Vienumi

Pārskats

Hosta vienumu sarakstam var piekļūt sadaļā *Datu ievākšana* → *Hosti*, noklikšķinot uz *Vienumi* attiecīgajam hostam.

Tiek parādīts esošo vienumu saraksts.



Parādītie dati:

Kolonna	Apraksts
Vienuma izvēlne	Noklikšķiniet uz trīs punktu ikonas, lai atvērtu vienuma izvēlni.

Kolonna	Apraksts
Hosts	Vienuma hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma. Šī kolonna tiek parādīta tikai tad, ja filtrā ir atlasīti vairāki hosti vai nav atlasīts neviens hosts.
Nosaukums	Vienuma nosaukums tiek parādīts kā zila saite uz vienuma detaļām. Noklikšķinot uz vienuma nosaukuma saites, tiek atvērta vienuma konfigurācijas forma. Ja hosta vienums pieder veidnei, pirms vienuma nosaukuma tiek parādīts veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiks atvērts vienumu saraksts veidnes līmenī. Ja vienums ir izveidots no vienuma prototipa, pirms tā nosaukuma oranžā krāsā tiek parādīts zema līmeņa atklāšanas kārtulas nosaukums. Noklikšķinot uz atklāšanas kārtulas nosaukuma, tiks atvērts vienumu prototipu saraksts.
Trigeri	Pārvietojot peles kursoru virs Trigeri, tiks parādīts informācijas logs ar trigeriem, kas saistīti ar vienumu. Trigeru skaits tiek parādīts pelēkā krāsā.
Atslēga	Tiek parādīta vienuma atslēga.
Intervāls	Tiek parādīts pārbaudes biežums. *Ņemiet vērā*, ka pasīvos vienumus var pārbaudīt arī nekavējoties, nospiežot *Izpildīt tagad* pogu.
Vēsture	Tiek parādīts, cik dienas tiks glabāta vienuma datu vēsture.
Tendences	Tiek parādīts, cik dienas tiks glabāta vienuma tendenču vēsture.
Tips	Tiek parādīts vienuma tips (Zabbix aģents, SNMP aģents, vienkāršā pārbaude utt.).
Statuss	Tiek parādīts vienuma statuss — *Iespējots*, *Atspējots* vai *Nav atbalstīts*. Statusu var mainīt manuāli, noklikšķinot uz tā — no Iespējots uz Atspējots (un atpakaļ); no Nav atbalstīts uz Atspējots (un atpakaļ). Atklātie vienumi, kas ir zaudēti, ir atzīmēti ar informācijas ikonu. Uznirstošais paskaidrojums sniedz detalizētu informāciju par to statusu.
Birkas	Tiek parādītas vienuma birkas. Var tikt parādītas līdz trim birkām (nosaukums:vērtība pāri). Ja birku ir vairāk, tiek parādīta saite "...", kas ļauj redzēt visas birkas, novietojot virs tās peles kursoru.
Informācija	Ja vienums darbojas pareizi, šajā kolonnā netiek parādīta neviena ikona. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Novietojiet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu vienumu, noklikšķiniet uz pogas *Izveidot vienumu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

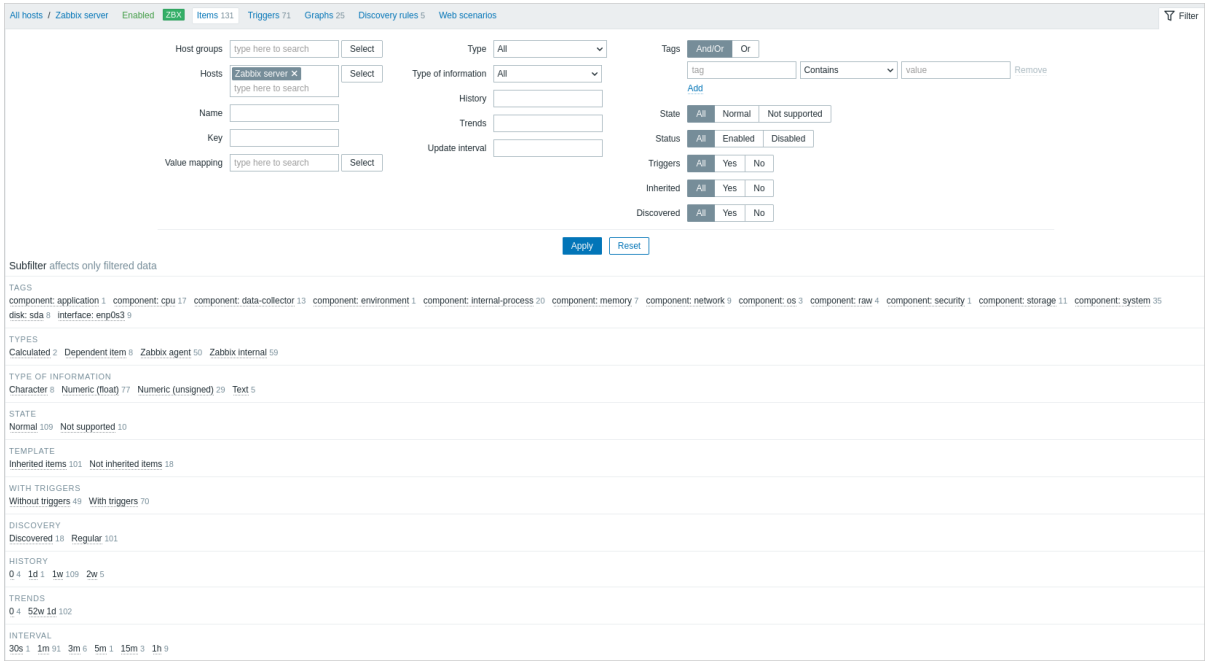
- *Iespējot* - mainīt vienuma statusu uz *Iespējots*
- *Atspējot* - mainīt vienuma statusu uz *Atspējots*
- *Izpildīt tagad* - nekavējoties izpildīt jaunu vienuma vērtību pārbaudi. Atbalstīts tikai **pasīvajām** pārbaudēm (skatiet sīkāku informāciju). Ņemiet vērā, ka, nekavējoties pārbaudot vērtības, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ vērtības neatspoguļos pavisam nesenās vienuma konfigurācijas izmaiņas.
- *Notīrīt vēsturi un tendences* - dzēst vienumu vēstures un tendenču datus.
- *Kopēt* - kopēt vienumus uz citiem hostiem vai veidnēm.
- *Masveida atjaunināšana* - vienlaikus atjaunināt vairākus rekvizītus vairākiem vienumiem.
- *Dzēst* - dzēst vienumus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos vienumus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makrokomandām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



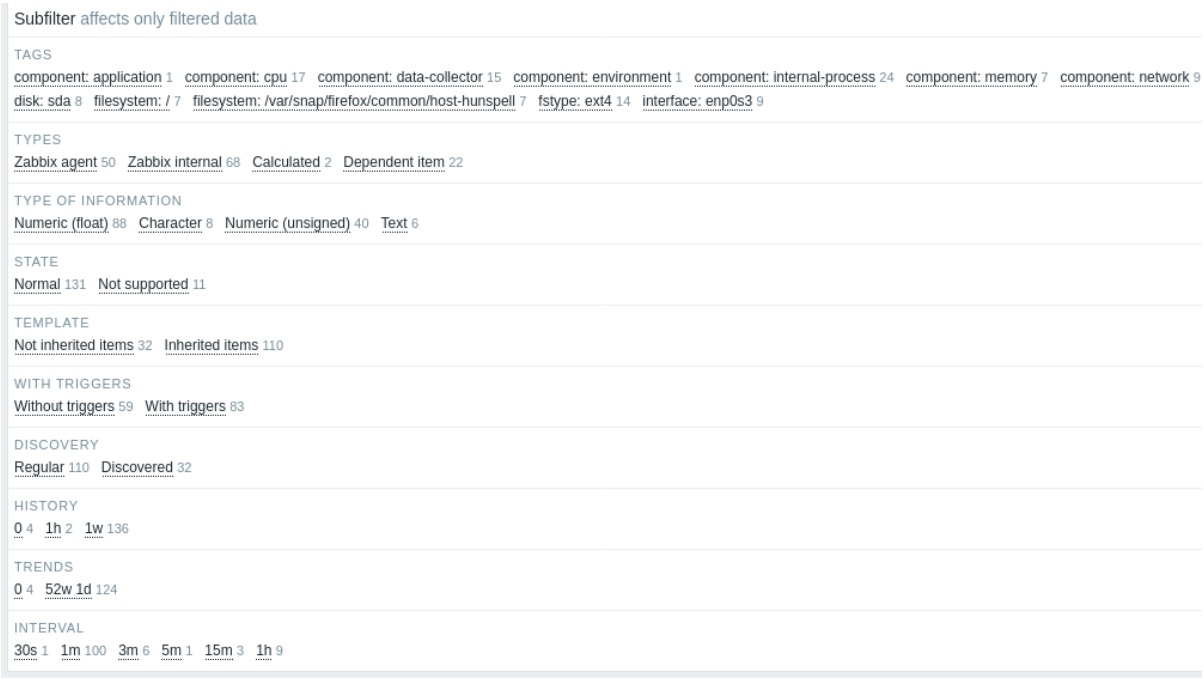
Parametrs	Apraksts
Hostu grupas	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas. Hostu grupas, kurās ir tikai veidnes, nevar atlasīt.
Hosti	Filtrēt pēc viena vai vairākiem hostiem.
Nosaukums	Filtrēt pēc vienuma nosaukuma.
Atslēga	Filtrēt pēc vienuma atslēgas.
Vērtību kartēšana	Filtrēt pēc izmantotās vērtību kartes. Šis parametrs netiek parādīts, ja opcija *Hosti* ir tukša.
Tips	Filtrēt pēc vienuma tipa (Zabbix aģents, SNMP aģents u.c.).
Informācijas tips	Filtrēt pēc informācijas tipa (bez zīmes skaitlisks, peldošā komata skaitlis u.c.).
Vēsture	Filtrēt pēc tā, cik ilgi tiek glabāta vienuma vēsture.
Tendences	Filtrēt pēc tā, cik ilgi tiek glabātas vienuma tendences.
Atjaunināšanas intervāls	Filtrēt pēc vienuma atjaunināšanas intervāla.
Birkas	Norādiet birkas, lai ierobežotu parādīto vienumu skaitu. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birku nosaukumu atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Neeksistē** — izslēgt norādītos birku nosaukumus **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina tipi: **Un/Vai** — visiem nosacījumiem jābūt izpildītiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai **Vai** — pietiek, ja ir izpildīts viens nosacījums
Stāvoklis	Filtrēt pēc vienuma stāvokļa — *Normāls* vai *Nav atbalstīts*.
Statuss	Filtrēt pēc vienuma statusa — *Iespējots* vai *Atspējots*.
Trigeri	Filtrēt vienumus ar (vai bez) trigeriem.
Mantots	Filtrēt vienumus, kas mantoti (vai nav mantoti) no veidnes.
Atklāšana	Filtrēt vienumus, ko zema līmeņa atklāšana ir atklājusi (vai nav atklājusi).

Apakšfiltra izmantošana

Apakšfiltrs ļauj papildus modificēt filtrēšanu no galvenā filtra.

Tas satur klikšķināmas saites ātrai piekļuvei saistītajiem vienumiem. Vienumi ir saistīti pēc kopīgas entītijas — birkas, vienuma tipa,

vienuma stāvokļa, vienuma atjaunināšanas intervāla utt. Noklikšķinot uz entītijas, tā tiek izcelta ar pelēku fonu, un vienumi tiek nekavējoties filtrēti (nav nepieciešams galvenajā filtrā noklikšķināt uz *Pielietot*). Noklikšķinot uz citas entītijas, tā tiek pievienota filtrētajiem rezultātiem. Atkārtoti noklikšķinot uz entītijas, filtrēšana tiek noņemta.



Apakšfiltri tiek ģenerēti, pamatojoties uz filtrētajiem datiem, kuru apjoms ir ierobežots līdz 1000 ierakstiem. Ja vēlaties apakšfiltrā redzēt vairāk ierakstu, jums jāpalielina parametra *Limit for search and filter results* vērtība (sadaļā *Administration -> General -> GUI*).

Atšķirībā no galvenā filtra, apakšfiltrs tiek atjaunināts ar katru tabulas atsvaidzināšanas pieprasījumu, lai vienmēr būtu pieejama aktuāla informācija par pieejamajām filtrēšanas opcijām un to skaitītāju vērtībām.

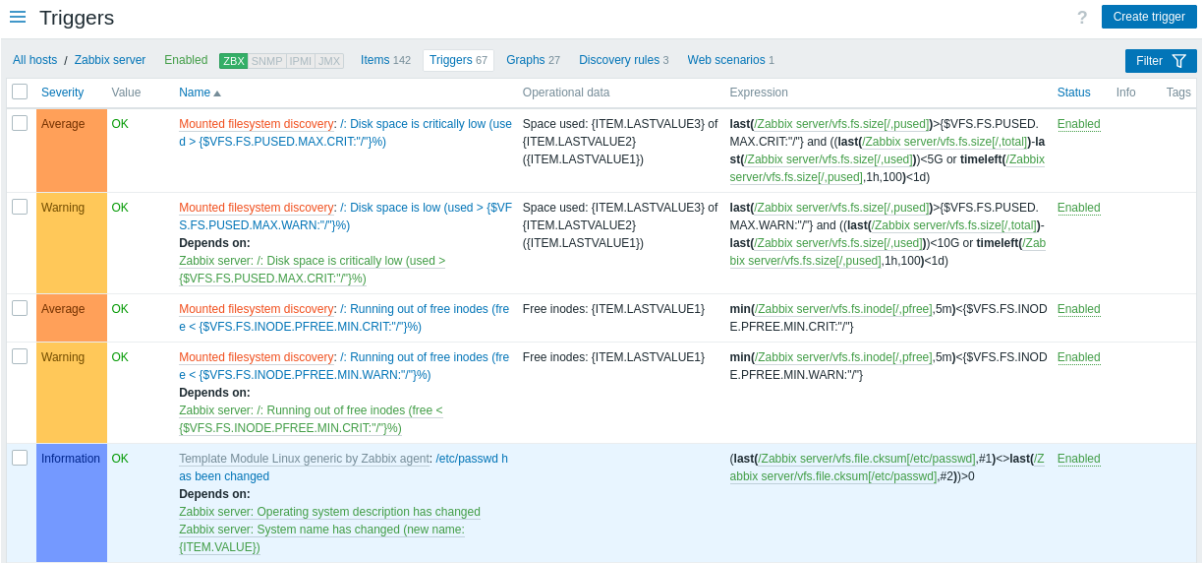
Attēloto entītiju skaits horizontāli ir ierobežots līdz 100. Ja to ir vairāk, beigās tiek parādīta trīs punktu ikona; tā nav klikšķināma.

Skaitlis blakus katrai klikšķināmajai entītijai norāda tajā grupēto vienumu skaitu (pamatojoties uz galvenā filtra rezultātiem). Noklikšķinot uz entītijas, skaitļi pie citām pieejamajām entītijām tiek parādīti ar pluszīmi, norādot, cik vienumu var pievienot pašreizējai atlasei. Entītijas bez vienumiem netiek rādītas, ja vien tās iepriekš nav atlasītas apakšfiltrā.

2 Trigeri

Pārskats

Hosta trigeru sarakstam var piekļūt sadaļā *Datu ievākšana → Hosti*, noklikšķinot uz *Trigeri* attiecīgajam hostam.



Attēlotie dati:

Kolonna	Apraksts
Svarīgums	Trigera svarīgums tiek attēlots gan ar nosaukumu, gan šūnas fona krāsu.
Vērtība	Tiek attēlots trigera stāvoklis: **OK** - OK stāvoklis **PROBLĒMA** - Problēmas stāvoklis
Hosts	Trigera hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma. Šī kolonna tiek rādīta tikai tad, ja filtrā ir atlasīti vairāki hosti vai nav atlasīts neviens hosts.
Nosaukums	Trigera nosaukums tiek attēlots kā zila saite uz trigera detalizēto informāciju. Noklikšķinot uz trigera nosaukuma saites, tiek atvērta trigera konfigurācijas forma. Ja hosta trigeris pieder veidnei, pirms trigera nosaukuma tiek attēlots veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiks atvērts trigeru saraksts veidnes līmenī. Ja trigeris ir izveidots no trigera prototipa, pirms tā nosaukuma oranžā krāsā tiek attēlots zema līmeņa atklāšanas kārtulas nosaukums. Noklikšķinot uz atklāšanas kārtulas nosaukuma, tiek atvērts trigeru prototipu saraksts.
Operacionālie dati	Trigera operacionālo datu definīcija, kas satur patvaļīgas virknes un makrosus, kuri dinamiski tiks atrisināti sadaļā *Uzraudzība* → *Problēmas*.
Izteiksme	Tiek attēlota trigera izteiksme. Izteiksmes hosta-vienuma daļa tiek attēlota kā saite, kas ved uz vienuma konfigurācijas formu.
Statuss	Tiek attēlots trigera statuss — *Iespējots*, *Atspējots* vai *Nezināms*. Noklikšķinot uz statusa, to var manuāli mainīt — no Iespējots uz Atspējots (un atpakaļ); no Nezināms uz Atspējots (un atpakaļ). Atspējota trigera problēmas vairs netiek rādītas lietotāja saskarnē, taču tās netiek dzēstas. Atklātie trigeri, kas ir zuduši, tiek atzīmēti ar informācijas ikonu. Uznirstošajā paskaidrojumā ir sniegta detalizēta informācija par to statusu.
Informācija	Ja viss darbojas pareizi, šajā kolonnā netiek attēlota neviena ikona. Kļūdu gadījumā tiek attēlota kvadrātveida ikona ar burtu ”i”. Virziet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.
Birkas	Ja trigeris satur birkas, šajā kolonnā tiek attēlots birkas nosaukums un vērtība.

Lai konfigurētu jaunu trigeri, noklikšķiniet uz pogas *Izveidot trigeri* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

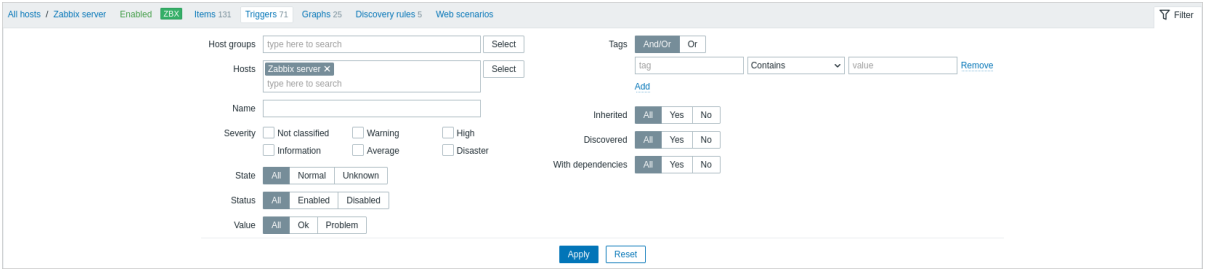
- *Iespējot* — mainīt trigera statusu uz *Iespējots*.
- *Atspējot* — mainīt trigera statusu uz *Atspējots*.
- *Kopēt* — kopēt trigerus uz citiem hostiem vai veidnēm.
- *Masveida atjaunināšana* — atjaunināt vairākus vairāku trigeru rekvizītus vienlaikus.
- *Dzēst* — dzēst trigerus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem trigeriem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tos trigerus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Ikona *Filtrs* ir pieejama augšējā labajā stūrī. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



Parametrs	Apraksts
Hostu grupas	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas. Hostu grupas, kurās ir tikai veidnes, nevar atlasīt.
Hosti	Filtrēt pēc viena vai vairākiem hostiem. Ja iepriekš augstāk jau ir atlasītas hostu grupas, hostu atlase ir ierobežota ar šīm grupām.

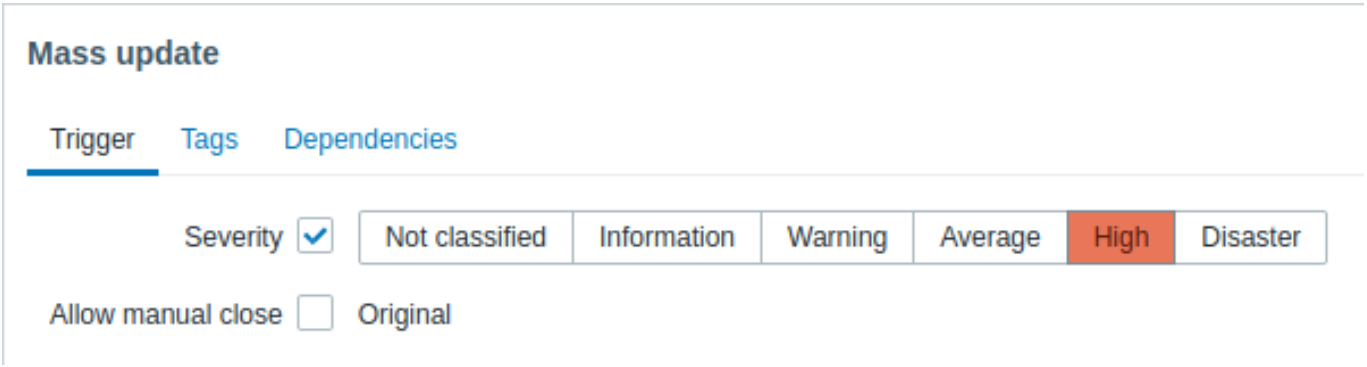
Parametrs	Apraksts
Nosaukums	Filtrēt pēc trigera nosaukuma.
Svarīgums	Atlasiet, lai filtrētu pēc viena vai vairākiem trigeru svarīguma līmeņiem.
Stāvoklis	Filtrēt pēc trigera izteiksmes stāvokļa (Normāls/Nezināms).
Statuss	Filtrēt pēc trigera aktivizācijas statusa (Iespējots/Atspējots).
Vērtība	Filtrēt pēc trigera vērtības.
Birkas	Filtrēt pēc trigera birkas nosaukuma un vērtības. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katram nosacījumam ir pieejami vairāki operatori: **Eksistē** - iekļaut norādītos birku nosaukumus **Vienāds ar** - iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Satur** - iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) **Neeksistē** - izslēgt norādītos birku nosaukumus **Nav vienāds ar** - izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi) **Nesatur** - izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga) Nosacījumiem ir divi aprēķina veidi: **Un/Vai** - visiem nosacījumiem jābūt izpildītiem, nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc Vai nosacījuma **Vai** - pietiek, ja ir izpildīts viens nosacījums Gan birkas nosaukuma, gan birkas vērtības laukos tiek atbalstītas makrokomandas un makrokomandu funkcijas.
Mantots	Filtrēt trigerus, kas mantoti (vai nav mantoti) no veidnes.
Atklāts	Filtrēt trigerus, kas atklāti (vai nav atklāti) ar zema līmeņa atklāšanu.
Ar atkarībām	Filtrēt trigerus ar (vai bez) atkarībām.

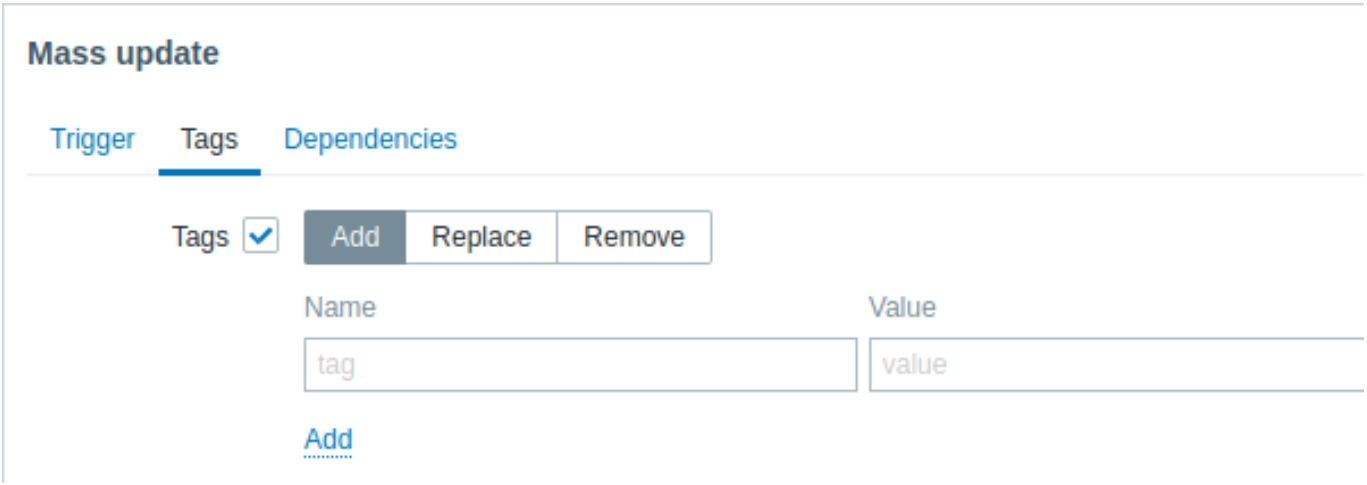
Masveida atjaunināšanas izmantošana

Ar masveida atjaunināšanu jūs varat vienlaikus mainīt kādu atribūtu vairākiem trigeriem, tādējādi ietaupot laiku, kas citādi būtu nepieciešams, lai atvērtu katru atsevišķo trigeri rediģēšanai.

Lai masveidā atjauninātu vairākus trigerus, rīkojieties šādi:

- Atzīmējiet to trigeru izvēles rūtiņas sarakstā, kurus vēlaties atjaunināt
- Noklikšķiniet uz *Mass update* zem saraksta
- Pārejiet uz cilni ar nepieciešamajiem atribūtiem (*Trigger*, *Tags* vai *Dependencies*)
- Atzīmējiet to atribūtu izvēles rūtiņas, kurus vēlaties atjaunināt

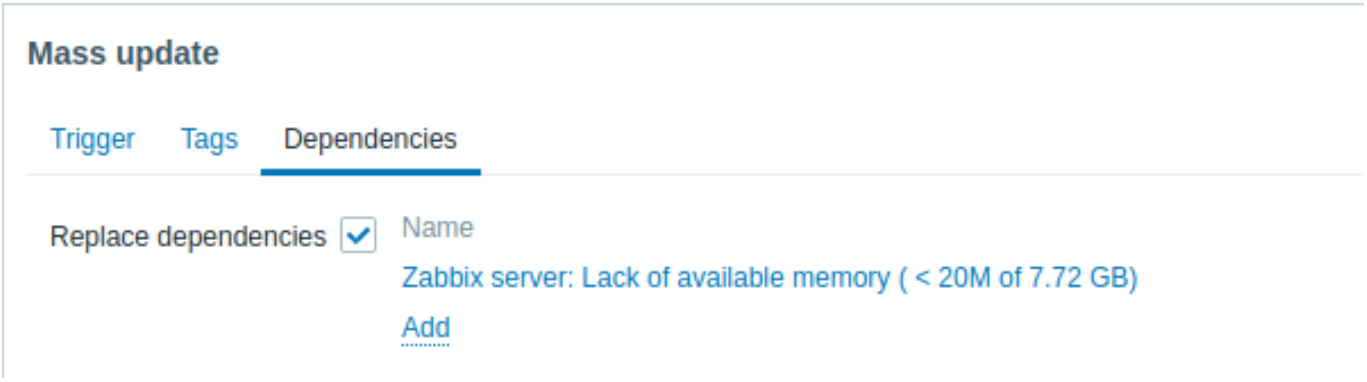




Izvēloties attiecīgo pogu birku atjaunināšanai, ir pieejamas šādas opcijas:

- *Add* - ļauj pievienot jaunas birkas trigeriem;
- *Replace* - noņems visas esošās birkas no trigeriem un aizstās tās ar zemāk norādīto(-ajām);
- *Remove* - noņems norādītās birkas no trigeriem.

Ņemiet vērā, ka birkas ar vienādu nosaukumu, bet atšķirīgām vērtībām netiek uzskatītas par "dublikātiem", un tās var pievienot vienam un tam pašam trigerim.



Replace dependencies - noņems visas esošās atkarības no trigeriem un aizstās tās ar norādīto(-ajām).

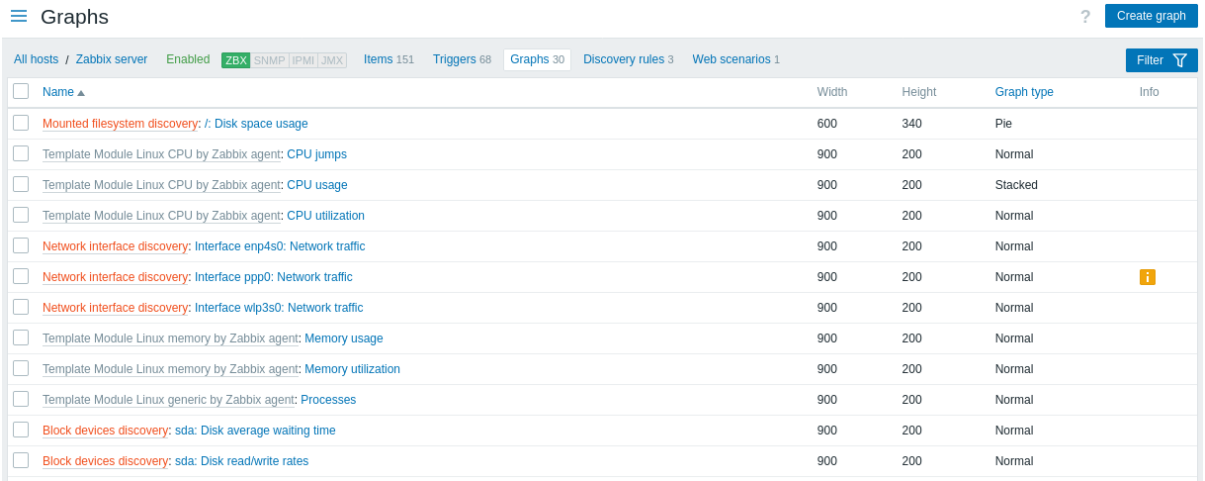
Noklikšķiniet uz *Update*, lai piemērotu izmaiņas.

3 Grafiki

Pārskats

Pielāgoto grafiku sarakstam hostam var piekļūt sadaļā *Datu ievākšana* → *Hosti*, noklikšķinot uz *Grafiki* attiecīgajam hostam.

Tiek parādīts esošo grafiku saraksts.



Parādītie dati:

Kolonna	Apraksts
Hosts	Grafa hosts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma. Šī kolonna tiek rādīta tikai tad, ja filtrā ir atlasīti vairāki hosti vai nav atlasīts neviens hosts.
Nosaukums	Pielāgotā grafa nosaukums tiek parādīts kā zila saite uz grafa detalizēto informāciju. Noklikšķinot uz grafa nosaukuma saites, tiek atvērta grafa konfigurācijas forma. Ja hosta grafiks pieder veidnei, veidnes nosaukums pirms grafa nosaukuma tiek parādīts kā pelēka saite. Noklikšķinot uz veidnes saites, tiks atvērts grafiku saraksts veidnes līmenī. Ja grafiks ir izveidots no grafa prototipa, pirms tā nosaukuma oranžā krāsā tiek parādīts zema līmeņa atklāšanas kārtulas nosaukums. Noklikšķinot uz atklāšanas kārtulas nosaukuma, tiek atvērts grafu prototipu saraksts.
Platums	Tiek parādīts grafa platums.
Augstums	Tiek parādīts grafa augstums.
Grafa tips	Tiek parādīts grafa tips — *Parasts*, *Sakrauts*, *Sektoru* vai *Izcelts*.
Informācija	Ja grafiks darbojas pareizi, šajā kolonnā netiek parādīta neviena ikona. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Virziet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu grafu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot grafu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Kopēt* — kopēt grafikus uz citiem hostiem vai veidnēm
- *Dzēst* — dzēst grafikus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafikiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

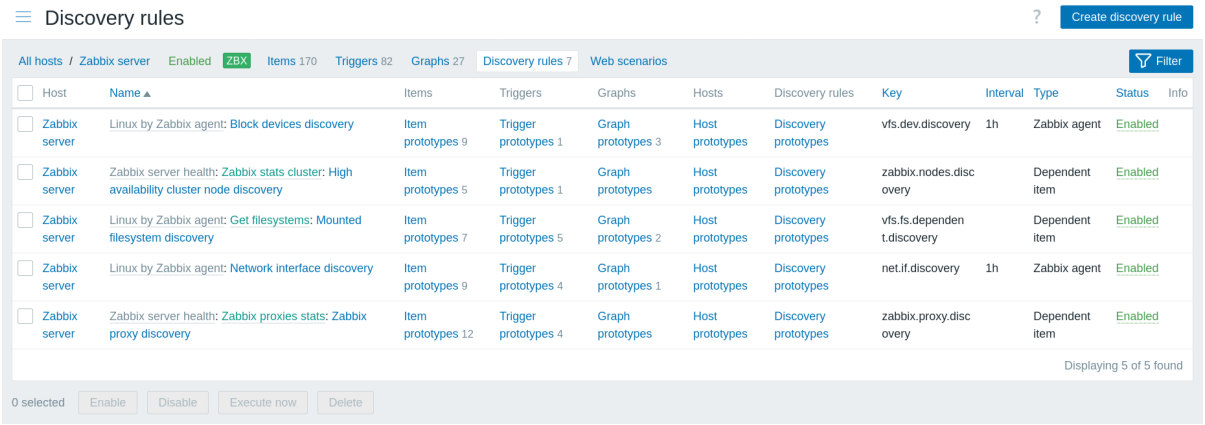
Jūs varat filtrēt grafikus pēc hostu grupas un hosta. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

4 Atklāšanas kārtulas

Pārskats

Hosts zema līmeņa atklāšanas noteikumu sarakstam var piekļūt no *Datu ievākšana* → *Hosti*, noklikšķinot uz *Atklāšana* attiecīgajam hostam.

Tiek parādīts esošo zema līmeņa atklāšanas noteikumu saraksts. Ir arī iespējams skatīt visus atklāšanas noteikumus neatkarīgi no hosta vai visus konkrētas hostu grupas atklāšanas noteikumus, mainot filtra iestatījumus.



Parādītie dati:

Column	Description
Hosts	Tiek parādīts redzamais hosta nosaukums. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma. Ja redzamais hosta nosaukums nav pieejams, tiek parādīts tehniskais hosta nosaukums.
Name	Noteikuma nosaukums tiek parādīts kā zila saite. Noklikšķinot uz noteikuma nosaukuma, tiek atvērta zema līmeņa atklāšanas noteikuma konfigurācijas forma. Ja atklāšanas noteikums pieder veidnei, pirms noteikuma nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts noteikumu saraksts veidnes līmenī.
Items	Tiek parādīta saite uz vienumu prototipu sarakstu. Esošo vienumu prototipu skaits tiek parādīts pelēkā krāsā.
Triggers	Tiek parādīta saite uz trigeru prototipu sarakstu. Esošo trigeru prototipu skaits tiek parādīts pelēkā krāsā.
Graphs	Tiek parādīta saite uz grafiku prototipu sarakstu. Esošo grafiku prototipu skaits tiek parādīts pelēkā krāsā.
Hosts	Tiek parādīta saite uz hostu prototipu sarakstu. Esošo hostu prototipu skaits tiek parādīts pelēkā krāsā.
Key	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Interval	Tiek parādīts atklāšanas izpildes biežums. *Ņemiet vērā*, ka atklāšanu var izpildīt arī nekavējoties, nospiežot pogu *Izpildīt tagad* zem saraksta.
Type	Tiek parādīts atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u.c.).
Status	Tiek parādīts atklāšanas noteikuma statuss — *Iespējots*, *Atspējots* vai *Nav atbalstīts*. Noklikšķinot uz statusa, to var mainīt — no Iespējots uz Atspējots (un atpakaļ); no Nav atbalstīts uz Atspējots (un atpakaļ).
Info	Ja viss ir kārtībā, šajā kolonnā netiek parādīta neviena ikona. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Virziet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu zema līmeņa atklāšanas noteikumu, noklikšķiniet uz pogas *Izveidot atklāšanas noteikumu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

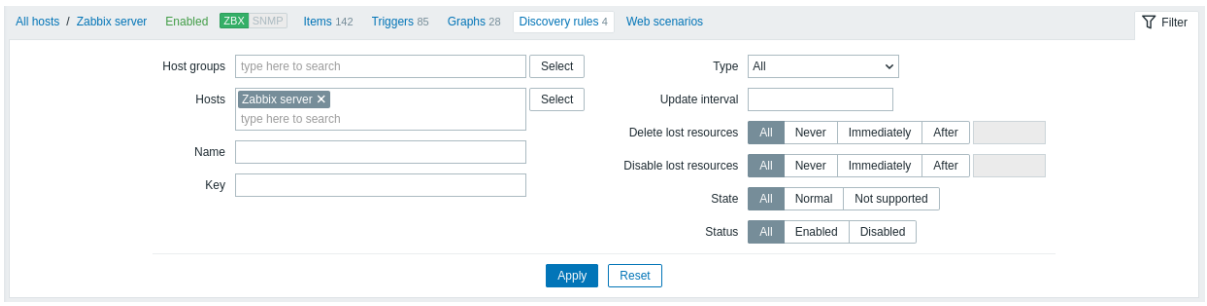
- *Iespējot* - mainīt zema līmeņa atklāšanas kārtulas statusu uz *Iespējots*.
- *Atspējot* - mainīt zema līmeņa atklāšanas kārtulas statusu uz *Atspējots*.
- *Izpildīt tagad* - nekavējoties veikt atklāšanu, pamatojoties uz atklāšanas kārtulām. Skatiet vairāk informācijas. Ņemiet vērā, ka, veicot atklāšanu nekavējoties, konfigurācijas kešatmiņa netiek atjaunināta, tādēļ rezultāts neatspoguļos pavisam nesen veiktās izmaiņas atklāšanas kārtulu konfigurācijā.
- *Dzēst* - dzēst zema līmeņa atklāšanas kārtulas.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām atklāšanas kārtulām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Jūs varat izmantot filtru, lai attēlotu tikai tos atklāšanas noteikumus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs atklāšanas noteikumu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt atklāšanas noteikumus pēc hostu grupas, hosta, nosaukuma, vienuma atslēgas, vienuma tipa un citiem parametriem.

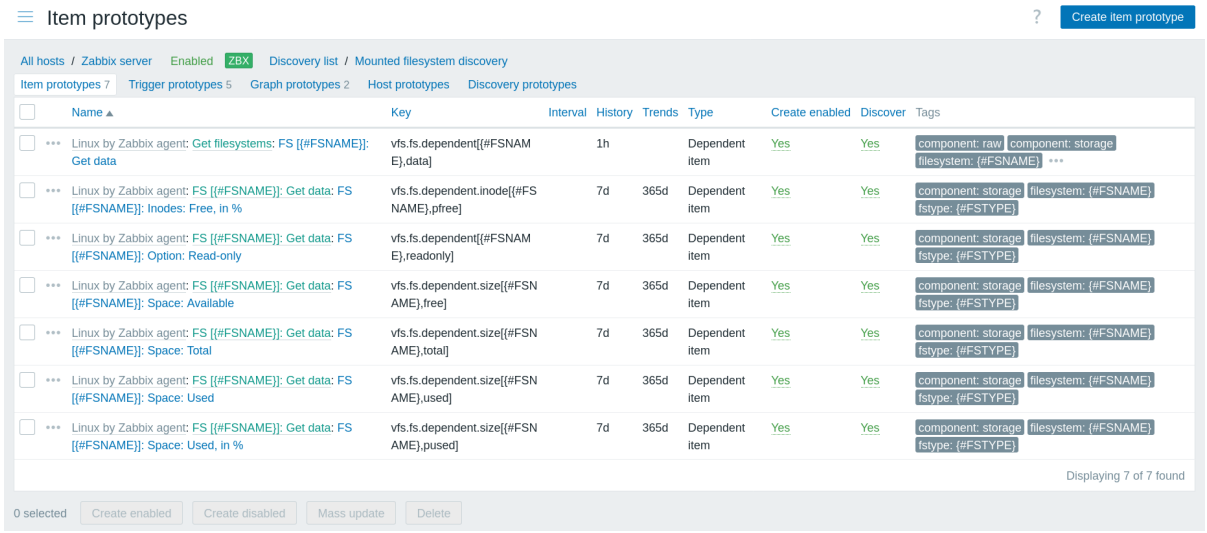


Parametrs	Apraksts
Hostu grupas	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās hostu grupas.
Hosti	Filtrēt pēc viena vai vairākiem hostiem.
Nosaukums	Filtrēt pēc atklāšanas noteikuma nosaukuma.
Atslēga	Filtrēt pēc atklāšanas vienuma atslēgas.
Tips	Filtrēt pēc atklāšanas vienuma tipa.
Atjaunināšanas intervāls	Filtrēt pēc atjaunināšanas intervāla. Nav pieejams Zabbix trapper un atkarīgajiem vienumiem.
Zaudēto resursu dzēšana	Filtrēt pēc perioda *Zaudēto resursu dzēšana*.
Zaudēto resursu atspējošana	Filtrēt pēc perioda *Zaudēto resursu atspējošana*.
SNMP OID	Filtrēt pēc SNMP OID. Pieejams tikai tad, ja kā tips ir atlasīts *SNMP aģents*.
Stāvoklis	Filtrēt pēc atklāšanas noteikuma stāvokļa (Visi/Normāls/Netiek atbalstīts).
Statuss	Filtrēt pēc atklāšanas noteikuma statusa (Visi/Iespējots/Atspējots).

1 Vienumu prototipi

Pārskats

Šajā sadaļā tiek attēloti zema līmeņa atklāšanas kārtulas vienumu prototipi hostā. Vienumu prototipi ir pamats reāliem hosta vienumiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Attēlotie dati:

Column	Description
Name	Vienuma prototipa nosaukums tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta vienuma prototipa konfigurācijas forma. Ja vienuma prototips pieder veidnei, veidnes nosaukums tiek attēlots pirms kārtulas nosaukuma kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts vienumu prototipu saraksts veidnes līmenī. Ja vienuma prototips ir tipa ”Dependent item”, pirms vienuma prototipa nosaukuma zaļā krāsā tiek attēlots galvenā vienuma nosaukums.
Key	Tiek attēlota vienuma prototipa atslēga.
Interval	Tiek attēlots pārbaudes biežums.
History	Tiek attēlots, cik dienas saglabāt vienuma datu vēsturi.
Trends	Tiek attēlots, cik dienas saglabāt vienuma tendenču vēsturi.
Type	Tiek attēlots vienuma prototipa tips (Zabbix aģents, SNMP aģents, vienkāršā pārbaude utt.).
Create enabled	Izveidot vienumu, pamatojoties uz šo prototipu, kā: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp ’Yes’ un ’No’, noklikšķinot uz tiem.

Column	Description
Discover	Atklāt vienumu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp ’Yes’ un ’No’, noklikšķinot uz tiem.
Tags	Tiek attēlotas vienuma prototipa birkas.

Lai konfigurētu jaunu vienuma prototipu, noklikšķiniet uz pogas *Create item prototype* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

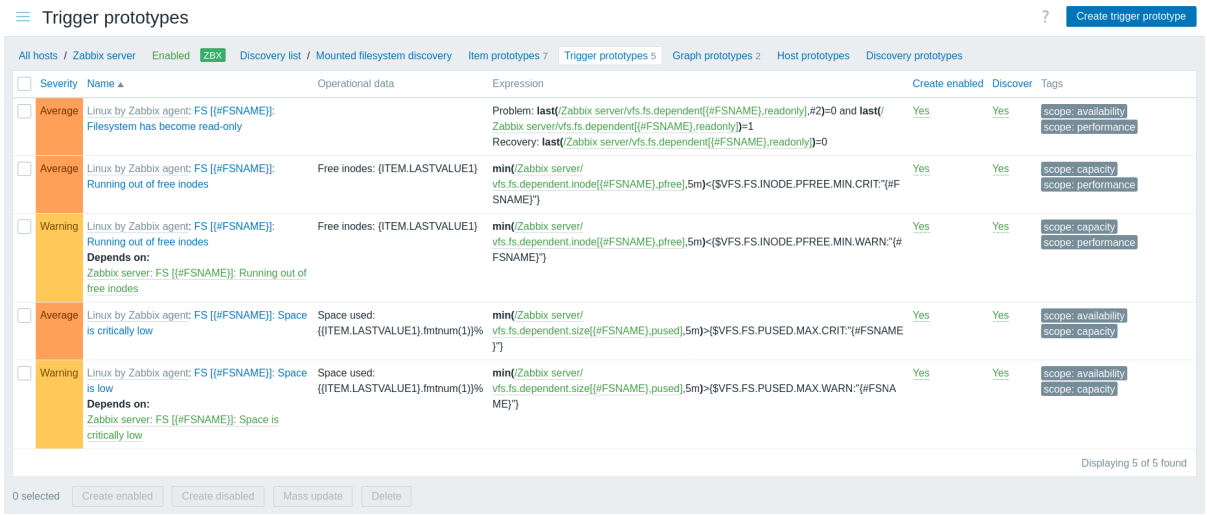
- *Izveidot iespējotus* - izveidot šos vienumus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos vienumus kā *Atspējotus*
- *Masveida atjaunināšana* - masveidā atjaunināt šos vienumu prototipus
- *Dzēst* - dzēst šos vienumu prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem vienumu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

2 Trigeru prototipi

Pārskats

Šajā sadaļā tiek attēloti hosta zema līmeņa atklāšanas kārtulas trigeru prototipi. Trigeru prototipi ir pamats reāliem hosta trigeriem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Attēlotie dati:

Kolonna	Apraksts
Nosaukums	Trigera prototipa nosaukums, kas tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta trigera prototipa konfigurācijas forma. Ja trigera prototips pieder saistītai veidnei, pirms trigera nosaukuma tiek attēlots veidnes nosaukums kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts trigeru prototipu saraksts saistītās veidnes līmenī.
Darbības dati	Tiek attēlots trigera darbības datu formāts, kas satur patvaļīgas virknes un makrosus, kuri dinamiski tiks atrisināti sadaļā *Uzraudzība → Problēmas*.
Izveidot iespējotu	Izveidot trigeri, pamatojoties uz šo prototipu, kā: **Jā** - iespējots **Nē** - atspējots. Varat pārslēgties starp ’Jā’ un ’Nē’, noklikšķinot uz tiem.
Atklāt	Atklāt trigeri, pamatojoties uz šo prototipu: **Jā** - atklāt **Nē** - neatklāt. Varat pārslēgties starp ’Jā’ un ’Nē’, noklikšķinot uz tiem.
Birkas	Tiek attēlotas trigera prototipa birkas.

Lai konfigurētu jaunu trigera prototipu, noklikšķiniet uz pogas *Izveidot trigera prototipu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

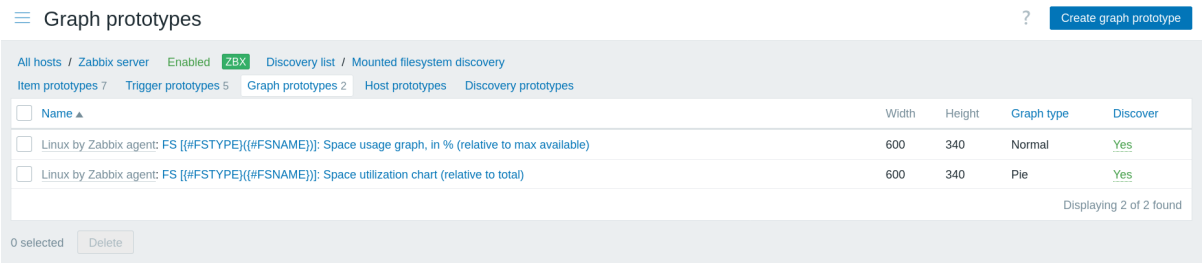
- *Izveidot iespējotus* - izveidot šos trigerus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos trigerus kā *Atspējotus*
- *Masveida atjaunināšana* - masveidā atjaunināt šos trigeru prototipus
- *Dzēst* - dzēst šos trigeru prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem trigeru prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

3 Grafiku prototipi

Pārskats

Šajā sadaļā tiek attēloti hosta zema līmeņa atklāšanas kārtulas grafiku prototipi. Grafiku prototipi ir pamats reāliem hosta grafikiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Attēlotie dati:

Kolonna	Apraksts
Nosaukums	Grafika prototipa nosaukums, kas tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta grafika prototipa konfigurācijas forma. Ja grafika prototips pieder saistītai veidnei, pirms grafika nosaukuma kā pelēka saite tiek attēlots veidnes nosaukums. Noklikšķinot uz veidnes saites, tiek atvērts grafiku prototipu saraksts saistītās veidnes līmenī.
Platums	Tiek attēlots grafika prototipa platums.
Augstums	Tiek attēlots grafika prototipa augstums.
Tips	Tiek attēlots grafika prototipa tips — *Parasts*, *Sakrauts*, *Sektoru* vai *Izvērsts*.
Atklāt	Atklāt grafiku, pamatojoties uz šo prototipu: **Jā** — atklāt **Nē** — neatklāt. Varat pārslēgties starp "Jā" un "Nē", noklikšķinot uz tiem.

Lai konfigurētu jaunu grafika prototipu, noklikšķiniet uz pogas *Izveidot grafika prototipu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

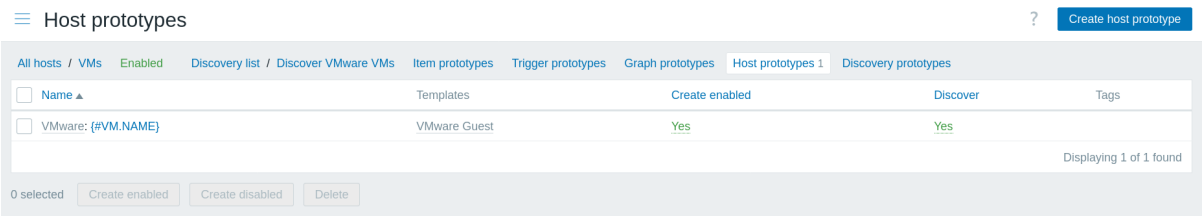
- *Dzēst* — dzēst šos grafiku prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem grafiku prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

4 Hostu prototipi

Pārskats

Šajā sadaļā tiek attēloti hosta zema līmeņa atklāšanas kārtulas hostu prototipi. Hostu prototipi ir pamats reāliem hostiem, kas tiek izveidoti zema līmeņa atklāšanas laikā.



Attēlotie dati:

Kolonna	Apraksts
Name	Hosta prototipa nosaukums, kas tiek attēlots kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta hosta prototipa konfigurācijas forma. Ja hosta prototips pieder piesaistītai veidnei, veidnes nosaukums pirms hosta nosaukuma tiek attēlots kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts hosta prototipu saraksts piesaistītās veidnes līmenī.
Templates	Tiek attēlotas hosta prototipa veidnes.
Create enabled	Izveidot hostu, pamatojoties uz šo prototipu, kā: **Yes** - iespējots **No** - atspējots. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Discover	Atklāt hostu, pamatojoties uz šo prototipu: **Yes** - atklāt **No** - neatklāt. Varat pārslēgties starp 'Yes' un 'No', noklikšķinot uz tiem.
Tags	Tiek attēlotas hosta prototipa birkas.

Lai konfigurētu jaunu hosta prototipu, noklikšķiniet uz pogas *Create host prototype* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

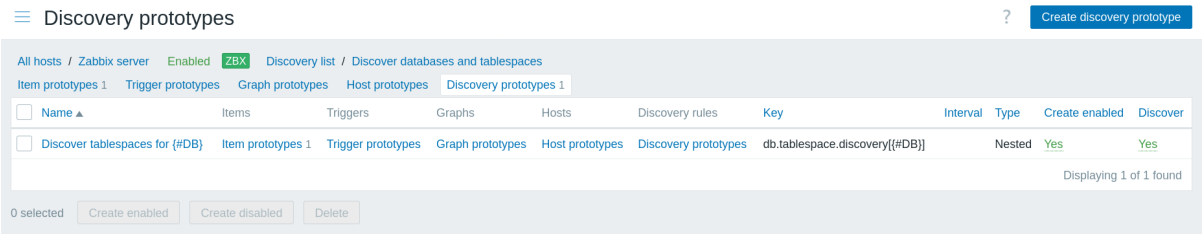
- *Izveidot iespējotus* - izveidot šos hostus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos hostus kā *Atspējotus*
- *Dzēst* - dzēst šos hostu prototipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem hostu prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

5 Atklāšanas prototipi

Pārskats

Šajā sadaļā tiek parādīti hosta zema līmeņa atklāšanas kārtulas atklāšanas prototipi. Atklāšanas prototipi ir vecākās atklāšanas kārtulas ligzdotas atklāšanas kārtulas.



Parādītie dati:

Column	Description
Name	Atklāšanas prototipa nosaukums tiek parādīts kā zila saite. Noklikšķinot uz nosaukuma, tiek atvērta atklāšanas prototipa konfigurācijas forma. Ja atklāšanas prototips pieder piesaistītai veidnei, veidnes nosaukums pirms hosta nosaukuma tiek parādīts kā pelēka saite. Noklikšķinot uz veidnes saites, tiek atvērts atklāšanas prototipu saraksts piesaistītās veidnes līmenī.
Items	Tiek parādīta saite uz vienumu prototipu sarakstu. Esošo vienumu prototipu skaits tiek parādīts pelēkā krāsā.

Column	Description
Triggers	Tiek parādīta saite uz trigeru prototipu sarakstu. Esošo trigeru prototipu skaits tiek parādīts pelēkā krāsā.
Graphs	Tiek parādīta saite uz grafiku prototipu sarakstu. Esošo grafiku prototipu skaits tiek parādīts pelēkā krāsā.
Hosts	Tiek parādīta saite uz hostu prototipu sarakstu. Esošo hostu prototipu skaits tiek parādīts pelēkā krāsā.
Discovery rules	Tiek parādīta saite uz atklāšanas prototipu sarakstu. Esošo atklāšanas prototipu skaits tiek parādīts pelēkā krāsā.
Key	Tiek parādīta atklāšanai izmantotā vienuma atslēga.
Interval	Tiek parādīts atklāšanas izpildes biežums. *Ņemiet vērā*, ka atklāšanu var izpildīt arī nekavējoties, nospiežot zem saraksta esošo pogu *Execute now*.
Type	Tiek parādīts atklāšanai izmantotais vienuma tips (Zabbix aģents, SNMP aģents u.c.).
Create enabled	Tiek parādīts atklāšanas prototipa izveides statuss — izveide iespējota (*Yes*) vai izveide atspējota (*No*). Noklikšķinot uz statusa, to var mainīt — no Yes uz No (un atpakaļ).
Discover	Tiek parādīts atklāšanas prototipa atklāšanas statuss — atklāt (*Yes*) vai neatklāt (*No*). Noklikšķinot uz statusa, to var mainīt — no Yes uz No (un atpakaļ).

Lai konfigurētu jaunu atklāšanas prototipu, noklikšķiniet uz pogas *Create discovery prototype* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

- *Izveidot iespējotus* - izveidot šos atklāšanas prototipus kā *Iespējotus*
- *Izveidot atspējotus* - izveidot šos atklāšanas prototipus kā *Atspējotus*
- *Dzēst* - dzēst šos atklāšanas prototipus

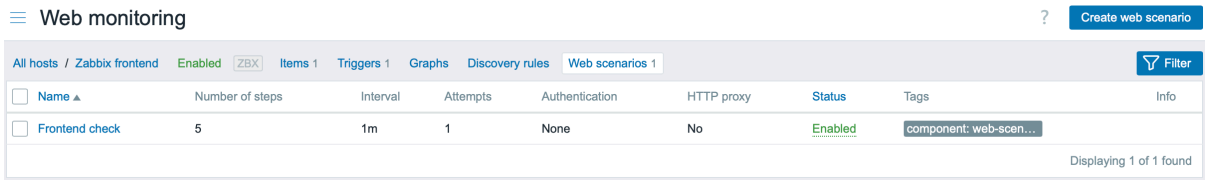
Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem atklāšanas prototipiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

5 Tīmekļa scenāriji

Pārskats

Web scenāriju sarakstam hostam var piekļūt sadaļā *Datu ievākšana* → *Hosti*, noklikšķinot uz *Web* attiecīgajam hostam.

Tiek parādīts esošo web scenāriju saraksts.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Web scenārija nosaukums. Noklikšķinot uz web scenārija nosaukuma, tiek atvērta web scenārija konfigurācijas forma. Ja hosta web scenārijs pieder veidnei, pirms web scenārija nosaukuma kā pelēka saite tiek parādīts veidnes nosaukums. Noklikšķinot uz veidnes saites, tiks atvērts web scenāriju saraksts veidnes līmenī.
Soļu skaits	Scenārijā iekļauto soļu skaits.
Atjaunināšanas intervāls	Cik bieži scenārijs tiek izpildīts.
Mēģinājumi	Cik mēģinājumu tiek veikts, izpildot web scenārija soļus.
Autentifikācija	Tiek parādīta autentifikācijas metode — Basic, NTLM, Kerberos, Digest vai None.
HTTP starpniekserveris	Parāda HTTP starpniekserveri vai 'No', ja tas netiek izmantots.
Statuss	Tiek parādīts web scenārija statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.

Kolonna	Apraksts
Birkas	Tiek parādītas web scenārija birkas. Var tikt parādītas līdz trim birkām (nosaukums:vērtība pāri). Ja birku ir vairāk, tiek parādīta saite "...", kas ļauj, novietojot peles kursoru virs tās, apskatīt visas birkas.
Informācija	Ja viss darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā tiek parādīta kvadrātveida ikona ar burtu "i". Novietojiet peles kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu web scenāriju, noklikšķiniet uz pogas *Izveidot web scenāriju* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

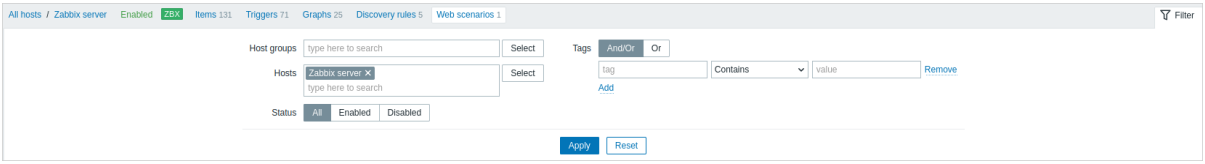
- *Iespējot* - mainīt scenārija statusu uz *Iespējots*
- *Atspējot* - mainīt scenārija statusu uz *Atspējots*
- *Notīrīt vēsturi un tendences* - notīrīt scenāriju vēstures un tendenču datus
- *Dzēst* - dzēst tīmekļa scenārijus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem tīmekļa scenārijiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos scenārijus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrokomandām.

Saite *Filtrs* ir pieejama virs tīmekļa scenāriju saraksta. Noklikšķinot uz tās, tiks atvērts filtrs, kurā varat norādīt vēlamos filtrēšanas kritērijus.



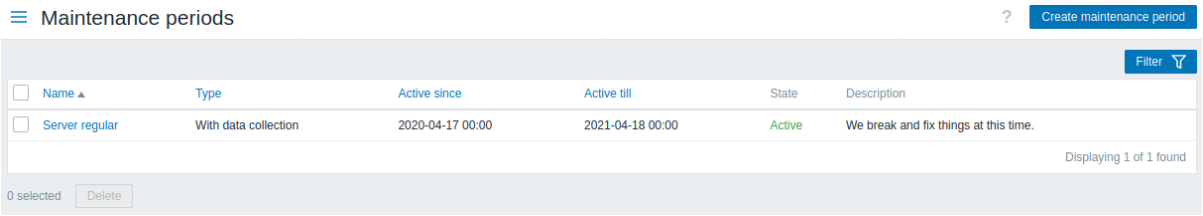
Parametrs	Apraksts
Hostu grupas	Filtrēt pēc vienas vai vairākām hostu grupām. Norādot vecākgrupas hostu grupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas.
Hosti	Filtrēt pēc viena vai vairākiem hostiem. Ja iepriekš ir atlasītas hostu grupas, hostu atlase ir ierobežota ar šīm grupām.
Statuss	Filtrēt pēc tīmekļa scenārija statusa.
Birkas	Filtrēt pēc tīmekļa scenārija birkas nosaukuma un vērtības. Ir iespējams gan iekļaut, gan izslēgt noteiktas birkas un birku vērtības. Var iestatīt vairākus nosacījumus. Birkas nosaukuma atbilstība vienmēr ir reģistrjutīga. Katrai nosacījuma rindai ir pieejami vairāki operatori: **Eksistē** — iekļaut norādītos birku nosaukumus; **Vienāds ar** — iekļaut norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Satur** — iekļaut norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga); **Neeksistē** — izslēgt norādītos birku nosaukumus; **Nav vienāds ar** — izslēgt norādītos birku nosaukumus un vērtības (reģistrjutīgi); **Nesatur** — izslēgt norādītos birku nosaukumus, kuru birku vērtības satur ievadīto virkni (apakšvirknes atbilstība, nereģistrjutīga). Nosacījumiem ir pieejami divi aprēķina veidi: **Un/Vai** — jāizpildās visiem nosacījumiem; nosacījumi ar vienādu birkas nosaukumu tiks grupēti pēc nosacījuma Vai; **Vai** — pietiek, ja izpildās viens nosacījums.

5 Uzturēšana

Pārskats

Sadaļā *Datu ievākšana → Uzturēšana* lietotāji var konfigurēt un pārvaldīt hostu uzturēšanas periodus.

Tiek parādīts esošo uzturēšanas periodu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Uzturēšanas perioda nosaukums. Noklikšķinot uz uzturēšanas perioda nosaukuma, tiek atvērta uzturēšanas perioda konfigurācijas forma.
Tips	Tiek parādīts uzturēšanas tips: *Ar datu ievākšanu* vai *Bez datu ievākšanas*
Aktīvs kopš	Datums un laiks, kad uzturēšanas periodu izpilde kļūst aktīva. Piezīme: Šis laiks neaktivizē uzturēšanas periodu; uzturēšanas periodi jāiestata atsevišķi.
Aktīvs līdz	Datums un laiks, kad uzturēšanas periodu izpilde pārstāj būt aktīva.
Stāvoklis	Uzturēšanas perioda stāvoklis: **Tuvojas** - drīz kļūs aktīvs **Aktīvs** - ir aktīvs **Beidzies** - vairs nav aktīvs
Apraksts	Tiek parādīts uzturēšanas perioda apraksts.

Lai konfigurētu jaunu uzturēšanas periodu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot uzturēšanas periodu*.

Masveida rediģēšanas opcijas

Poga zem saraksta piedāvā vienu masveida rediģēšanas opciju:

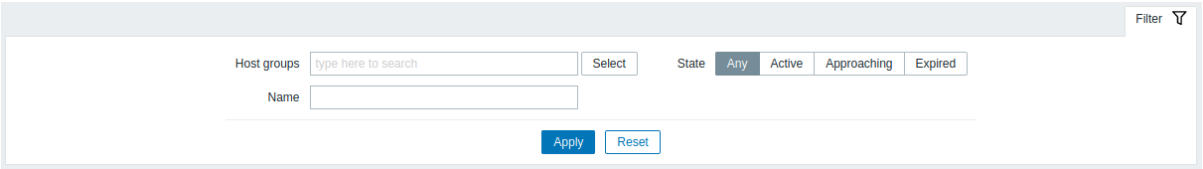
- *Dzēst* - dzēst uzturēšanas periodus

Lai izmantotu šo opciju, atzīmējiet izvēles rūtiņas pie attiecīgajiem uzturēšanas periodiem un noklikšķiniet uz *Dzēst*.

Filtra izmantošana

Jūs varat izmantot filtru, lai parādītu tikai tos uzturēšanas periodus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs uzturēšanas periodu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt uzturēšanas periodus pēc hostu grupas, nosaukuma un stāvokļa.



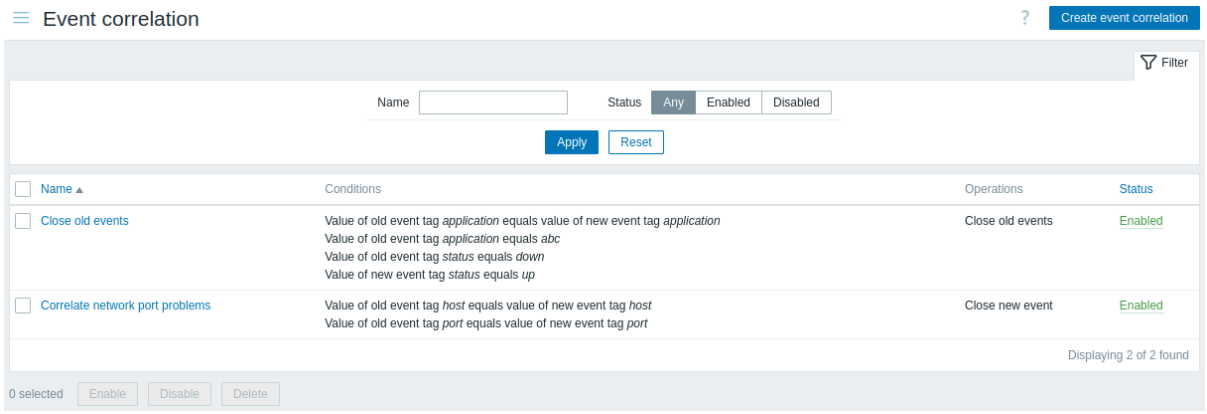
Rindu aprēķināšana apkopes laikā

Attention:
Zabbix starpniekserveris nav informēts par apkopes periodiem; sīkāku informāciju skatiet Rindu aprēķināšana apkopes laikā.

6 Notikumu korelācija

Pārskats

Sadaļā *Datu ievākšana → Notikumu korelācija* lietotāji var konfigurēt un uzturēt globālās korelācijas noteikumus Zabbix notikumiem.



Attēlotie dati:

Kolonna	Apraksts
Nosaukums	Korelācijas noteikuma nosaukums. Noklikšķinot uz korelācijas noteikuma nosaukuma, tiek atvērta noteikuma konfigurācijas forma.
Nosacījumi	Tiek attēloti korelācijas noteikuma nosacījumi.
Darbības	Tiek attēlotas korelācijas noteikuma darbības.
Statuss	Tiek attēlots korelācijas noteikuma statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.

Lai konfigurētu jaunu korelācijas noteikumu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot notikumu korelāciju*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

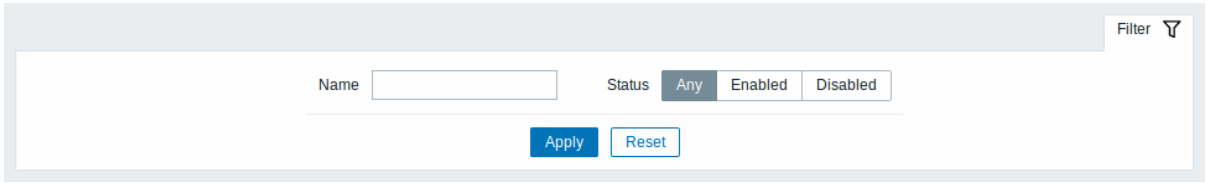
- *Iespējot* - mainīt korelācijas noteikuma statusu uz *Iespējots*
- *Atspējot* - mainīt korelācijas noteikuma statusu uz *Atspējots*
- *Dzēst* - dzēst korelācijas noteikumus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem korelācijas noteikumiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos korelācijas noteikumus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs korelācijas noteikumu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt korelācijas noteikumus pēc nosaukuma un statusa.

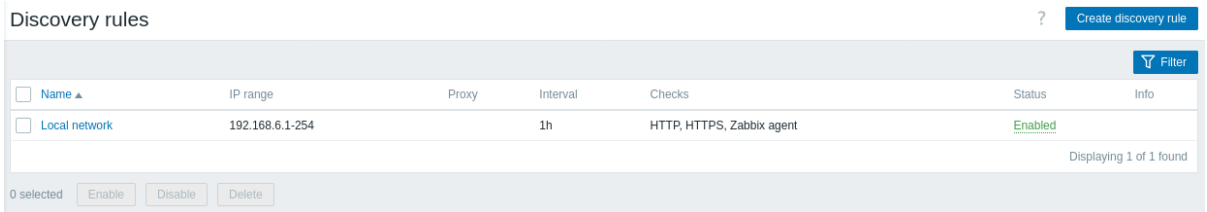


7 Atklāšana

Pārskats

Sadaļā *Datu ievākšana → Atklāšana* lietotāji var konfigurēt un uzturēt atklāšanas kārtulas.

Tiek parādīts esošo atklāšanas kārtulu saraksts ar detalizētu informāciju par tām.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Atklāšanas kārtulas nosaukums. Noklikšķinot uz atklāšanas kārtulas nosaukuma, tiek atvērta atklāšanas kārtulas konfigurācijas forma.
IP diapazons	Tiek parādīts tīkla skenēšanai izmantojamo IP adrešu diapazons.
Starpniekserveris	Tiek parādīts starpniekservera nosaukums, ja atklāšanu veic starpniekserveris.
Intervāls	Tiek parādīts atklāšanas izpildes biežums.
Pārbaudes	Tiek parādīti atklāšanai izmantoto pārbaužu tipi.
Statuss	Tiek parādīts atklāšanas kārtulas statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Informācija	Ja viss darbojas pareizi, šajā kolonnā nekas netiek parādīts. Kļūdu gadījumā tiek parādīta sarkana informācijas ikona ar burtu "i". Virziet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu atklāšanas kārtulu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot atklāšanas kārtulu*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

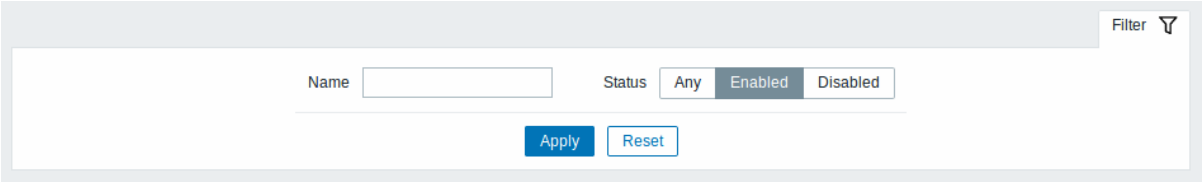
- *Iespējot* — mainīt atklāšanas kārtulas statusu uz *Iespējots*
- *Atspējot* — mainīt atklāšanas kārtulas statusu uz *Atspējots*
- *Dzēst* — dzēst atklāšanas kārtulas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām atklāšanas kārtulām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos atklāšanas noteikumus, kas jūs interesē. Lai uzlabotu meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs atklāšanas noteikumu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt atklāšanas noteikumus pēc nosaukuma un statusa.



7 Brīdinājumi

Pārskats

Šajā izvēlnē ir pieejamas sadaļas, kas saistītas ar brīdinājumu konfigurēšanu Zabbix.

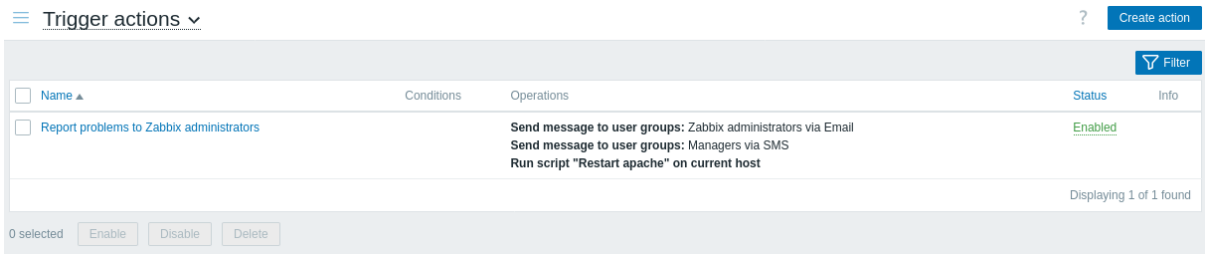
1 Darbības

Pārskats

Sadaļā *Brīdinājumi* → *Darbības* lietotāji var konfigurēt un uzturēt darbības.

Parādītās darbības ir darbības, kas piešķirtas atlasītajam notikumu avotam (trigeri, pakalpojumi, atklāšana, automātiskā reģistrācija, iekšējās darbības).

Lai pārslēgtos uz citu notikumu avotu, noklikšķiniet uz *Darbības* sadaļā *Brīdinājumi* izvēlnē. Ir iespējams arī pārslēgties starp avotiem, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Darbības nosaukums. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurēšanas forma.
Nosacījumi	Tiek parādīti darbības nosacījumi.
Operācijas	Tiek parādītas darbības operācijas. Operāciju sarakstā tiek parādīts arī paziņojumiem izmantotais medija tips (e-pasts, SMS vai skripts), kā arī paziņojuma saņēmēja vārds un uzvārds (iekavās aiz lietotājvārda). Darbības operācija atkarībā no atlasītā operācijas tipa var būt gan paziņojums, gan attālināta komanda.
Statuss	Tiek parādīts darbības statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt. Papildinformāciju par to, kas notiek, ja darbība tiek atspējota eskalācijas laikā, skatiet sadaļā Eskalācijas.
Informācija	Ja viss darbojas pareizi, šajā kolonnā ikona netiek parādīta. Kļūdu gadījumā, piemēram, ja pēc jaunināšanas/zema līmeņa atklāšanas trūkst darbības operāciju vai nosacījumu, tiek parādīta brīdinājuma ikona. Virziet kursoru virs ikonas, lai redzētu uznirstošo paskaidrojumu ar kļūdas aprakstu.

Lai konfigurētu jaunu darbību, noklikšķiniet uz pogas *Izveidot darbību* augšējā labajā stūrī.

Lietotājiem bez Super admin tiesībām darbības tiek parādītas atbilstoši atļauju iestatījumiem. Tas nozīmē, ka dažos gadījumos lietotājs bez Super admin tiesībām nevar skatīt pilnu darbību sarakstu noteiktu atļauju ierobežojumu dēļ. Darbība tiek parādīta lietotājam bez Super admin tiesībām, ja ir izpildīti šādi nosacījumi:

- Lietotājam ir lasīšanas-rakstīšanas piekļuve hostu grupām, hostiem, veidnēm un trigeriem darbības nosacījumos
- Lietotājam ir lasīšanas-rakstīšanas piekļuve hostu grupām, hostiem un veidnēm darbības operācijās, atkopšanas operācijās un atjaunināšanas operācijās
- Lietotājam ir lasīšanas piekļuve lietotāju grupām un lietotājiem darbības operācijās, atkopšanas operācijās un atjaunināšanas operācijās

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

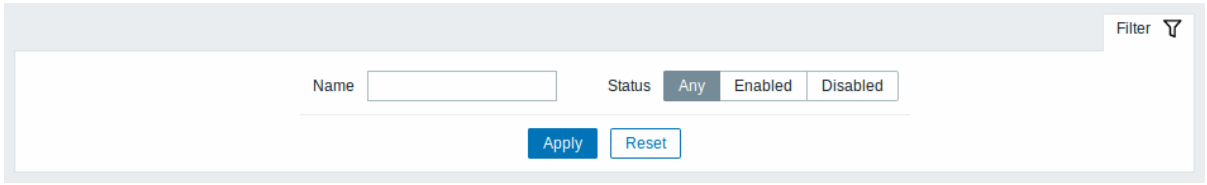
- *Iespējot* - mainīt darbības statusu uz *Iespējots*
- *Atspējot* - mainīt darbības statusu uz *Atspējots*
- *Dzēst* - dzēst darbības

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām darbībām, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tās darbības, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makrosiem.

Saite *Filtrs* ir pieejama virs darbību saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt darbības pēc nosaukuma un statusa.



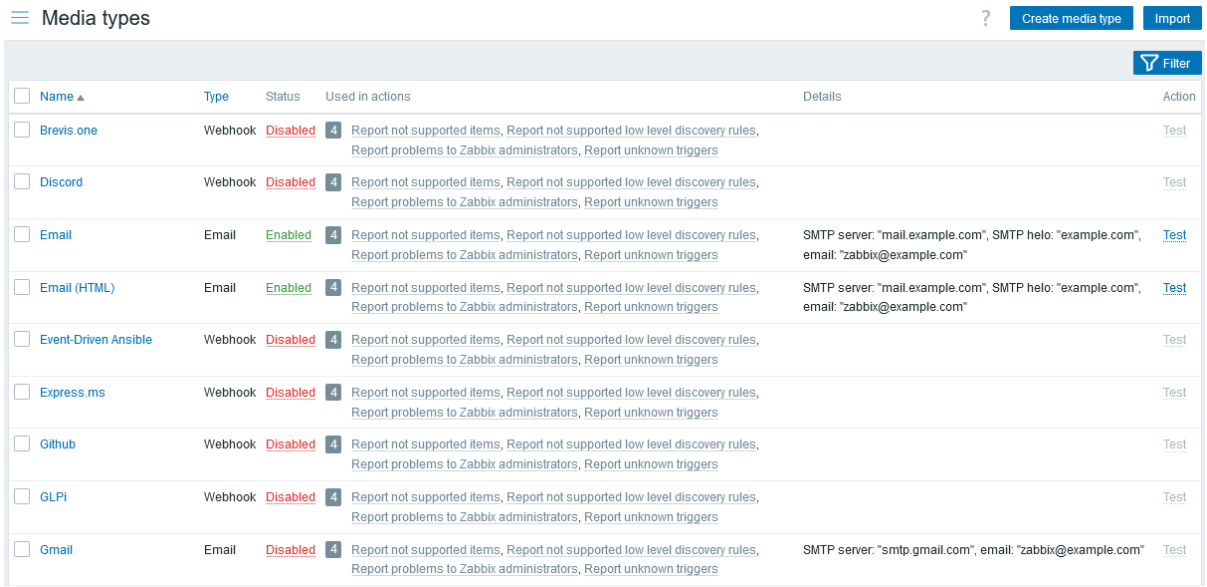
2 Mediju tipi

Pārskats

Sadaļā *Brīdinājumi* → *Paziņošanas veidi* lietotāji var konfigurēt un uzturēt paziņošanas veidu informāciju.

Paziņošanas veida informācija satur vispārīgus norādījumus par vides izmantošanu kā paziņojumu piegādes kanālu. Konkrēta informācija, piemēram, atsevišķas e-pasta adreses, uz kurām nosūtīt paziņojumu, tiek glabāta pie katra lietotāja atsevišķi.

Tiek parādīts esošo paziņošanas veidu saraksts ar to detalizēto informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Paziņošanas veida nosaukums. Noklikšķinot uz nosaukuma, tiek atvērta paziņošanas veida konfigurācijas forma.
Tips	Tiek parādīts vides tips (e-pasts, SMS u.c.).
Statuss	Tiek parādīts paziņošanas veida statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Izmantots darbībās	Tiek parādītas darbības, kurās tiek izmantots paziņošanas veids, pirms tam norādot šo darbību kopējo skaitu. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurācijas forma. Ja lietotājam nav atļauju šai darbībai, nosaukums nav klikšķināms.
Detaļas	Tiek parādīta detalizēta informācija par paziņošanas veidu.
Darbības	Ir pieejama šāda darbība: **Testēt** — noklikšķiniet, lai atvērtu testēšanas formu, kurā var ievadīt paziņošanas veida parametrus (piemēram, saņēmēja adresi, testa tēmu un ziņojuma tekstu) un nosūtīt testa ziņojumu, lai pārbaudītu, vai konfigurētais paziņošanas veids darbojas. Skatiet arī: Paziņošanas veida testēšana.

Lai konfigurētu jaunu paziņošanas veidu, noklikšķiniet uz pogas *Izveidot paziņošanas veidu* augšējā labajā stūrī.

Lai importētu paziņošanas veidu, noklikšķiniet uz pogas *Importēt* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

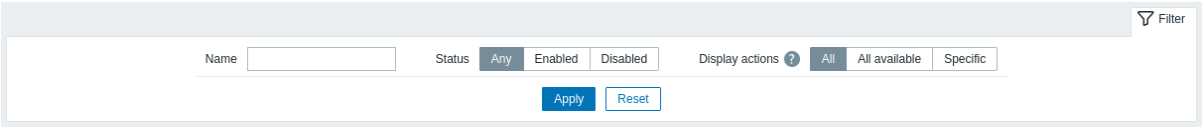
- *Iespējot* - mainīt mediju tipa statusu uz *Iespējots*
- *Atspējot* - mainīt mediju tipa statusu uz *Atspējots*
- *Eksportēt* - eksportēt mediju tipus YAML, XML vai JSON failā
- *Dzēst* - dzēst mediju tipus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem mediju tipiem un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai attēlotu tikai tos mediju tipus, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs mediju tipu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt mediju tipus pēc nosaukuma un statusa. Papildus varat izmantot filtru, lai kolonnā *Izmantots darbībās* attēlotu darbības atbilstoši to mediju tipu izmantošanas tvērumam (noteikts darbības operācijas detaļās ar parametru *Send to media type*).



3 Skripti

Pārskats

Sadaļā *Brīdinājumi* > *Skripti* var konfigurēt un uzturēt lietotāja definētus globālos skriptus.

Globālie skripti atkarībā no konfigurētās darbības jomas un arī lietotāja atļaujām ir pieejami izpildei:

- no hosta izvēlnes dažādās lietotāja saskarnes vietās (*Panelis*, *Problēmas*, *Jaunākie dati*, *Kartes* u.c.)
- no notikuma izvēlnes
- var tikt palaisti kā darbības operācija

Skripti tiek izpildīti uz Zabbix aģenta, Zabbix servera (starpniekservera) vai tikai uz Zabbix servera.
Skatiet arī Komandu izpilde.

Gan Zabbix aģentā, gan Zabbix starpniekserverī attālinātie skripti pēc noklusējuma ir atspējoti.
Tos var iespējot šādi:

- Attālinātām komandām, kas tiek izpildītas uz Zabbix aģenta:
  - katrai atļautajai komandai aģenta konfigurācijā pievienojot parametru AllowKey=system.run[<command>,*], kur * apzīmē wait un nowait režīmu;
- Attālinātām komandām, kas tiek izpildītas uz Zabbix starpniekservera:
  - **Brīdinājums: Nav nepieciešams iespējot attālinātās komandas uz Zabbix starpniekservera, ja attālinātās komandas tiek izpildītas uz Zabbix aģenta, kuru uzrauga Zabbix starpniekserveris.** Tomēr, ja ir nepieciešams izpildīt attālinātās komandas uz Zabbix starpniekservera, iestatiet *EnableRemoteCommands* parametru uz '1' starpniekservera konfigurācijā.

Globālo skriptu izpildi uz Zabbix servera var atspējot, servera konfigurācijā iestatot `EnableGlobalScripts=0`.
Jaunām instalācijām, sākot ar Zabbix 7.0, globālo skriptu izpilde uz Zabbix servera pēc noklusējuma ir atspējota.

Tiek parādīts esošo skriptu saraksts ar to detalizēto informāciju.

Parādītie dati:

Kolonna	Apraksts
Nosaukums	Skripta nosaukums. Noklikšķinot uz skripta nosaukuma, tiek atvērta skripta konfigurācijas forma.
Darbības joma	Skripta darbības joma — darbības operācija, manuāla hosta darbība vai manuāla notikuma darbība. Šis iestatījums nosaka, kur skripts ir pieejams.
Izmantots darbībās	Tiek parādītas visas darbības, kurās skripts tiek izmantots, pirms tam norādot šo darbību kopējo skaitu. Noklikšķinot uz darbības nosaukuma, tiek atvērta darbības konfigurācijas forma. Ja lietotājam nav atļauju šai darbībai, nosaukums nav klikšķināms.

Kolonna	Apraksts
Tips	Tiek parādīts skripta tips — *URL*, *Webhook*, *Skripts*, *SSH*, *Telnet* vai *IPMI* komanda.
Izpildīt uz	Tiek parādīts, vai skripts tiks izpildīts uz Zabbix aģenta, Zabbix starpniekservera vai servera, vai tikai uz Zabbix servera.
Komandas	Tiek parādītas visas komandas, kas tiks izpildītas skripta ietvaros. Webhook gadījumā šeit nekas netiek parādīts.
Lietotāju grupa	Tiek parādīta lietotāju grupa, kurai skripts ir pieejams (vai *Visi* visām lietotāju grupām).
Hostu grupa	Tiek parādīta hostu grupa, kurai skripts ir pieejams (vai *Visi* visām hostu grupām).
Hosta piekļuve	Tiek parādīts hostu grupas atļauju līmenis — *Lasīt* vai *Rakstīt*. Tikai lietotājiem ar nepieciešamo atļauju līmeni būs piekļuve skripta izpildei.

Lai konfigurētu jaunu skriptu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot skriptu*.

Masveida rediģēšanas opcijas

Poga zem saraksta piedāvā vienu masveida rediģēšanas opciju:

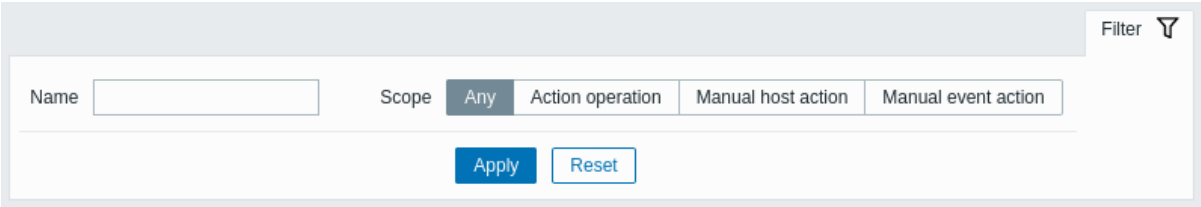
- *Dzēst* - dzēst skriptus

Lai izmantotu šo opciju, atzīmējiet izvēles rūtiņas pie attiecīgajiem skriptiem un noklikšķiniet uz *Dzēst*.

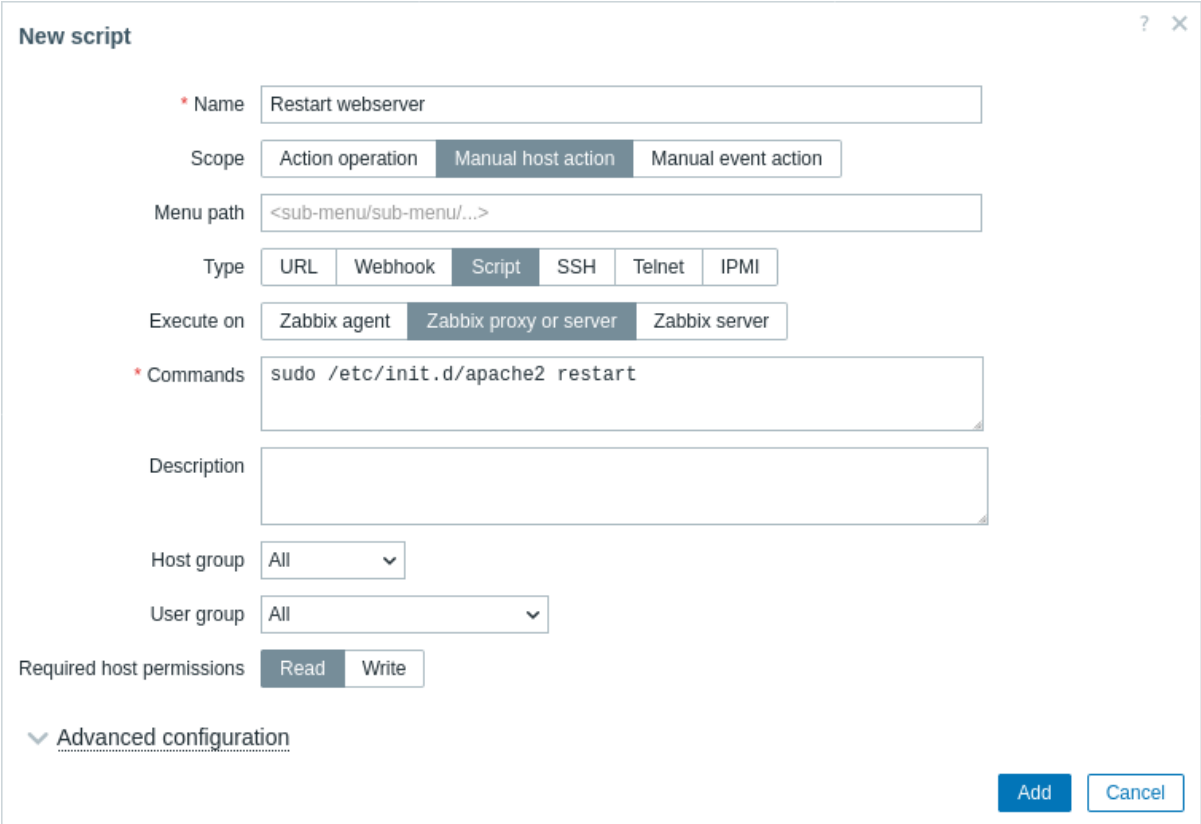
Filtra izmantošana

Jūs varat izmantot filtru, lai parādītu tikai tos skriptus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs skriptu saraksta. Ja uz tās noklikšķināt, kļūst pieejams filtrs, kurā varat filtrēt skriptus pēc nosaukuma un tvēruma.



Globāla skripta konfigurēšana



Note:
Drošības uzlabošanai globālajos skriptos ieteicams izmantot makro funkcijas vienkāršu makro vietā, jo makro netiek automātiski atsaistīti.

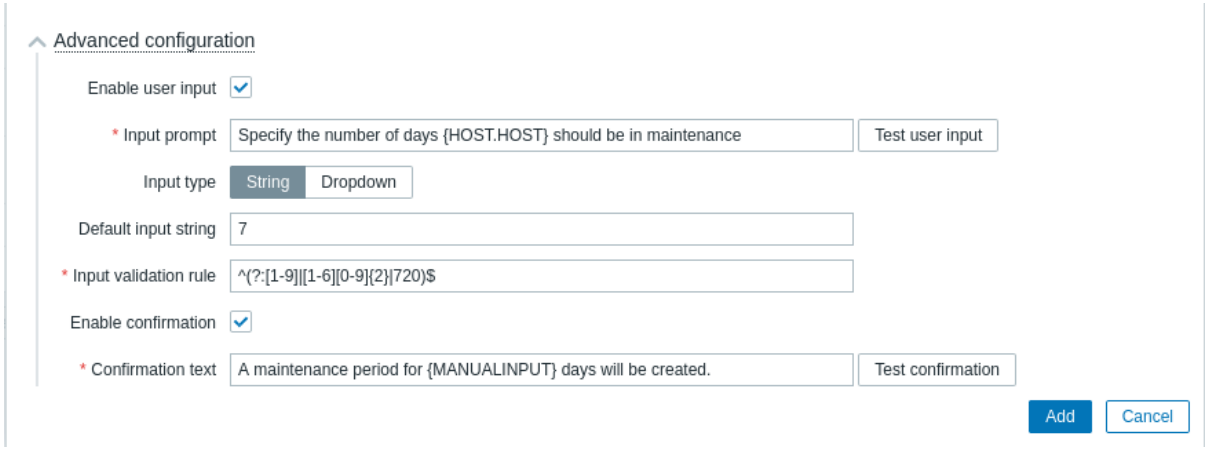
Skripta atribūti:

Parametrs	Apraksts
Nosaukums	Unikāls skripta nosaukums. Piem., `Clear /tmp filesystem`
Tvērums	Skripta tvērums — darbības operācija, manuāla hosta darbība vai manuāla notikuma darbība. Šis iestatījums nosaka, kur skriptu var izmantot — darbību operāciju attālinātajās komandās, no hosta izvēlnes vai no notikuma izvēlnes. Iestatot tvērumu uz 'Action operation', skripts kļūst pieejams visiem lietotājiem, kuriem ir piekļuve *Brīdinājumi* > *Darbības*. Ja skripts faktiski tiek izmantots darbībā, tā tvērumu nevar mainīt no 'action operation' uz citu vērtību. **Makro atbalsts** Tvērums ietekmē pieejamo makro diapazonu. Piemēram, ar lietotāju saistītie makro ({USER.*}) tiek atbalstīti skriptos, lai varētu nodot informāciju par lietotāju, kurš palaida skriptu. Taču tie netiek atbalstīti, ja skripta tvērums ir action operation, jo darbību operācijas tiek izpildītas automātiski. Makro {MANUALINPUT} ļauj norādīt manuālu ievadi skripta izpildes laikā. Tas tiek atbalstīts manuālas hosta darbības un manuālas notikuma darbības skriptos. Lai noskaidrotu, kuri citi makro tiek atbalstīti, meklējiet 'Trigger-based notifications and commands/Trigger-based commands', 'Manual host action scripts' un 'Manual event action scripts' tabulā atbalstītie makro. Ņemiet vērā: ja makro var tikt atrisināts uz vērtību ar atstarpēm (piemēram, hosta nosaukumu), neaizmirstiet to vajadzības gadījumā ievietot pēdiņās.
Izvēlnes ceļš	Vēlamais izvēlnes ceļš uz skriptu. Piemēram, `Default` vai `Default/` parādīs skriptu attiecīgajā direktorijā. Izvēlnes var būt ligzdotas, piemēram, `Main menu/Sub menu1/Sub menu2`. Piekļūstot skriptiem caur hosta/notikuma izvēlni uzraudzības sadaļās, tie tiks sakārtoti atbilstoši norādītajām direktorijām. Šis lauks tiek rādīts tikai tad, ja kā *Tvērums* ir atlasīts 'Manual host action' vai 'Manual event action'.
Tips	Noklikšķiniet uz attiecīgās pogas, lai atlasītu skripta tipu: **URL**, **Webhook**, **Skripts**, **SSH**, **Telnet** vai **IPMI** komanda. Tips **URL** ir pieejams tikai tad, ja kā *Tvērums* ir atlasīts 'Manual host action' vai 'Manual event action'.
Skripta tips: **URL**	
URL	Norādiet URL ātrai piekļuvei no hosta izvēlnes vai notikuma izvēlnes. Tiek atbalstīti makro un pielāgoti lietotāja makro. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Izmantojiet makro {MANUALINPUT} šajā laukā, lai skripta izpildes laikā varētu norādīt manuālu ievadi, piemēram: `http://{MANUALINPUT}/zabbix/zabbix.php?action=dashboard.view` Makro vērtības nedrīkst būt URL kodētas.
Atvērt jaunā logā	Nosaka, vai URL jāatver jaunā vai tajā pašā pārlūka cilnē.
Skripta tips: **Webhook**	
Parametri	Norādiet webhook mainīgos kā atribūta-vērtības pārus. Skatiet arī: Webhook medija konfigurācija. Makro un pielāgoti lietotāja makro tiek atbalstīti parametru vērtībās. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš).
Skripts	Ievadiet JavaScript kodu modālajā redaktorā, kas tiek atvērts, noklikšķinot parametra laukā vai uz zīmuļa ikonas blakus tam. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Skatiet arī: Webhook medija konfigurācija, Papildu Javascript objekti.
Noildze	JavaScript izpildes noildze (1-60s, noklusējums 30s). Tiek atbalstīti laika sufiksi, piem., 30s, 1m.
Skripta tips: **Script**	

Parametrs	Apraksts
Izpildīt uz	Noklikšķiniet uz attiecīgās pogas, lai izpildītu čaulas skriptu uz: **Zabbix aģents** — skriptu izpildīs Zabbix aģents (ja system.run vienums ir atļauts) hostā **Zabbix starpniekserveris vai serveris** — skriptu izpildīs Zabbix starpniekserveris vai serveris atkarībā no tā, vai hosts tiek uzraudzīts ar starpniekserveri vai serveri. Tas tiks izpildīts uz starpniekservera, ja to atļauj EnableRemoteCommands. Tas tiks izpildīts uz servera, ja globālie skripti ir iespējoti ar servera parametru EnableGlobalScripts. **Zabbix serveris** — skriptu izpildīs tikai Zabbix serveris. Šī opcija nebūs pieejama, ja globālie skripti ir atspējoti ar servera parametru EnableGlobalScripts.
Komandas	Ievadiet pilnu ceļu uz komandām, kas jāizpilda skriptā. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **SSH**	
Autentifikācijas metode	Atlasiet autentifikācijas metodi — parole vai publiskā atslēga.
Lietotājvārds	Ievadiet lietotājvārdu.
Parole	Ievadiet paroli. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta 'Password'.
Publiskās atslēgas fails	Ievadiet ceļu uz publiskās atslēgas failu. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta 'Public key'.
Privātās atslēgas fails	Ievadiet ceļu uz privātās atslēgas failu. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta 'Public key'.
Paroles frāze	Ievadiet paroles frāzi. Šis lauks ir pieejams, ja kā autentifikācijas metode ir atlasīta 'Public key'.
Ports	Ievadiet attālā SSH pakalpojuma portu mērķa hostā, pie kura Zabbix pieslēgsies.
Komandas	Ievadiet komandas. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **Telnet**	
Lietotājvārds	Ievadiet lietotājvārdu.
Parole	Ievadiet paroli.
Ports	Ievadiet attālā Telnet pakalpojuma portu mērķa hostā, pie kura Zabbix pieslēgsies.
Komandas	Ievadiet komandas. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Tiek atbalstīti pielāgoti lietotāja makro.
Skripta tips: **IPMI**	
Komanda	Ievadiet IPMI komandu. Makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* iepriekš). Tiek atbalstīti pielāgoti lietotāja makro.
Apraksts	Ievadiet skripta aprakstu.
Hostu grupa	Atlasiet hostu grupu, kurai skripts būs pieejams (vai *All* visām hostu grupām).
Lietotāju grupa	Atlasiet lietotāju grupu, kurai skripts būs pieejams (vai *All* visām lietotāju grupām). Šis lauks tiek rādīts tikai tad, ja kā *Tvērums* ir atlasīts 'Manual host action' vai 'Manual event action'.
Nepieciešamās hosta atļaujas	Atlasiet atļauju līmeni hostu grupai — *Read* vai *Write*. Tikai lietotāji ar nepieciešamo atļauju līmeni varēs izpildīt skriptu. Šis lauks tiek rādīts tikai tad, ja kā *Tvērums* ir atlasīts 'Manual host action' vai 'Manual event action'.
Paplašinātā konfigurācija	Noklikšķiniet uz etiķetes *Paplašinātā konfigurācija*, lai parādītu paplašinātās konfigurācijas opcijas. Šis lauks tiek rādīts tikai tad, ja kā *Tvērums* ir atlasīts 'Manual host action' vai 'Manual event action'.

Papildu konfigurācija

Papildu konfigurācijas opcijas ir pieejamas sakļaujamā sadaļā *Papildu konfigurācija*:



Parametrs	Apraksts
Iespējot lietotāja ievadi	Atzīmējiet izvēles rūtiņu, lai pirms skripta izpildes iespējotu manuālu lietotāja ievadi. Manuālā lietotāja ievade aizstās makro {MANUALINPUT} vērtību skriptā. Skatiet arī: Manuālā lietotāja ievade.
Ievades uzvedne	Ievadiet pielāgotu tekstu, kas aicina ievadīt pielāgotu lietotāja ievadi. Šis teksts tiks parādīts virs ievades lauka uznirstošajā logā *Manuālā ievade*. Lai skatītu uznirstošā loga *Manuālā ievade* priekšskatījumu, noklikšķiniet uz *Pārbaudīt lietotāja ievadi*. Priekšskatījums arī ļauj pārbaudīt, vai ievades virkne atbilst ievades validācijas noteikumam (skatiet parametrus zemāk). Makro un lietotāja makro atbalsts ir atkarīgs no skripta tvēruma (skatiet *Tvērums* vispārīgajos skripta konfigurācijas parametros).
Ievades tips	Atlasiet manuālās ievades tipu: **Virkne** — viena virkne; **Nolaižamais saraksts** — vērtība tiek atlasīta no vairākām nolaižamā saraksta opcijām.
Nolaižamā saraksta opcijas	Ievadiet unikālas vērtības lietotāja ievades nolaižamajam sarakstam ar komatiem atdalītā sarakstā. Lai nolaižamajā sarakstā iekļautu tukšu opciju, pievienojiet papildu komatu saraksta sākumā, vidū vai beigās. Šis lauks tiek parādīts tikai tad, ja kā *Ievades tips* ir atlasīts 'Nolaižamais saraksts'.
Noklusējuma ievades virkne	Ievadiet noklusējuma virkni lietotāja ievadei (vai neatstājiet nevienu). Šis lauks tiks validēts atbilstoši regulārajai izteiksmei, kas norādīta laukā *Ievades validācijas noteikums*. Šeit ievadītā vērtība pēc noklusējuma tiks parādīta uznirstošajā logā *Manuālā ievade*. Šis lauks tiek parādīts tikai tad, ja kā *Ievades tips* ir atlasīta 'Virkne'.
Ievades validācijas noteikums	Ievadiet regulāro izteiksmi, lai validētu lietotāja ievades virkni. Tiek atbalstītas globālās regulārās izteiksmes. Šis lauks tiek parādīts tikai tad, ja kā *Ievades tips* ir atlasīta 'Virkne'.
Iespējot apstiprinājumu	Atzīmējiet izvēles rūtiņu, lai pirms skripta izpildes parādītu apstiprinājuma ziņojumu. Šī funkcija var būt īpaši noderīga potenciāli bīstamām darbībām (piemēram, pārstartēšanas skriptam) vai darbībām, kas var aizņemt ilgu laiku.
Apstiprinājuma teksts	Ievadiet pielāgotu apstiprinājuma tekstu apstiprinājuma uznirstošajam logam, kas iespējots ar iepriekš minēto izvēles rūtiņu (piemēram, *Attālinātā sistēma tiks pārstartēta. Vai tiešām turpināt?*). Lai redzētu, kā teksts izskatīsies, noklikšķiniet uz *Pārbaudīt apstiprinājumu* blakus laukam. Makro un pielāgotie lietotāja makro tiek atbalstīti. *Piezīme:* pārbaudot apstiprinājuma ziņojumu, makro netiks izvērsti.

Ja ir konfigurēta gan manuālā lietotāja ievade, gan apstiprinājuma ziņojums, tie tiks parādīti secīgos uznirstošajos logos.

Manuāla lietotāja ievade

Manuāla lietotāja ievade ļauj norādīt pielāgotu parametru katrā skripta izpildes reizē. Tas novērš nepieciešamību izveidot vairākus līdzīgus lietotāja skriptus, kas atšķiras tikai ar vienu parametru.

Piemēram, skripta izpildes laikā var būt nepieciešams norādīt citu vesela skaitļa vērtību vai citu URL adresi.

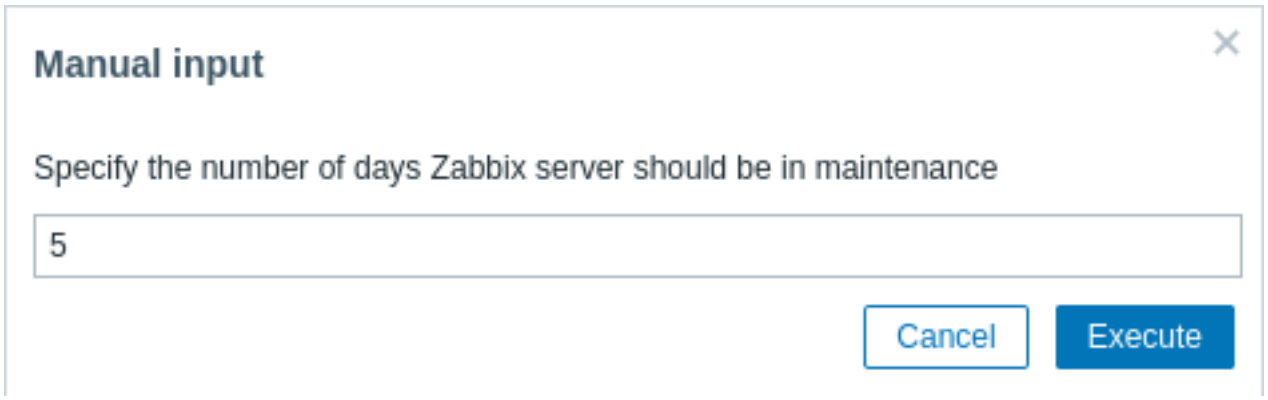
Lai iespējotu manuālu lietotāja ievadi:

- izmantojiet makrosu {MANUALINPUT} skriptā (komandās, skriptā, skripta parametrā), kur tas ir nepieciešams; vai URL skriptu URL laukā;

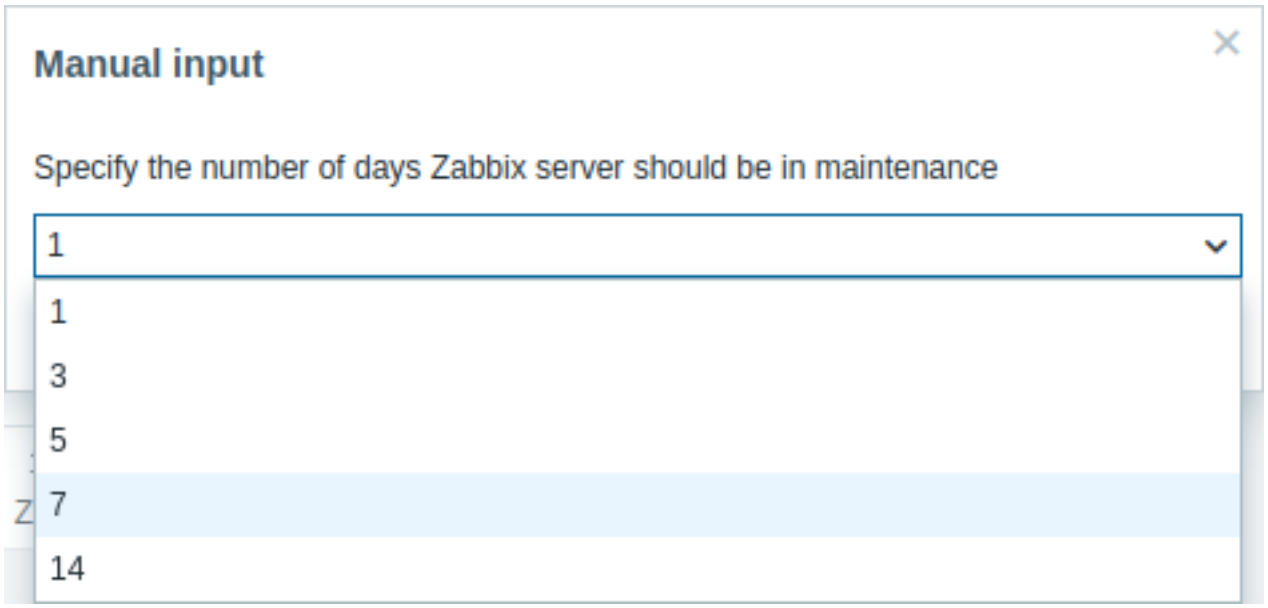
- paplašinātajā skripta konfigurācijā iespējojiet manuālu lietotāja ievadi un konfigurējiet ievades opcijas.

Ja lietotāja ievade ir iespējota, pirms skripta izpildes lietotājam tiks parādīts uznirstošais logs *Manual input*, kurā būs jānorāda pielāgota vērtība. Norādītā vērtība skriptā aizstās {MANUALINPUT}.

Atkarībā no konfigurācijas lietotājam tiks lūgts ievadīt virknes vērtību:



Vai atlasīt vērtību no nolaižamā saraksta ar iepriekš noteiktām opcijām:



Manuāla lietotāja ievade ir pieejama tikai skriptiem, kuru tvērums ir 'Manual host action' vai 'Manual event action'.

Skripta izpilde un rezultāts

Skripti, ko palaiž Zabbix serveris, tiek izpildīti secībā, kas aprakstīta komandu izpildes lapā.

Skripta rezultāts tiek parādīts uznirstošajā logā, kas parādās pēc skripta palaišanas. Skripta atgrieztā vērtība ir standarta izvade:

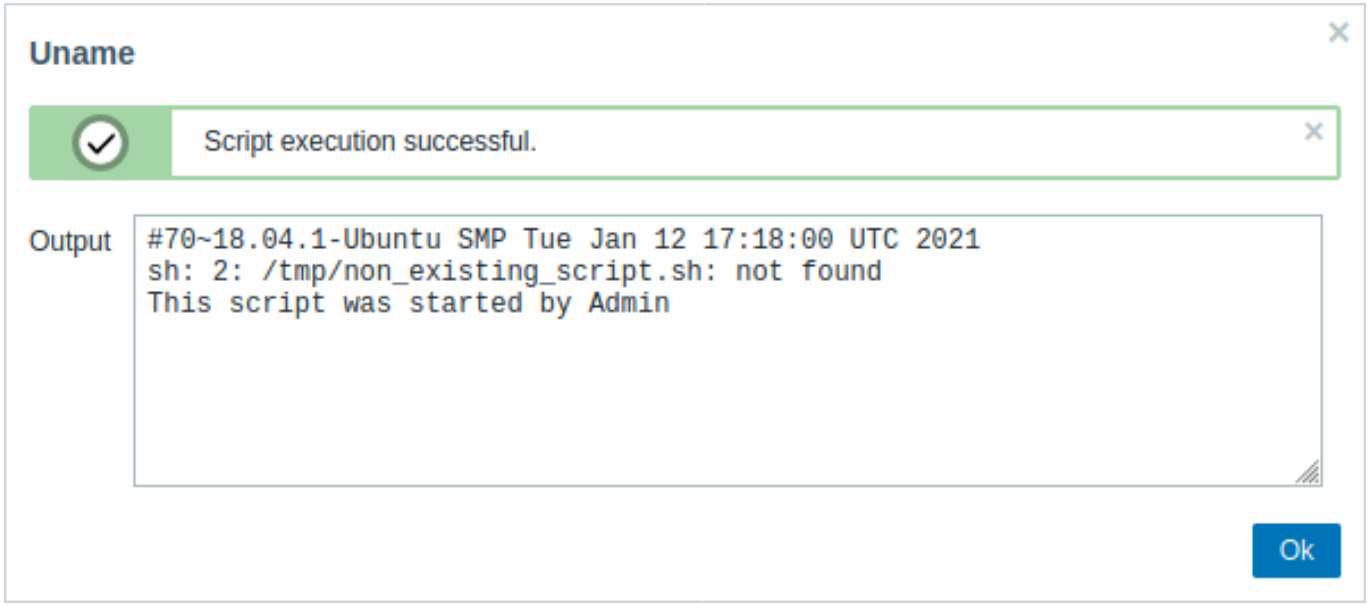
- Ja skripts beidzas veiksmīgi (izejas kods 0), atgrieztā vērtība ir ierobežota līdz 16MB (ieskaitot beigu atstarpes, kas tiek apgrieztas).
- Ja skripts beidzas ar kļūdu (izejas kods nav nulle), atgrieztā vērtība ir standarta kļūdu izvade, kas ierobežota līdz 2KB.

Pēc noklusējuma Zabbix nesaglabā paplašinātu skripta izvadi. Lai saglabātu pilnu izvades informāciju, varat ieviest reģistrēšanu pašā skriptā (piemēram, novirzīt izvadi uz lokālu žurnālfailu).

Ņemiet vērā, ka skriptiem, kas tiek izpildīti vai nu uz Zabbix servera, vai Zabbix starpniekservera, ir spēkā arī datu bāzes ierobežojumi.

Tālāk ir sniegts skripta un rezultātu loga piemērs:

```
uname -v
/tmp/non_existing_script.sh
echo "This script was started by {USER.USERNAME}"
```



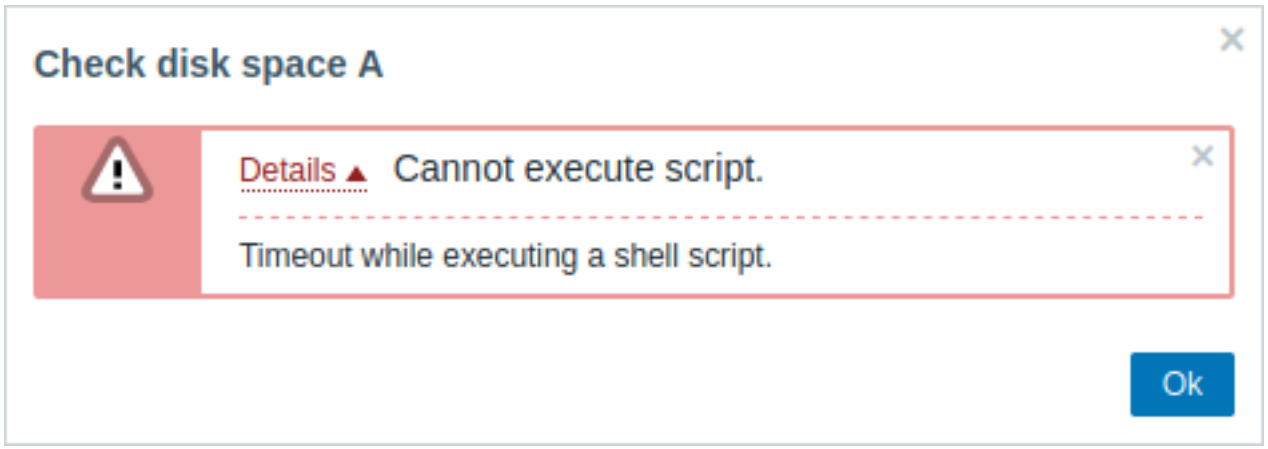
Skripta rezultātā netiek parādīts pats skripts.

Skripta noildze Zabbix aģentā

Var rasties situācija, kad skripta izpildes laikā iestājas noildze.

Tālāk skatiet piemēru skriptam, kas darbojas Zabbix aģentā, un rezultāta logu:

```
sleep 5
df -h
```



Kļūdas ziņojums šajā gadījumā ir šāds:

```
Timeout while executing a shell script.
```

Lai izvairītos no šādām situācijām, ieteicams optimizēt pašu skriptu (iepriekš minētajā piemērā — "5"), nevis pielāgot `Timeout` parametru Zabbix aģenta konfigurācijā un Zabbix servera konfigurācijā.

Ja `Timeout` parametrs ir mainīts Zabbix aģenta konfigurācijā, tiks parādīts šāds kļūdas ziņojums:

```
Get value from agent failed: ZBX_TCP_READ() timed out.
```

Tas nozīmē, ka izmaiņas ir veiktas Zabbix aģenta konfigurācijā, taču `Timeout` parametrs ir jāmaina arī Zabbix servera konfigurācijā.

Skripta noildze aktīvajiem aģentiem

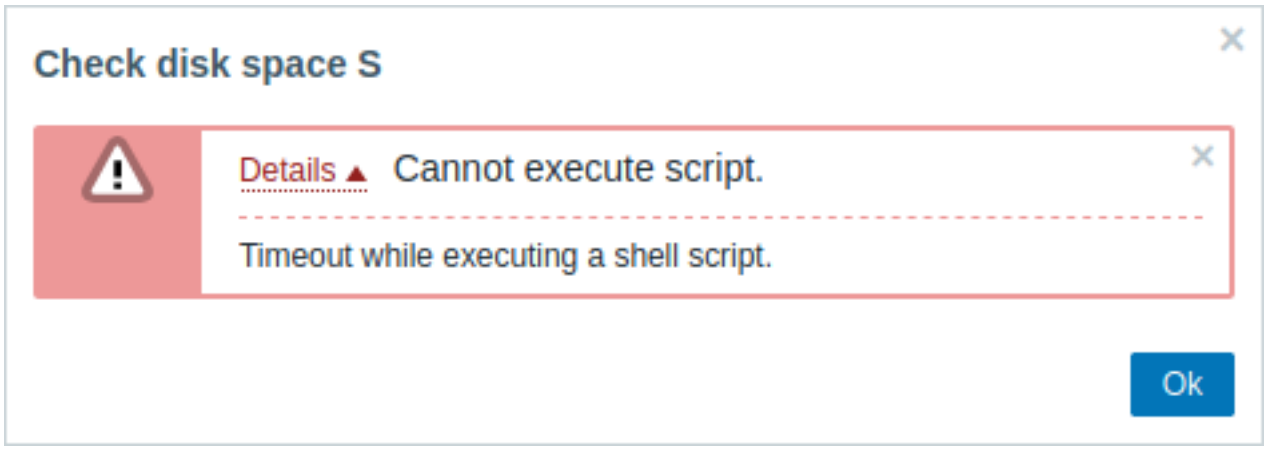
Ir iespējams izpildīt skriptus tikai uz aģentiem, kas darbojas aktīvajā režīmā. Kad skripta izpildi ierosina darbības operācija vai manuāla skripta izpilde, komanda tiek iekļauta aktīvās pārbaudes konfigurācijā un izpildīta, tiklīdz aktīvais aģents to saņem.

Manuālie skripti tiek nosūtīti aktīvajam aģentam kopā ar servera/starpniekservera skripta izpildes noildzi. Lūdzu, palieliniet noklusējuma serveris/starpniekserveris noildzi skripta izpildei. Noildzei jābūt lielākai par aktīvās pārbaudes atsvaidzināšanas biežumu, pretējā gadījumā noildze tiek pārsniegta, pirms aktīvais aģents saņem skriptu un spēj atgriezt rezultātu.

Skripta noildze uz Zabbix servera/starpniekservera

Skatiet piemēru skriptam, kas darbojas uz Zabbix servera, un rezultāta logu zemāk:

```
sleep 11
df -h
```



Ieteicams arī optimizēt pašu skriptu (nevis pielāgot `TrapperTimeout` parametru atbilstošai vērtībai (mūsu gadījumā, > 11), modificējot Zabbix servera konfigurāciju).

8 Lietotāji

Pārskats

Šajā izvēlnē ir sadaļas, kas saistītas ar lietotāju konfigurēšanu Zabbix. Šī izvēlne ir pieejama tikai SuperAdmin lietotāja tipa lietotājiem.

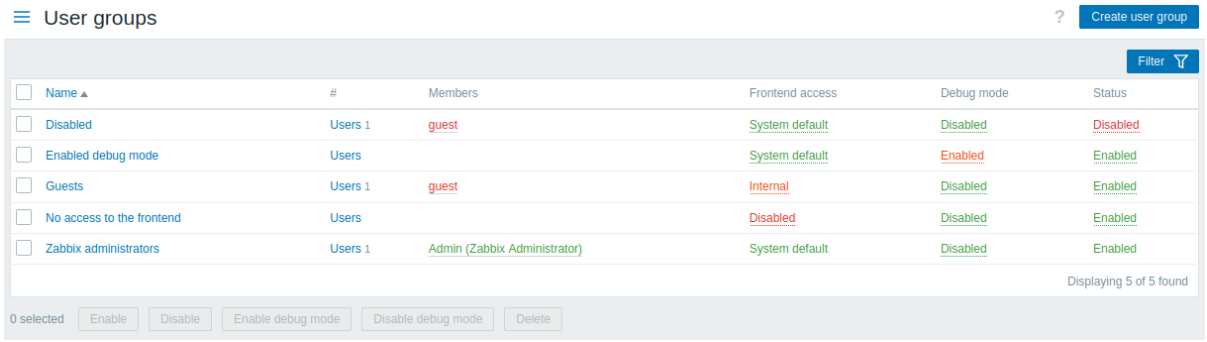
1 Lietotāju grupas

Pārskats

Sadaļā *Lietotāji* → *Lietotāju grupas* tiek pārvaldītas sistēmas lietotāju grupas.

Lietotāju grupas

Tiek parādīts esošo lietotāju grupu saraksts ar to detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Nosaukums	Lietotāju grupas nosaukums. Noklikšķinot uz lietotāju grupas nosaukuma, tiek atvērta lietotāju grupas konfigurācijas forma.
#	Lietotāju skaits grupā. Noklikšķinot uz *Lietotāji*, tiks parādīti attiecīgie lietotāji, atlasīti lietotāju sarakstā.
Dalībnieki	Atsevišķu lietotāju lietotājvārdi lietotāju grupā (ar vārdu un uzvārdu iekavās). Noklikšķinot uz lietotājvārda, tiks atvērta lietotāja konfigurācijas forma. Lietotāji no atspējotām grupām tiek parādīti sarkanā krāsā.

Kolonna	Apraksts
Piekļuve lietotāja saskarnei	Tiek parādīts piekļuves līmenis lietotāja saskarnei: **Sistēmas noklusējums** — lietotāji tiek autentificēti ar Zabbix, LDAP vai HTTP (atkarībā no globāli iestatītās autentifikācijas metodes); **Iekšēja** — lietotāji tiek autentificēti ar Zabbix; tiek ignorēta, ja HTTP autentifikācija ir globālais noklusējums; **LDAP** — lietotāji tiek autentificēti ar LDAP; tiek ignorēta, ja HTTP autentifikācija ir globālais noklusējums; **Atspējota** — piekļuve Zabbix lietotāja saskarnei šai grupai ir aizliegta. Noklikšķinot uz pašreizējā līmeņa, to var mainīt.
Atkļūdošanas režīms	Tiek parādīts atkļūdošanas režīma statuss — *Iespējots* vai *Atspējots*. Noklikšķinot uz statusa, to var mainīt.
Statuss	Tiek parādīts lietotāju grupas statuss — *Iespējota* vai *Atspējota*. Noklikšķinot uz statusa, to var mainīt.

Lai konfigurētu jaunu lietotāju grupu, noklikšķiniet uz pogas *Izveidot lietotāju grupu* augšējā labajā stūrī.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

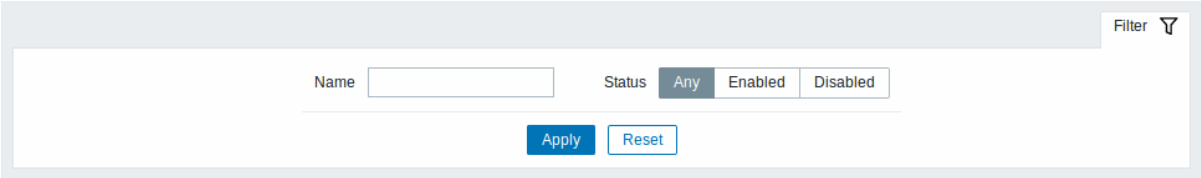
- *Iespējot* - mainīt lietotāju grupas statusu uz *Iespējots*
- *Atspējot* - mainīt lietotāju grupas statusu uz *Atspējots*
- *Iespējot atkļūdošanas režīmu* - iespējot atkļūdošanas režīmu lietotāju grupām
- *Atspējot atkļūdošanas režīmu* - atspējot atkļūdošanas režīmu lietotāju grupām
- *Dzēst* - dzēst lietotāju grupas

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām lietotāju grupām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Jūs varat izmantot filtru, lai parādītu tikai tās lietotāju grupas, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

Saite *Filtrs* ir pieejama virs lietotāju grupu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt lietotāju grupas pēc nosaukuma un statusa.



2 Lietotāju lomas

Pārskats

Sadaļā *Lietotāji* > *Lietotāju lomas* jūs varat izveidot lietotāju lomas.

Lietotāju lomas ļauj izveidot detalizētas atļaujas, pamatojoties uz sākotnēji izvēlēto lietotāja tipu (*Lietotājs*, *Administrators*, *Super administrators*).

Izvēloties lietotāja tipu, tiek piešķirtas visas šim lietotāja tipam pieejamās atļaujas (pēc noklusējuma atzīmētas).

Atļaujas var tikai atsaukt no kopas, kas ir pieejama attiecīgajam lietotāja tipam; tās nevar paplašināt ārpus tā, kas ir pieejams šim lietotāja tipam.

Izvēles rūtiņas nepieejamām atļaujām ir pelēkotas; lietotāji nevarēs piekļūt elementam pat tad, ja pārlūkprogrammā ievadīs tiešu šī elementa URL.

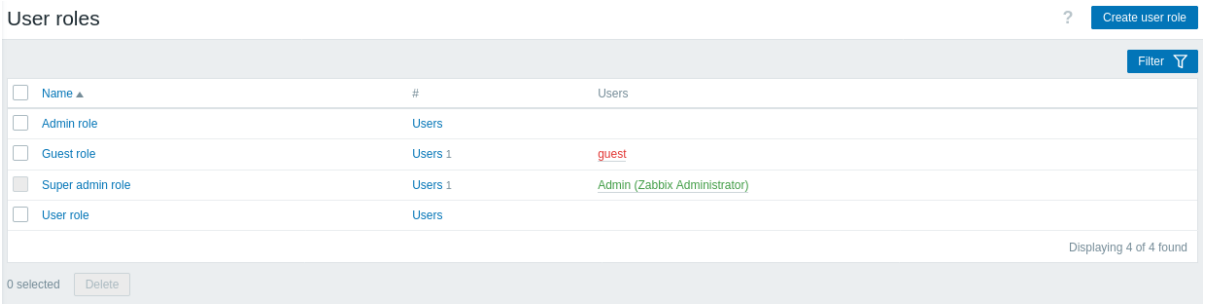
Note:
Piekļuves ierobežošana dažiem lietotāja saskarnes elementiem tikai novērš šīs lapas atvēršanu — tā neatceļ iespēju piekļūt pamatā esošajiem datiem citās saskarnes daļās.

Lietotāju lomas var piešķirt sistēmas lietotājiem. Katram lietotājam var būt piešķirta tikai viena loma.

Noklusējuma lietotāju lomas

Pēc noklusējuma Zabbix ir konfigurēts ar četrām lietotāju lomām, kurām ir iepriekš definēts atļauju kopums:

- Viesa loma
- Lietotāja loma
- Administratora loma
- Super administratora loma

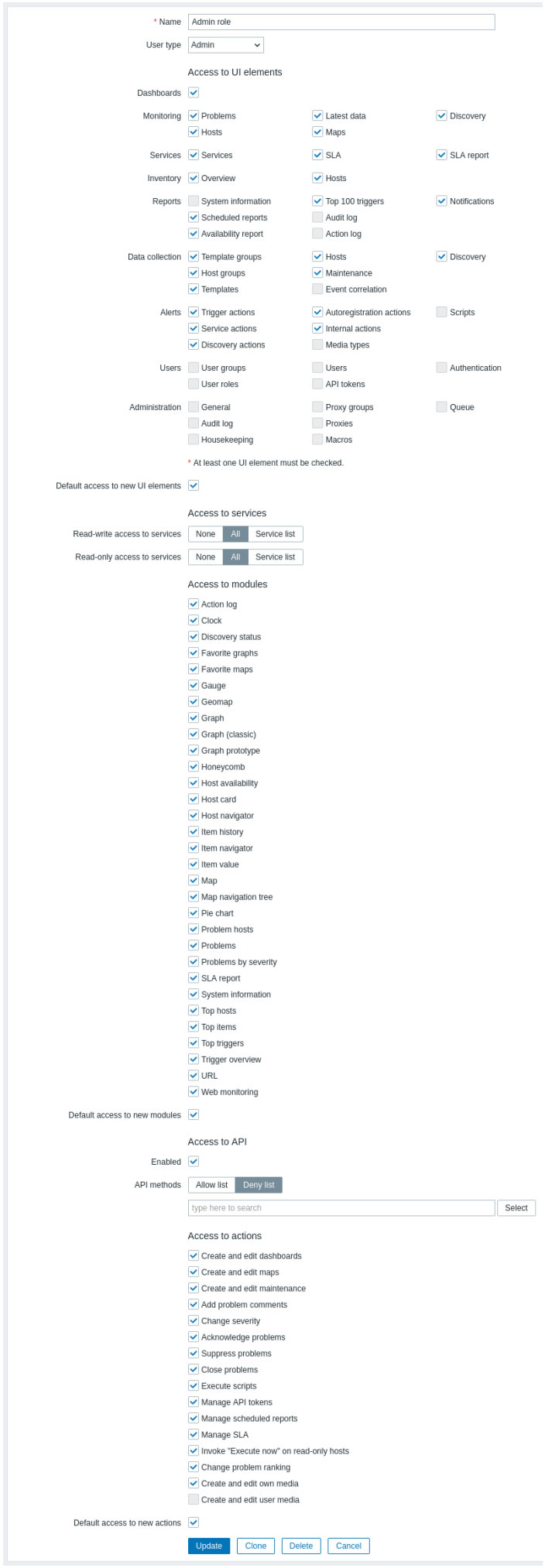


Tās ir balstītas uz galvenajiem lietotāju tipiem Zabbix. Tiek parādīts visu lietotāju saraksts, kuriem ir piešķirta attiecīgā loma. Lietotāji, kas iekļauti atspējotās grupās, ir norādīti sarkanā krāsā. *Viesa loma* ir lietotāja tipa loma, kurai vienīgās atļaujas ir skatīt dažas lietotāja saskarnes sadaļas.

Note:
Noklusējuma *Super administratora lomu* nevar mainīt vai dzēst, jo Zabbix ir jāpastāv vismaz vienam Super administratora lietotājam ar neierobežotām privilēģijām. *Super administratora* tipa lietotāji var mainīt savas lomas iestatījumus, bet ne lietotāja tipu.

Konfigurācija

Lai izveidotu jaunu lomu, augšējā labajā stūrī noklikšķiniet uz pogas *Izveidot lietotāja lomu*. Lai atjauninātu esošu lomu, noklikšķiniet uz lomas nosaukuma, lai atvērtu konfigurācijas formu.


* 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

Tiek parādītas pieejamās atļaujas. Lai atsauktu noteiktu atļauju, noņemiet atzīmi no tās izvēles rūtiņas.

Tālāk ir aprakstītas pieejamās atļaujas kopā ar noklusējuma vērtībām katrai iepriekš konfigurētajai lietotāja lomai Zabbix sistēmā.

Noklusējuma atļaujas

Piekļuve lietotāja saskarnes elementiem

Noklusējuma piekļuve izvēlnes sadaļām ir atkarīga no lietotāja tipa. Skatiet atļauju lapu, lai iegūtu sīkāku informāciju.

Piekļuve citām opcijām

Parametrs	**Apraksts**	**Noklusējuma lietotāju lomas**			
		Super admin loma	*Admin loma*	*Lietotāja loma*	*Viesa loma*
Noklusējuma piekļuve jauniem lietotāja saskarnes elementiem	Šī opcija nosaka, kā jaunās izvēlnes sadaļas būs pieejamas pēc Zabbix jauninašanas. Esošās moduļu izvēlnes sadaļas netiek ietekmētas.	Jā	Jā	Jā	Jā
Piekļuve pakalpojumiem					
Lasīšanas-rakstīšanas piekļuve pakalpojumiem	Atlasiet lasīšanas-rakstīšanas piekļuvi pakalpojumiem: **Nav** - nav piekļuves vispār **Visi** - piekļuve visiem pakalpojumiem ir lasīšanas-rakstīšanas režīmā **Pakalpojumu saraksts** - atlasiet pakalpojumus lasīšanas-rakstīšanas piekļuvei Ja lasīšanas-rakstīšanas piekļuve ir piešķirta, tai ir prioritāte pār tikai lasīšanas piekļuves iestatījumiem, un tā tiek dinamiski mantota bērnpakalpojumiem.	Visi	Visi	Nav	Nav
Lasīšanas-rakstīšanas piekļuve pakalpojumiem ar birku	Norādiet birkas nosaukumu un pēc izvēles arī vērtību, lai papildus piešķirtu lasīšanas-rakstīšanas piekļuvi pakalpojumiem, kas atbilst šai birkai. Šī opcija ir pieejama, ja parametrā *Lasīšanas-rakstīšanas piekļuve pakalpojumiem* ir atlasīts 'Pakalpojumu saraksts'. Ja lasīšanas-rakstīšanas piekļuve ir piešķirta, tai ir prioritāte pār tikai lasīšanas piekļuves iestatījumiem, un tā tiek dinamiski mantota bērnpakalpojumiem.				
Tikai lasīšanas piekļuve pakalpojumiem	Atlasiet tikai lasīšanas piekļuvi pakalpojumiem: **Nav** - nav piekļuves vispār **Visi** - piekļuve visiem pakalpojumiem ir tikai lasīšanas režīmā **Pakalpojumu saraksts** - atlasiet pakalpojumus tikai lasīšanas piekļuvei Tikai lasīšanas piekļuvei nav prioritātes pār lasīšanas-rakstīšanas piekļuvi, un tā tiek dinamiski mantota bērnpakalpojumiem.			Visi	Visi

Tikai lasīšanas piekļuve pakalpojumiem ar birku	Norādiet birkas nosaukumu un pēc izvēles arī vērtību, lai papildus piešķirtu tikai lasīšanas piekļuvi pakalpojumiem, kas atbilst šai birkai. Šī opcija ir pieejama, ja parametrā *Tikai lasīšanas piekļuve pakalpojumiem* ir atlasīts 'Pakalpojumu saraksts'. Tikai lasīšanas piekļuvei nav prioritātes pār lasīšanas-rakstīšanas piekļuvi, un tā tiek dinamiski mantota bērnpakalpojumiem.				
Piekļuve moduļiem					
<Moduļa nosaukums>	Atļaut/aizliegt piekļuvi konkrētam modulim. Šajā sadaļā tiek rādīti tikai iespējotie moduļi. Nav iespējams piešķirt vai ierobežot piekļuvi modulim, kas pašlaik ir atspējots.	Jā	Jā	Jā	Jā
Noklusējuma piekļuve jauniem moduļiem	Šī opcija nosaka, kā jauni moduļi un logrīki būs pieejami pēc Zabbix jauninašanas. Tā attiecas arī uz moduļiem un logrīkiem, kas pievienoti sadaļā *Administration > General > Modules*.				
Piekļuve API					
Iespējots API metodes	Iespējot/atspējot piekļuvi API. Atlasiet vai nu *Atļauto sarakstu*, lai atļautu, vai *Aizliegto sarakstu*, lai aizliegtu API metodes, kas norādītas meklēšanas laukā. Ņemiet vērā, ka nav iespējams atļaut dažas API metodes un citas aizliegt. Meklēšanas laukā sāciet rakstīt metodes nosaukumu, pēc tam atlasiet metodi no automātiskās pabeigšanas saraksta. Varat arī nospiest pogu Select un atlasīt metodes no pilnā saraksta, kas pieejams šim lietotāja tipam. Ņemiet vērā, ka, ja noteikta darbība blokā Piekļuve darbībām nav atzīmēta, lietotāji nevarēs izmantot ar šo darbību saistītās API metodes. Tiek atbalstītas aizstājējzīmes. Piemēri: `dashboard.*` (visas `dashboard.` API servisa metodes) `*` (jebkura metode), `*.export` (metodes ar nosaukumu `.export` no visiem API servisiem). Ja nav norādīta neviena metode, *Atļauto/Aizliegto saraksta* noteikums tiks ignorēts.	Jā	Jā	Jā	Nē
Piekļuve darbībām					
Izveidot un rediģēt paneļus	Noņemot šo izvēles rūtiņu, tiks atsauktas arī tiesības izmantot `.create`, `.update` un `.delete` API metodes atbilstošajiem elementiem.	Jā	Jā	Jā	Nē
Izveidot un rediģēt kartes					
Izveidot un rediģēt uzturēšanu				Nē	
Pievienot problēmu komentārus	Noņemot šo izvēles rūtiņu, tiks atsauktas arī tiesības veikt atbilstošo darbību, izmantojot `event.acknowledge` API metodi.			Jā	
Mainīt svarīguma pakāpi					
Apstiprināt problēmas					

Apspiest problēmas Aizvērt problēmas			
Izpildīt skriptus	Noņemot šo izvēles rūtiņu, tiks atsauktas arī tiesības izmantot `script.execute` API metodi.		
Pārvaldīt API marķierus	Noņemot šo izvēles rūtiņu, tiks atsauktas arī tiesības izmantot visas `token.` API metodes.		
Pārvaldīt plānotās atskaites	Noņemot šo izvēles rūtiņu, tiks atsauktas arī tiesības izmantot visas `report.` API metodes.		Nē
Pārvaldīt SLA	Iespējot/atspējot tiesības pārvaldīt SLA.		
Izsaukt ”Execute now” tikai lasāmiem hostiem	Atļaut izmantot opciju ”Execute now” jaunākajos datos tikai lasāmu hostu vienumiem.		Jā
Mainīt problēmas rangu	Atļaut mainīt problēmas rangu no cēloņa uz simptomu un otrādi.		
Izveidot un rediģēt savu mediju	Atļaut izveidot/rediģēt savu mediju.		
Izveidot un rediģēt lietotāju mediju	Atļaut izveidot/rediģēt lietotāju mediju. Šī opcija ir pieejama tikai Super admin lietotājiem.	Nē	Nē
Noklusējuma piekļuve jaunām darbībām	Šī opcija nosaka, kā jaunas darbības būs pieejamas pēc Zabbix jaunināšanas.	Jā	Jā

Skatiet arī:

- Lietotāja konfigurēšana

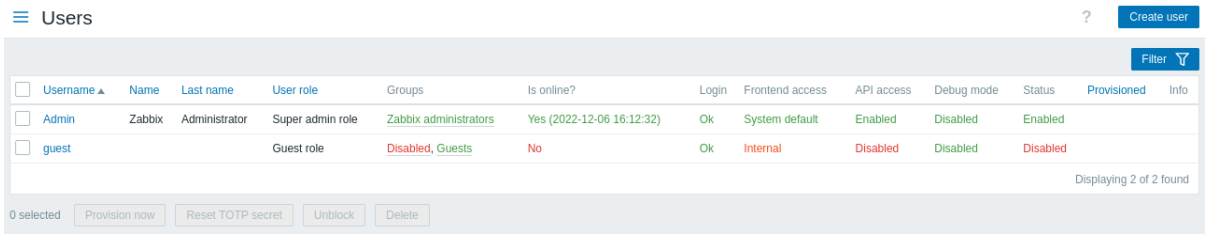
3 Lietotāji

Pārskats

Sadaļā *Lietotāji* → *Lietotāji* tiek pārvaldīti sistēmas lietotāji.

Lietotāji

Tiek parādīts esošo lietotāju saraksts ar detalizētu informāciju par tiem.



Parādītie dati:

Kolonna	Apraksts
Username	Lietotājvārds pieslēgšanās veikšanai Zabbix. Noklikšķinot uz lietotājvārda, tiek atvērta lietotāja konfigurācijas forma.
Name	Lietotāja vārds.
Last name	Lietotāja uzvārds.
User role	Tiek parādīta lietotāja loma.
Groups	Tiek uzskaitītas grupas, kuru dalībnieks ir lietotājs. Noklikšķinot uz lietotāju grupas nosaukuma, tiek atvērta lietotāju grupas konfigurācijas forma. Atspējotās grupas tiek parādītas sarkanā krāsā.
Is online?	Tiek parādīts lietotāja tiešsaistes statuss — *Yes* vai *No*. Iekavās tiek parādīts pēdējās lietotāja aktivitātes laiks.

Kolonna	Apraksts
Login	Tiek parādīts lietotāja pieteikšanās statuss — *Ok* vai *Blocked*. Lietotājs var tikt uz laiku bloķēts, ja tiek pārsniegts neveiksmīgu pieteikšanās mēģinājumu skaits, kas iestatīts sadaļā *Administration → General → Other* (pēc noklusējuma pieci). Noklikšķinot uz *Blocked*, varat atbloķēt lietotāju.
Frontend access	Tiek parādīts lietotāja saskarne piekļuves līmenis — *System default*, *Internal*, *LDAP* vai *Disabled* — atkarībā no tā, kas iestatīts visai lietotāju grupai.
API access	Tiek parādīts API piekļuves statuss — *Enabled* vai *Disabled* — atkarībā no tā, kas iestatīts lietotāja lomai.
Debug mode	Tiek parādīts atkļūdošanas režīma statuss — *Enabled* vai *Disabled* — atkarībā no tā, kas iestatīts visai lietotāju grupai.
Status	Tiek parādīts lietotāja statuss — *Enabled* vai *Disabled* — atkarībā no tā, kas iestatīts visai lietotāju grupai.
Provisioned	Tiek parādīts datums, kad lietotājs pēdējo reizi tika provisionēts. Izmanto lietotājiem, kas izveidoti ar JIT provisionēšanu no LDAP/SAML.
Info	Tiek parādīta informācija par kļūdām. Dzeltenais brīdinājums tiek parādīts lietotājiem bez lietotāju grupām. Sarkanais brīdinājums tiek parādīts lietotājiem bez lomām, kā arī lietotājiem bez lomām un lietotāju grupām.

Lai konfigurētu jaunu lietotāju, augšējā labajā stūrī noklikšķiniet uz pogas *Create user*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā vairākas masveida rediģēšanas opcijas:

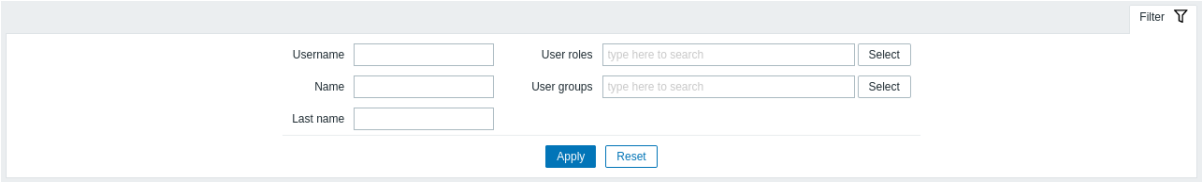
- *Provision now* — atjaunināt lietotāja informāciju no LDAP (šī opcija ir iespējota tikai tad, ja ir atlasīts LDAP lietotājs)
- *Reset TOTP secret* — atiestatīt lietotāja TOTP noslēpumus visām TOTP metodēm un dzēst lietotāja sesiju (šī opcija ir iespējota tikai tad, ja ir iespējots MFA; lietotājiem bez TOTP noslēpumiem viņu sesija netiks dzēsta)
- *Unblock* — atkārtoti iespējot sistēmas piekļuvi bloķētiem lietotājiem
- *Delete* — dzēst lietotājus

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem lietotājiem, pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Varat izmantot filtru, lai parādītu tikai tos lietotājus, kas jūs interesē. Lai nodrošinātu labāku meklēšanas veiktspēju, dati tiek meklēti ar neatrisinātiem makrosiem.

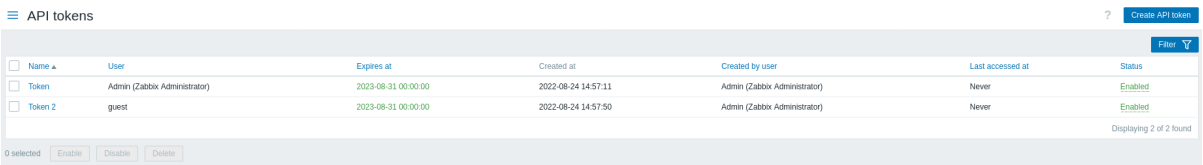
Saite *Filtrs* ir pieejama virs lietotāju saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt lietotājus pēc lietotājvārda, vārda, uzvārda, lietotāja lomas un lietotāju grupas.



4 API pilnvaru marķieri

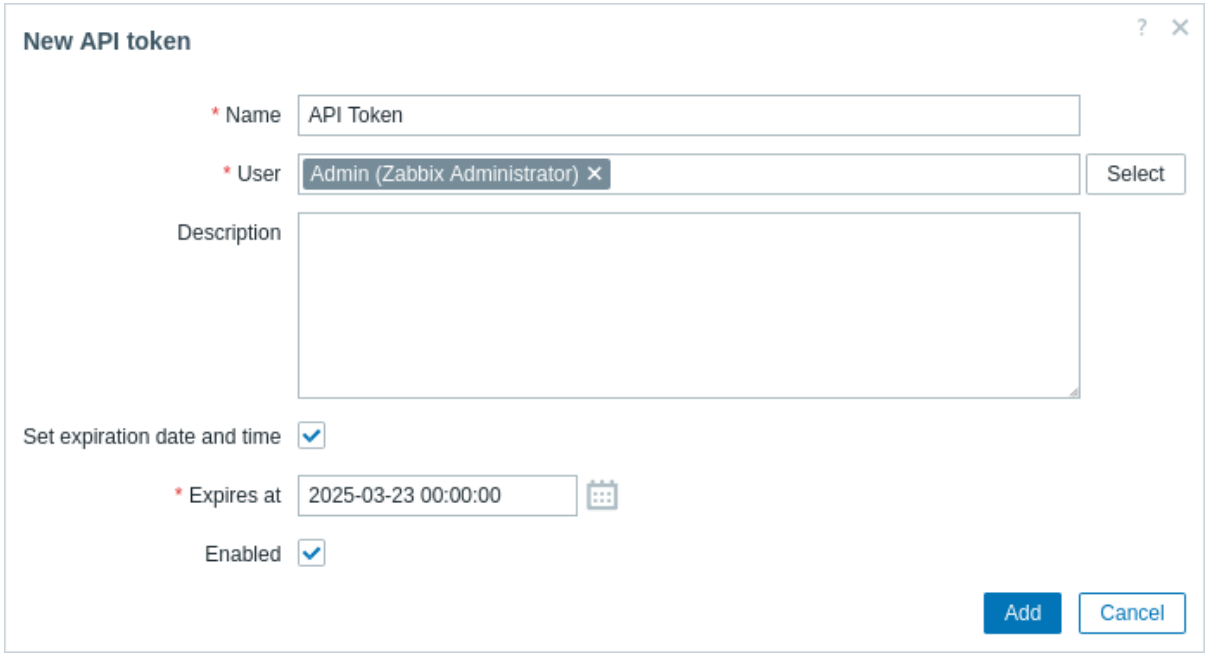
Pārskats

Šajā sadaļā var izveidot un pārvaldīt API marķierus.



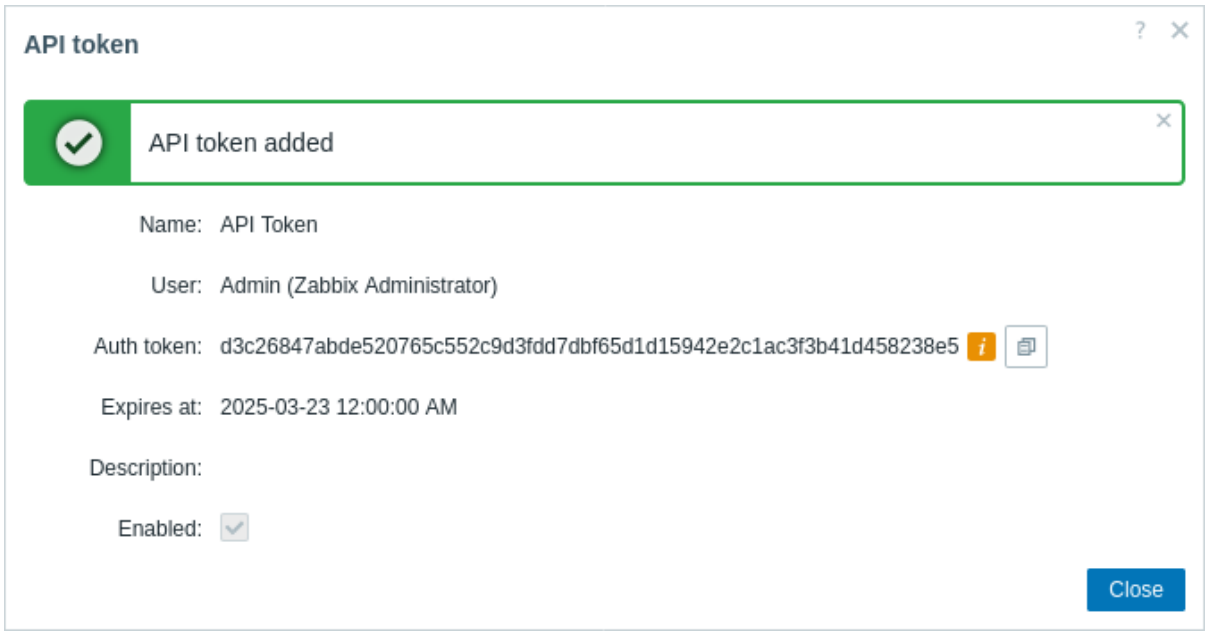
API marķierus var filtrēt pēc nosaukuma, lietotājiem, kuriem marķieri ir piešķirti, derīguma termiņa, lietotājiem, kuri izveidojuši marķierus, vai statusa (iespējots/atspējots). Noklikšķiniet uz marķiera statusa sarakstā, lai ātri iespējotu/atspējotu marķieri. Varat arī masveidā iespējot/atspējot marķierus, atlasot tos sarakstā un pēc tam noklikšķinot uz pogām *Enable/Disable* zem saraksta.

Lai izveidotu jaunu marķieri, nospiediet pogu *Create API token* augšējā labajā stūrī, pēc tam aizpildiet obligātos laukus marķiera konfigurācijas ekrānā:



Parametrs	Apraksts
Name	Marķiera redzamais nosaukums.
User	Lietotājs, kuram marķieris jāpiešķir. Lai ātri atlasītu lietotāju, sāciet rakstīt lietotājvārdu, vārdu vai uzvārdu, pēc tam izvēlieties vajadzīgo lietotāju no automātiskās pabeigšanas saraksta. Varat arī nospiest pogu *Select* un izvēlēties lietotāju no pilnā lietotāju saraksta. Marķieri var piešķirt tikai vienam lietotājam.
Description	Neobligāts marķiera apraksts.
Set expiration date and time	Noņemiet atzīmi no šīs izvēles rūtiņas, ja marķierim nevajadzētu būt derīguma termiņam.
Expiry date	Noklikšķiniet uz kalendāra ikonas, lai atlasītu marķiera derīguma termiņa datumu, vai ievadiet datumu manuāli formātā YYYY-MM-DD hh:mm:ss
Enabled	Noņemiet atzīmi no šīs izvēles rūtiņas, ja marķieris jāizveido atspējotā stāvoklī.

Nospiediet *Add*, lai izveidotu marķieri.



Nokopējiet *Auth token* vērtību un saglabājiet to drošā vietā **pirms lapas aizvēršanas**, pēc tam nospiediet Close. Marķieris parādīsies sarakstā.

Warning:
Auth token vērtību vēlāk vairs nevarēs apskatīt. Tā ir pieejama tikai uzreiz pēc marķiera izveides. Ja pazaudēsiet saglabāto marķieri, tas būs jāģenerē no jauna, un šīs darbības rezultātā tiks izveidota jauna autorizācijas virkne.

Noklikšķiniet uz marķiera nosaukuma, lai rediģētu nosaukumu, aprakstu, derīguma termiņa iestatījumus vai marķiera statusu. Ņemiet vērā, ka nav iespējams mainīt lietotāju, kuram marķieris ir piešķirts. Nospiediet pogu *Update*, lai saglabātu izmaiņas. Ja marķieris ir pazaudēts vai kļuvis pieejams nepiederošām personām, varat nospiest pogu *Regenerate*, lai ģenerētu jaunu marķiera vērtību. Tiks parādīts apstiprinājuma dialoglodziņš ar lūgumu apstiprināt šo darbību, jo pēc turpināšanas iepriekš ģenerētais marķieris kļūs nederīgs.

Lietotāji bez piekļuves izvēlnes sadaļai *Administration* var skatīt un mainīt sev piešķirto marķieru informāciju tikai sadaļā *User profile* → *API tokens* section, ja viņu lietotāja lomā ir atļauta atļauja *Manage API tokens*.

5 Autentifikācija

Pārskats

Sadaļa *Lietotāji* → *Autentifikācija* ļauj norādīt lietotāju autentifikācijas metodi Zabbix sistēmai un iekšējās paroles prasības.

Pieejamās autentifikācijas metodes ir iekšējā, HTTP, LDAP, SAML un MFA autentifikācija.

Noklusējuma autentifikācija

Pēc noklusējuma Zabbix visiem lietotājiem izmanto **iekšējo** Zabbix autentifikāciju.

Ir iespējams mainīt noklusējuma autentifikācijas metodi uz LDAP visā sistēmā. Lai to izdarītu, atveriet cilni *LDAP* un konfigurējiet LDAP parametrus, pēc tam atgriezieties cilnē *Authentication* un pārslēdziet atlasītāju *Default authentication* uz LDAP.

Ņemiet vērā, ka autentifikācijas metodi var precīzi pielāgot lietotāju grupas līmenī. Pat ja LDAP autentifikācija ir iestatīta globāli, dažas lietotāju grupas joprojām var tikt autentificētas ar Zabbix. Šīm grupām lietotāja saskarnes piekļuvei jābūt iestatītai uz Internal.

Ir iespējams arī iespējot LDAP autentifikāciju tikai noteiktām lietotāju grupām, ja globāli tiek izmantota iekšējā autentifikācija. Šajā gadījumā LDAP autentifikācijas datus var norādīt un izmantot konkrētām lietotāju grupām, kuru lietotāja saskarnes piekļuvei pēc tam jābūt iestatītai uz LDAP. Ja lietotājs ir iekļauts vismaz vienā lietotāju grupā ar LDAP autentifikāciju, šis lietotājs nevarēs izmantot iekšējo autentifikācijas metodi.

HTTP, SAML 2.0 un MFA autentifikācijas metodes var izmantot papildus noklusējuma autentifikācijas metodei.

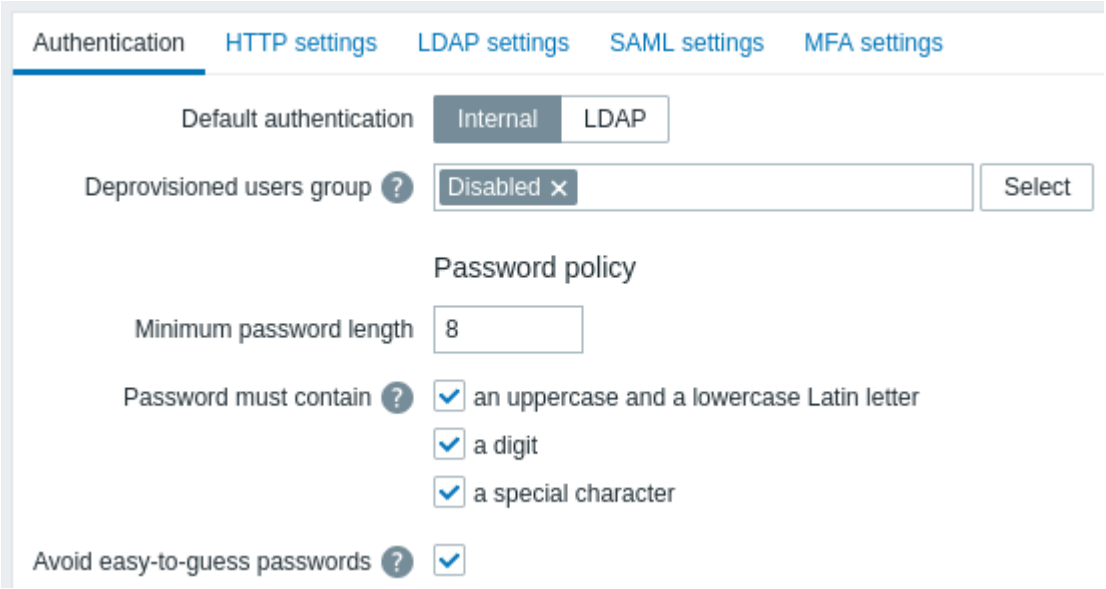
Zabbix atbalsta tieši laikā (JIT) nodrošināšanu, kas ļauj izveidot lietotāju kontus Zabbix brīdī, kad ārējs lietotājs autentificējas pirmo reizi, un nodrošināt šos lietotāju kontus. JIT nodrošināšana tiek atbalstīta LDAP un SAML.

Skatiet arī:

- HTTP autentifikācija
- LDAP autentifikācija
- SAML autentifikācija
- MFA autentifikācija

Konfigurācija

Cilne *Autentifikācija* ļauj iestatīt noklusējuma autentifikācijas metodi, norādīt grupu deprovisionētiem lietotājiem un iestatīt paroles sarežģītības prasības Zabbix lietotājiem.



Konfigurācijas parametri:

Parametrs	Apraksts
Noklusējuma autentifikācija	Izvēlieties noklusējuma autentifikācijas metodi Zabbix sistēmā — *Iekšējā* vai *LDAP*.
Deprovisionēto lietotāju grupa	Norādiet lietotāju grupu deprovisionētiem lietotājiem. Šis iestatījums ir nepieciešams tikai JIT provisionēšanai attiecībā uz lietotājiem, kas tika izveidoti Zabbix no LDAP vai SAML sistēmām, bet kuri vairs nav jāprovisionē. Jānorāda atspējota lietotāju grupa.
Minimālais paroles garums	Pēc noklusējuma minimālais paroles garums ir iestatīts uz 8. Atbalstītais diapazons: 1-70. Ņemiet vērā, ka paroles, kas garākas par 72 rakstzīmēm, tiks saīsinātas.
Parolei jāsatur	Atzīmējiet vienu vai vairākas izvēles rūtiņas, lai pieprasītu noteiktu rakstzīmju izmantošanu parolē: - lielais un mazais latīņu alfabēta burts - cipars - speciālā rakstzīme Novietojiet kursoru virs jautājuma zīmes, lai redzētu paskaidrojumu ar katras opcijas rakstzīmju sarakstu.
Izvairīties no viegli uzminamām parolēm	Ja atzīmēts, parole tiks pārbaudīta atbilstoši šādām prasībām: - tā nedrīkst saturēt lietotāja vārdu, uzvārdu vai lietotājvārdu - tā nedrīkst būt viena no bieži lietotajām vai kontekstam specifiskajām parolēm. Bieži lietoto un kontekstam specifisko paroļu saraksts tiek ģenerēts automātiski no NCSC "Top 100k passwords" saraksta, SecLists "Top 1M passwords" saraksta un Zabbix kontekstam specifisko paroļu saraksta. Iekšējiem lietotājiem nebūs atļauts iestatīt paroles, kas ir iekļautas šajā sarakstā, jo šādas paroles tiek uzskatītas par vājām to plašās lietošanas dēļ.

Izmaiņas paroles sarežģītības prasībās neietekmēs esošās lietotāju paroles, taču, ja esošs lietotājs izvēlēsies mainīt paroli, jaunajai parolei būs jāatbilst pašreizējām prasībām. Paskaidrojums ar prasību sarakstu tiks parādīts blakus laukam *Parole* user profile un lietotāja konfigurācijas veidlapā, kas pieejama izvēlnē *Users* → *Users*.

1 HTTP

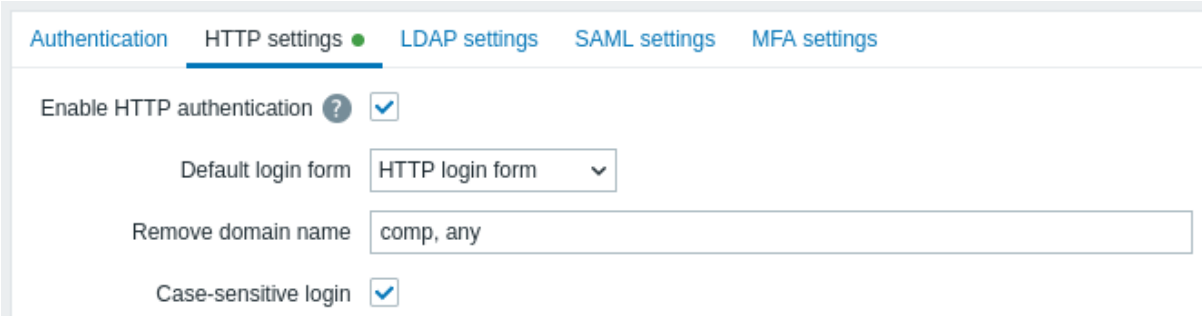
Pārskats

Uz HTTP vai tīmekļa serveri balstītu autentifikāciju (piemēram: BasicAuthentication, NTLM/Kerberos) var izmantot, lai pārbaudītu lietotājvārdus un paroles. Ņemiet vērā, ka lietotājam ir jāeksistē arī Zabbix, taču viņa Zabbix parole netiks izmantota.

Attention:
Esiet uzmanīgi! Pirms tās ieslēgšanas pārliecinieties, ka tīmekļa servera autentifikācija ir konfigurēta un darbojas pareizi.

HTTP autentifikāciju var atspējot lietotāja saskarnē, konfigurējot atbilstošo opciju cilnē *HTTP settings* sadaļā *Users > Authentication*. Ja HTTP autentifikācija ir atspējota, cilne ar HTTP autentifikācijas opcijām lietotāja saskarnē netiks parādīta. Ņemiet vērā, ka lietotāja saskarnes pārinstalēšana (palaižot setup.php) atiestatīs autentifikācijas iestatījumus, tostarp HTTP autentifikācijas konfigurāciju.

Konfigurācija



Konfigurācijas parametri:

Parametrs	Apraksts
Iespējot HTTP autentifikāciju	Atzīmējiet izvēles rūtiņu, lai iespējotu HTTP autentifikāciju. Novietojot peles kursoru virs , tiks parādīts paskaidrojuma lodziņš ar brīdinājumu, ka tīmekļa servera autentifikācijas gadījumā visi lietotāji (pat ar lietotāja saskarnes piekļuvi, kas iestatīta uz LDAP/Internal) tiks autentificēti tīmekļa serverī, nevis Zabbix.
Noklusējuma pieteikšanās forma	Norādiet, vai neautentificētus lietotājus novirzīt uz: **Zabbix pieteikšanās formu** - standarta Zabbix pieteikšanās lapu. **HTTP pieteikšanās formu** - HTTP pieteikšanās lapu. Ieteicams iespējot uz tīmekļa serveri balstītu autentifikāciju tikai lapai `index_http.php`. Ja *Noklusējuma pieteikšanās forma* ir iestatīta uz 'HTTP pieteikšanās lapa', lietotājs tiks pieteikts automātiski, ja tīmekļa servera autentifikācijas modulis iestatīs derīgu lietotāja pieteikšanās vārdu mainīgajā `$_SERVER`. Atbalstītās `$_SERVER` atslēgas ir `PHP_AUTH_USER`, `REMOTE_USER`, `AUTH_USER`.
Noņemt domēna nosaukumu	Ar komatiem atdalīts domēnu nosaukumu saraksts, kas jānoņem no lietotājvārda. Piemēram, `comp,any` - ja lietotājvārds ir 'Admin@any' vai 'comp\Admin', lietotājs tiks pieteikts kā 'Admin'; ja lietotājvārds ir 'notacompany\Admin', pieteikšanās tiks liegta.
Reģistrjutīga pieteikšanās	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu reģistrjutīgu pieteikšanos lietotājvārdiem (pēc noklusējuma iespējota). Atspējojot reģistrjutīgu pieteikšanos, ir iespējams, piemēram, pieteikties kā "admin", pat ja Zabbix lietotājs ir "Admin" vai "ADMIN". Lūdzu, ņemiet vērā, ka, ja reģistrjutīga pieteikšanās ir atspējota un pastāv vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), šo lietotāju pieteikšanās vienmēr tiks liegta ar šādu kļūdas ziņojumu: "Autentifikācija neizdevās: norādītie akreditācijas dati nav unikāli."

Note:
Iekšējiem lietotājiem, kuri nevar pieteikties, izmantojot HTTP akreditācijas datus (ja HTTP pieteikšanās forma ir iestatīta kā noklusējuma), kā rezultātā tiek saņemta 401 kļūda, varat pievienot rindu `ErrorDocument 401 /index.php?form=default` pamata autentifikācijas direktīvām, kas novirzīs uz parasto Zabbix pieteikšanās formu.

2 LDAP

Pārskats

Ārējo LDAP autentifikāciju var izmantot, lai pārbaudītu lietotājvārdus un paroles.

Zabbix LDAP autentifikācija darbojas vismaz ar Microsoft Active Directory un OpenLDAP.

Ja ir konfigurēta tikai LDAP pierakstīšanās, tad lietotājam ir jāeksistē arī Zabbix, taču tā Zabbix parole netiks izmantota. Ja autentifikācija ir veiksmīga, tad Zabbix salīdzinās lokālu lietotājvārdu ar lietotājvārda atribūtu, ko atgriež LDAP.

Lietotāju nodrošināšana

Ir iespējams konfigurēt JIT (just-in-time) **lietotāju nodrošināšanu** LDAP lietotājiem. Šajā gadījumā nav nepieciešams, lai lietotājs jau eksistētu Zabbix. Lietotāja kontu var izveidot, kad lietotājs pirmo reizi piesakās Zabbix.

Kad LDAP lietotājs ievada savu LDAP pieteikšanās vārdu un paroli, Zabbix pārbauda, vai šis lietotājs eksistē *noklusējuma* LDAP serverī. Ja lietotājs eksistē un viņam vēl nav konta Zabbix, Zabbix tiek izveidots jauns lietotājs, un lietotājs var pieteikties.

Lietotājs, kas izveidots, izmantojot JIT nodrošināšanu, tiek saistīts ar LDAP serveri (direktoriju), kas izveides brīdī ir iestatīts kā noklusējuma serveris. Vēlāka noklusējuma LDAP servera maiņa nemaina un neatjaunina LDAP serveri, kas ir saistīts ar lietotājiem, kuri jau tika nodrošināti.

Attention:
Ja JIT nodrošināšana ir iespējota, cilnē *Authentication* ir jānorāda lietotāju grupa atnodrošinātajiem lietotājiem.

JIT nodrošināšana arī ļauj atjaunināt nodrošināto lietotāju kontus, pamatojoties uz izmaiņām LDAP. Piemēram, ja lietotājs tiek pārvietots no vienas LDAP grupas uz citu, lietotājs tiks pārvietots arī no vienas grupas uz citu Zabbix; ja lietotājs tiek noņemts no LDAP grupas, lietotājs tiks noņemts arī no grupas Zabbix un, ja tas nepieder nevienai citai grupai, pievienots atnodrošināto lietotāju grupai. Nodrošināto lietotāju konti tiek atjaunināti, pamatojoties uz konfigurēto nodrošināšanas periodu vai kad lietotājs piesakās Zabbix.

Ņemiet vērā, ka fona nodrošināšanu veic Zabbix lietotāja saskarne, kad lietotājs ar to mijiedarbojas vai vismaz pārlūkprogrammā ir atvērta lietotāja saskarnes lapa. Nav īpašu fona procesu lietotāju nodrošināšanai.

LDAP atbalsta trīs veidus, kā piesaistīties direktorijam autentifikācijai un meklēšanai:

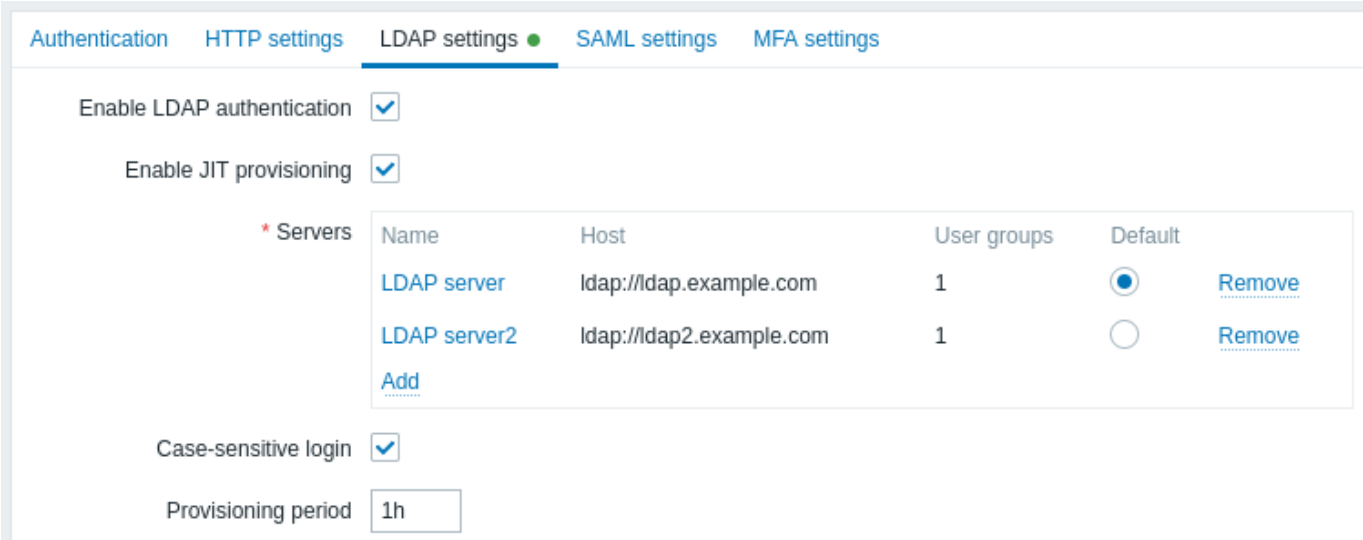
- Anonīma piesaiste — netiek norādīts *Bind DN* / *Bind password*, un LDAP serveris atļauj anonīmus vaicājumus.
- Īpašs piesaistes lietotājs (servisa konts) — konkrēts LDAP konts tiek iestatīts laukos *Bind DN* / *Bind password* un Zabbix to izmanto meklēšanai un nodrošināšanai. Šī ir ieteicamā un elastīgākā iespēja, jo Zabbix var veikt meklēšanu un fona nodrošināšanu bez galalietotāja akreditācijas datiem.
- Tieša lietotāja piesaiste — Zabbix veic piesaisti, izmantojot akreditācijas datus, ko lietotājs ievada pieteikšanās laikā (nav konfigurēts *Bind DN* / *Bind password*); to konfigurē, Base DN iekļaujot vietturi, piemēram, `uid=%{user}`. Šajā režīmā Zabbix piekļūst lietotāja parolei tikai interaktīvās pieteikšanās laikā. Tādēļ nodrošināšanas darbības, kurām nepieciešama autentificēšanās LDAP ārpus lietotāja pieteikšanās sesijas (piemēram, izmantojot pogu *Provision now* vai veicot fona nodrošināšanu, kad lietotājs nav aktīvi pieteicies), nevar autentificēties un tādēļ nedarbosies. Tiešās lietotāja piesaistes gadījumā nodrošināšana un atjauninājumi notiek tikai brīdī, kad lietotājs piesakās.

Vairāki serveri

Ja nepieciešams, var definēt vairākus LDAP serverus. Piemēram, citu serveri var izmantot citas lietotāju grupas autentificēšanai. Kad LDAP serveri ir konfigurēti, lietotāju grupas konfigurācijā kļūst iespējams atlasīt attiecīgajai lietotāju grupai nepieciešamo LDAP serveri.

Ja lietotājs ir vairākās lietotāju grupās un vairākos LDAP serveros, autentificēšanai tiks izmantots pirmais serveris LDAP serveru sarakstā, kas sakārtots pēc nosaukuma augošā secībā.

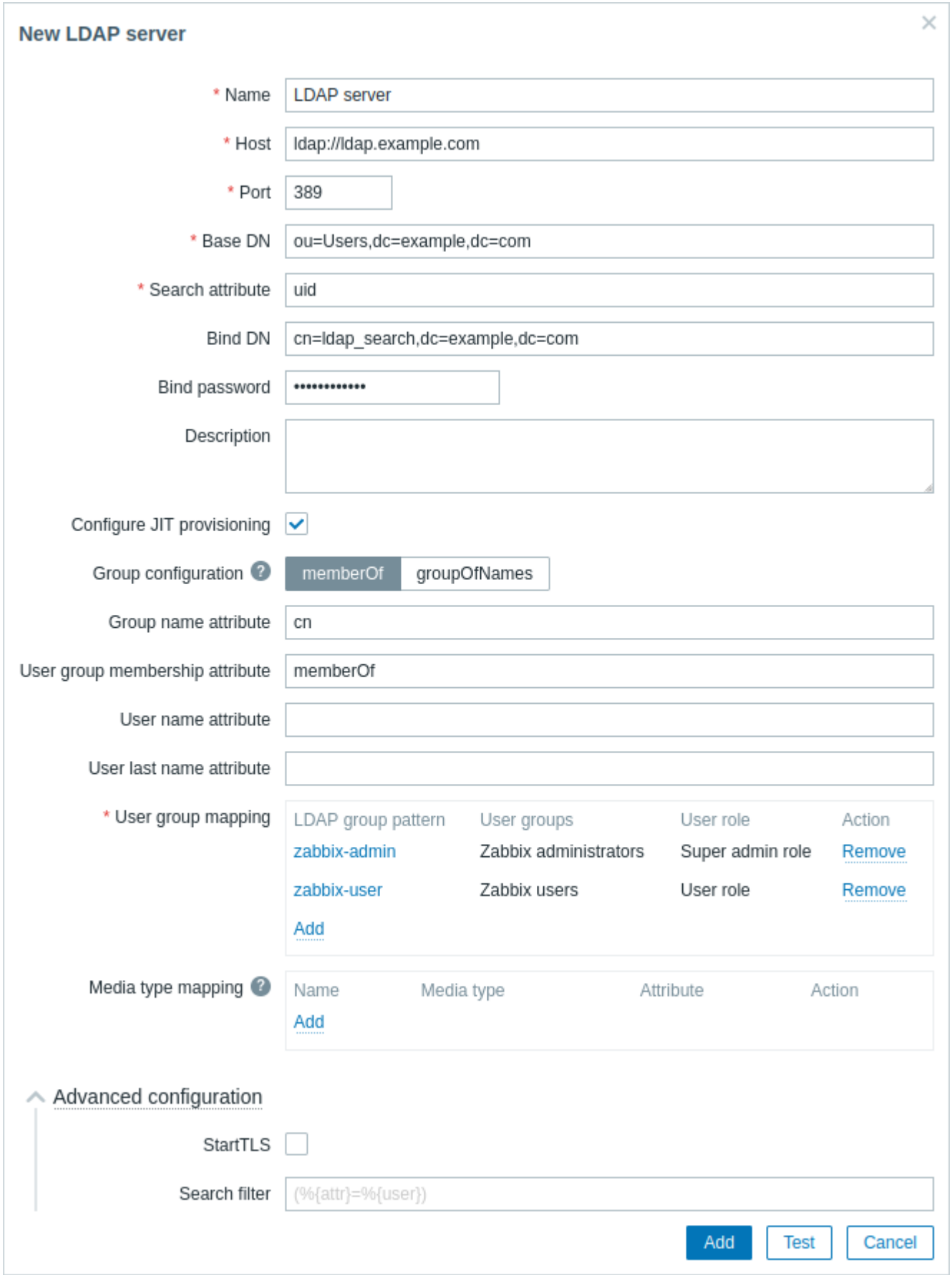
Konfigurācija



Konfigurācijas parametri:

Parametrs	Apraksts
Iespējot LDAP autentifikāciju	Atzīmējiet izvēles rūtiņu, lai iespējotu LDAP autentifikāciju.
Iespējot JIT nodrošināšanu	Atzīmējiet izvēles rūtiņu, lai iespējotu JIT nodrošināšanu.
Serveri	Noklikšķiniet uz *Pievienot*, lai konfigurētu LDAP serveri (skatiet LDAP servera konfigurācija zemāk).
Reģistrjutīgs pieteikšanās vārds	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu reģistrjutīgu pieteikšanos lietotājvārdiem (pēc noklusējuma iespējots). Atspējojot reģistrjutīgu pieteikšanos, ir iespējams, piemēram, pieteikties kā ”admin”, pat ja Zabbix lietotājs ir ”Admin” vai ”ADMIN”. Lūdzu, ņemiet vērā, ka, ja reģistrjutīga pieteikšanās ir atspējota un pastāv vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), pieteikšanās šiem lietotājiem vienmēr tiks liegta ar šādu kļūdas ziņojumu: ”Autentifikācija neizdevās: norādītie akreditācijas dati nav unikāli.”
Nodrošināšanas periods	Iestatiet nodrošināšanas periodu, t.i., cik bieži pieteicies lietotājs tiks nodrošināts, strādājot ar lietotāja saskarne.

LDAP servera konfigurācija



LDAP servera konfigurācijas parametri:

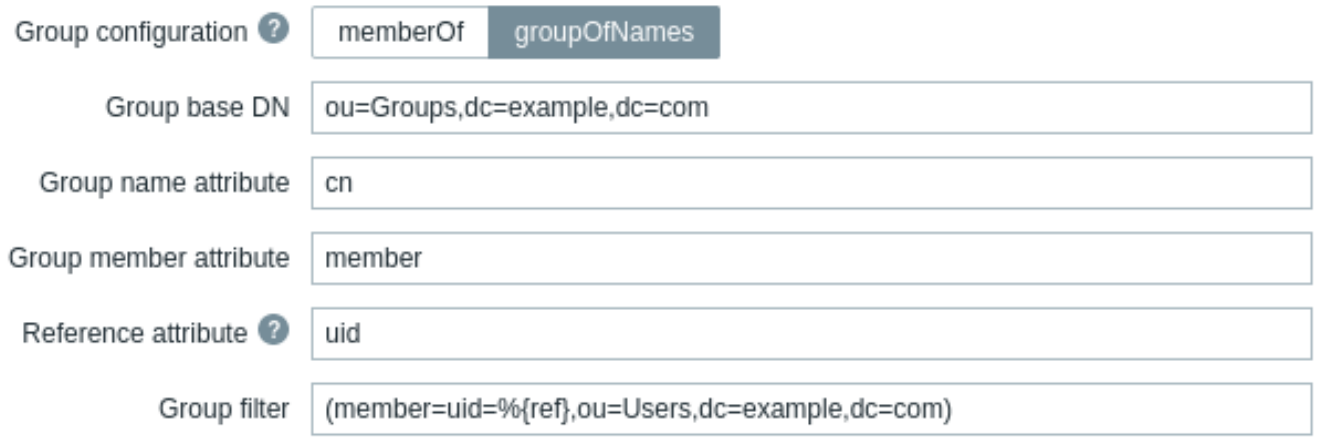
Parametrs	Apraksts
Name	LDAP servera nosaukums Zabbix konfigurācijā.

Parametrs	Apraksts
Host	LDAP servera resursdatora nosaukums, IP vai URI. Piemēri: ldap.example.com, 127.0.0.1, ldap://ldap.example.com Drošam LDAP serverim izmantojiet *ldaps* protokolu un resursdatora nosaukumu. Piemērs: ldaps://ldap.example.com Ar OpenLDAP 2.x.x un jaunākām versijām var izmantot pilnu LDAP URI formā ldap://hostname:port vai ldaps://hostname:port.
Port	LDAP servera ports. Noklusējuma vērtība ir 389. Drošam LDAP savienojumam porta numurs parasti ir 636. Netiek izmantots, ja tiek lietoti pilni LDAP URI.
Base DN	Bāzes ceļš uz lietotāju kontiem LDAP serverī: ou=Users,ou=system (OpenLDAP gadījumā), DC=company,DC=com (Microsoft Active Directory gadījumā) uid=%{user},dc=example,dc=com (tiešai lietotāja piesaistei, skatiet piezīmi zemāk)
Search attribute	LDAP konta atribūts, ko izmanto meklēšanai: uid (OpenLDAP gadījumā), sAMAccountName (Microsoft Active Directory gadījumā)
Bind DN	LDAP konts piesaistei un meklēšanai LDAP serverī, piemēri: uid=ldap_search,ou=system (OpenLDAP gadījumā), CN=ldap_search,OU=user_group,DC=company,DC=com (Microsoft Active Directory gadījumā) Tiek atbalstīta arī anonīma piesaiste. Ņemiet vērā, ka anonīma piesaiste var potenciāli atvērt domēna konfigurāciju neautorizētiem lietotājiem (informācija par lietotājiem, datoriem, serveriem, grupām, pakalpojumiem utt.). Drošības apsvērumu dēļ atspējojiet anonīmās piesaistes LDAP hostos un tā vietā izmantojiet autentificētu piekļuvi.
Bind password	LDAP parole kontam, ko izmanto piesaistei un meklēšanai LDAP serverī.
Description	LDAP servera apraksts.
Configure JIT provisioning	Atzīmējiet šo izvēles rūtiņu, lai parādītu ar JIT provisioning saistītās opcijas.
Group configuration	Atlasiet grupu konfigurācijas metodi: **memberOf** - meklējot lietotājus un to grupas dalības atribūtu **groupOfNames** - meklējot grupas, izmantojot member atribūtu Ņemiet vērā, ka memberOf ir ieteicamāks, jo tas ir ātrāks; izmantojiet groupOfNames, ja jūsu LDAP serveris neatbalsta `memberOf` vai ir nepieciešama grupu filtrēšana.
Group name attribute	Norādiet atribūtu, no kura iegūt grupas nosaukumu no visiem objektiem atribūtā `memberOf` (skatiet lauku *User group membership attribute*) Grupas nosaukums ir nepieciešams lietotāju grupu kartēšanai.
User group membership attribute	Norādiet atribūtu, kas satur informāciju par grupām, kurām lietotājs pieder (piemēram, `memberOf`). Piemēram, atribūts memberOf var saturēt šādu informāciju: `memberOf=cn=zabbix-admin,ou=Groups,dc=example,dc=com` Šis lauks ir pieejams tikai metodei memberOf.
User name attribute	Norādiet atribūtu, kas satur lietotāja vārdu.
User last name attribute	Norādiet atribūtu, kas satur lietotāja uzvārdu.
User group mapping	Kartējiet LDAP lietotāju grupas šablonu uz Zabbix lietotāju grupu un lietotāja lomu. Tas ir nepieciešams, lai noteiktu, kādu lietotāju grupu/lomu provisionētais lietotājs saņems Zabbix. Noklikšķiniet uz *Add*, lai pievienotu kartējumu. Lauks *LDAP group pattern* atbalsta aizstājējzīmes. Grupas nosaukumam jāsakrīt ar esošu grupu. Ja LDAP lietotājs atbilst vairākām Zabbix lietotāju grupām, lietotājs kļūst par visu šo grupu dalībnieku. Ja lietotājs atbilst vairākām Zabbix lietotāju lomām, lietotājs saņems to, kurai ir augstākais atļauju līmenis.
Media type mapping	Kartējiet lietotāja LDAP medija atribūtus (piemēram, e-pastu) uz Zabbix lietotāja medijiem paziņojumu sūtīšanai (atribūta vērtība tiek izmantota kā medija lauks *Send to*).
Advanced configuration	Noklikšķiniet uz etiķetes *Advanced configuration*, lai parādītu paplašinātās konfigurācijas opcijas (skatiet zemāk).
StartTLS	Atzīmējiet izvēles rūtiņu, lai, pieslēdzoties LDAP serverim, izmantotu StartTLS operāciju. Savienojums neizdosies, ja serveris neatbalsta StartTLS. StartTLS nevar izmantot ar serveriem, kas izmanto *ldaps* protokolu.

Parametrs	Apraksts
Search filter	Definējiet pielāgotu virkni lietotāja autentificēšanai LDAP. Tiek atbalstīti šādi vietturi: `%{attr}` - meklēšanas atribūta nosaukums (uid, sAMAccountName) `%{user}` - autentificējamā lietotāja lietotājvārda vērtība Piemēram, lai veiktu reģistrjutīgu meklēšanu reģistrnejutīgā LDAP vai Microsoft Active Directory vidē, virkni var definēt šādi: `(%{attr}:caseExactMatch:=%{user})`. Ja filtrs nav pielāgots, LDAP izmantos noklusējuma vērtību: `(%{attr}=%{user})`.

Note:
Lai konfigurētu LDAP serveri **tiešai lietotāja piesaistei**, pievienojiet atribūtu uid=%{user} parametram *Base DN* (piemēram,*uid=%{user},dc=example,dc=com*) un atstājiet parametrus *BindDN* un *Bind password* tukšus. Autentificēšanas laikā vietturis %{user} tiks aizstāts ar pieteikšanās laikā ievadīto lietotājvārdu.
 Izmantojot tiešo lietotāja piesaisti, Zabbix ir piekļuve tikai lietotāja akreditācijas datiem interaktīvās pieteikšanās laikā. Tāpēc provisionēšanas uzdevumi, kas darbojas bez interaktīvas pieteikšanās (piemēram, poga *Provision now* vai API metode `user.provision`), ignorēs lietotājus, kuri autentificējas, izmantojot tiešo lietotāja piesaisti, jo Zabbix nevar veikt piesaisti viņu vārdā. Izmantojiet anonīmu piesaisti vai īpašu piesaistes lietotāju (servisa kontu), ja jums nepieciešams, lai provisionēšana un LDAP meklēšana darbotos no lietotāja saskarnes, neprasot gala lietotājam būt pieteikušamies.

Šie lauki ir specifiski metodei "groupOfNames" kā *Group configuration* metodei:



Parametrs	Apraksts
Group base DN	Bāzes ceļš uz grupām LDAP serverī.
Group name attribute	Norādiet atribūtu, no kura iegūt grupas nosaukumu norādītajā bāzes ceļā uz grupām. Grupas nosaukums ir nepieciešams lietotāju grupu kartēšanai.
Group member attribute	Norādiet atribūtu, kas satur informāciju par grupas dalībniekiem LDAP (piemēram, `member`).
Reference attribute	Norādiet atsauces atribūtu grupas filtram (skatiet lauku *Group filter*). Pēc tam izmantojiet `%{ref}` grupas filtrā, lai iegūtu vērtības šeit norādītajam atribūtam.
Group filter	Norādiet filtru, lai iegūtu grupu, kuras dalībnieks ir lietotājs. Piemēram, `(member=uid=%{ref},ou=Users,dc=example,dc=com)` atbildīs "User1", ja grupas member atribūts ir `uid=User1,ou=Users,dc=example,dc=com`, un atgriezīs grupu, kuras dalībnieks ir "User1".

Warning:
Ja rodas problēmas ar sertifikātiem, lai drošs LDAP savienojums (ldaps) darbotos, iespējams, būs jāpievieno rinda `TLS_REQCERT allow` konfigurācijas failam /etc/openldap/ldap.conf. Tas var samazināt savienojuma ar LDAP katalogu drošību.

Note:
Ieteicams izveidot atsevišķu LDAP kontu (*Bind DN*), lai veiktu piesaisti un meklēšanu LDAP serverī ar minimālām tiesībām LDAP, nevis izmantot reālus lietotāju kontus (kas tiek izmantoti pieteikšanās veikšanai Zabbix lietotāja saskarnē).
 Šāda pieeja nodrošina lielāku drošību un neprasa mainīt *Bind password*, kad lietotājs LDAP serverī nomaina savu paroli.
 Iepriekš redzamajā tabulā tas ir konta nosaukums *ldap_search*.

Piekļuves testēšana

Poga *Test* ļauj pārbaudīt lietotāja piekļuvi:

Parametrs	Apraksts
Login	LDAP lietotāja vārds testēšanai (iepriekš aizpildīts ar pašreizējā lietotāja vārdu no Zabbix lietotāja saskarne). Šim lietotāja vārdam jāeksistē LDAP serverī. Zabbix neaktivizēs LDAP autentifikāciju, ja tas nespēs autentificēt testa lietotāju.
User password	LDAP lietotāja parole testēšanai.

3 SAML

Pārskats

SAML 2.0 autentifikāciju var izmantot, lai pierakstītos Zabbix.

Ja ir konfigurēta tikai SAML pierakstīšanās, tad lietotājam arī jāeksistē Zabbix, tomēr tā Zabbix parole netiks izmantota. Ja autentifikācija ir veiksmīga, tad Zabbix salīdzinās lokālo lietotājvārdu ar lietotājvārda atribūtu, ko atgriež SAML.

Lietotāju nodrošināšana

Ir iespējams konfigurēt JIT (just-in-time) **lietotāju nodrošināšanu** SAML lietotājiem. Šajā gadījumā nav nepieciešams, lai lietotājs jau eksistētu Zabbix. Lietotāja kontu var izveidot, kad lietotājs pirmo reizi piesakās Zabbix.

Attention:
Ja JIT nodrošināšana ir iespējota, cilnē *Authentication* ir jānorāda lietotāju grupa atnodrošinātajiem lietotājiem.

Papildus JIT nodrošināšanai ir iespējams arī iespējot un konfigurēt SCIM (System for Cross-domain Identity Management) nodrošināšanu — *nepārtrauktu* lietotāju kontu pārvaldību tiem lietotājiem, kuri ir izveidoti ar lietotāju nodrošināšanu. SCIM nodrošināšanai ir nepieciešams Zabbix API token (ar Super admin atļaujām) autentifikācijai Zabbix.

Piemēram, ja lietotājs tiek pārvietots no vienas SAML grupas uz citu, lietotājs tiks pārvietots arī no vienas grupas uz citu Zabbix; ja lietotājs tiek noņemts no SAML grupas, lietotājs tiks noņemts arī no grupas Zabbix un, ja tas nepieder nevienai citai grupai, pievienots lietotāju grupai atnodrošinātajiem lietotājiem.

Ja SCIM ir iespējots un konfigurēts, SAML lietotājs tiks nodrošināts brīdī, kad lietotājs piesakās Zabbix, un pēc tam nepārtraukti atjaunināts, pamatojoties uz izmaiņām SAML. Jau esošie SAML lietotāji netiks nodrošināti, un tiks atjaunināti tikai nodrošinātie lietotāji. Ņemiet vērā, ka lietotājam tiks pievienoti tikai derīgi mediji, kad lietotājs tiks nodrošināts vai atjaunināts.

Ja SCIM nav iespējots, SAML lietotājs tiks nodrošināts (un vēlāk atjaunināts) brīdī, kad lietotājs piesakās Zabbix.

Note:
Ja SAML autentifikācija ir iespējota, lietotāji varēs izvēlēties starp lokālu pieteikšanos vai SAML vienoto pierakstīšanos. Ja tiek izmantota JIT nodrošināšana, tad ir iespējama tikai vienotā pierakstīšanās.

Identitātes nodrošinātāja iestatīšana

Lai strādātu ar Zabbix, SAML identitātes nodrošinātājs (onelogin.com, auth0.com, okta.com u.c.) ir jākonfigurē šādi:

- *Assertion Consumer URL* jāiestata uz `<path_to_zabbix_ui>/index_sso.php?acs`
- *Single Logout URL* jāiestata uz `<path_to_zabbix_ui>/index_sso.php?sls`

`<path_to_zabbix_ui>` piemēri: `https://example.com/zabbix/ui`, `http://another.example.com/zabbix`, `http://<any_publ`

Zabbix iestatīšana

Attention:
Ja vēlaties izmantot SAML autentifikāciju lietotāja saskarnē, ir jāinstalē php-openssl.

Lai izmantotu SAML autentifikāciju, Zabbix jākonfigurē šādi:

1. Nodrošiniet SP privāto atslēgu un sertifikātus. Lai aktivizētu SAML, izmantojot lietotāja saskarni, ir jānorāda IdP sertifikāts (ielīmējot vai izvēloties failu). Ja ir atlasītas opcijas *Sign* vai *Encrypt*, ir jānorāda SP sertifikāts un SP privātā atslēga.

Ja zabbix.conf.php ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā lietotāja saskarnē varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu — faili failu sistēmā nav nepieciešami.

Ja zabbix.conf.php ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē ui/conf/certs vai ceļā, kas konfigurēts zabbix.conf.php), un lietotāja saskarne nesaglabās sertifikātus datubāzē. Ņemiet vērā, ka, ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir aizkomentēts, tiek pieņemts, ka tiek izmantota glabāšana failos.

Pēc noklusējuma Zabbix meklēs šādās vietās:

- ui/conf/certs/sp.key - SP privātās atslēgas fails
- ui/conf/certs/sp.crt - SP sertifikāta fails
- ui/conf/certs/idp.crt - IDP sertifikāta fails

Attention:
Sertifikātu materiālu var importēt, izmantojot lietotāja saskarni, ielīmējot tekstu vai augšupielādējot failus. Lai gan nešifrēts imports dažās vidēs ērtības labad ir atļauts, ir stingri ieteicams izmantot šifrētu pārraidi (piemēram, iespējot HTTPS/TLS Zabbix lietotāja saskarnei). Ja SAML sertifikāti/atslēgas tiek glabāti datubāzē, iespējojiet TLS/SSL datubāzes savienojumiem, lai aizsargātu datplūsmu starp lietotni un datubāzi.
 Šifrējiet datubāzes dublējumkopijas, kas var saturēt SAML sertifikātus vai privātās atslēgas, un ierobežojiet piekļuvi dublējumkopiju failiem un datubāzes tabulām, kurās tiek glabāts SAML materiāls. Ierobežojiet datubāzes lietotāja privilēģijas līdz nepieciešamajam minimumam.

2. Visus svarīgākos iestatījumus var konfigurēt Zabbix lietotāja saskarnē. Tomēr ir iespējams norādīt arī papildu iestatījumus konfigurācijas failā.

Konfigurācijas parametri, kas pieejami Zabbix lietotāja saskarnē:

Parameter	Description
Enable SAML authentication	Atzīmējiet izvēles rūtiņu, lai iespējotu SAML autentifikāciju.
Enable JIT provisioning	Atzīmējiet izvēles rūtiņu, lai iespējotu JIT lietotāju nodrošināšanu.
IDP entity ID	Unikālais entītijas identifikators SAML identitātes nodrošinātājā.
SSO service URL	URL, uz kuru lietotāji tiks novirzīti, piesakoties sistēmā.
SLO service URL	URL, uz kuru lietotāji tiks novirzīti, izrakstoties no sistēmas. Ja tas ir atstāts tukšs, SLO pakalpojums netiks izmantots.

Parameter	Description
Username attribute	SAML atribūts, kas tiks izmantots kā lietotājvārds, piesakoties Zabbix. Atbalstīto vērtību sarakstu nosaka identitātes nodrošinātājs. Piemēri: 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	Unikālais pakalpojuma nodrošinātāja identifikators (ja tas nesakrīt, darbība tiks noraidīta). Ir iespējams norādīt URL vai jebkuru datu virkni.
SP name ID format	Pieprasīt atbildē noteiktu name ID formātu. Piemēri: 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	Pakalpojuma nodrošinātāja (SP) sertifikāta saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai.
SP private key	Pakalpojuma nodrošinātāja (SP) privātās atslēgas saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai. Nodrošina drošu autentifikāciju un datu apmaiņu ar identitātes nodrošinātāju (IdP).
SP certificate	Pakalpojuma nodrošinātāja (SP) sertifikāta saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai.
Sign	Atzīmējiet izvēles rūtiņas, lai atlasītu entītijas, kurām jāiespējo SAML paraksts: *Messages* *Assertions* *AuthN requests* *Logout requests* *Logout responses*
Encrypt	Atzīmējiet izvēles rūtiņas, lai atlasītu entītijas, kurām jāiespējo SAML šifrēšana: *Name ID* *Assertions*
Case-sensitive login	Noņemiet atzīmi no izvēles rūtiņas, lai atspējotu reģistrjutīgu pieteikšanos lietotājvārdiem (pēc noklusējuma iespējota). Atspējojot reģistrjutīgu pieteikšanos, ir iespējams, piemēram, pieteikties kā ”admin”, pat ja Zabbix lietotājs ir ”Admin” vai ”ADMIN”. Lūdzu, ņemiet vērā, ka, ja reģistrjutīga pieteikšanās ir atspējota un ir vairāki Zabbix lietotāji ar līdzīgiem lietotājvārdiem (piemēram, Admin un admin), šo lietotāju pieteikšanās vienmēr tiks liegta ar šādu kļūdas ziņojumu: ”Authentication failed: supplied credentials are not unique.”
Configure JIT provisioning	Atzīmējiet šo izvēles rūtiņu, lai parādītu opcijas, kas saistītas ar JIT lietotāju nodrošināšanu.
Group name attribute	Norādiet grupas nosaukuma atribūtu JIT lietotāju nodrošināšanai.
User name attribute	Norādiet lietotāja vārda atribūtu JIT lietotāju nodrošināšanai.
User last name attribute	Norādiet lietotāja uzvārda atribūtu JIT lietotāju nodrošināšanai.
User group mapping	Kartējiet SAML lietotāju grupas šablonu uz Zabbix lietotāju grupu un lietotāja lomu. Tas ir nepieciešams, lai noteiktu, kuru lietotāju grupu/lomu nodrošinātais lietotājs saņems Zabbix. Noklikšķiniet uz *Add*, lai pievienotu kartējumu. Lauks *SAML group pattern* atbalsta aizstājējzīmes. Grupas nosaukumam jāsakrīt ar esošas grupas nosaukumu. Ja SAML lietotājs atbilst vairākām Zabbix lietotāju grupām, lietotājs kļūst par visu šo grupu dalībnieku. Ja lietotājs atbilst vairākām Zabbix lietotāju lomām, lietotājs saņems augstāko atļauju līmeni no tām.
Media type mapping	Kartējiet lietotāja SAML mediju atribūtus (piemēram, e-pastu) uz Zabbix lietotāja medijiem paziņojumu sūtīšanai.
Enable SCIM provisioning	Atzīmējiet šo izvēles rūtiņu, lai iespējotu SCIM 2.0 nodrošināšanu.

Skatiet SAML identitātes nodrošinātāju konfigurēšanas piemērus pierakstīšanās un lietotāju nodrošināšanai Zabbix vidē ar:

- Microsoft Azure AD
- Okta
- Onelogin

Piezīmes par SCIM nodrošināšanu

SCIM nodrošināšanai identitātes nodrošinātāja pusē norādiet ceļu uz Zabbix lietotāja saskarne un pievienojiet tam `api_scim.php`, piemēram:

`https://<path-to-zabbix-ui>/api_scim.php`

Lietotāja atribūti, kas tiek izmantoti Zabbix (lietotājvārds, lietotāja vārds, lietotāja uzvārds un mediju atribūti), ir jāpievieno kā pielāgoti atribūti, un, ja nepieciešams, ārējai nosaukumvietai jābūt tādai pašai kā lietotāja shēmai: `urn:ietf:params:scim:schemas:core:2.0`

Papildu iestatījumi

Papildu SAML parametrus var konfigurēt Zabbix lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*):

- $SSO['SP_KEY'] = '<ceļš uz SP privātās atslēgas failu>';
- $SSO['SP_CERT'] = '<ceļš uz SP sertifikāta failu>';
- $SSO['IDP_CERT'] = '<ceļš uz IDP sertifikāta failu>';
- $SSO['SETTINGS']

Note:
Masīvam `$SSO['SETTINGS']` jāatbilst tai pašai struktūrai, ko sagaida *SAML PHP Toolkit* bibliotēka (iekļauta Zabbix komplektā). Pilnu pieejamo konfigurācijas opciju aprakstu skatiet oficiālajā bibliotēkas dokumentācijā.

Tikai tālāk norādītās opcijas var iestatīt kā daļu no $SSO['SETTINGS']:

- *strict*
- *baseurl*
- *compress*
- *contactPerson*
- *organization*
- *sp* (tikai šajā sarakstā norādītās opcijas)
  - *attributeConsumingService*
  - *x509certNew*
- *idp* (tikai šajā sarakstā norādītās opcijas)
  - *singleLogoutService* (tikai viena opcija)
    - *responseUrl*
  - *certFingerprint*
  - *certFingerprintAlgorithm*
  - *x509certMulti*
- *security* (tikai šajā sarakstā norādītās opcijas)
  - *signMetadata*
  - *wantNameId*
  - *requestedAuthnContext*
  - *requestedAuthnContextComparison*
  - *wantXMLValidation*
  - *relaxDestinationValidation*
  - *destinationStrictlyMatches*
  - *rejectUnsolicitedResponsesWithInResponseTo*
  - *signatureAlgorithm*
  - *digestAlgorithm*
  - *lowercaseUrlencoding*

Visas pārējās opcijas tiks ņemtas no datubāzes, un tās nevar pārrakstīt. Opcija *debug* tiks ignorēta.

Papildus tam, ja Zabbix UI atrodas aiz starpniekservera vai slodzes balansētāja, var izmantot pielāgoto opciju *use_proxy_headers*:

- *false* (noklusējums) - ignorēt opciju;
- *true* - izmantot X-Forwarded-* HTTP galvenes bāzes URL izveidei.

Ja savienojumam ar Zabbix instanci tiek izmantots slodzes balansētājs, kur slodzes balansētājs izmanto TLS/SSL, bet Zabbix to neizmanto, jānorāda parametri 'baseurl', 'strict' un 'use_proxy_headers' šādi:

```
$SSO['SETTINGS'] = [
    'strict' => false,
```

```
    'baseurl' => 'https://zabbix.example.com/zabbix/',
    'use_proxy_headers' => true
];
```

Konfigurācijas piemērs:

```
$SSO['SETTINGS'] = [
    'security' => [
        'signatureAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#rsa-sha384'
        'digestAlgorithm' => 'http://www.w3.org/2001/04/xmldsig-more#sha384',
        // ...
    ],
    // ...
];
```

Lietotāja saskarnes konfigurācija ar Kerberos/ADFS

Zabbix lietotāja saskarnes konfigurācijas failu (*zabbix.conf.php*) var izmantot, lai konfigurētu SSO ar Kerberos autentifikāciju un ADFS:

```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => [
            'urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos',
        ],
        'requestedAuthnContextComparison' => 'exact'
    ]
];
```

Šajā gadījumā SAML konfigurācijas laukā *SP name ID* iestatiet:

```
urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
```

4 MFA

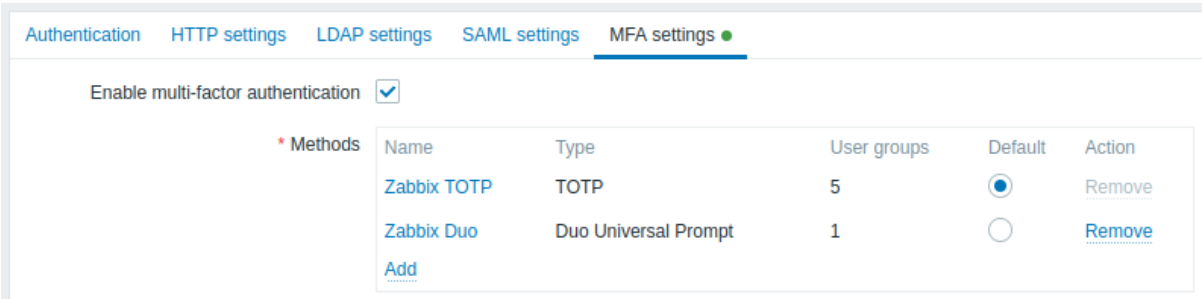
Pārskats

Daudzfaktoru autentifikāciju (MFA) var izmantot, lai pierakstītos Zabbix, nodrošinot papildu drošības slāni papildus tikai lietotājvārdam un parolei.

Izmantojot MFA, lietotājam jābūt izveidotam Zabbix, piesakoties jānorāda Zabbix akreditācijas dati, kā arī papildus jāpierāda sava identitāte citā veidā, parasti ar kodu, ko ģenerē autentifikatora lietotne lietotāja tālrunī.

Ir pieejamas vairākas MFA metodes, kas ļauj lietotājiem izvēlēties iespēju, kura vislabāk atbilst viņu drošības prasībām un vēlmēm. Šīs metodes ir Time-Based One-Time Password (TOTP) un Duo Universal Prompt.

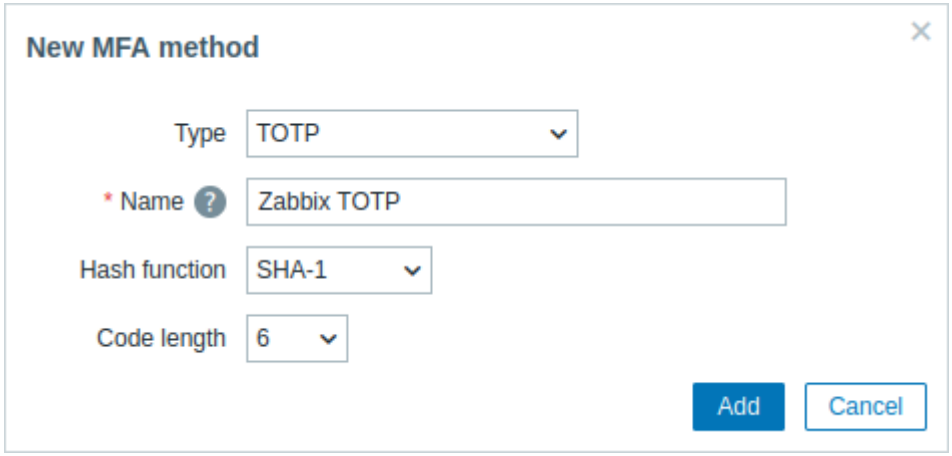
Konfigurācija



Konfigurācijas parametri:

Parametrs	Apraksts
Iespējot daudzfaktoru autentifikāciju	Atzīmējiet izvēles rūtiņu, lai iespējotu daudzfaktoru autentifikāciju.
Metodes	Noklikšķiniet uz *Pievienot*, lai konfigurētu MFA metodi (skatiet metodes konfigurāciju zemāk).

Metodes konfigurācija



Metodes konfigurācijas parametri:

Parametrs	Apraksts
Type	Atlasiet MFA metodes tipu: **TOTP** — izmantojiet autentifikatora lietotni, lai ģenerētu uz laiku balstītas vienreizējās paroles; **Duo Universal Prompt** — izmantojiet Duo autentifikācijas pakalpojumu, lai nodrošinātu daudzfaktoru autentifikāciju.
Name	Ievadiet nosaukumu, kas autentifikatora lietotnēs visiem MFA lietotājiem tiek parādīts kā konta nosaukums (piemēram, "Zabbix").
Hash function	Atlasiet jaucējfunkciju (SHA-1, SHA-256 vai SHA-512) TOTP kodu ģenerēšanai. Šis parametrs ir pieejams, ja MFA metodes tips ir iestatīts uz "TOTP". Ņemiet vērā, ka SHA-256 vai SHA-512 izvēle var būtiski ierobežot saderību, jo daudzas lietotnes pašlaik šīs funkcijas neatbalsta.
Code length	Atlasiet verifikācijas koda garumu (6 vai 8). Šis parametrs ir pieejams, ja MFA metodes tips ir iestatīts uz "TOTP".
API hostname	Ievadiet API resursdatora nosaukumu, ko nodrošina Duo autentifikācijas pakalpojums. Šis parametrs ir pieejams, ja MFA metodes tips ir iestatīts uz "Duo Universal Prompt".
Client ID	Ievadiet klienta ID, ko nodrošina Duo autentifikācijas pakalpojums. Šis parametrs ir pieejams, ja MFA metodes tips ir iestatīts uz "Duo Universal Prompt".
Client secret	Ievadiet klienta slepeno atslēgu, ko nodrošina Duo autentifikācijas pakalpojums. Šis parametrs ir pieejams, ja MFA metodes tips ir iestatīts uz "Duo Universal Prompt".

Konfigurācijas piemēri

Šajā sadaļā ir sniegti piemēri MFA konfigurēšanai, izmantojot Time-Based One-Time Password (TOTP) un Duo Universal Prompt.

TOTP

TOTP gadījumā lietotājiem ir jāapstiprina sava identitāte, izmantojot autentifikatora lietotni (piemēram, lietotni Google Authenticator).

1. Dodieties uz MFA iestatījumiem Zabbix sadaļā *Lietotāji* → *Autentifikācija* un iespējojiet daudzfaktoru autentifikāciju.

2. Pievienojiet jaunu MFA metodi ar šādu konfigurāciju:

- Tips: TOTP
- Nosaukums: Zabbix TOTP
- Jaucējfunkcija: SHA-1
- Koda garums: 6

3. Noklikšķiniet uz pogas *Pievienot* un pēc tam uz pogas *Atjaunināt*.

4. Dodieties uz *Lietotāji* → *Lietotāju grupas* un izveidojiet jaunu lietotāju grupu ar šādu konfigurāciju:

- Grupas nosaukums: TOTP grupa
- Lietotāji: Admin
- Daudzfaktoru autentifikācija: Noklusējuma (vai "Zabbix TOTP", ja tā nav iestatīta kā noklusējuma)

5. Izrakstieties no Zabbix un piesakieties vēlreiz, izmantojot savus akreditācijas datus. Pēc veiksmīgas pieteikšanās jums tiks piedāvāts reģistrēties MFA, parādot QR kodu un slepeno atslēgu.

6. Noskenējiet QR kodu vai ievadiet slepeno atslēgu lietotnē Google Authenticator. Lietotne ģenerēs verifikācijas kodu, kas jums jāievada, lai pabeigtu pieteikšanās procesu.

7. Turpmākajās pieteikšanās reizēs iegūstiet verifikācijas kodu no lietotnes Google Authenticator un ievadiet to pieteikšanās laikā.

Duo Universal Prompt

Duo Universal Prompt gadījumā lietotājiem ir jāapstiprina sava identitāte, izmantojot autentifikācijas lietotni Duo Mobile.

Attention:
Duo Universal Prompt MFA metodei ir nepieciešams instalēt php-curl paplašinājumu, piekļuvei Zabbix, izmantojot HTTPS, un atļaujai izejošajiem savienojumiem ar Duo serveriem. Turklāt, ja esat iespējojis Content Security Policy (CSP) tīmekļa serverī, noteikti pievienojiet "duo.com" CSP direktīvai sava virtuālā hosta konfigurācijas failā.

1. Reģistrējiet bezmaksas Duo administratora kontu vietnē Duo Signup.

2. Atveriet Duo administrēšanas paneli, dodieties uz *Applications* → *Protect an Application*, meklējiet lietotni *Web SDK* un noklikšķiniet uz *Protect*.

3. Pierakstiet akreditācijas datus (Client ID, Client secret, API hostname), kas nepieciešami MFA metodes konfigurēšanai Zabbix.

4. Dodieties uz MFA iestatījumiem Zabbix sadaļā *Users* → *Authentication* un iespējojiet daudzfaktoru autentifikāciju.

5. Pievienojiet jaunu MFA metodi ar šādu konfigurāciju:

   - Tips: Duo Universal Prompt
   - Nosaukums: Zabbix Duo
   - API hostname: (izmantojiet API hostname no Duo)
   - Client ID: (izmantojiet Client ID no Duo)
   - Client secret: (izmantojiet Client secret no Duo)

6. Noklikšķiniet uz pogas *Add* un pēc tam uz pogas *Update*.

7. Dodieties uz *Users* → *User groups* un izveidojiet jaunu lietotāju grupu ar šādu konfigurāciju:

   - Grupas nosaukums: Duo grupa
   - Lietotāji: Admin
   - Daudzfaktoru autentifikācija: Default (vai "Zabbix Duo", ja tā nav iestatīta kā noklusējuma vērtība)

8. Izrakstieties no Zabbix un piesakieties vēlreiz, izmantojot savus akreditācijas datus. Pēc veiksmīgas pieteikšanās jums tiks piedāvāts reģistrēties MFA un jūs tiksiet novirzīts uz Duo. Pabeidziet Duo iestatīšanu un apstipriniet savu lietotāju, izmantojot Duo lietotni savā tālrunī, lai pieteiktos.

9. Turpmākajās pieteikšanās reizēs izmantojiet atbilstošo MFA metodi, ko nodrošina Duo lietotne (piemēram, verifikācijas koda iegūšanu, atbildēšanu uz paspiegādes paziņojumiem vai aparatūras atslēgu izmantošanu), un pieteikšanās laikā ievadiet nepieciešamo informāciju.

9 Administrēšana

Pārskats

Izvēlne Administration ir paredzēta Zabbix administratīvajām funkcijām. Šī izvēlne ir pieejama tikai SuperAdmin lietotāja tipa lietotājiem.

1 Vispārīgi

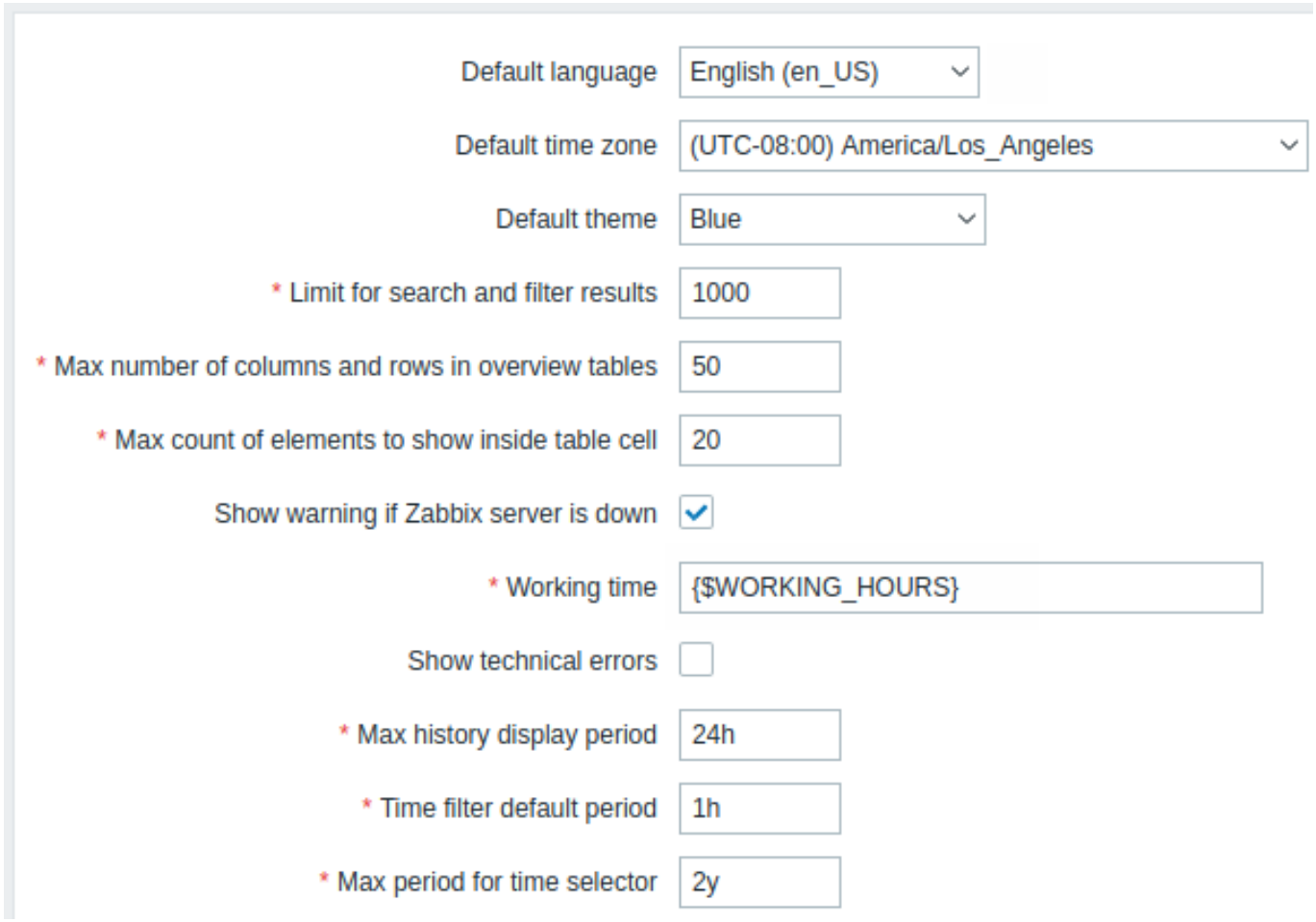
Pārskats

Sadaļā *Administrēšana* → *Vispārīgi* ir pieejamas vairākas apakšsadaļas, kas paredzētas ar lietotāja saskarni saistīto noklusējuma iestatījumu noteikšanai un Zabbix pielāgošanai.

Pieejamo apakšsadaļu saraksts tiek parādīts, nospiežot *Vispārīgi* izvēlnes sadaļā *Administrēšana*. Ir iespējams arī pārslēgties starp apakšsadaļām, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.

GUI

Šajā sadaļā ir pieejama vairāku ar lietotāja saskarni saistītu noklusējuma iestatījumu pielāgošana.



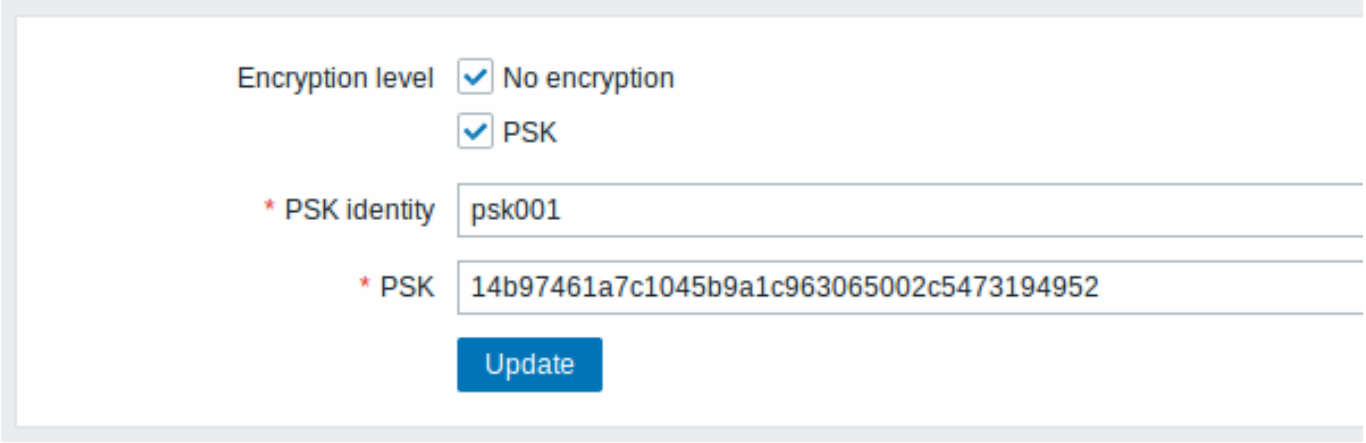
Konfigurācijas parametri:

Parametrs	Apraksts
Noklusējuma valoda	Noklusējuma valoda lietotājiem, kuri savos profilos nav norādījuši valodu, un vieslietotājiem. Papildinformāciju skatiet sadaļā Papildu lietotāja saskarnes valodu instalēšana.
Noklusējuma laika josla	Noklusējuma laika josla lietotājiem, kuri savos profilos nav norādījuši laika joslu, un vieslietotājiem.
Noklusējuma tēma	Noklusējuma tēma lietotājiem, kuri savos profilos nav norādījuši tēmu, un vieslietotājiem.
Meklēšanas un filtra rezultātu ierobežojums	Maksimālais elementu (rindu) skaits, kas tiks parādīts tīmekļa saskarnes sarakstā, piemēram, sadaļā *Datu ievākšana* > *Hosti*. *Piezīme*: ja iestatīta, piemēram, vērtība "50", visos ietekmētajos lietotāja saskarnes sarakstos tiks parādīti tikai pirmie 50 elementi. Ja kādā sarakstā ir vairāk nekā piecdesmit elementi, tas tiks norādīts ar zīmi "+" tekstā *"Displaying 1 to 50 of **50+** found"*. Tāpat, ja tiek izmantota filtrēšana un joprojām ir vairāk nekā 50 atbilstību, tiks parādītas tikai pirmās 50. Ņemiet vērā, ka šī parametra vērtības palielināšana var samazināt veiktspēju un palielināt atmiņas patēriņu tīmekļa servera pusē.
Maksimālais kolonnu un rindu skaits pārskata tabulās	Maksimālais kolonnu un rindu skaits, ko parādīt paneļa logrīkā *Trigeru pārskats*. Tas pats ierobežojums attiecas gan uz kolonnām, gan rindām. Ja pastāv vairāk rindu un/vai kolonnu, nekā tiek parādīts, sistēma tabulas apakšdaļā parādīs brīdinājumu: "Not all results are displayed. Please provide more specific search criteria."
Maksimālais elementu skaits parādīšanai tabulas šūnā	Ierakstiem, kas tiek parādīti vienā tabulas šūnā, netiks rādīts vairāk par šeit konfigurēto skaitu.
Rādīt brīdinājumu, ja Zabbix serveris nav pieejams	Šis parametrs iespējo brīdinājuma ziņojuma parādīšanu pārlūkprogrammas logā, ja Zabbix serveris nav sasniedzams (iespējams, nedarbojas). Ziņojums paliek redzams arī tad, ja lietotājs ritina lapu uz leju. Novietojot kursoru virs ziņojuma, tas tiek īslaicīgi paslēpts, lai atklātu zem tā esošo saturu.
Darba laiks	Šis visas sistēmas parametrs nosaka darba stundas. Grafikos darba laiks tiek attēlots ar baltu fonu, bet ārpus darba laika — ar pelēku. Laika formāta aprakstu skatiet lapā Laika perioda specifikācija. Tiek atbalstīti lietotāja makrosi.

Parametrs	Apraksts
Rādīt tehniskās kļūdas	Ja atzīmēts, visi reģistrētie lietotāji varēs redzēt tehniskās kļūdas (PHP/SQL). Ja nav atzīmēts, šī informācija ir pieejama tikai Zabbix Super Admins un lietotājiem, kuri pieder lietotāju grupām ar iespējotu atkļūdošanas režīmu.
Maksimālais vēstures attēlošanas periods	Maksimālais laika periods, par kuru attēlot vēsturiskos datus sadaļā *Uzraudzība > Jaunākie dati*, hosta tīmekļa scenārija informāciju sadaļā *Uzraudzība > Hosti* un paneļa logrīkā *Top items*. Atļautais diapazons: 24 stundas (noklusējums) - 1 nedēļa. Tiek atbalstīti laika sufiksi, piemēram, 1w (viena nedēļa), 36h (36 stundas).
Laika filtra noklusējuma periods	Laika periods, kas pēc noklusējuma tiek izmantots grafikos un paneļos. Atļautais diapazons: 1 minūte - 10 gadi (noklusējums: 1 stunda). Tiek atbalstīti laika sufiksi, piemēram, 10m (desmit minūtes), 5w (piecas nedēļas). Piezīme: kad lietotājs, skatot grafiku, maina laika periodu, šis laika periods tiek saglabāts kā lietotāja iestatījums, aizstājot globālo noklusējuma vērtību vai iepriekšējo lietotāja izvēli.
Maksimālais periods laika atlasītājam	Maksimālais pieejamais laika periods grafikiem un paneļiem. Lietotāji nevarēs vizualizēt senākus datus. Atļautais diapazons: 1 gads - 10 gadi (noklusējums: 2 gadi). Tiek atbalstīti laika sufiksi, piemēram, 1y (viens gads), 365w (365 nedēļas).

Automātiskā reģistrācija

Šajā sadaļā varat konfigurēt šifrēšanas līmeni aktīvā aģenta automātiskajai reģistrācijai.



Parametri, kas atzīmēti ar zvaigznīti, ir obligāti.

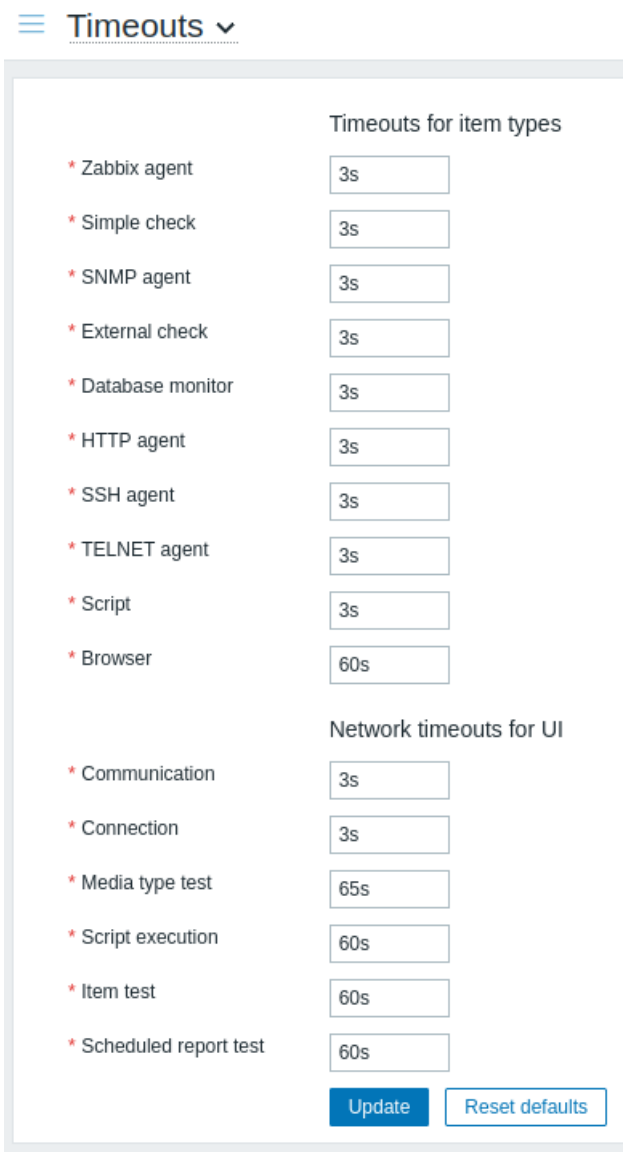
Konfigurācijas parametri:

Parametrs	Apraksts
Šifrēšanas līmenis	Atlasiet vienu vai abas šifrēšanas līmeņa opcijas: **Bez šifrēšanas** — ir atļauti nešifrēti savienojumi **PSK** — ir atļauti ar TLS šifrēti savienojumi ar iepriekš koplietotu atslēgu
PSK identitāte	Ievadiet iepriekš koplietotās atslēgas identitātes virkni. Šis lauks ir pieejams tikai tad, ja kā *Šifrēšanas līmenis* ir atlasīts 'PSK'. Neievietojiet sensitīvu informāciju PSK identitātē, jo tā tiek pārraidīta nešifrētā veidā pa tīklu, lai informētu saņēmēju, kuru PSK izmantot.
PSK	Ievadiet iepriekš koplietoto atslēgu (pāra skaits heksadecimālu rakstzīmju). Maksimālais garums: 512 heksadecimālas rakstzīmes (256 baitu PSK), ja Zabbix izmanto GnuTLS vai OpenSSL bibliotēku, 64 heksadecimālas rakstzīmes (32 baitu PSK), ja Zabbix izmanto mbed TLS (PolarSSL) bibliotēku. Piemērs: 1f87b595725ac58dd977beef14b97461a7c1045b9a1c963065002c5473194952 Šis lauks ir pieejams tikai tad, ja kā *Šifrēšanas līmenis* ir atlasīts 'PSK'.

Skatiet arī: Droša automātiskā reģistrācija

Noildzes

Šajā sadaļā ir iespējams iestatīt globālās vienumu tipu noildzes un tīkla noildzes. Visi lauki šajā formā ir obligāti.

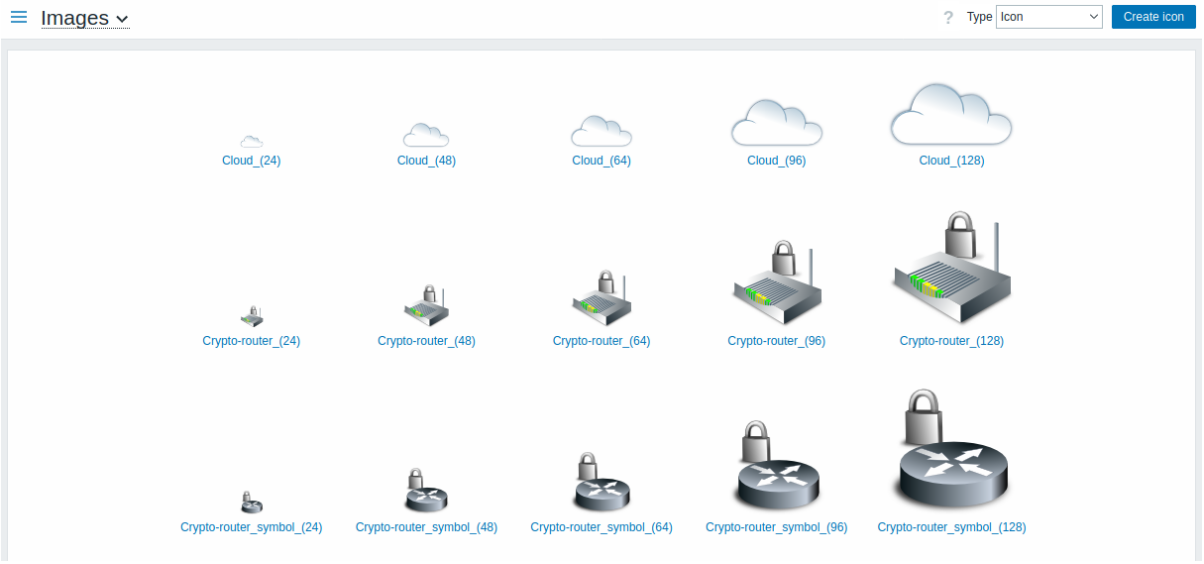


Parametrs		Apraksts
Noildzes vienumu tipiem		Cik sekundes Zabbix jāgaida, pirms pārtraukt pārbaudi (atkarībā no tās tipa). Papildu informāciju un ierobežojumus skatiet sadaļā vienuma noildze. Atļautais diapazons: 1 - 600s (noklusējums: 3s; noklusējums pārlūka vienumiem: 60s). Tiek atbalstīti laika sufiksi, piemēram, 30s, 1m, un lietotāja makrosi. Ņemiet vērā, ka starpniekservera līmeņa noildzes, ja tās ir konfigurētas, aizstāj šos globālos iestatījumus. Turklāt, ja ir konfigurētas noildzes konkrētiem vienumiem, tās aizstāj gan globālos, gan starpniekservera iestatījumus.
Tīkla noildzes lietotāja saskarnei		
	Saziņa	Cik sekundes gaidīt, pirms aizvērt neaktīvu ligzdu (ja savienojums ar Zabbix serveri ir izveidots iepriekš, bet lietotāja saskarne šajā laikā nevar pabeigt datu lasīšanas/sūtīšanas darbību, savienojums tiks pārtraukts). Atļautais diapazons: 1 - 300s (noklusējums: 3s).
	Savienojums	Cik sekundes gaidīt, pirms pārtraukt mēģinājumu izveidot savienojumu ar Zabbix serveri. Atļautais diapazons: 1 - 30s (noklusējums: 3s).
	Mediju tipa tests	Cik sekundes gaidīt atbildi, testējot mediju tipu. Atļautais diapazons: 1 - 300s (noklusējums: 65s).

Parametrs		Apraksts
	Skripta izpilde	Cik sekundes gaidīt atbildi, izpildot skriptu. Atļautais diapazons: 1 - 300s (noklusējums: 60s). Šī noildze attiecas uz visu skripta izpildes ķēdi, kurai var būt dažāds garums. Piemēram, ja skripts tiek izpildīts aģentā, tas ir pilns ceļš caur serveri (iespējams, arī starpniekserveri) uz aģentu un atpakaļ. Skriptiem, kas tiek izpildīti aģentā, tiek piemērota servera vai starpniekservera noildze. Skriptiem, kas tiek izpildīti tikai aktīvajā aģentā, visticamāk, būs jāpalielina noklusējuma servera/starpniekservera noildze. Servera/starpniekservera noildzei jābūt lielākai par aktīvās pārbaudes atsvaidzināšanas biežumu, pretējā gadījumā servera/starpniekservera noildze var tikt pārsniegta, pirms aktīvais aģents vispār saņem skriptu. Skatiet arī: Skripta noildze
	Vienuma tests	Cik sekundes gaidīt atgrieztos datus, testējot vienumu. Atļautais diapazons: 1 - 600s (noklusējums: 60s).
	Plānotās atskaites tests	Cik sekundes gaidīt atgrieztos datus, testējot plānoto atskaiti. Atļautais diapazons: 1 - 300s (noklusējums: 60s).

Attēli

Sadaļā Attēli tiek parādīti visi Zabbix pieejamie attēli. Attēli tiek glabāti datubāzē.

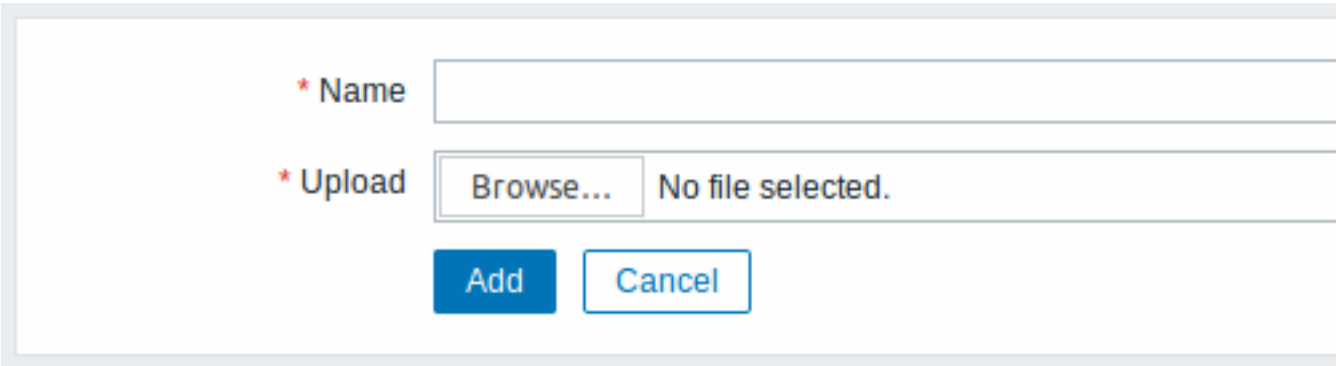


Nolaižamais saraksts *Tips* ļauj pārslēgties starp ikonu un fona attēliem:

- Ikonas tiek izmantotas, lai attēlotu tīkla kartes elementus
- Foni tiek izmantoti kā tīkla karšu fona attēli

Attēla pievienošana

Jūs varat pievienot savu attēlu, noklikšķinot uz pogas *Izveidot ikonu* vai *Izveidot fonu* augšējā labajā stūrī.



Attēla atribūti:

Parameter	Description
Name	Unikāls attēla nosaukums.
Upload	Atlasiet failu (PNG, JPEG, GIF vai WebP) no lokālās sistēmas, lai augšupielādētu to Zabbix. *Ņemiet vērā*, ka var būt iespējams augšupielādēt arī citus formātus, kas augšupielādes laikā tiks pārveidoti uz PNG. Attēlu apstrādei tiek izmantota GD bibliotēka, tāpēc atbalstītie formāti ir atkarīgi no izmantotās bibliotēkas versijas (Zabbix nepieciešama versija 2.0.28 vai jaunāka).

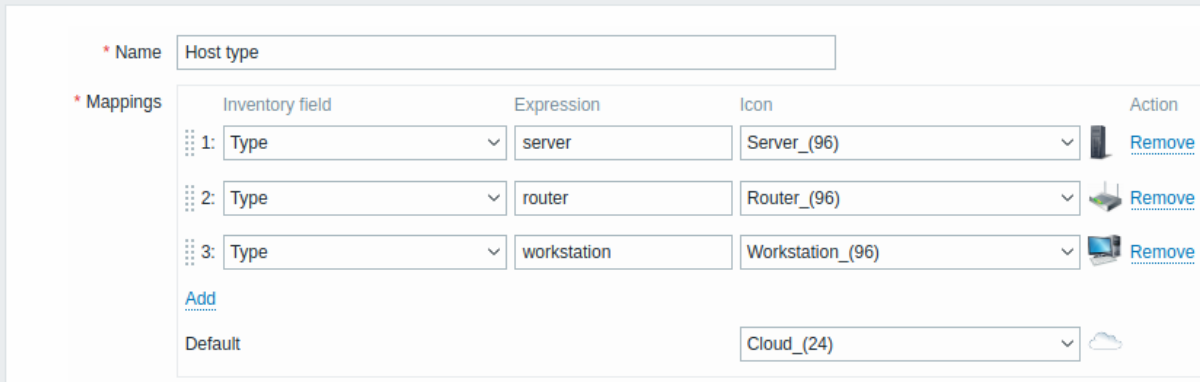
Note:
Augšupielādējamā faila maksimālais izmērs ir ierobežots ar ZBX_MAX_IMAGE_SIZE vērtību, kas ir 1024x1024 baiti jeb 1 MB.
Attēla augšupielāde var neizdoties, ja attēla izmērs ir tuvu 1 MB un MySQL konfigurācijas parametra `max_allowed_packet` noklusējuma vērtība ir 1 MB. Šādā gadījumā palieliniet max_allowed_packet parametru.

Ikonu kartēšana

Šajā sadaļā var izveidot noteiktu hostu kartēšanu ar noteiktām ikonām. Lai izveidotu kartēšanu, tiek izmantota hosta inventāra lauku informācija.

Pēc tam kartējumus var izmantot tīkla kartes konfigurācijā, lai atbilstošās ikonas atbilstošajiem hostiem tiktu piešķirtas automātiski.

Lai izveidotu jaunu ikonu karti, augšējā labajā stūrī noklikšķiniet uz *Izveidot ikonu karti*.



Konfigurācijas parametri:

Parametrs	Apraksts
Name	Unikāls ikonu kartes nosaukums.
Mappings	Kartējumu saraksts. Kartējumu secība nosaka, kuram no tiem būs prioritāte. Kartējumus sarakstā var pārvietot uz augšu un uz leju, izmantojot vilkšanu un nomešanu.
Inventory field	Hosta inventāra lauks, kurā tiks meklēta atbilstība.
Expression	Regulārā izteiksme, kas apraksta atbilstību.
Icon	Ikona, ko izmantot, ja tiek atrasta izteiksmei atbilstoša vērtība.
Default	Noklusējuma ikona, ko izmantot.

Regulārās izteiksmes

Šajā sadaļā var izveidot pielāgotas regulārās izteiksmes, kuras var izmantot vairākās vietās lietotāja saskarnē. Sīkāku informāciju skatiet sadaļā Regulārās izteiksmes.

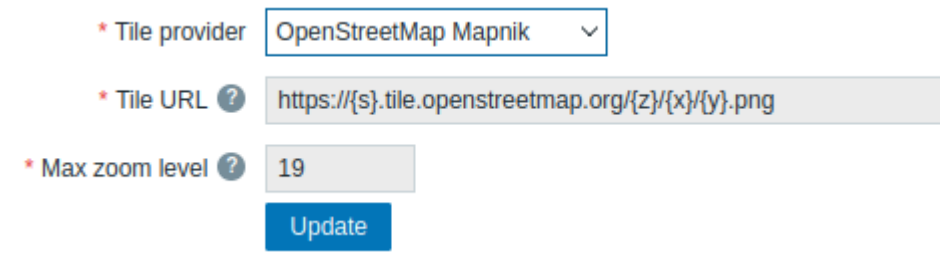
Trigeru attēlošanas opcijas

Šī sadaļa ļauj pielāgot, kā trigera stāvoklis tiek attēlots lietotāja saskarnē, kā arī trigera nopietnības nosaukumus un krāsas.

Parametrs	Apraksts
Izmantot pielāgotas notikumu statusa krāsas	Atzīmējot šo parametru, tiek ieslēgta apstiprinātu/neapstiprinātu problēmu krāsu pielāgošana.
Neapstiprināti PROBLEM notikumi, *Apstiprināti PROBLEM notikumi*, *Neapstiprināti RESOLVED notikumi*, *Apstiprināti RESOLVED notikumi*	Ievadiet jaunu krāsas kodu vai noklikšķiniet uz krāsas, lai izvēlētos jaunu no piedāvātās paletes. Ja izvēles rūtiņa *blinking* ir atzīmēta, trigeri kādu laiku mirgos pēc stāvokļa maiņas, lai tie būtu labāk pamanāmi.
Rādīt OK trigerus	Laika periods, cik ilgi rādīt OK trigerus. Atļautais diapazons: 0 - 24 stundas. Tiek atbalstīti laika sufiksi, piemēram, 5m, 2h, 1d.
Mainoties statusam, trigeri mirgo	Trigera mirgošanas ilgums. Atļautais diapazons: 0 - 24 stundas. Tiek atbalstīti laika sufiksi, piemēram, 5m, 2h, 1d.
Neklasificēts, *Informācija*, *Brīdinājums*, *Vidējs*, *Augsts*, *Katastrofa*	Pielāgoti nopietnības nosaukumi un/vai krāsas, kas tiks attēlotas sistēmas noklusējuma vērtību vietā. Ievadiet jaunu krāsas kodu vai noklikšķiniet uz krāsas, lai izvēlētos jaunu no piedāvātās paletes. Ņemiet vērā, ka šeit ievadītie pielāgotie nopietnības nosaukumi tiks izmantoti visās lokalizācijās. Ja noteiktiem lietotājiem tie ir jātulko citās valodās, skatiet lapu Trigera nopietnību pielāgošana.

Ģeogrāfiskās kartes

Šajā sadaļā var izvēlēties ģeogrāfisko karšu flīžu pakalpojuma sniedzēju un konfigurēt pakalpojuma sniedzēja iestatījumus Geomap paneļa logrīkam. Lai nodrošinātu vizualizāciju, izmantojot ģeogrāfiskās kartes, Zabbix izmanto atvērtā pirmkoda JavaScript interaktīvo karšu bibliotēku Leaflet. Lūdzu, ņemiet vērā, ka Zabbix nekontrolē trešo pušu flīžu pakalpojumu sniedzēju, tostarp iepriekš definēto flīžu pakalpojumu sniedzēju, nodrošināto attēlu kvalitāti.



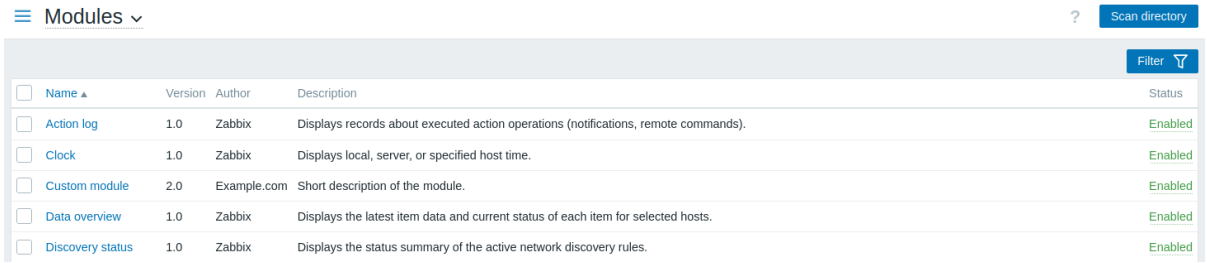
Parametrs	Apraksts
Flīžu pakalpojuma sniedzējs	Izvēlieties vienu no pieejamajiem flīžu pakalpojuma sniedzējiem vai izvēlieties *Cits*, lai pievienotu citu flīžu pakalpojuma sniedzēju vai paša mitinātas flīzes (skatiet Pielāgota flīžu pakalpojuma sniedzēja izmantošana).
Flīžu URL	URL veidne (līdz 2048 rakstzīmēm) flīžu slāņa ielādei un attēlošanai ģeogrāfiskajās kartēs. Šo lauku var rediģēt tikai tad, ja *Flīžu pakalpojuma sniedzējs* ir iestatīts uz *Cits*. Tiek atbalstīti šādi vietturi: *{s}* apzīmē vienu no pieejamajiem apakšdomēniem; *{z}* apzīmē URL tālummaiņas līmeņa parametru; *{x}* un *{y}* apzīmē flīzes koordinātas; *{r}* var izmantot, lai URL pievienotu "@2x" un ielādētu retina flīzes. Piemērs: `https://{s}.example.com/{z}/{x}/{y}{r}.png`
Atsauces teksts	Flīžu pakalpojuma sniedzēja atsauces teksts, kas tiek parādīts nelielā teksta lodziņā kartē. Šis lauks ir redzams tikai tad, ja *Flīžu pakalpojuma sniedzējs* ir iestatīts uz *Cits*.
Maksimālais tālummaiņas līmenis	Kartes maksimālais tālummaiņas līmenis. Šo lauku var rediģēt tikai tad, ja *Flīžu pakalpojuma sniedzējs* ir iestatīts uz *Cits*.

Pielāgota flīžu pakalpojuma sniedzēja izmantošana

Geomap logrīks spēj ielādēt rastra flīžu attēlus no pielāgota pašmitināta vai trešās puses flīžu pakalpojuma sniedzēja. Lai izmantotu pielāgotu trešās puses flīžu pakalpojuma sniedzēju vai pašmitinātu flīžu mapi vai serveri, laukā *Tile provider* atlasiet *Other* un norādiet pielāgoto URL laukā *Tile URL*, izmantojot atbilstošus vietturus.

Moduļi

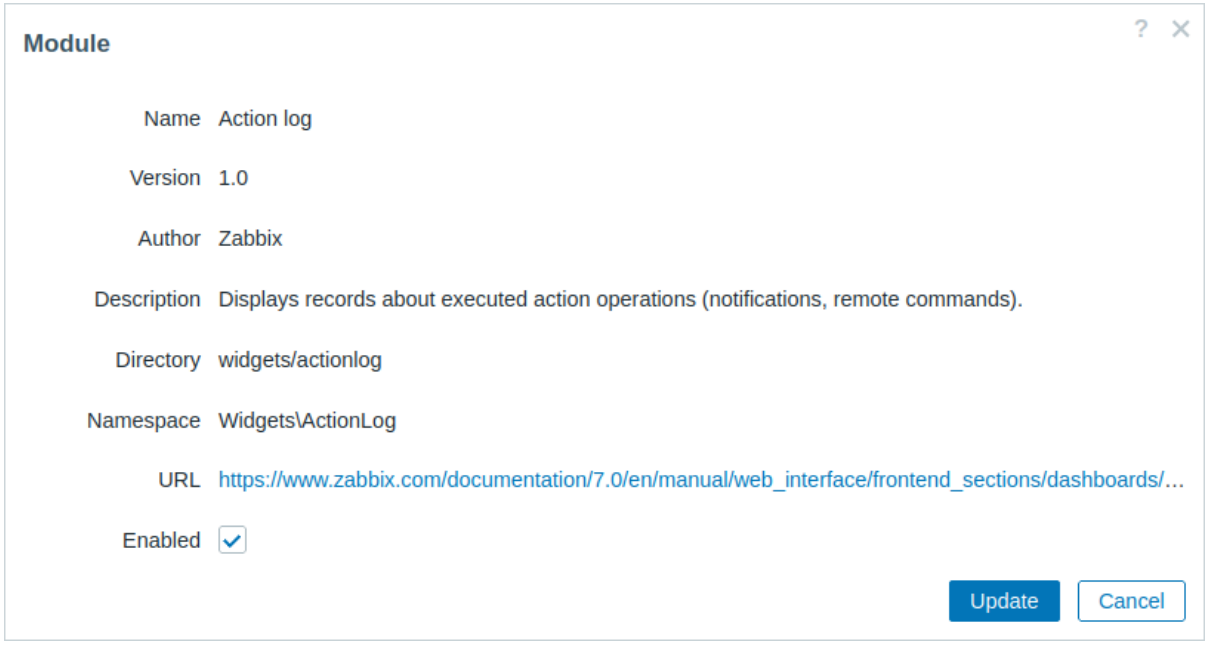
Šī sadaļa ļauj administrēt pielāgotus, kā arī iebūvētus lietotāja saskarnes moduļus.



Noklikšķiniet uz *Scan directory*, lai reģistrētu/noņemtu reģistrāciju jebkuriem pielāgotiem moduļiem. Reģistrētie moduļi tiks parādīti sarakstā; nereģistrētie moduļi tiks noņemti no saraksta.

Noklikšķiniet uz moduļa statusa sarakstā, lai iespējotu/atspējotu moduli. Varat arī masveidā iespējot/atspējot moduļus, atlasot tos sarakstā un pēc tam noklikšķinot uz pogām *Enable/Disable* zem saraksta.

Noklikšķiniet uz moduļa nosaukuma sarakstā, lai uznirstošajā logā skatītu tā detalizēto informāciju.

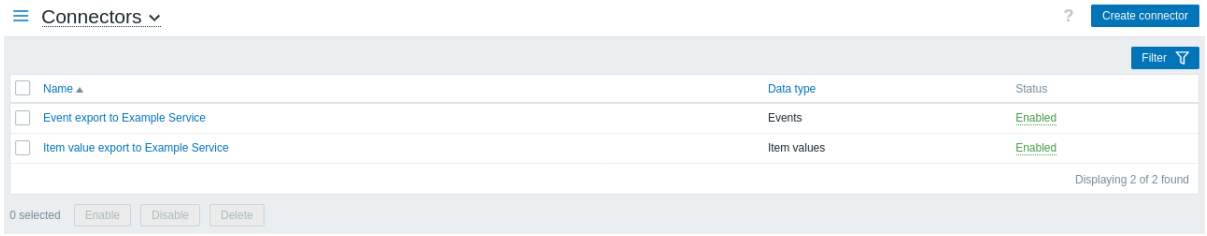


Moduļa statusu var atjaunināt arī moduļa detalizētās informācijas uznirstošajā logā; lai to izdarītu, atzīmējiet/noņemiet atzīmi izvēles rūtiņā *Enabled* un pēc tam noklikšķiniet uz *Update*.

Varat filtrēt moduļus pēc nosaukuma vai statusa (iespējots/atspējots).

Savienotāji

Šajā sadaļā var konfigurēt savienotājus Zabbix datu straumēšanai uz ārējām sistēmām, izmantojot HTTP.



Noklikšķiniet uz *Izveidot savienotāju*, lai konfigurētu jaunu savienotāju.

Savienotājus var filtrēt pēc nosaukuma vai statusa (iespējots/atspējots). Noklikšķiniet uz savienotāja statusa sarakstā, lai iespējotu/atspējotu savienotāju.
Savienotājus var arī masveidā iespējot/atspējot, atlasot tos sarakstā un pēc tam noklikšķinot uz pogām *Iespējot/Atspējot* zem saraksta.

Citi

Šajā sadaļā var konfigurēt dažādus citus lietotāja saskarnes parametrus.

Parametrs	Apraksts
Lietotāja saskarnes URL	Zabbix tīmekļa saskarnes URL (līdz 2048 rakstzīmēm). Šo parametru Zabbix tīmekļa serviss izmanto saziņai ar lietotāja saskarni, un tas ir jānorāda, lai iespējotu plānotos pārskatus.
Grupa atklātajiem hostiem	Hosti, kas atklāti ar tīkla atklāšanu un aģenta automātisko reģistrāciju, tiks automātiski ievietoti šeit izvēlētajā hostu grupā.
Noklusējuma hosta inventarizācijas režīms	Noklusējuma režīms hosta inventarizācijai. Tas tiks izmantots ikreiz, kad serveris vai lietotāja saskarne izveido jaunu hostu vai hosta prototipu, ja vien hosta atklāšanas/automātiskās reģistrācijas laikā to nepārraksta darbība *Iestatīt hosta inventarizācijas režīmu*.

Parametrs	Apraksts
Lietotāju grupa datubāzes nedarbošanās ziņojumam	Lietotāju grupa trauksmes ziņojuma nosūtīšanai vai 'Nav'. Zabbix serveris ir atkarīgs no aizmugursistēmas datubāzes pieejamības. Bez datubāzes tas nevar darboties. Ja datubāze nav pieejama, Zabbix var paziņot par to izvēlētajiem lietotājiem. Paziņojumi tiks nosūtīti šeit norādītajai lietotāju grupai, izmantojot iespējotos lietotāju mediju ierakstus. Paziņojumi tiek pārraidīti, izmantojot šādus mediju tipus (ja tie ir iespējoti): e-pasts; SMS; pielāgoti brīdinājumu skripti. Pat ja webhook medija ieraksts ir konfigurēts un iespējots, tas neizraisīs paziņojumus. Zabbix serveris netiks apturēts; tas gaidīs, līdz datubāze atkal būs pieejama, lai turpinātu apstrādi. Paziņojums sastāv no šāda satura: `[MySQL|PostgreSQL] database <DB Name> [on <DB Host>:<DB Port>] is not available: <error message depending on the type of DBMS (database)>` <DB Host> netiek pievienots ziņojumam, ja tas ir definēts kā tukša vērtība, un <DB Port> netiek pievienots, ja tā ir noklusējuma vērtība ("0"). Brīdinājumu pārvaldnieks (īpašs Zabbix servera process) mēģina izveidot jaunu savienojumu ar datubāzi ik pēc 10 sekundēm. Ja datubāze joprojām nav pieejama, brīdinājumu pārvaldnieks atkārto brīdinājumu sūtīšanu, bet ne biežāk kā reizi 15 minūtēs.
Reģistrēt neatbilstošos SNMP slazdus žurnālā	Reģistrēt SNMP slazdu žurnālā, ja nav atrastas atbilstošas SNMP saskarnes.

Autorizācija

Parametrs	Apraksts
Pieteikšanās mēģinājumi	Neveiksmīgu pieteikšanās mēģinājumu skaits, pēc kura tiek bloķēta iespēja pieteikties.
Pieteikšanās bloķēšanas intervāls	Laika periods, kurā pieteikšanās būs aizliegta, ja tiek pārsniegts *Pieteikšanās mēģinājumi* limits. Atļautais diapazons: 30 - 3600 sekundes. Tiek atbalstīti laika sufiksi, piemēram, 90s, 5m, 1h.

Noslēpumu glabāšana

Parameter	Description
Vault provider	Izvēlieties noslēpumu pārvaldības programmatūru lietotāja makro vērtību glabāšanai — *HashiCorp Vault* (noklusējums) vai *CyberArk Vault*.
Resolve secret vault macros by	Atrisināt noslēpumu glabātuves makro, izmantojot: **Zabbix serveris** — noslēpumi tiek iegūti ar Zabbix serveris un, ja nepieciešams, pārsūtīti starpniekserveriem (noklusējums); **Zabbix serveris and proxy** — noslēpumi tiek iegūti gan ar Zabbix serveris, gan starpniekserveriem, ļaujot tiem atrisināt makro neatkarīgi.

Skatiet arī: Noslēpumu glabāšana.

Drošība

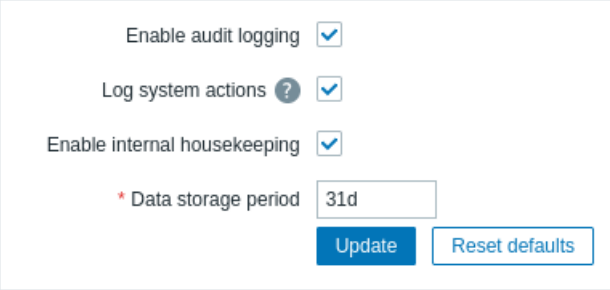
Parametrs	Apraksts
Validate URI schemes	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu URI shēmu validāciju (pēc noklusējuma iespējota). Ja atzīmēta, varat norādīt ar komatiem atdalītu atļauto URI shēmu sarakstu (pēc noklusējuma: http,https,ftp,file,mailto,tel,ssh). Attiecas uz visiem laukiem lietotāja saskarnē, kuros tiek izmantoti URI (piemēram, kartes elementu URL).

Parametrs	Apraksts
Use X-Frame-Options HTTP header	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu HTTP X-Frame-options galveni (nav ieteicams). Ja atzīmēta, varat norādīt HTTP X-Frame-options galvenes vērtību. Atbalstītās vērtības: **SAMEORIGIN** (pēc noklusējuma) vai **'self'** (jāliek vienpēdiņās) — lapu var attēlot rāmī tikai tajā pašā avotā kā pati lapa; **DENY** vai **'none'** (jāliek vienpēdiņās) — lapu nevar attēlot rāmī neatkarīgi no vietnes, kas mēģina to darīt; **virkne ar atstarpēm atdalītiem hostu nosaukumiem**; pievienojot sarakstam **'self'** (jāliek vienpēdiņās), lapa var tikt attēlota rāmī tajā pašā avotā kā pati lapa. Ņemiet vērā, ka, izmantojot **'self'** vai **'none'** bez vienpēdiņām, tie tiks uzskatīti par hostu nosaukumiem.
Use iframe sandboxing	Noņemiet atzīmi no šīs izvēles rūtiņas, lai atspējotu iegūtā URL satura ievietošanu smilškastē (nav ieteicams). Ja atzīmēta, varat norādīt iframe smilškastes izņēmumus; nenorādītie ierobežojumi joprojām tiks piemēroti. Ja šis lauks ir tukšs, tiek piemēroti visi sandbox atribūta ierobežojumi. Papildinformāciju skatiet atribūta `sandbox` aprakstā.

2 Audita žurnāls

Pārskats

Šajā sadaļā var konfigurēt audita žurnāla iestatījumus.



Ir pieejami šādi parametri:

Parametrs	Apraksts
Iespējot audita žurnalēšanu	Iespējot (noklusējums) vai atspējot audita žurnalēšanu.
Reģistrēt sistēmas darbības	Iespējot (noklusējums) vai atspējot servera (sistēmas lietotāja) veikto zema līmeņa atklāšanas, tīkla atklāšanas un automātiskās reģistrācijas darbību audita žurnalēšanu.
Iespējot iekšējo uzturēšanu	Iespējot (noklusējums) vai atspējot audita žurnāla ierakstu iekšējo uzturēšanu.
Datu glabāšanas periods	Dienu skaits, cik ilgi audita žurnāla ieraksti jāsaglabā, pirms tos noņem uzturēšanas process. Obligāts, ja uzturēšana ir iespējota. Noklusējums: 31 diena.

Audita žurnāla saspiešanas parametri ir pieejami sadaļā *Administrēšana > Uzturēšana*, blokā *Vēstures, tendenču un audita žurnāla saspiešana*, kas kļūst redzams, ja tiek izmantots TimescaleDB.

3 Uzturēšana

Pārskats

Sadaļa *Administration > Housekeeping* ļauj konfigurēt Zabbix iekšējo datu tīrīšanu, kas automātiski noņem no datubāzes novecojušus vai lietotāja dzēstus datus. Tas novērš neierobežotu datubāzes pieaugumu, kas patērētu nevajadzīgu diska vietu un pasliktinātu datubāzes veiktspēju.

Datu tīrīšanu var iespējot un konfigurēt šādiem datu tipiem:

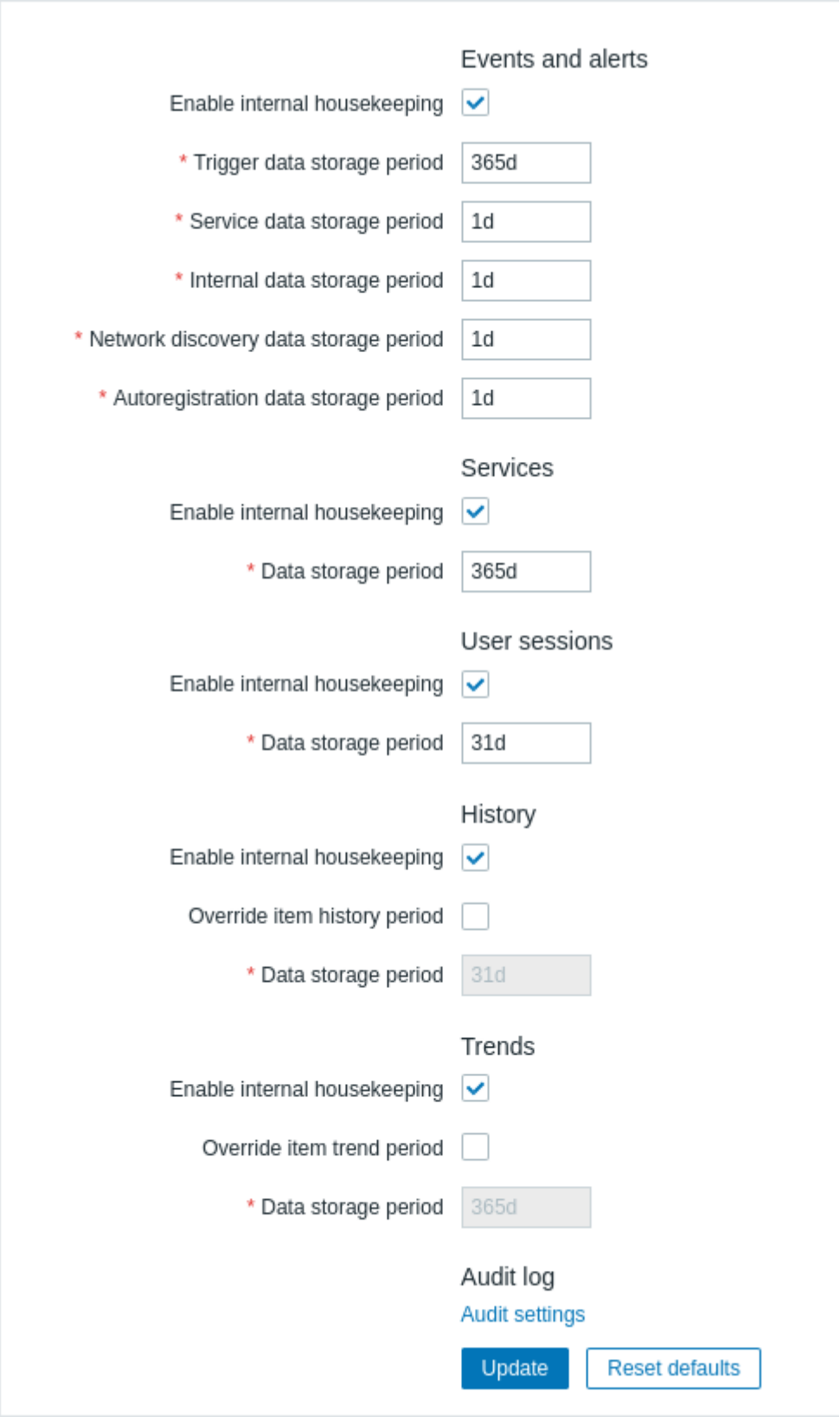
- Notikumi un brīdinājumi no trigeriem, pakalpojumiem, iekšējiem datiem, tīkla atklāšanas un automātiskās reģistrācijas
- Pakalpojumi

- Lietotāju sesijas
- Vēsture un tendences vienumiem

Note:
Audita žurnāla datu tīrīšana tiek konfigurēta atsevišķi.

Konfigurācija

Ir pieejami šādi parametri:



Parametrs	Apraksts
Iespējot iekšējo housekeeper	Iespējojiet vai atspējojiet iekšējo housekeeper (pēc noklusējuma iespējots). Ja tas ir iespējots, housekeeping procedure automātiski noņem no datubāzes datus, kas pārsniedz *Datu glabāšanas periodu*.
Datu glabāšanas periods	Norādiet, cik ilgi saglabāt datus, pirms housekeeper tos noņem. Obligāts, ja ir iespējots iekšējais housekeeper. Diapazons: 1 diena (1 stunda vēsturei) - 25 gadi; vai ”0”. Tiek atbalstīti laika sufiksi (piemēram, 1d, 1w). *Notikumiem un brīdinājumiem* datu glabāšanas periods tiek iestatīts atsevišķi trigeriem, pakalpojumiem, iekšējiem datiem, tīkla atklāšanai un automātiskajai reģistrācijai. Housekeeper noņem tikai tos notikumus, kas nav saistīti ar problēmām. Piemēram, problēmas/atkopšanas notikums, kas ir vecāks par *Datu glabāšanas periodu*, netiks noņemts, ja tas ir saistīts ar neatrisinātu problēmu. Kad housekeeper noņem novecojušas entītijas, tas vispirms noņem problēmas un pēc tam notikumus. *Vēsturei* un *Tendencēm* datu glabāšanas periodi nosaka arī to, cik ilgi dati paliek redzami sadaļā *Uzraudzība > Jaunākie dati*, pat ja iekšējais housekeeper ir atspējots. Ņemiet vērā, ka, dzēšot vienumu vai trigeri, tiek dzēstas arī ar tiem saistītās problēmas un notikumi.
Pārrakstīt vienuma vēstures periodu	Ja iespējots, vienuma konfigurācijā norādītais vēstures/tendenču glabāšanas periods tiek pārrakstīts ar iestatījumu *Datu glabāšanas periods* (izņemot vienumus, kuriem ir iespējota opcija *Neglabāt*). Šo opciju var izmantot arī tad, ja iekšējais housekeeper ir atspējots un tiek izmantots ārējs housekeeper.
Pārrakstīt vienuma tendenču periodu	

Ja tiek izmantots **TimescaleDB**, kļūst pieejama sadaļa *Vēstures, tendenču un audita žurnāla saspiešana*.

Note:
Lai pilnībā izmantotu automātiskās particionēšanas priekšrocības ar TimescaleDB, iespējojiet *Pārrakstīt vienuma vēstures periodu*, *Pārrakstīt vienuma tendenču periodu* un *Iespējot iekšējo housekeeper* vēsturei un tendencēm. Ja šīs opcijas ir atspējotas, vēstures un tendenču tabulās glabātie dati joprojām tiks particionēti, taču housekeeper nenodzēsīs novecojušās particijas un tiks parādīti konfigurācijas brīdinājumi. Ja ir iespējota novecojušo particiju dzēšana, Zabbix serveris un lietotāja saskarne vairs nesekos dzēstajiem vienumiem, un šo vienumu vēsture tiks notīrīta, kad tiks dzēsta novecojusi particija.

Poga *Atiestatīt noklusējuma vērtības* ļauj atjaunot visas veiktās izmaiņas.

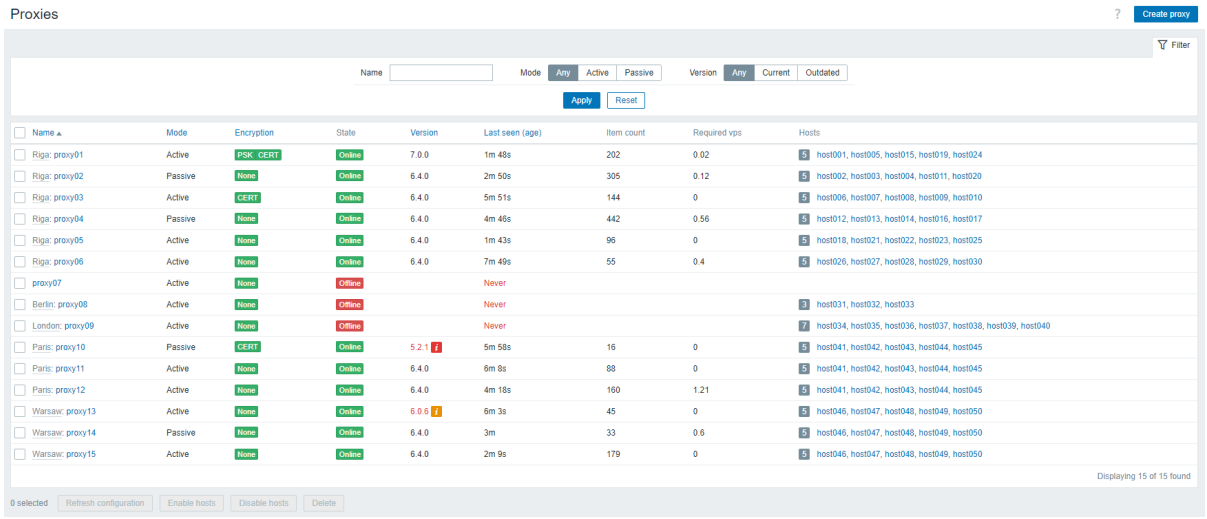
4 Starpniekserveri

Pārskats

Sadaļā *Administrēšana → Starpniekserveri* Zabbix lietotāja saskarnē var konfigurēt starpniekserverus izkliedētajai uzraudzībai.

Starpniekserveri

Tiek parādīts esošo starpniekserveru saraksts ar to detalizēto informāciju.



Parādītie dati:

Kolonna	Apraksts
Name	Starpniekservera nosaukums. Noklikšķinot uz starpniekservera nosaukuma, tiek atvērta starpniekservera konfigurācijas forma. Ja starpniekserveris pieder starpniekserveru grupai, grupas nosaukums tiek parādīts pirms starpniekservera nosaukuma kā pelēka saite. Noklikšķinot uz grupas nosaukuma, tiek atvērta starpniekserveru grupas konfigurācijas forma.
Mode	Starpniekservera režīms — *Active* vai *Passive*.
Encryption	Šifrēšanas statuss savienojumiem no starpniekservera: **None** — bez šifrēšanas; **PSK** — izmantojot iepriekš koplietotu atslēgu; **Cert** — izmantojot sertifikātu.
State	Starpniekservera stāvoklis: **Unknown** — starpniekserveris tika izveidots, kamēr Zabbix serveris nedarbojās, vai arī serveris vēl nav atjauninājis stāvokli; **Online** — starpniekserveris ir sazinājies ar Zabbix serveri pārslēgšanās atteices perioda laikā (pasīvais starpniekserveris atbildēja uz servera pieprasījumu; aktīvais starpniekserveris nosūtīja pieprasījumu); **Offline** — starpniekserveris nav sazinājies ar Zabbix serveri pārslēgšanās atteices perioda laikā.
Version	Starpniekservera versija (trīsciparu versijas numurs). Ja starpniekserveris ir novecojis vai netiek atbalstīts, versijas numurs tiek izcelts (sarkanā krāsā) un tiek parādīta informācijas statusa ikona (dzeltena vai sarkana). Lai skatītu detalizētu informāciju, novietojiet kursoru virs ikonas.
Last seen (age)	Laiks, kad serveris pēdējo reizi redzēja starpniekserveri.
Item count	Iespējoto vienumu skaits iespējotajos hostos, kas piešķirti starpniekserverim.
Required vps	Nepieciešamā starpniekservera veiktspēja (vērtību skaits, kas jāsavāc sekundē).
Hosts	Iespējoto hostu skaits, kas piešķirti starpniekserverim, un starpniekservera uzraudzīto hostu saraksts. Noklikšķinot uz hosta nosaukuma, tiek atvērta hosta konfigurācijas forma.

Lai konfigurētu jaunu starpniekserveri, augšējā labajā stūrī noklikšķiniet uz pogas *Create proxy*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

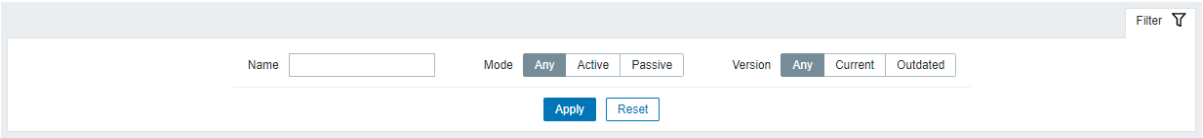
- *Atsvaidzināt konfigurāciju* - atsvaidzināt starpniekserveru konfigurāciju;
- *Iespējot hostus* - mainīt starpniekservera uzraudzīto hostu statusu uz *Uzraudzīts*;
- *Atspējot hostus* - mainīt starpniekservera uzraudzīto hostu statusu uz *Neuzraudzīts*;
- *Dzēst* - dzēst starpniekserverus.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajiem starpniekserveriem un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Jūs varat izmantot filtru, lai parādītu tikai tos starpniekserverus, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makrosu vērtībām.

Saite *Filtrs* ir pieejama virs starpniekserveru saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt starpniekserverus pēc nosaukuma, režīma un versijas. Ņemiet vērā, ka filtra opcija *Novecojis* parāda gan novecojušus (daļēji atbalstītus), gan neatbalstītus starpniekserverus.



5 Starpniekserveru grupas

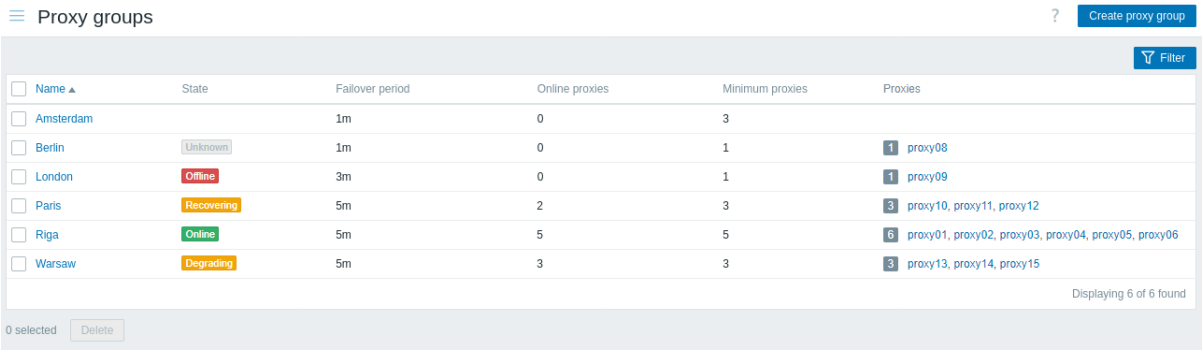
Pārskats

Sadaļā *Administrēšana* → *Starpniekserveru grupas* ir iespējams konfigurēt starpniekserveru grupas.

Starpniekserveru grupas tiek izmantotas starpniekserveru slodzes līdzsvarošanā ar automatizētu hostu sadali starp starpniekserveriem un augstu pieejamību starp starpniekserveriem.

Starpniekserveru grupas

Tiek parādīts esošo starpniekserveru grupu saraksts ar detalizētu informāciju.



Parādītie dati:

Kolonna	Apraksts
Name	Starpniekserveru grupas nosaukums. Noklikšķinot uz starpniekserveru grupas nosaukuma, tiek atvērta starpniekserveru grupas konfigurācijas forma.
State	Starpniekserveru grupas stāvoklis (tiek parādīts, ja grupā ir vismaz viens starpniekserveris): **Unknown** — grupa tika izveidota, kamēr Zabbix serveris nedarbojās, vai arī serveris vēl nav atjauninājis stāvokli; **Online** — minimālais starpniekserveru skaits ir sazinājies ar Zabbix serveri atteices pārslēgšanās periodā; **Degrading** — grupa drīzumā pāries bezsaistes režīmā, jo tiešsaistē esošo starpniekserveru skaits ir zem *Minimum proxies* sliekšņa; **Offline** — mazāk nekā minimālais starpniekserveru skaits ir sazinājies ar Zabbix serveri atteices pārslēgšanās periodā; **Recovering** — grupa drīzumā atgriezīsies tiešsaistē, jo tiešsaistē esošo starpniekserveru skaits ir sasniedzis *Minimum proxies* slieksni. Starpniekserveru grupas stāvokli var uzraudzīt ar zabbix[proxy group,<name>,state] un zabbix[proxy group,discovery] vienumiem.
Failover period	Periods, kura laikā starpniekserverim starpniekserveru grupā ir jāsazinās ar Zabbix serveri, lai tas tiktu uzskatīts par tiešsaistē esošu.
Online proxies	Tiešsaistē esošo starpniekserveru skaits (tiek parādīts sarkanā krāsā, ja tas ir zem grupas minimuma).
Minimum proxies	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa paliktu tiešsaistē.
Proxies	Starpniekserveru skaits grupā un grupas starpniekserveru saraksts ar saitēm uz starpniekserveru konfigurācijas formu. Maksimālo uzskaitīto starpniekserveru skaitu ierobežo *Max count of elements to show inside table cell* vērtība.

Lai konfigurētu jaunu starpniekserveru grupu, augšējā labajā stūrī noklikšķiniet uz pogas *Create proxy groups*.

Masveida rediģēšanas opcijas

Pogas zem saraksta piedāvā dažas masveida rediģēšanas opcijas:

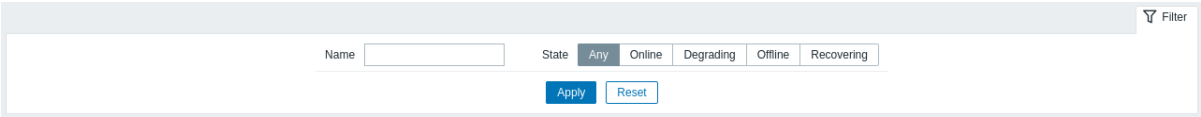
- *Delete* - dzēst starpniekserveru grupas.

Lai izmantotu šīs opcijas, atzīmējiet izvēles rūtiņas pie attiecīgajām starpniekserveru grupām un pēc tam noklikšķiniet uz vajadzīgās pogas.

Filtra izmantošana

Jūs varat izmantot filtru, lai attēlotu tikai tās starpniekserveru grupas, kas jūs interesē. Labākai meklēšanas veiktspējai dati tiek meklēti ar neatrisinātām makro vērtībām.

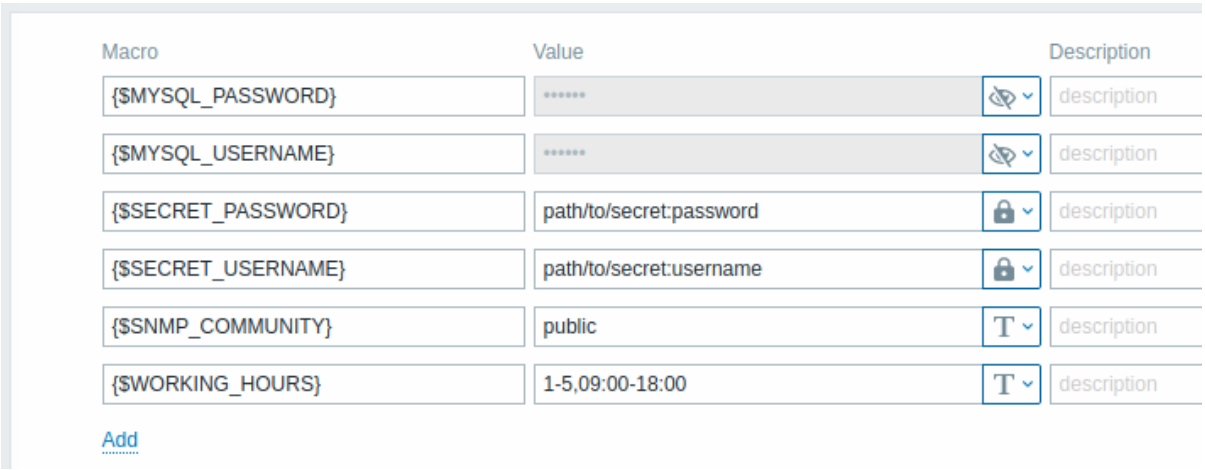
Saite *Filtrs* ir pieejama virs starpniekserveru grupu saraksta. Noklikšķinot uz tās, kļūst pieejams filtrs, kurā varat filtrēt starpniekserveru grupas pēc nosaukuma un statusa.



6 Makrosi

Pārskats

Šajā sadaļā var definēt sistēmas mēroga lietotāja makrosus kā nosaukuma-vērtības pārus. Ņemiet vērā, ka makrosu vērtības var glabāt kā vienkāršu tekstu, slepenu tekstu vai Vault noslēpumu. Tiek atbalstīta arī apraksta pievienošana.



7 Rinda

Pārskats

Sadaļā *Administrēšana* → *Rinda* tiek attēloti vienumi, kas gaida atjaunināšanu.

Ideālā gadījumā, atverot šo sadaļu, visam vajadzētu būt ”zaļam”, kas nozīmē, ka rindā nav neviena vienuma. Ja visi vienumi tiek atjaunināti bez aizkaves, neviens negaida rindā. Tomēr nepietiekamas servera veiktspējas dēļ daži vienumi var aizkavēties, un informācija par to tiek attēlota šajā sadaļā. Plašāku informāciju skatiet sadaļā Rinda.

Note:
Zabbix starpniekserveris neņem vērā uzturēšanas periodus; sīkāku informāciju skatiet sadaļā Rindu aprēķināšana uzturēšanas laikā.

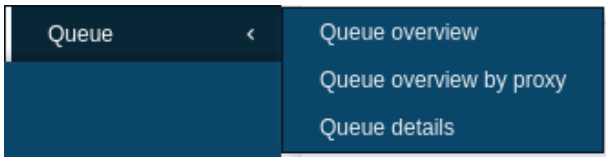
Note:
Rinda ir pieejama tikai tad, ja darbojas Zabbix serveris. Vienumi netiek ieskaitīti rindā, ja vienuma saskarne kļūst nepieejama savienojuma problēmu dēļ vai ja aģents nedarbojas pareizi.

Sadaļā *Administrēšana* → *Rinda* ir pieejamas šādas lapas:

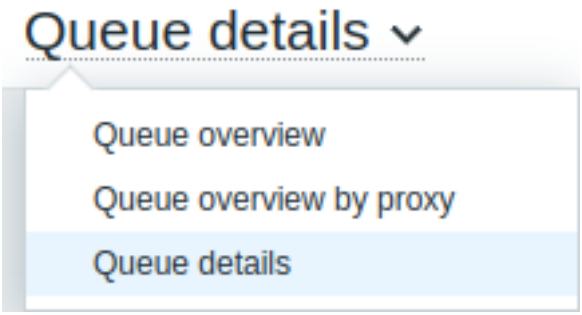
- Rindas pārskats — attēlo rindu pēc vienuma tipa;
- Rindas pārskats pēc starpniekservera — attēlo rindu pēc starpniekservera;

- Rindas detalizēta informācija — attēlo aizkavēto vienumu sarakstu.

Pieejamo lapu saraksts tiek parādīts, nospiežot *Rinda* izvēlnes sadaļā *Administrēšana*. Ir iespējams arī pārslēgties starp lapām, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.



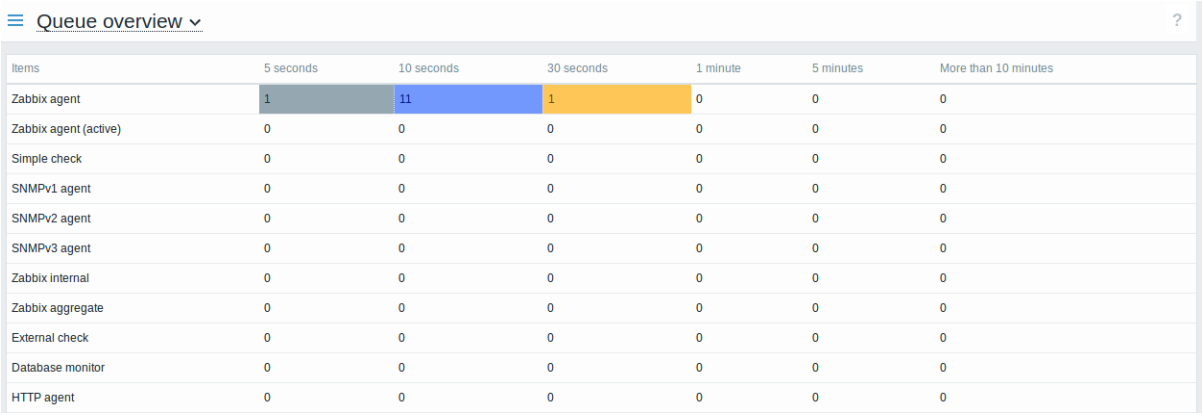
Trešā līmeņa izvēlne.



Virsraksta nolaižamais saraksts.

Pārskats pēc vienuma tipa

Šajā ekrānā ir viegli noteikt, vai problēma ir saistīta ar vienu vai vairākiem vienumu tipiem.

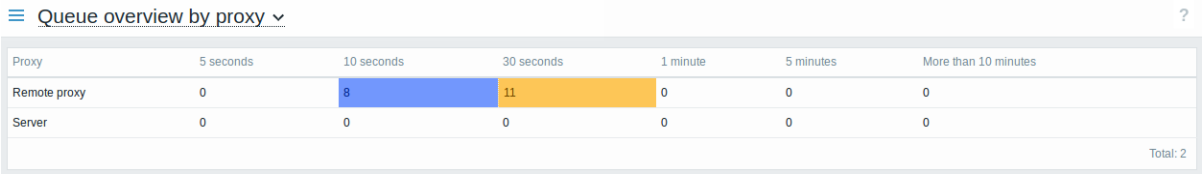

Queue overview

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

Katra rinda satur vienuma tipu. Katra kolonna attēlo gaidošo vienumu skaitu — attiecīgi gaida 5–10 sekundes/10–30 sekundes/30–60 sekundes/1–5 minūtes/5–10 minūtes vai vairāk nekā 10 minūtes.

Pārskats pēc starpniekservera

Šajā ekrānā ir viegli noteikt, vai problēma ir saistīta ar kādu no starpniekserveriem vai serveri.


Queue overview by proxy

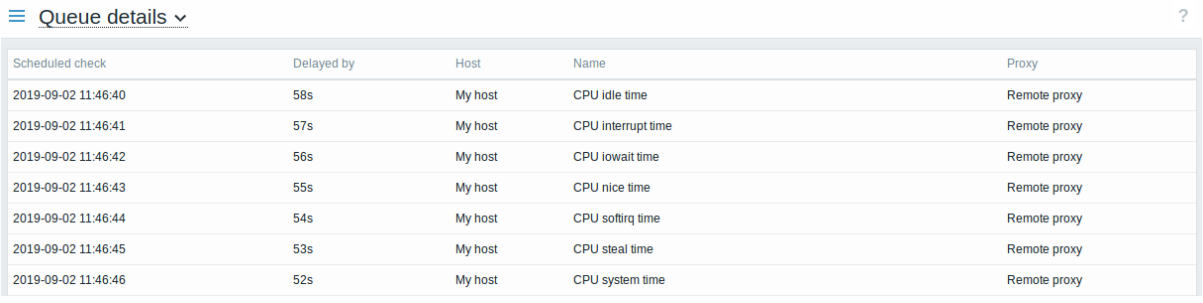
Proxy	5 seconds	10 seconds	30 seconds	1 minute	5 minutes	More than 10 minutes
Remote proxy	0	8	11	0	0	0
Server	0	0	0	0	0	0

Total: 2

Katrā rindā ir attēlots starpniekserveris, bet saraksta beigās — serveris. Katra kolonna parāda gaidošo vienumu skaitu — attiecīgi gaida 5–10 sekundes/10–30 sekundes/30–60 sekundes/1–5 minūtes/5–10 minūtes vai vairāk nekā 10 minūtes.

Gaidīšanas vienumu saraksts

Šajā ekrānā ir uzskaitīts katrs gaidošais vienums.


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

Attēlotie dati:

Kolonna	Apraksts
Plānotā pārbaude	Tiek attēlots laiks, kad pārbaudei bija jānotiek.
Aizkavēts par	Tiek attēlots aizkaves ilgums.
Hosts	Tiek attēlots vienuma hosts.
Nosaukums	Tiek attēlots gaidošā vienuma nosaukums.
Starpniekserveris	Tiek attēlots starpniekservera nosaukums, ja hosts tiek uzraudzīts ar starpniekserveri.

Iespējamie kļūdu ziņojumi

Jūs varat saskarties ar situāciju, kad netiek attēloti nekādi dati un parādās šāds kļūdas ziņojums:

Kļūdas ziņojums šajā gadījumā ir šāds:

```
Cannot display item queue. Permission denied
```

Tas notiek, ja PHP konfigurācijas parametri failā *zabbix.conf.php* — `$ZBX_SERVER` vai abi `$ZBX_SERVER` un `$ZBX_SERVER_PORT` — norāda uz esošu Zabbix serveri, kas izmanto citu datubāzi.

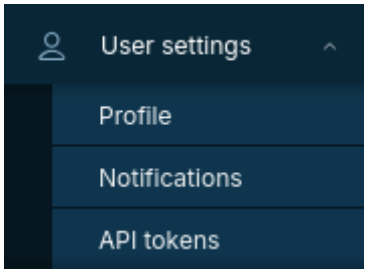
3 Lietotāja iestatījumi

Pārskats

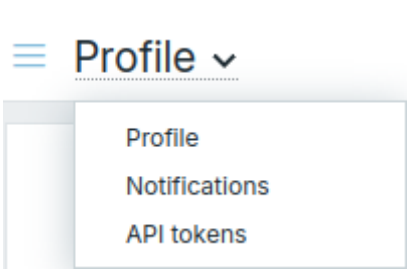
Atkarībā no lietotāja lomas atļaujām sadaļā *Lietotāja iestatījumi* var būt šādas lapas:

- *Profils* vai *Lietotāja profils* — noteiktu Zabbix lietotāja saskarnes funkciju pielāgošanai;
- *Paziņojumi* — pašreizējā lietotāja paziņojumu pielāgošanai;
- *API marķieri* — pašreizējam lietotājam piešķirto API marķieru pārvaldībai.

Pieejamo lapu saraksts tiek parādīts, noklikšķinot uz lietotāja ikonas Zabbix izvēlnes apakšdaļas tuvumā (nav pieejams *guest* lietotājam). Starp lapām iespējams arī pārslēgties, izmantojot virsraksta nolaižamo sarakstu augšējā kreisajā stūrī.



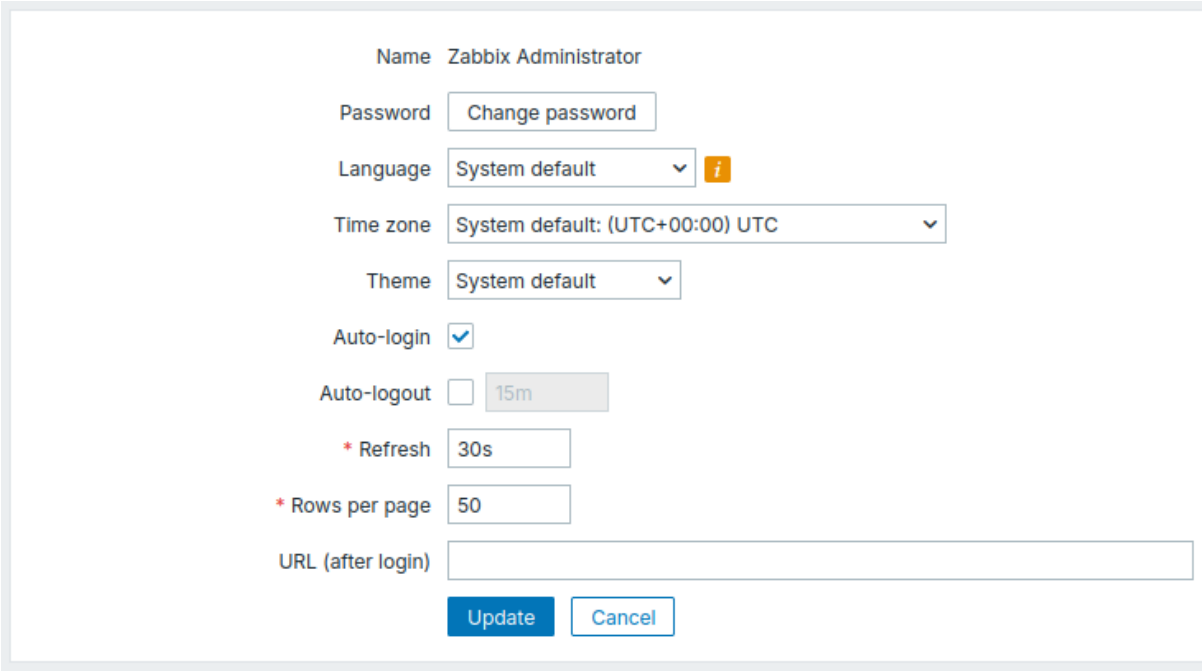
Trešā līmeņa izvēlne.



Virsraksta nolaižamais saraksts.

Lietotāja profils

Sadaļā *Lietotāja profils* ir pieejamas opcijas, lai iestatītu pielāgotu lietotāja saskarnes valodu, krāsu tēmu, sarakstos attēloto rindu skaitu u.c. Šeit veiktās izmaiņas tiks piemērotas tikai pašreizējam lietotājam.



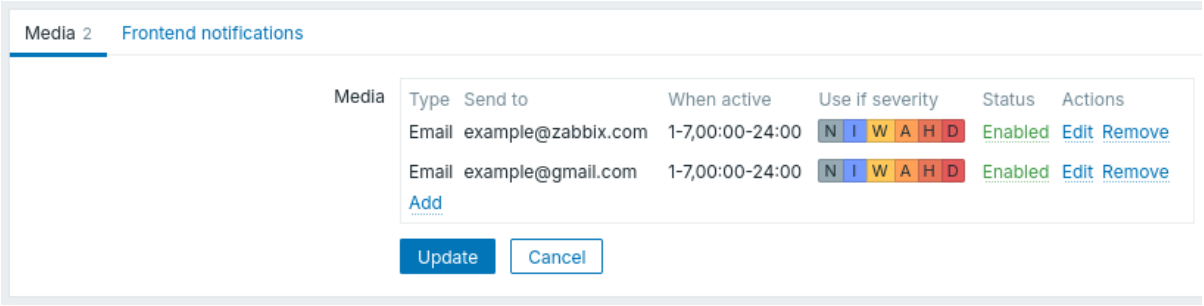
Parametrs	Apraksts
Name	Tiek attēlots lietotāja vārds un uzvārds. Ja vārds un uzvārds nav norādīti, tiek attēlots lietotājvārds.
Password	Noklikšķiniet uz pogas *Change password*, lai atvērtu trīs laukus: *Old password*, *New password*, *New password (once again)*. Ja parole tiek veiksmīgi nomainīta, lietotājs tiks izrakstīts no visām aktīvajām sesijām. Ņemiet vērā, ka paroli var mainīt tikai lietotājiem, kuri izmanto Zabbix iekšējo autentifikāciju.
Language	Izvēlieties vēlamo lietotāja saskarnes valodu vai atlasiet **System default**, lai izmantotu noklusējuma sistēmas iestatījumus. Izvēloties *English (en_US)*, lietotāja saskarnē tiks iespējots arī ASV laika/datuma formāts. Papildinformāciju skatiet sadaļā Papildu lietotāja saskarnes valodu instalēšana.
Time zone	Izvēlieties laika joslu, lai lietotāja līmenī ignorētu globālo laika joslu, vai atlasiet **System default**, lai izmantotu globālos laika joslas iestatījumus.
Theme	Izvēlieties krāsu tēmu tieši savam profilam: **System default** - izmantot noklusējuma sistēmas iestatījumus; **Blue** - standarta zilā tēma; **Blue (classic)** - standarta zilā tēma, izmantojot mantotos fontus; **Dark** - alternatīva tumšā tēma; **Dark (classic)** - alternatīva tumšā tēma, izmantojot mantotos fontus; **High-contrast light** - gaišā tēma ar augstu kontrastu; **High-contrast dark** - tumšā tēma ar augstu kontrastu.
Auto-login	Atzīmējiet šo izvēles rūtiņu, lai Zabbix jūs atcerētos un automātiski pieteiktu uz 30 dienām. Piesakoties ar iespējotu *Remember for 30 days*: - Automātiskās izrakstīšanās taimeris tiek atiestatīts (sesija saglabājas līdz 30 dienu perioda beigām). - Automātiskā pieteikšanās tiek iespējota turpmākajos apmeklējumos 30 dienu laikā. Piesakoties bez *Remember for 30 days*: - Parametrs *Auto-login* tiek notīrīts. - Automātiskā izrakstīšanās notiek saskaņā ar konfigurēto taimautu kā parasti. Tam tiek izmantotas pārlūkprogrammas sīkdatnes.

Parametrs	Apraksts
Auto-logout	Ja šī izvēles rūtiņa ir atzīmēta, pēc iestatītā sekunžu skaita jūs tiksiet automātiski izrakstīts (minimums 90 sekundes, maksimums 1 diena). Tiek atbalstīti laika sufiksi, piemēram: 90s, 5m, 2h, 1d. Ņemiet vērā, ka šī opcija nedarbosies šādos gadījumos: * Kad izvēlnes *Monitoring* lapas fonā veic informācijas atsvaidzināšanu. Ja tiek atstātas atvērtas lapas, kas atsvaidzina datus noteiktā laika intervālā (paneļi, grafiki, jaunākie dati u.c.), sesijas darbības laiks attiecīgi tiek pagarināts, tādējādi atspējojot automātiskās izrakstīšanās funkciju. * Ja pieteikšanās notiek ar atzīmētu opciju *Remember me for 30 days*; šajā gadījumā automātiskās izrakstīšanās taimeris tiek ignorēts, saglabājot jūsu sesiju aktīvu visu periodu. * Ja autentifikācija tiek veikta, izmantojot SSO identitātes nodrošinātāju (IdP), automātiskā izrakstīšanās beidz tikai Zabbix sesiju. SSO sesija ar IdP paliek aktīva, tāpēc lietotājs var atkal pieteikties, neievadot paroli. *Auto-logout* var pieņemt arī vērtību "0", kas nozīmē, ka pēc profila iestatījumu atjaunināšanas automātiskās izrakstīšanās funkcija tiek atspējota.
Refresh	Iestatiet, cik bieži tiks atsvaidzināta informācija izvēlnes *Monitoring* lapās (minimums 0 sekundes, maksimums 1 stunda). Tiek atbalstīti laika sufiksi, piemēram: 30s, 90s, 1m, 1h.
Rows per page	Iestatiet, cik rindu tiks attēlots vienā sarakstu lapā. Mazāks rindu skaits (un mazāk attēlojamu ierakstu) nodrošina ātrāku ielādes laiku.
URL (after login)	Iestatiet konkrētu URL, kas tiks attēlots pēc pieteikšanās. Noklusējuma *Dashboards* vietā tas var būt, piemēram, *Monitoring* > *Triggers* URL.

Paziņojumi

Sadaļa *Paziņojumi* nodrošina iespējas pielāgot pašreizējā lietotāja paziņojumus.

Cilnē *Mediji* varat norādīt lietotāja mediju informāciju, piemēram, izmantojamos mediju tipus un adreses, kā arī to, kad tos izmantot paziņojumu piegādei.



Ja medija tips ir atspējots:

- pēc nosaukuma tiek parādīta dzeltena informācijas ikona;
- kolonnā *Statuss* tiek parādīts "Atspējots".

Note:
Lietotāja tiesības mainīt savu mediju informāciju var tikt piešķirtas/atsauktas atkarībā no viņa lietotāja lomas (skatiet opciju *Izveidot un rediģēt savus medijus*). Super administratora tiesības mainīt citu lietotāju mediju informāciju arī var tikt piešķirtas/atsauktas atkarībā no viņa lietotāja lomas (skatiet opciju *Izveidot un rediģēt lietotāja medijus*)

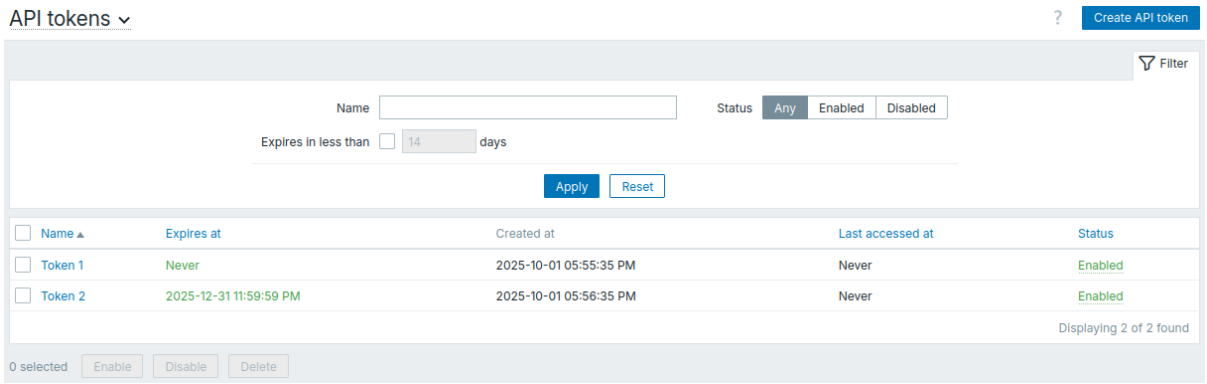
Ņemiet vērā, ka nodrošinātajiem lietotājiem:

- nodrošinātā lietotāja medijus nevar dzēst;
- nodrošinātā lietotāja medijus var atspējot/iespējot;
- tādus nodrošinātā lietotāja mediju laukus kā *Kad aktīvs*, *Izmantot, ja nopietnība* un *Iespējots* var rediģēt manuāli;
- nodrošinātajiem lietotājiem var manuāli pievienot papildu lietotāja medijus (piemēram, papildu e-pasta adreses);
- manuāli pievienotos lietotāja medijus var dzēst.

Cilne *Lietotāja saskarnes paziņojumi* ļauj iestatīt globālos paziņojumus.

API marķieri

Sadaļa *API marķieri* ļauj skatīt lietotājam piešķirtos marķierus, rediģēt marķieru informāciju un izveidot jaunus marķierus. Šī sadaļa lietotājam ir pieejama tikai tad, ja lietotāja lomas iestatījumos ir atļauta darbība *Pārvaldīt API marķierus*.



API marķierus var filtrēt pēc nosaukuma, derīguma termiņa vai statusa (*Iespējots/Atspējots*). Noklikšķiniet uz marķiera statusa sarakstā, lai ātri iespējotu/atspējotu marķieri. Varat arī iespējot/atspējot vairākus marķierus vienlaikus, atlasot tos sarakstā un pēc tam noklikšķinot uz pogām *Iespējot/Atspējot* zem saraksta.

Attention:
Lietotāji nevar skatīt tiem Zabbix piešķirto marķieru *Auth token* vērtību. *Auth token* vērtība tiek parādīta tikai vienu reizi — uzreiz pēc marķiera izveides. Ja marķieris ir pazaudēts, tas ir jāģenerē no jauna.

1 Globālie paziņojumi

Pārskats

Globālie paziņojumi nodrošina iespēju attēlot aktuālās problēmas reāllaikā tieši jūsu pašreizējā ekrānā Zabbix lietotāja saskarnē.

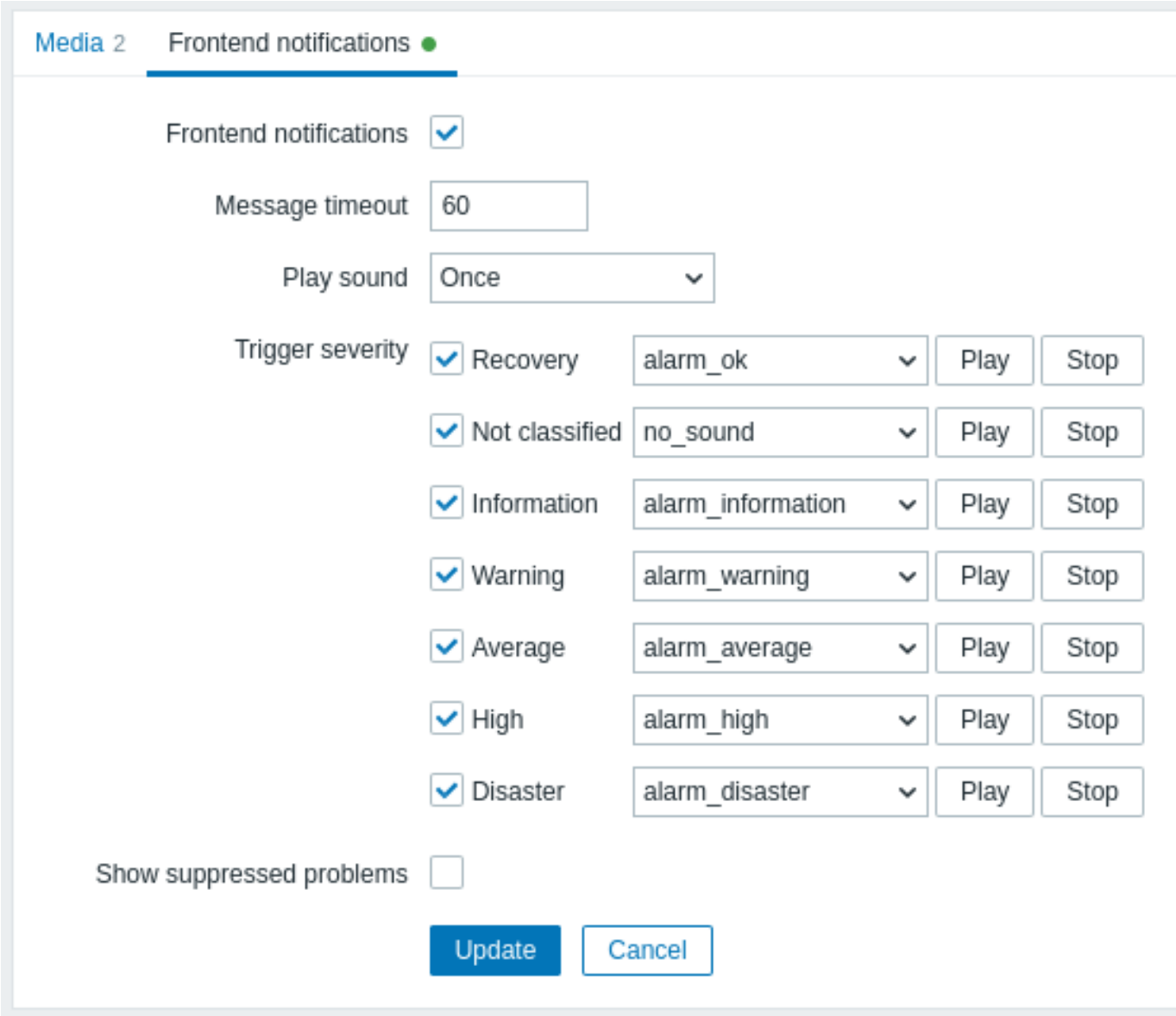
Bez globālajiem paziņojumiem, strādājot ārpus sadaļām *Problēmas* vai *Panelis*, jūs nesaņemtu nekādu informāciju par pašreizējām problēmām. Globālie paziņojumi nodrošina, ka šī informācija tiek attēlota neatkarīgi no jūsu pašreizējās atrašanās vietas Zabbix lietotāja saskarnē.

Globālie paziņojumi ietver gan ziņojuma attēlošanu, gan skaņas atskaņošanu.

Attention:
Skaņu automātiskā atskaņošana jaunākajās pārlūkprogrammu versijās var būt atspējota (pēc noklusējuma). Šādos gadījumos šis iestatījums ir jāiespējo manuāli.

Konfigurācija

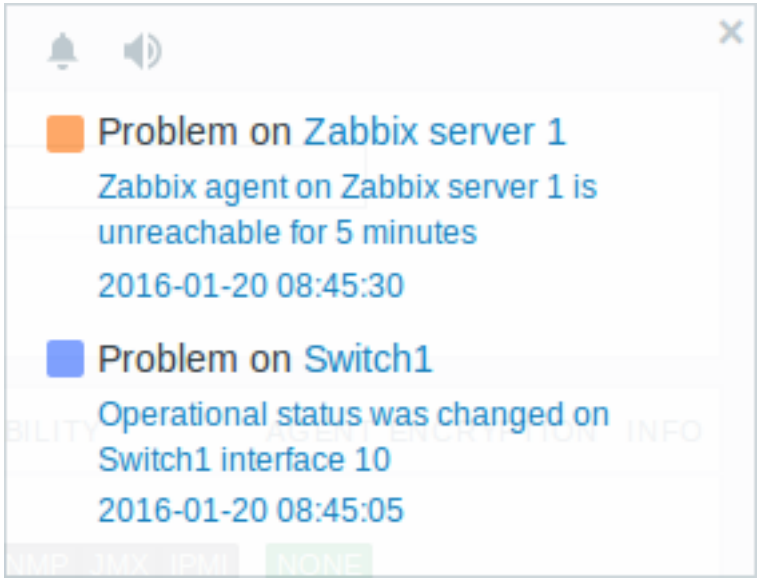
Globālos paziņojumus var iespējot katram lietotājam sadaļas Paziņojumi cilnē **Lietotāja saskarnes paziņojumi**.



Parametrs	Apraksts
Lietotāja saskarnes paziņojumi	Atzīmējiet izvēles rūtiņu, lai iespējotu globālos paziņojumus.
Ziņojuma noildze	Iestatiet laiku, cik ilgi ziņojums tiks rādīts. Pēc noklusējuma ziņojumi paliek ekrānā 60 sekundes. Tiek atbalstīti laika sufiksi, piemēram: 30s, 5m, 2h, 1d.
Atskaņot skaņu	Iestatiet laiku, cik ilgi tiks atskaņota skaņa. **Vienreiz** — skaņa tiek atskaņota vienu reizi pilnībā; **10 sekundes** — skaņa tiek atkārtota 10 sekundes; **Ziņojuma noildze** — skaņa tiek atkārtota, kamēr ziņojums ir redzams.
Trigera nopietnība	Iestatiet trigeru nopietnības līmeņus, kuriem tiks aktivizēti globālie paziņojumi un skaņas. Varat arī atlasīt dažādām nopietnības pakāpēm atbilstošas skaņas. Ja nav atzīmēts neviens nopietnības līmenis, netiks rādīts neviens ziņojums. Turklāt atkopšanas ziņojumi tiks rādīti tikai atzīmētajiem nopietnības līmeņiem. Piemēram, ja ir atzīmēti *Atkopšana* un *Katastrofa*, globālie paziņojumi tiks rādīti *Katastrofas* nopietnības trigeru problēmām un atkopšanām.
Rādīt apspiestās problēmas	Atzīmējiet izvēles rūtiņu, lai rādītu paziņojumus par problēmām, kas citādi tiktu apspiestas (netiktu rādītas) hosta uzturēšanas dēļ.

Parādītie globālie ziņojumi

Kad pienāk ziņojumi, tie tiek parādīti peldošā sadaļā labajā pusē. Šo sadaļu var brīvi pārvietot, velkot sadaļas galveni.



Šai sadaļai ir pieejamas vairākas vadīklas:

- poga **Snooze** apklusina pašlaik aktīvo trauksmes skaņu;
- poga **Mute/Unmute** pārslēdz starp trauksmes skaņu atskaņošanu un pilnīgu to neatskaņošanu.

2 Skaņa pārlūkprogrammās

Pārskats

Skaņa tiek izmantota globālajos paziņojumos.

Lai skaņas tiktu atskaņotas Zabbix lietotāja saskarnē, lietotāja profila cilnē *Frontend notifications* ir jābūt iespējotai opcijai *Frontend notifications*, un jābūt atzīmētiem visiem trigeru nopietnības līmeņiem. Papildus tam skaņām jābūt iespējotām globālo paziņojumu uznirstošajā logā.

Ja kāda iemesla dēļ ierīcē nevar atskaņot audio, poga globālo paziņojumu uznirstošajā logā pastāvīgi paliks "mute" stāvoklī, un, novietojot kursoru virs pogas , tiks parādīts ziņojums "Cannot support notification audio for this device".

Skaņas, tostarp noklusējuma audio klipi, tiek atbalstītas tikai MP3 formātā.

Zabbix lietotāja saskarnes skaņas ir veiksmīgi pārbaudītas jaunākajās Firefox un Opera pārlūkprogrammu versijās Linux vidē, kā arī Chrome, Firefox, Microsoft Edge un Opera pārlūkprogrammās Windows vidē.

Attention:
Skaņu automātiskā atskaņošana jaunākajās pārlūkprogrammu versijās var būt atspējota (pēc noklusējuma). Šādos gadījumos šis iestatījums ir jāiespējo manuāli.

4 Globālā meklēšana

Ir iespējams meklēt Zabbix lietotāja saskarnē hostus, hostu grupas, veidnes un veidņu grupas.

Meklēšanas ievades lauks atrodas izvēlnē zem Zabbix logotipa. Meklēšanu var sākt, nospiežot *Enter* vai noklikšķinot uz meklēšanas ikonas.

Ja ir hosts, kura nosaukumā jebkurā daļā ir ievadītā virkne, parādīsies nolaižamais saraksts, kurā būs uzskaitīti visi šādi hosti (ar atbilstošā daļa izcelta oranžā krāsā). Nolaižamajā sarakstā tiks iekļauts arī hosts, ja šī hosta redzamais nosaukums atbilst kā meklēšanas virkne ievadītajam tehniskajam nosaukumam; atbilstošais hosts tiks parādīts, bet bez izcelšanas.

Meklējamie atribūti

Hostus var meklēt pēc šādām īpašībām:

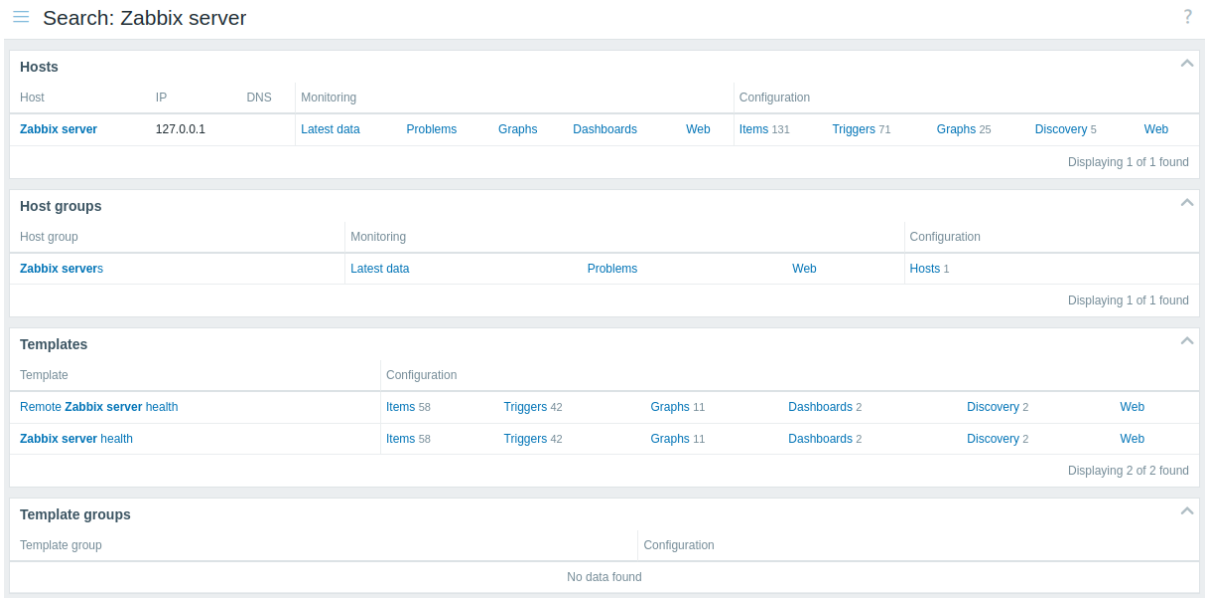
- Hosta nosaukums
- Redzamais nosaukums
- IP adrese
- DNS nosaukums

Veidnes var meklēt pēc nosaukuma vai redzamā nosaukuma. Ja meklējat pēc nosaukuma, kas atšķiras no redzamā nosaukuma (veidnei/hostam), tad meklēšanas rezultātos tas tiek parādīts zem redzamā nosaukuma iekavās.

Hostu un veidņu grupas var meklēt pēc nosaukuma. Norādot vecākgrupu, tiek netieši atlasītas arī visas tajā ietvertās apakšgrupas.

Meklēšanas rezultāti

Meklēšanas rezultāti sastāv no četriem atsevišķiem blokiem: hostiem, hostu grupām, veidnēm un veidņu grupām.



Ir iespējams sakļaut/izvērst katru atsevišķo bloku. Ierakstu skaits tiek parādīts katra bloka apakšā, piemēram, *Parādīti 13 no 13 atrastajiem*. Ja ierakstu nav, ierakstu skaits netiek parādīts. Kopējais vienā blokā parādīto ierakstu skaits ir ierobežots līdz 100.

Katrs ieraksts nodrošina saites uz uzraudzības un konfigurācijas datiem. Skatiet saišu pilno sarakstu.

Visiem konfigurācijas datiem (piemēram, vienumiem, trigeriem, grafikiem) atrasto entītiju skaits tiek parādīts ar skaitli blakus entītijas nosaukumam pelēkā krāsā. **Ņemiet vērā**, ka, ja entītiju skaits ir nulle, skaitlis netiek parādīts.

Iespējotie hosti tiek parādīti zilā krāsā, atspējotie hosti — sarkanā.

Pieejamās saites

Katram ierakstam ir pieejamas šādas saites:

- Hosti
  - Uzraudzība
    - Jaunākie dati
    - Problēmas
    - Grafiki
    - Hostu paneļi
    - Tīmekļa scenāriji
  - Konfigurācija
    - Vienumi
    - Trigeri
    - Grafiki
    - Atklāšanas kārtulas
    - Tīmekļa scenāriji
- Hostu grupas
  - Uzraudzība
    - Jaunākie dati
    - Problēmas
    - Tīmekļa scenāriji
  - Konfigurācija
    - Hosti
- Veidnes
  - Konfigurācija
    - Vienumi
    - Trigeri
    - Grafiki
    - Veidņu paneļi
    - Atklāšanas kārtulas
    - Tīmekļa scenāriji
- Veidņu grupas
  - Konfigurācija
    - Veidnes

5 Lietotāja saskarnes uzturēšanas režīms

Pārskats

Ir iespējams uz laiku atspējot Zabbix lietotāja saskarne, lai ierobežotu piekļuvi. Tas ir noderīgi, lai aizsargātu Zabbix datubāzi no lietotāju iniciētām izmaiņām, saglabājot tās integritāti.

Kamēr Zabbix lietotāja saskarne ir uzturēšanas režīmā, jūs varat droši apturēt datubāzi un veikt uzturēšanas darbus.

Lietotāji no norādītajām IP adresēm uzturēšanas režīma laikā varēs mijiedarboties ar lietotāja saskarni kā parasti.

Konfigurācija

Lai iespējotu uzturēšanas režīmu, atveriet failu `maintenance.inc.php` (atrodas Zabbix HTML dokumentu direktorijas `/conf` mapē tīmekļa serverī) un noņemiet komentārus šādām rindām:

```
// Uzturēšanas režīms.
define('ZBX_DENY_GUI_ACCESS', 1);

// IP adrešu masīvs, kurām ir atļauts pieslēgties lietotāja saskarnei (neobligāti).
$ZBX_GUI_ACCESS_IP_RANGE = array('127.0.0.1');

// Brīdinājuma ekrānā parādāmais ziņojums (neobligāti).
$ZBX_GUI_ACCESS_MESSAGE = 'Mēs veicam MySQL datubāzes jaunināšanu līdz plkst. 15:00. Lūdzu, uzgaidiet...';
```

Note:
Vairumā gadījumu fails `maintenance.inc.php` atrodas Zabbix HTML dokumentu direktorijas `/conf` mapē tīmekļa serverī.
Tomēr dažas operētājsistēmas un tīmekļa serveri var izmantot citu atrašanās vietu.
Piemēram, atrašanās vieta:

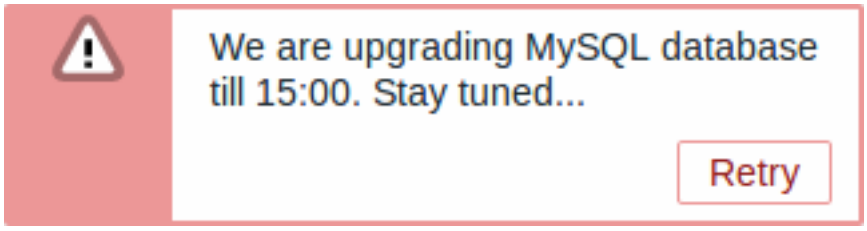
- SUSE un RedHat ir `/etc/zabbix/web/maintenance.inc.php`.
- Debian bāzētām sistēmām ir `/usr/share/zabbix/conf/`.

Skatiet arī PHP failu kopēšana.

Parametrs	Detalizēta informācija
ZBX_DENY_GUI_ACCESS	Ja ir definēts ar jebkādu vērtību, tiks iespējots uzturēšanas režīms. Lai atspējotu uzturēšanas režīmu, izkomentējiet vai izdzēsiet to.
ZBX_GUI_ACCESS_IP_RANGE	IP adrešu masīvs, kurām ir atļauts pieslēgties lietotāja saskarnei (neobligāti). Piemēram: `array('192.168.1.1', '192.168.1.2')`
ZBX_GUI_ACCESS_MESSAGE	Ziņojums, lai informētu lietotājus par uzturēšanu (neobligāti). Ja nav definēts, tiks izmantots noklusējuma ziņojums *'Zabbix is under maintenance'*.

Attēlošana

Lietotāji redzēs šādu ekrānu, mēģinot piekļūt Zabbix lietotāja saskarnei uzturēšanas režīmā. Ekrāns tiek atsvaidzināts ik pēc 30 sekundēm, lai pēc uzturēšanas beigām bez lietotāja iejaukšanās atgrieztos normālā stāvoklī.



IP adreses, kas definētas *ZBX_GUI_ACCESS_IP_RANGE*, varēs piekļūt lietotāja saskarnei kā parasti.

6 Lapas parametri

Pārskats

Lielākā daļa Zabbix tīmekļa lietotāja saskarnes lapu atbalsta dažādus HTTP GET parametrus, kas nosaka, kas tiks attēlots. Tos var nodot, pēc URL norādot parameter=value pārus, kas no URL ir atdalīti ar jautājuma zīmi (?) un cits no cita — ar zīmi "&" (&).

Uzraudzība > Problēmas

Tiek atbalstīti šādi parametri:

Parametrs	Apraksts	Piemērs
show	Filtra opcija *Rādīt*. Iespējamās vērtības: 1 - nesenās problēmas; 2 - visas; 3 - problēmas stāvoklī.	show=1
name	Filtra opcija *Problēma*: brīvas formas virkne.	name=Zabbix aģents

Parametrs	Apraksts	Piemērs
severities	Filtra opcija *Svarīgums*: atlasīto svarīguma līmeņu masīvs formātā `severities[*]=*` (aizstājiet * ar svarīguma līmeni). Iespējamās vērtības: 0 - nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa.	severities[3]=3
inventory	Filtra opcija *Hosta inventārs*: inventāra lauku masīvs `[field],[value]`	inventory[0][field]=type&invento
evaltype	Filtra opcija *Birkas*: birku novērtēšanas metode. Iespējamās vērtības: 0 - un/vai; 2 - vai.	evaltype=0
tags	Filtra opcija *Birkas*: definēto birku masīvs `[tag], [operator], [value]` Iespējamās operator vērtības: 0 - satur; 1 - vienāds ar; 2 - nesatur; 3 - nav vienāds ar; 4 - eksistē; 5 - neeksistē.	tags[0][tag]=target&tags[0][ope
show_tags	Filtra opcija *Rādīt birkas*. Iespējamās vērtības: 0 - nevienu; 1 - vienu; 2 - divas; 3 - trīs.	show_tags=3
tag_name_format	Filtra opcija *Birkas nosaukums*. Iespējamās vērtības: 0 - pilns nosaukums; 1 - saīsināts nosaukums; 2 - nav.	tag_name_format=1
tag_priority	Filtra opcija *Birku attēlošanas prioritāte*: ar komatiem atdalīta birku attēlošanas prioritāšu virkne	tag_priority=customer, target
show_suppressed	Filtra opcija *Rādīt apspiestās problēmas*. Iespējamās vērtības: 1 - rādīt; 2 - nerādīt.	show_suppressed=1
acknowledgement_status	Filtra opcija *Apstiprināšanas statuss*. Iespējamās vērtības: 0 - visi; 1 - neapstiprināti; 2 - apstiprināti.	acknowledgement_status=0
acknowledged_by_me	Filtra opcija *Manis*; tiek atbalstīta tikai ar `acknowledgement_status=2`. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	acknowledged_by_me=1
compact_view	Filtra opcija *Kompaktais skats*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	compact_view=1

Parametrs	Apraksts	Piemērs
highlight_row	Filtra opcija *Izcelt visu rindu*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	highlight_row=1
filter_name	Filtra īpašību opcija *Nosaukums*: brīvas formas virkne	filter_name=Databases
filter_show_counter	Filtra īpašību opcija *Rādīt ierakstu skaitu*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	filter_show_counter=1
filter_custom_time	Filtra īpašību opcija *Iestatīt pielāgotu laika periodu*. Iespējamās vērtības: 0 - atspējots; 1 - iespējots.	filter_custom_time=1
sort	Kolonna, pēc kuras kārtot. Iespējamās vērtības: clock - kārtot pēc kolonnas *Laiks*; host - kārtot pēc kolonnas *Hosts*; severity - kārtot pēc kolonnas *Svarīgums*; name - kārtot pēc kolonnas *Problēma*.	sort=clock
sortorder	Rezultātu kārtošanas secība. Iespējamās vērtības: DESC - dilstošā secībā; ASC - augošā secībā.	sortorder=DESC
age_state	Filtra opcija *Vecums mazāks par*; tiek atbalstīta tikai ar `show`=3. Iespējamās vērtības: 0 - atspējot parametru `age`; 1 - iespējot parametru `age`.	age_state=1
age	Filtra opcija *Vecums mazāks par*: integer, dienu skaits; tiek atbalstīta tikai ar `age_state`=1 un `show`=3.	age=7
groupids	Filtra opcija *Hostu grupas*: hostu grupu ID masīvs.	groupids[]=4
hostids	Filtra opcija *Hosti*: hostu ID masīvs	hostids[]=10084
triggerids	Filtra opcija *Trigeri*: trigeru ID masīvs	triggerids[]=22382
show_timeline	Filtra opcija *Rādīt laika skalu*; netiek atbalstīta ar `compact_view`=1. Iespējamās vērtības: 0 - nerādīt; 1 - rādīt.	show_timeline=1
details	Filtra opcija *Rādīt detalizētu informāciju*. Iespējamās vērtības: 0 - nerādīt; 1 - rādīt.	details=1
from	Datumu diapazona sākums, var būt relatīvs (piemēram, now-1m); tiek atbalstīts tikai ar `filter_custom_time`=1	from=now-2h
to	Datumu diapazona beigas, var būt relatīvas (piemēram, now-1m); tiek atbalstītas tikai ar `filter_custom_time`=1	to=now

Skatiet arī: Problēmu lapas filtra opcijas.

Kioska režīms

Kioska režīmu atbalstītajās lietotāja saskarnes lapās var aktivizēt, izmantojot URL parametrus. Piemēram, paneļos:

- `/zabbix.php?action=dashboard.view&kiosk=1` - aktivizēt kioska režīmu
- `/zabbix.php?action=dashboard.view&kiosk=0` - aktivizēt parasto režīmu

Slaidrāde

Informācijas panelī ir iespējams aktivizēt slaidrādi:

- `/zabbix.php?action=dashboard.view&slideshow=1` - aktivizēt slaidrādi

7 Definīcijas

Pārskats

Lai gan daudzas lietas lietotāja saskarnē var konfigurēt, izmantojot pašu lietotāja saskarni, daži pielāgojumi pašlaik ir iespējami tikai, rediģējot definīciju failu.

Šis fails ir `defines.inc.php`, kas atrodas Zabbix HTML dokumentu direktorija mapē /include.

Parametri

Parametri šajā failā, kas varētu būt interesanti lietotājiem:

- ZBX_MIN_PERIOD

Minimālais grafika periods sekundēs. Pēc noklusējuma viena minūte.

- GRAPH_YAXIS_SIDE_DEFAULT

Noklusējuma Y ass atrašanās vieta vienkāršajos grafikos un noklusējuma vērtība nolaižamajā sarakstā, pievienojot vienumus pielāgotiem grafikiem. Iespējamās vērtības: 0 - pa kreisi, 1 - pa labi.

Noklusējums: 0

- ZBX_SESSION_NAME

Virkne, kas tiek izmantota kā Zabbix lietotāja saskarnes sesijas sīkdatnes nosaukums.

Noklusējums: zbx_sessionid

- ZBX_DATA_CACHE_TTL

TTL noildze sekundēs, kas tiek izmantota, lai padarītu nederīgu Vault atbildes datu kešatmiņu. Iestatiet 0, lai atspējotu Vault atbilžu kešatmiņu.

Noklusējums: 60

- SUBFILTER_VALUES_PER_GROUP

Apakšfiltra vērtību skaits katrā grupā (piemēram, jaunāko datu apakšfiltrā).

Noklusējums: 1000

- ZBX_MAX_WIDGET_LINES

Maksimālais attēlojamo logrīka rindu skaits.

Noklusējums: 1000

8 Savas tēmas izveide

Pārskats

Pēc noklusējuma Zabbix nodrošina vairākas iepriekš definētas tēmas. Jūs varat sekot šeit sniegtajai soli pa solim procedūrai, lai izveidotu savu tēmu. Ja esat izveidojis ko patiešām labu, droši dalieties ar sava darba rezultātu ar Zabbix kopienu.

1. darbība

Lai definētu savu tēmu, jums būs jāizveido CSS fails un jāsaglabā tas mapē `assets/styles/` (piemēram, *custom-theme.css*). Jūs varat vai nu nokopēt failus no citas tēmas un izveidot savu tēmu, balstoties uz to, vai arī sākt no nulles.

2. darbība

Pievienojiet savu tēmu tēmu sarakstam, ko atgriež metode APP::getThemes(). To var izdarīt, pārrakstot metodi ZBase::getThemes() klasē APP. To var izdarīt, pievienojot šādu kodu pirms noslēdzošās figūriekavas failā *include/classes/core/APP.php*:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Pielāgota tēma')
    ]);
}
```

Attention:
Ņemiet vērā, ka nosaukumam, ko norādāt pirmajā pēdiņu pārī, jāsakrīt ar tēmas faila nosaukumu bez paplašinājuma.

Lai pievienotu vairākas tēmas, vienkārši uzskaitiet tās zem pirmās tēmas, piemēram:

```
public static function getThemes() {
    return array_merge(parent::getThemes(), [
        'custom-theme' => _('Pielāgota tēma'),
        'anothertheme' => _('Vēl viena tēma'),
        'onemoretheme' => _('Vēl viena tēma')
    ]);
}
```

Ņemiet vērā, ka katrai tēmai, izņemot pēdējo, beigās jābūt komatam.

Note:
Lai mainītu grafiku krāsas, ieraksts jāpievieno datubāzes tabulā *graph_theme*.

3. darbība

Aktivizējiet jauno tēmu.

Zabbix lietotāja saskarnē varat vai nu iestatīt šo tēmu kā noklusējuma tēmu, vai arī mainīt savu tēmu lietotāja profilā.

Izbaudiet jauno izskatu un lietošanas pieredzi!

9 Atkļūdošanas režīms

Pārskats

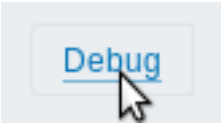
Atkļūdošanas režīmu var izmantot, lai diagnosticētu veiktspējas problēmas lietotāja saskarnes lapās.

Konfigurācija

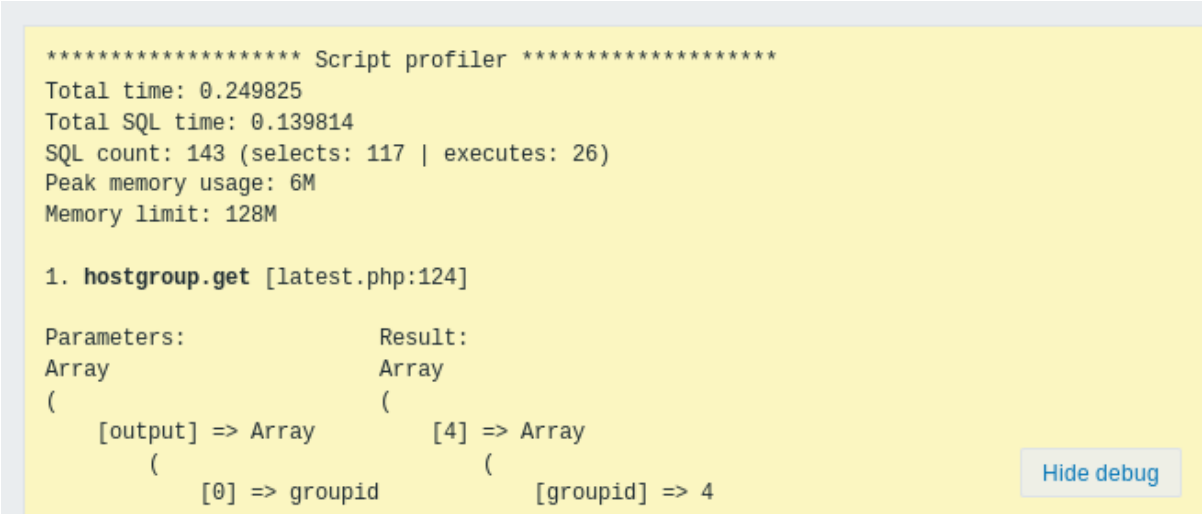
Atkļūdošanas režīmu var aktivizēt atsevišķiem lietotājiem, kuri pieder lietotāju grupai:

- konfigurējot lietotāju grupu;
- skatot konfigurētās lietotāju grupas.

Ja lietotāju grupai ir iespējots *Atkļūdošanas režīms*, tās lietotāji pārlūkprogrammas loga apakšējā labajā stūrī redzēs pogu *Atkļūdot*:



Noklikšķinot uz pogas *Atkļūdot*, zem lapas satura tiek atvērts jauns logs, kurā ir lapas SQL statistika, kā arī API izsaukumu un atsevišķu SQL priekšrakstu saraksts:



Ja rodas lapas veiktspējas problēmas, šo logu var izmantot, lai meklētu problēmas pamatcēloni.

Warning:
Iespējots *Atkļūdošanas režīms* negatīvi ietekmē lietotāja saskarne veiktspēju.

10 Zabbix izmantotās sīkdatnes

Pārskats

Šajā lapā ir sniegts Zabbix izmantoto sīkdatņu saraksts.

Nosaukums	Apraksts	Vērtības	Beidzas/Max-Age	HttpOnly[1]	Secure[2]
ZBX_SESSIONID	Zabbix lietotāja saskarnes sesijas dati, saglabāti kā JSON, kas kodēts ar base64		Sesija (beidzas, kad beidzas pārlūkošanas sesija)	+	+ (tikai tad, ja tīmekļa serverī ir iespējots HTTPS)
tab	Aktīvās cilnes numurs; šī sīkdatne tiek izmantota tikai lapās ar vairākām cilnēm (piemēram, *Host*, *Trigger* vai *Action* konfigurācijas lapā) un tiek izveidota, kad lietotājs pārvietojas no primārās cilnes uz citu cilni (piemēram, *Tags* vai *Dependencies* cilni). 0 tiek izmantots primārajai cilnei.	Piemērs: 1	Sesija (beidzas, kad beidzas pārlūkošanas sesija)	-	-
browserwarning_ignore	Vai ignorēt brīdinājums par novecojušas pārlūkprogrammas izmantošanu ir jāignorē.	yes	Sesija (beidzas, kad beidzas pārlūkošanas sesija)	-	-
system-message-ok	Ziņojums, kas jāparāda, tiklīdz lapa tiek pārlādēta.	Vienkārša teksta ziņojums	Sesija (beidzas, kad beidzas pārlūkošanas sesija) vai tiklīdz lapa tiek pārlādēta	+	-
system-message-error	Kļūdas ziņojums, kas jāparāda, tiklīdz lapa tiek pārlādēta.	Vienkārša teksta ziņojums	Sesija (beidzas, kad beidzas pārlūkošanas sesija) vai tiklīdz lapa tiek pārlādēta	+	-

Note:
'HttpOnly' karodziņa piespiedu iestatīšana Zabbix sīkdatnēm ar tīmekļa servera direktīvu netiek atbalstīta.

Piezīmes

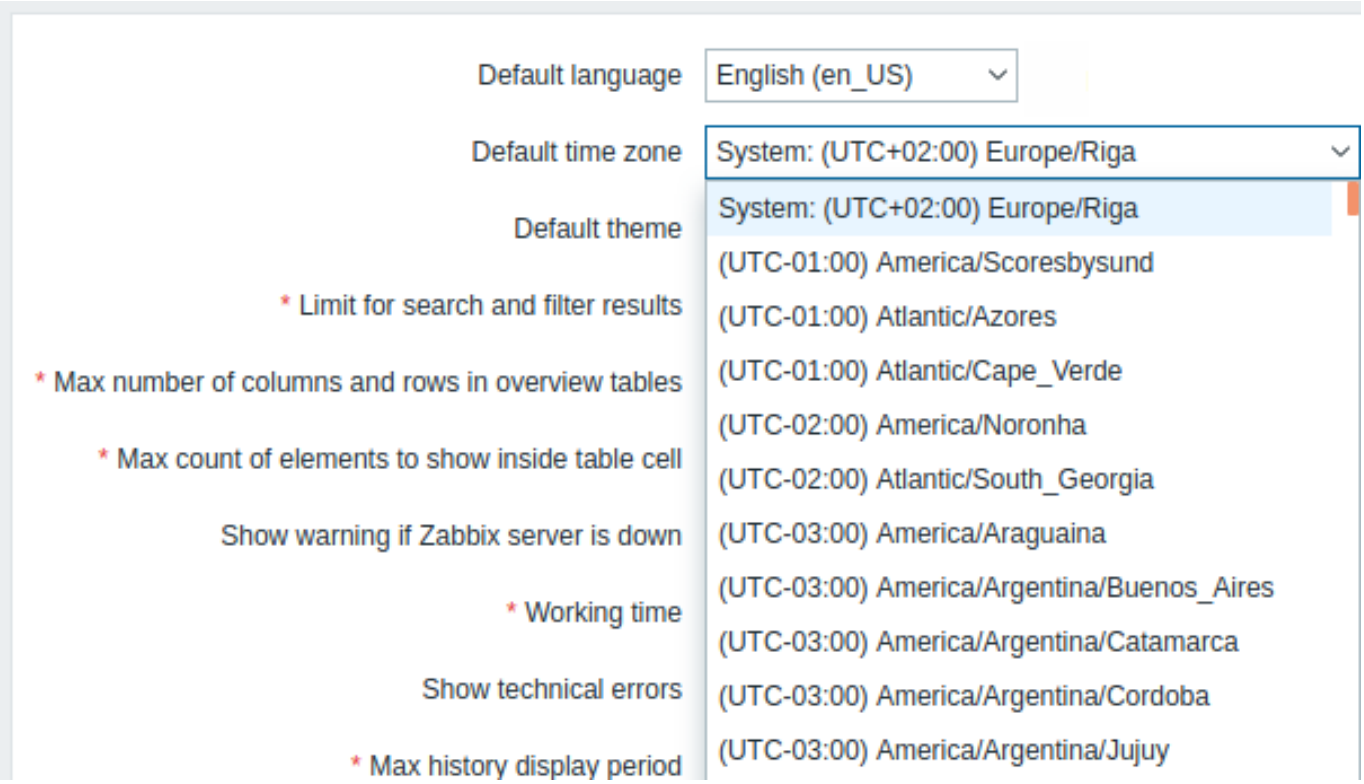
[1] Ja `HttpOnly` ir 'true', sīkdatne būs pieejama tikai caur HTTP protokolu. Tas nozīmē, ka sīkdatne nebūs pieejama skriptu valodām, piemēram, JavaScript. Šis iestatījums var efektīvi palīdzēt samazināt identitātes zādzību XSS uzbrukumu dēļ (lai gan to neatbalsta visas pārlūkprogrammas).

[2] `Secure` norāda, ka sīkdatne no klienta ir jāpārsūta tikai pa drošu HTTPS savienojumu. Ja iestatīts uz 'true', sīkdatne tiks iestatīta tikai tad, ja pastāv drošs savienojums.

11 Laika joslas

Pārskats

Lietotāja saskarnes laika joslu var iestatīt globāli lietotāja saskarnē un pielāgot atsevišķiem lietotājiem.



Ja ir atlasīts *System*, lietotāja saskarnei tiks izmantota tīmekļa servera laika josla (tostarp php.ini parametra 'date.timezone' vērtība, ja tā ir iestatīta), savukārt Zabbix serveris izmantos tās iekārtas laika joslu, kurā tas darbojas.

Note:
Zabbix serveris izmantos norādīto globālo/lietotāja laika joslu tikai tad, kad paziņojumos tiks izvērstas makrokomandas (piemēram, {EVENT.TIME} var tikt izvērsts citā laika joslā katram lietotājam) un laika ierobežojumam, kad tiek sūtīti paziņojumi (skatiet iestatījumu "When active" lietotāja mediju konfigurācijā).

Laika joslas izvēle neietekmē lietotāja saskarnes laika/datuma formātu. Tā vietā varat pielāgot saskarnes valodu (vai nu instalēšanas laikā, vai sadaļā lietotāja iestatījumi) — atlasot *English (en_US)*, lietotāja saskarnē tiks iespējots arī ASV laika/datuma formāts.

Konfigurācija

Globālā laika josla:

- var tikt iestatīta manuāli, instalējot lietotāja saskarni
- var tikt mainīta sadaļā *Administrēšana* → *Vispārīgi* → *GUI*

Lietotāja līmeņa laika josla:

- var tikt iestatīta, konfigurējot/atjauninot lietotāju
- to katrs lietotājs var iestatīt savā lietotāja profilā

Skatiet arī: Laika joslu saskaņošana, izmantojot plānošanas intervālus.

12 Paroles atiestatīšana

Pārskats Šajā sadaļā ir aprakstītas darbības, kas jāveic, lai Zabbix atiestatītu lietotāju paroles.

Soļi Ja esat aizmirsis savu Zabbix paroli un nevarat pieteikties, sazinieties ar savu Zabbix administratoru.

Super administratora lietotājs var mainīt paroles visiem lietotājiem lietotāja konfigurācijas formā.

Ja Super administrators ir aizmirsis savu paroli un nevar pieteikties, jāizpilda šāds SQL vaicājums, lai Super administratora lietotājam piemērotu noklusējuma paroli (aizstājiet 'Admin' ar atbilstošo Super administratora lietotājvārdu):

```
UPDATE users SET passwd = '$2a$10$ZXIvHAEP2ZM.dLXTm6uPHOMVlARXX7cqjbhM6Fn0cANzkCQBWpMrS' WHERE username =
```

Pēc šī vaicājuma izpildes lietotāja parole tiks iestatīta uz *zabbix*. Noteikti nomainiet noklusējuma paroli pirmajā pieteikšanās reizē.

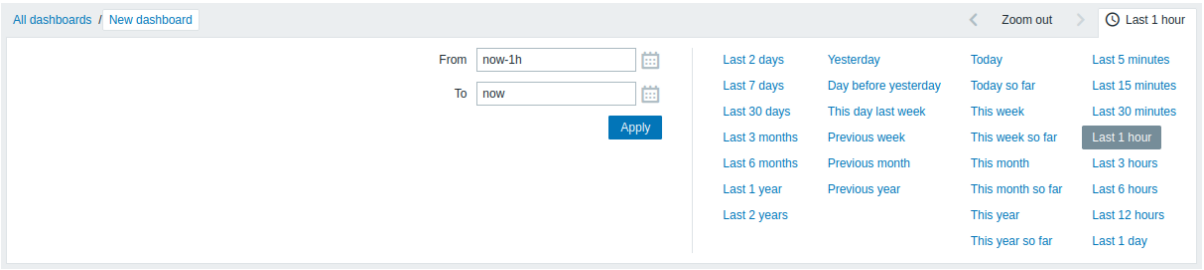
13 Laika perioda un hostu atlasītāji

Pārskats

Šajā lapā ir aprakstīti informācijas paneļos pieejamie laika perioda un hostu atlasītāji. Šie atlasītāji ļauj dinamiski pielāgot attēlotos datus visos saderīgajos informācijas paneļa logrīkos, nerediģējot katra atsevišķā logrīka iestatījumus.

Laika perioda atlasītājs

Laika perioda atlasītājs ļauj atlasīt bieži nepieciešamos periodus ar vienu peles klikšķi. To var izvērst vai sakļaut, noklikšķinot uz laika perioda atlasītāja pogas paneļa augšējā labajā stūrī.



Tādas opcijas kā *Šodien*, *Šī nedēļa* u.c. parāda visu periodu, ieskaitot stundas/dienas nākotnē. Tādas opcijas kā *Šodien līdz šim brīdim*, *Šī nedēļa līdz šim brīdim* u.c. parāda tikai jau pagājušās stundas.

Kad periods ir atlasīts, to var pārvietot atpakaļ vai uz priekšu laikā, noklikšķinot uz bultu pogām. Poga *Tālināt* ļauj samazināt mērogu periodam par 50% katrā virzienā.

Note:
Grafikiem (izņemot tos, kuriem ir *Pielāgots* laika periods) parādītā laika perioda atlasīšana ir iespējama arī, iezīmējot apgabalu grafikā ar peles kreiso pogu. Kad atlaižat peles kreiso pogu, grafiks pietuvinās iezīmētajam apgabalam. Tālināšana ir iespējama arī, veicot dubultklikšķi grafikā.

Lauki *No/Līdz* parāda atlasīto periodu vai nu absolūtā laika sintaksē (formātā `Y-m-d H:i:s`), vai relatīvā laika sintaksē. Relatīvs laika periods var saturēt vienu vai vairākas matemātiskas darbības (- vai +), piemēram, `now-1d` vai `now-1d-2h+5m`.

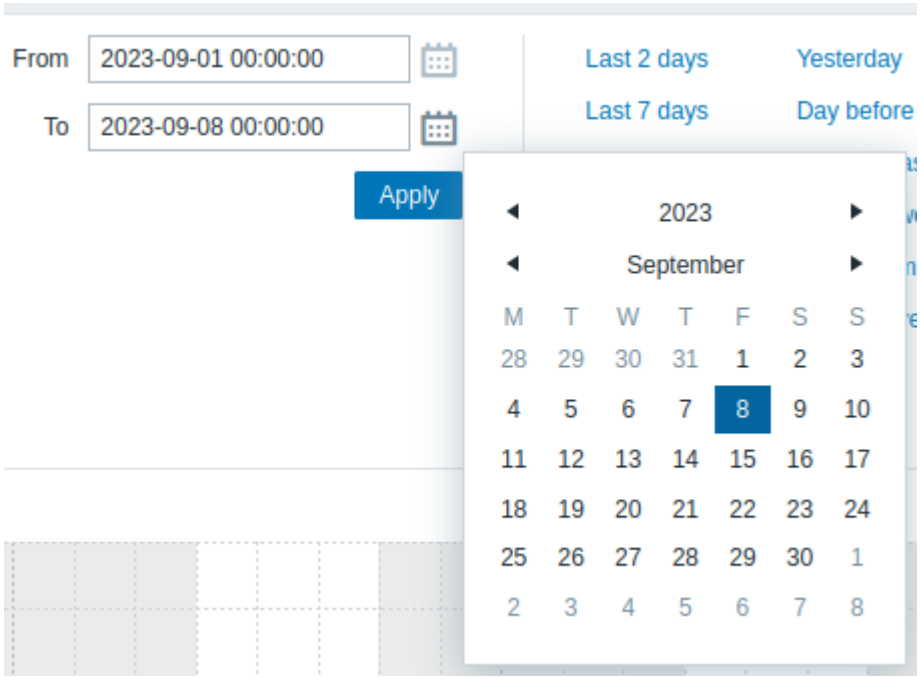
Tiek atbalstīti šādi relatīvā laika saīsinājumi:

- `now`
- `s` (sekundes)
- `m` (minūtes)
- `h` (stundas)
- `d` (dienas)
- `w` (nedēļas)
- `M` (mēneši)
- `y` (gadi)

Laika perioda atlasītājā tiek atbalstīta precizitāte (piemēram, `/M` izteiksmē `now-1d/M`). Precizitātes informācija:

Precizitāte	No	Līdz
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	Nedēļas pirmdiena 00:00:00	Nedēļas svētdiena 23:59:59
M	Mēneša pirmā diena 00:00:00	Mēneša pēdējā diena 23:59:59
y	Gada 1. janvāris 00:00:00	Gada 31. decembris 23:59:59

Laika periodu ir iespējams atlasīt arī, izmantojot datuma atlasītāju. Lai to atvērtu, noklikšķiniet uz kalendāra ikonas blakus laukiem *No/Līdz*.



Note:
Datuma atlasītājā var pārvietoties starp gadu/mēnesi/datumu, izmantojot Tab, Shift+Tab un tastatūras bulttaustiņus. Nospiežot Enter, atlase tiek apstiprināta.

Piemēri:

No	Līdz	Atlasītais periods
now/d	now/d	šodien 00:00 - 23:59
now/d	now/d+1d	no šodienas 00:00 līdz rītdienas 23:59
now/w	now/w	šīs nedēļas pirmdiena 00:00:00 - svētdiena 23:59:59
now-1y/w	now-1y/w	nedēļa pirms viena gada: pirmdiena 00:00:00 - svētdiena 23:59:59

Attention:
Izmantojot "now/M+1M" parametram *Līdz*, var tikt pievienota 31 diena, kā rezultātā datums var nobīdīties par 1-3 dienām atkarībā no dienu skaita mēnesī. Piemēram, ja tas tiek izmantots janvārī, rezultāts var būt 02. marts gaidītā 28. februāra vietā. Lai izvairītos no šīs problēmas, izmantojiet "now/M-3d+1M/M", kas precīzi pielāgojas mēneša garumam. Līdzīgi, ja konfigurējat *No* parametru virzībai atpakaļ, izmantojiet "now/M+3d-1M/M".

Hostu atlasītājs

Hostu atlasītājs ļauj atlasīt hostu, lai tā dati tiktu parādīti saderīgos paneļa logrīkos.

Tas kļūst pieejams pēc paneļa rediģēšanas un saglabāšanas, ja vismaz viens logrīks ir konfigurēts ar parametru *Override host*, kas iestatīts izmantot paneli kā datu avotu.

17 Labākā prakse

Pārskats Šajā sadaļā ir izklāstīta labākā prakse Zabbix iestatīšanai.

Lai gan šī prakse nav obligāta Zabbix darbībai, tās ieviešana ir ļoti ieteicama, lai nodrošinātu optimālu un drošu lietošanas pieredzi.

1 Drošības labākā prakse

Pārskats

Šajā sadaļā ir apkopota labākā prakse drošai Zabbix iestatīšanai.

Šajā sadaļā aprakstītā prakse nav obligāta Zabbix darbībai, taču tā ir ieteicama labākai sistēmas drošībai.

UTF-8 kodējums

UTF-8 ir vienīgais Zabbix atbalstītais kodējums. Ir zināms, ka tas darbojas bez jebkādām drošības ievainojamībām. Lietotājiem jāņem vērā, ka, izmantojot dažus citus kodējumus, pastāv zināmas drošības problēmas.

Windows instalētāja ceļi

Izmantojot Windows instalētājus, ieteicams izmantot instalētāja piedāvātos noklusējuma ceļus. Pielāgotu ceļu izmantošana bez atbilstošām atļaujām var apdraudēt instalācijas drošību.

Makrosi lietotāja definētos globālajos skriptos

Lai uzlabotu drošību, ieteicams lietot makro funkcijas, nevis vienkāršus makrosus lietotāja definētos globālajos skriptos, jo makrosi netiek automātiski atsaistīti.

Zabbix drošības ieteikumi un CVE datubāze

Skatiet *Zabbix drošības ieteikumus un CVE datubāzi*.

HTML e-pasta veidnes

Veidojot vai rediģējot ziņojumu veidnes, kas tiek izmantotas HTML e-pastiem, vienmēr aptiniet katru makrosu ar `htmlencode()` makrosa funkciju. Piemēram:

```
<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
```

Izmantojot `htmlencode()`, tiek nodrošināts, ka visas HTML rakstzīmes makrosu vērtībās tiek aizstātas ar drošām entītijām, un tiek novērsta HTML ievietošana paziņojumos (piemēram, ja uzbrucējs paziņojumā ievieto ļaunprātīgu vai pikšķerēšanas saiti).

Piezīme: Zabbix nodrošinātajos noklusējuma HTML e-pasta ziņojumos makrosiem jau ir piemērots `htmlencode()`. Šie norādījumi attiecas uz esošu veidņu rediģēšanu vai jaunu veidņu izveidi — pirms veidnes izmantošanas HTML e-pasta sūtīšanai pārbaudiet, vai makrosi ir kodēti.

1 Piekļuves kontrole

Pārskats

Šajā sadaļā ir apkopota labākā prakse drošai piekļuves kontroles iestatīšanai.

Mazāko privilēģiju princips

Lietotāju kontiem vienmēr jādarbojas ar pēc iespējas mazākām privilēģijām. Tas nozīmē, ka lietotāju kontiem Zabbix lietotāja saskarnē, datubāzes lietotājiem vai lietotājam, kas paredzēts Zabbix serveris/starpniekserveris/aģents procesiem, jābūt tikai tām privilēģijām, kas ir būtiski nepieciešamas paredzēto funkciju izpildei.

> **Attention:**
> Papildu privilēģiju piešķiršana lietotājam 'zabbix' ļaus tam piekļūt konfigurācijas failiem un izpildīt darbības, kas var apdraudēt infrastruktūras drošību.

Konfigurējot lietotāju kontu privilēģijas, jāņem vērā Zabbix lietotāja saskarnes lietotāju tipi. Ņemiet vērā, ka, lai gan lietotāja tipam *Admin* ir mazāk privilēģiju nekā lietotāja tipam *Super Admin*, tas joprojām var pārvaldīt konfigurāciju un izpildīt pielāgotus skriptus.

Note:
Daļa informācijas ir pieejama arī lietotājiem bez īpašām privilēģijām. Piemēram, lai gan *Alerts* → *Scripts* ir pieejams tikai *Super Admin* lietotājiem, skriptus var iegūt arī, izmantojot Zabbix API. Šādā gadījumā skriptu atļauju ierobežošana un sensitīvas informācijas neiekļaušana skriptos (piemēram, piekļuves akreditācijas dati) var palīdzēt novērst globālajos skriptos pieejamās sensitīvās informācijas atklāšanu.

Drošs lietotājs Zabbix aģentam

Pēc noklusējuma Zabbix servera, starpniekservera un aģenta (vai aģenta 2) procesi izmanto vienu `zabbix` lietotāju. Lai nepieļautu, ka Zabbix aģents/aģents 2 (darbojoties tajā pašā datorā, kur serveris/starpniekserveris) piekļūst sensitīvai informācijai servera/starpniekservera konfigurācijā (piemēram, datubāzes akreditācijas datiem), aģents jāpalaiž ar citu lietotāju:

Zabbix aģentam:

1. Izveidojiet drošu grupu un lietotāju (piemēram, `zabbix-agent`).
2. Norādiet šo lietotāju aģenta konfigurācijas faila parametrā User.
3. Restartējiet aģentu, lai samazinātu privilēģijas līdz jaunajam lietotājam.

Zabbix aģentam 2 konfigurācija jāpiemēro servisa līmenī, jo aģenta 2 konfigurācijas fails neatbalsta parametru `User`. Piemēru skatiet ZBX-26442.

Atsauciet rakstīšanas piekļuvi SSL konfigurācijai (Windows)

Ja esat kompilējis Zabbix aģentu operētājsistēmā Windows un OpenSSL atrodas neaizsargātā direktorijā (piemēram, `C:\zabbix`, `c:\openssl-64bit`, `C:\OpenSSL-Win64-111-static` vai `C:\dev\openssl`), noteikti atsauciet rakstīšanas piekļuvi šai direktorijai lietotājiem bez administratora tiesībām. Pretējā gadījumā aģents ielādē SSL iestatījumus no ceļa, ko var modificēt lietotāji bez paaugstinātām privilēģijām, tādējādi radot iespējamu drošības ievainojamību.

Zabbix komponentu drošības pastiprināšana

Daļu funkcionalitātes var izslēgt, lai pastiprinātu Zabbix komponentu drošību:

- globālo skriptu izpildi Zabbix serverī var atspējot, servera konfigurācijā iestatot EnableGlobalScripts=0;
- globālo skriptu izpilde Zabbix starpniekserverī pēc noklusējuma ir atspējota (to var iespējot, starpniekservera konfigurācijā iestatot EnableRemoteCommands=1);
- globālo skriptu izpilde Zabbix aģentos pēc noklusējuma ir atspējota (to var iespējot, aģenta konfigurācijā pievienojot parametru AllowKey=system.run[<command>,*] katrai atļautajai komandai);
- lietotāja HTTP autentifikāciju var atspējot, lietotāja saskarnes konfigurācijas failā (zabbix.conf.php) iestatot `$ALLOW_HTTP_AUTH=false`. Ņemiet vērā, ka, pārinstalējot lietotāja saskarni (palaižot setup.php), šis parametrs tiks noņemts.

Piekļuve UNC ceļiem operētājsistēmā Windows, izmantojot Zabbix aģentu

Zabbix aģenti operētājsistēmā Windows seko UNC ceļiem (SMB koplietojumiem, piemēram, `\\server\share\file.txt`) tādos vienumos kā `vfs.file.*`, `vfs.dir.*`, `modbus.get` un `perf_counter*`. Dažos kontekstos tas var radīt drošības risku.

Kad operētājsistēmai Windows tiek pieprasīts piekļūt UNC ceļam, tā mēģina autentificēties attiecīgajā serverī. Tas nozīmē, ka ļaunprātīgs pieprasījums Zabbix aģentam var atklāt NTLM jaucējkodu pieprasītāja serverim. Lietotāji vajadzības gadījumā var to mazināt, izmantojot konfigurācijas parametrus AllowKey un DenyKey.

1 MySQL/MariaDB drošības nodrošināšana

Pārskats

Šajā sadaļā ir apkopota paraugprakse MySQL/MariaDB datubāzes drošības nodrošināšanai.

Note:
Pamata iestatīšanai skatiet noklusējuma MySQL/MariaDB datubāzes izveides norādījumus, kas ietver `zabbix` lietotāja izveidi ar pilnām privilēģijām Zabbix datubāzē. Šis lietotājs ir datubāzes īpašnieks, kuram ir arī nepieciešamās privilēģijas datubāzes struktūras modificēšanai, jauninot Zabbix.

Lai uzlabotu drošību, ieteicams izveidot papildu datubāzes lomas un lietotājus ar minimālām privilēģijām. Šīs lomas un lietotāji jākonfigurē, pamatojoties uz mazāko privilēģiju principu, tas ir, tiem jābūt tikai tādām privilēģijām, kas ir būtiskas paredzēto funkciju veikšanai.

Lietotāju lomu izveide

Izveidojiet šādas lomas ar atbilstošajām privilēģijām:

- **zbx_srv** - loma Zabbix serveris un starpniekserveris darbināšanai:

```
CREATE ROLE 'zbx_srv';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_srv';
FLUSH PRIVILEGES;
```

- **zbx_web** - loma Zabbix lietotāja saskarne un API darbināšanai:

```
CREATE ROLE 'zbx_web';
GRANT DELETE, INSERT, SELECT, UPDATE ON zabbix.* TO 'zbx_web';
FLUSH PRIVILEGES;
```

- **zbx_bckp** - loma tabulu dublēšanai:

```
CREATE ROLE 'zbx_bckp';
GRANT LOCK TABLES, TRIGGER, SELECT ON zabbix.* TO 'zbx_bckp';
GRANT process ON *.* TO 'zbx_bckp';
FLUSH PRIVILEGES;
```

Note:
Tabulu atjaunošana un jaunināšana jāveic datubāzes īpašniekam.

- **zbx_part** - loma ar samazinātu privilēģiju kopu datubāzes particionēšanai; ņemiet vērā, ka šo lomu var izveidot tikai pēc datubāzes izveides, jo tā piešķir privilēģijas konkrētām datubāzes tabulām:

```
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';
-- MariaDB gadījumā: izlaidiet nākamo rindu (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;
```

Pēc lomu izveides tās var piešķirt lietotājiem.

Lietotāju lomu piešķiršana

Lai piešķirtu izveidotās lietotāju lomas, izveidojiet lietotājus un piešķiriet tiem atbilstošās lomas. Aizstājiet <user>, <host>, <role> un <password> pēc nepieciešamības.

```
CREATE USER '<user>'@'<host>' IDENTIFIED BY '<password>';
GRANT '<role>' TO '<user>'@'<host>';
SET DEFAULT ROLE '<role>' TO '<user>'@'<host>';
-- For MariaDB: SET DEFAULT ROLE '<role>' FOR '<user>'@'<host>'
FLUSH PRIVILEGES;
```

Piemēram, lai izveidotu un piešķirtu lomu Zabbix serveris un starpniekserveris darbināšanai:

```
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 PostgreSQL/TimescaleDB drošības nodrošināšana

Pārskats

Šajā sadaļā ir sniegta paraugprakse PostgreSQL datubāzes drošības nodrošināšanai.

Note:
Pamata iestatīšanai skatiet noklusējuma PostgreSQL datubāzes izveides norādījumus, kas ietver lietotāja 'zabbix' izveidi ar pilnām privilēģijām Zabbix datubāzē. Šis lietotājs ir datubāzes īpašnieks, kuram ir arī nepieciešamās privilēģijas datubāzes struktūras modificēšanai, jauninot Zabbix.

Lai uzlabotu drošību, ieteicams konfigurēt drošu shēmas lietošanas modeli, kā arī izveidot papildu datubāzes lomas un lietotājus ar minimālām privilēģijām. Šīs lomas un lietotāji jākonfigurē, pamatojoties uz mazāko privilēģiju principu, tas ir, tiem jābūt tikai tādām privilēģijām, kas ir būtiskas paredzēto funkciju veikšanai.

Datubāzes iestatīšana

Izveidojiet lietotāju, kas būs datubāzes īpašnieks, un izveidojiet Zabbix datubāzi; datubāzes īpašnieks ir lietotājs, kas ir norādīts datubāzes izveides laikā:

```
createuser -U postgres -h localhost --pwprompt usr_owner
createdb -U postgres -h localhost -O usr_owner -E Unicode -T template0 zabbix
```

Note:
Tīra datubāzes instalēšana vai jaunināšana jāveic datubāzes īpašniekam. Tas ir tāpēc, ka tiesības dzēst datubāzes objektu vai mainīt tā definīciju ir privilēģija, kas ir raksturīga datubāzes īpašniekam un ko nevar piešķirt vai atsaukt.

Attention:
Šajā lapā norādītās komandas jāizpilda, kamēr savienojums ar PostgreSQL ir izveidots tieši ar `zabbix` datubāzi.

Izveidojiet `zabbix` shēmu un iestatiet datubāzes īpašnieku (`usr_owner`) kā šīs shēmas īpašnieku:

```
CREATE SCHEMA zabbix AUTHORIZATION usr_owner;
```

Konfigurējiet drošu shēmas lietošanas modeli:

```
REVOKE CREATE ON SCHEMA public FROM PUBLIC;
REVOKE ALL ON DATABASE zabbix FROM PUBLIC;
-- Piezīme: search_path jānorāda uz "zabbix" shēmu:
ALTER ROLE ALL IN DATABASE zabbix SET search_path = "zabbix";
```

Pēc datubāzes iestatīšanas turpiniet ar lietotāju lomu izveidi.

Lietotāju lomu izveide

Izveidojiet šādas lomas ar atbilstošajām privilēģijām:

- **zbx_srv** - loma Zabbix servera un starpniekservera darbināšanai:

```
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** - loma Zabbix lietotāja saskarnes un API darbināšanai:

```
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** - loma tabulu dublēšanai:

```
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:
Tabulu atjaunošana ir iespējama tikai datubāzes īpašniekam.

Pēc lomu izveides tās var piešķirt lietotājiem.

Lietotāju lomu piešķiršana

Lai piešķirtu izveidotās lietotāju lomas, izveidojiet lietotājus un piešķiriet tiem atbilstošās lomas. Aizstājiet <user>, <role> un <password> pēc nepieciešamības.

```
CREATE USER <user> WITH ENCRYPTED password '<password>';
GRANT <role> TO <user>;
```

Piemēram, lai izveidotu un piešķirtu lomu Zabbix servera un starpniekservera darbināšanai:

```
CREATE USER usr_srv WITH ENCRYPTED password 'password';
GRANT zbx_srv TO usr_srv;
```

Datubāzes particionēšana ar TimescaleDB

Datubāzes particionēšanu atvieglo TimescaleDB. Lai izmantotu TimescaleDB, Zabbix serverim ir nepieciešamas datubāzes īpašnieka privilēģijas.

Ja PostgreSQL zabbix shēma jau ir izveidota zabbix datubāzē, varat iespējot TimescaleDB ar šādu komandu:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb WITH SCHEMA zabbix CASCADE;" | sudo -u postgres psql zabb
```

2 Kriptogrāfija

Pārskats

Šajā sadaļā ir apkopota paraugprakse drošai kriptogrāfijas iestatīšanai.

SSL iestatīšana Zabbix lietotāja saskarnei

Sistēmās, kuru pamatā ir RHEL, instalējiet mod_ssl pakotni:

```
dnf install mod_ssl
```

Izveidojiet direktoriju SSL atslēgām:

```
mkdir -p /etc/httpd/ssl/private
chmod 700 /etc/httpd/ssl/private
```

Izveidojiet SSL sertifikātu:

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/httpd/ssl/private/apache-selfsigned.key -
```

Atbilstoši aizpildiet uzvednēs prasīto informāciju. Svarīgākā rinda ir tā, kurā tiek pieprasīts Common Name. Jums jāievada domēna nosaukums, kuru vēlaties saistīt ar savu serveri. Ja jums nav domēna nosaukuma, tā vietā varat ievadīt publisko IP adresi.

```
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 []:
```

Rediģējiet Apache SSL konfigurācijas failu (/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
```

Lai piemērotu izmaiņas, restartējiet Apache servisu:

```
systemctl restart httpd.service
```

3 Tīmekļa serveris

Pārskats

Šajā sadaļā ir apkopota labākā prakse, kā drošā veidā iestatīt tīmekļa serveri.

Saknes URL pāradresācijas uz Zabbix SSL piespiedu iestatīšana

Sistēmās, kuru pamatā ir RHEL, pievienojiet Apache konfigurācijai (`/etc/httpd/conf/httpd.conf`) virtuālo hostu un iestatiet pastāvīgu dokumenta saknes pāradresāciju uz Zabbix SSL URL. Ņemiet vērā, ka `example.com` jāaizstāj ar faktisko servera nosaukumu.

```
#### Pievienojiet rindas:

<VirtualHost *:*>
    ServerName example.com
    Redirect permanent / https://example.com
</VirtualHost>
```

Lai piemērotu izmaiņas, restartējiet Apache servisu:

```
systemctl restart httpd.service
```

HTTP Strict Transport Security (HSTS) iespējošana tīmekļa serverī

Lai aizsargātu Zabbix lietotāja saskarne pret protokola pazemināšanas uzbrukumiem, iesakām tīmekļa serverī iespējot HSTS politiku.

Lai Apache konfigurācijā iespējotu HSTS politiku jūsu Zabbix lietotāja saskarnei, izpildiet šīs darbības:

1. Atrodiet sava virtuālā hosta konfigurācijas failu:

- `/etc/httpd/conf/httpd.conf` uz RHEL bāzētās sistēmās
- `/etc/apache2/sites-available/000-default.conf` uz Debian/Ubuntu

2. Pievienojiet šādu direktīvu sava virtuālā hosta konfigurācijas failam:

```
<VirtualHost *:*>
    Header set Strict-Transport-Security "max-age=31536000"
</VirtualHost>
```

3. Restartējiet Apache servisu, lai piemērotu izmaiņas:

```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Drošu un SameSite sesijas sīkdatņu piespiedu izmantošana Zabbix

Konfigurējot Zabbix, ir būtiski piespiest izmantot drošos un SameSite atribūtus sesijas sīkdatnēm, lai uzlabotu drošību un novērstu starpvietņu pieprasījumu viltošanas (CSRF) uzbrukumus. Tomēr `SameSite=Strict` piespiedu izmantošana noteiktos scenārijos var radīt problēmas, piemēram:

- Informācijas paneļa URL logrīki rāda "lietotājs nav pieteicies", iegulstot tā paša domēna iframe.
- Lietotājiem, kas piekļūst informācijas panelim, izmantojot HTTP, nevis HTTPS, var rasties pieteikšanās problēmas.
- Nav iespējams kopīgot URL uz noteiktām Zabbix izvēlnes sadaļām vai hostiem.

Lai mazinātu šīs problēmas, lietotājiem jābūt iespējai pielāgot SameSite politiku.

1. Drošās sīkdatnes

Iestatot `secure` karogu, tiek nodrošināts, ka sīkdatnes tiek pārsūtītas tikai caur HTTPS, novēršot to noplūdi nešifrētos savienojumos.

Lai iespējotu drošās sīkdatnes Zabbix, pievienojiet vai modificējiet šādu iestatījumu tīmekļa servera konfigurācijā:

Apache gadījumā:

```
Header always edit Set-Cookie ^(.*)$ $1;Secure
```

Nginx gadījumā:

```
proxy_cookie_path / "/; Secure";
```

Pārliecinieties, ka jūsu Zabbix lietotāja saskarnei piekļūst, izmantojot HTTPS; pretējā gadījumā sīkdatnes ar `Secure` karogu netiks nosūtītas.

2. SameSite atribūta konfigurēšana

Tīmekļa servera iestatījumi var arī piespiest izmantot SameSite atribūtu:

Apache gadījumā:

```
<IfModule mod_headers.c>
    Header onsuccess edit Set-Cookie (.*) "$1; SameSite=Strict"
</IfModule>
```

Nginx gadījumā (versija 1.19.3+):

```
proxy_cookie_flags ~ samesite=Strict; # Aizstājiet ~ ar 'zbx_session', lai nodrošinātu lielāku precizitāti
```

Content Security Policy (CSP) iespējošana tīmekļa serverī

Lai aizsargātu Zabbix lietotāja saskarne pret Cross Site Scripting (XSS), datu injekciju un līdzīgiem uzbrukumiem, iesakām iespējot Content Security Policy tīmekļa serverī. Lai to izdarītu, konfigurējiet tīmekļa serveri tā, lai tas atgrieztu HTTP header.

> **Attention:**
> Tālāk norādītā CSP header konfigurācija ir paredzēta tikai noklusējuma Zabbix lietotāja saskarne instalācijai un gadījumiem, kad viss saturs tiek ielādēts no vietnes domēna (izņemot apakšdomēnus). Citāda CSP header konfigurācija var būt nepieciešama, ja, piemēram, konfigurējat *URL* logrīku, lai attēlotu saturu no vietnes apakšdomēniem vai ārējiem domēniem, pārslēdzaties no *OpenStreetMap* uz citu kartes dzini vai pievienojat ārēju CSS vai logrīkus. Ja izmantojat Duo Universal Prompt daudzfaktoru autentifikācijas metodi, noteikti pievienojiet "duo.com" CSP direktīvai sava virtuālā hosta konfigurācijas failā.

Lai iespējotu CSP jūsu Zabbix lietotāja saskarne Apache konfigurācijā, izpildiet šīs darbības:

1. Atrodiet sava virtuālā hosta konfigurācijas failu:

- `/etc/httpd/conf/httpd.conf` RHEL bāzētās sistēmās
- `/etc/apache2/sites-available/000-default.conf` Debian/Ubuntu

2. Pievienojiet šādu direktīvu sava virtuālā hosta konfigurācijas failam:

```
<VirtualHost *:*>
    Header set Content-Security-Policy: "default-src 'self' *.openstreetmap.org; script-src 'self' 'unsafe
</VirtualHost>
```

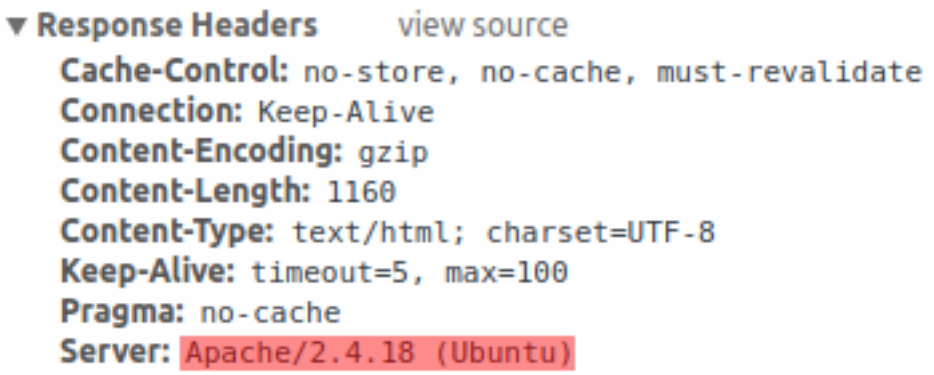
3. Restartējiet Apache servisu, lai piemērotu izmaiņas:

```
#### On RHEL-based systems:
systemctl restart httpd.service

#### On Debian/Ubuntu
systemctl restart apache2.service
```

Tīmekļa servera informācijas atklāšanas atspējošana

Lai uzlabotu drošību, ieteicams atspējot visus tīmekļa servera parakstus.

Pēc noklusējuma tīmekļa serveris atklāj programmatūras parakstu:



Parakstu var atspējot, pievienojot Apache konfigurācijas failam šādus parametrus:

```
ServerSignature Off
ServerTokens Prod
```

PHP parakstu (X-Powered-By HTTP galveni) var atspējot, mainot `php.ini` konfigurācijas failu (pēc noklusējuma paraksts ir atspējots):

```
expose_php = Off
```

Lai konfigurācijas faila izmaiņas stātos spēkā, ir nepieciešams restartēt tīmekļa serveri.

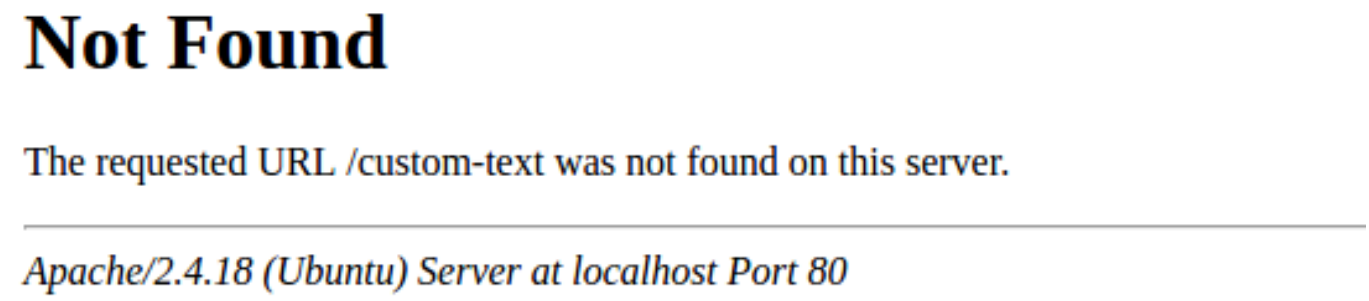
Papildu drošībai varat izmantot rīku *mod_security* kopā ar Apache (pakotne *libapache2-mod-security2*). Šis rīks ļauj noņemt servera parakstu, nevis tikai noņemt versiju no servera paraksta. Servera parakstu var mainīt uz jebkuru vērtību, pēc *mod_security* instalēšanas iestatot "SecServerSignature" uz jebkuru vēlamo vērtību.

Lūdzu, skatiet sava tīmekļa servera dokumentāciju, lai uzzinātu, kā noņemt/mainīt programmatūras parakstus.

Tīmekļa servera noklusējuma kļūdu lapu atspējošana

Lai izvairītos no informācijas atklāšanas, ieteicams atspējot noklusējuma kļūdu lapas.

Pēc noklusējuma tīmekļa serveris izmanto iebūvētās kļūdu lapas:



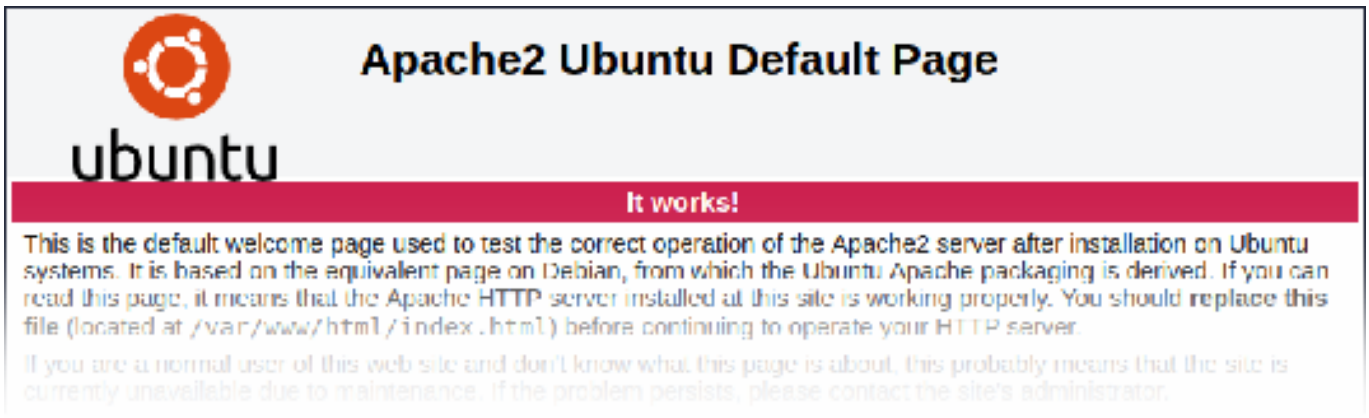
Šīs noklusējuma kļūdu lapas būtu jāaizstāj/jānoņem. Piemēram, Apache tīmekļa serverī var izmantot direktīvu "ErrorDocument", lai definētu pielāgotu kļūdu lapu/tekstu.

Lūdzu, skatiet sava tīmekļa servera dokumentāciju, lai uzzinātu, kā aizstāt/noņemt noklusējuma kļūdu lapas.

Tīmekļa servera testa lapas noņemšana

Lai izvairītos no informācijas atklāšanas, ir ieteicams noņemt tīmekļa servera testa lapu.

Pēc noklusējuma Apache tīmekļa servera tīmekļa saknes direktorijā ir `index.html` testa lapa:



Lūdzu, skatiet sava tīmekļa servera dokumentāciju, lai uzzinātu, kā noņemt noklusējuma testa lapas.

Iestatīt X-Frame-Options HTTP atbildes galveni

Pēc noklusējuma Zabbix ir konfigurēts ar parametru *Use X-Frame-Options HTTP header*, kam iestatīta vērtība `SAMEORIGIN`. Tas nozīmē, ka saturu var ielādēt tikai rāmī, kam ir tāda pati izcelsme kā pašai lapai.

Zabbix lietotāja saskarnes elementi, kas iegūst saturu no ārējiem URL (proti, URL paneļa logrīks), parāda iegūto saturu smilškastē ar iespējotiem visiem smilškastes ierobežojumiem.

Šie iestatījumi uzlabo Zabbix lietotāja saskarnes drošību un nodrošina aizsardzību pret XSS un klikšķu pārtveršanas uzbrukumiem. *Super admin* lietotāji pēc vajadzības var mainīt parametrus *Use iframe sandboxing* un *Use X-Frame-Options HTTP header*. Pirms noklusējuma iestatījumu maiņas, lūdzu, rūpīgi izvērtējiet riskus un ieguvumus. Nav ieteicams pilnībā izslēgt iframe smilškasti vai X-Frame-Options HTTP atbildes galveni.

Faila ar biežāk lietoto paroļu sarakstu paslēpšana

Lai palielinātu paroļu brutālā spēka uzbrukumu sarežģītību, ieteicams ierobežot piekļuvi failam `ui/data/top_passwords.txt`. Šajā failā ir ietverts biežāk lietoto un kontekstam specifisko paroļu saraksts, un tas neļauj lietotājiem iestatīt šādas paroles (ja paroļu politikā ir iespējots parametrs *Izvairīties no viegli uzminamām parolēm*).

Lai ierobežotu piekļuvi failam `top_passwords.txt`, modificējiet sava tīmekļa servera konfigurāciju.

Apache vidē piekļuvi failam var ierobežot, izmantojot `.htaccess` failu:

```
<Files "top_passwords.txt">
    Order Allow,Deny
    Deny from all
</Files>
```

NGINX vidē piekļuvi failam var ierobežot, izmantojot `location` direktīvu:

```
location = /data/top_passwords.txt {
    deny all;
    return 404;
}
```

2 Konfigurācijas labākā prakse

Pārskats

Šajā sadaļā ir izklāstīta labākā prakse Zabbix konfigurēšanai, lai nodrošinātu optimālu veiktspēju un ērtu lietošanu. Ieteikumi ir balstīti uz Zabbix izstrādātāju padomiem un Zabbix pasniedzēju un atbalsta inženieru praktisko pieredzi.

Katra Zabbix instalācija ir unikāla, un daļa no šīm vadlīnijām var nebūt piemērota jūsu konkrētajai konfigurācijai. Tomēr ieteicams pēc iespējas vairāk ievērot šīs vadlīnijas, lai izvairītos no bieži sastopamām iespējamām problēmām.

> **Note:**
> Ja uzskatāt, ka šo lapu varētu uzlabot, mēs labprāt uzklausīsim jūsu viedokli! Lūdzu, iezīmējiet attiecīgo tekstu un nospiediet **ctrl+Enter**, lai ziņotu par kļūdu vai dalītos ar savu atsauksmi.

Hosti un vienumi Hosta definēšana

Hosts Zabbix sistēmā nav fiziska mašīna vai ierīce, bet gan loģiska vienība. Uzraudzības nolūkos varat izveidot atsevišķus hostus datubāzei vai, piemēram, virtuālajai mašīnai. Alternatīvi varat izveidot vispārīgu hostu *Jāņa klēpjdators* un uzraudzīt visus rādītājus šajā hostā.

Labākā prakse ir izveidot atsevišķu hostu katrai neatkarīgai instancei, piemēram, virtuālajai mašīnai, datubāzei, konteineram vai tīkla komutatoram. Izmantojot šo pieeju, jūs:

1. Izvairīsieties no jucekļa uzraudzības datos, jo katram hostam būs atsevišķi vienumi, trigeri un brīdinājumu paziņojumi.
2. Precīzāk pielāgosiet lietotāju piekļuves līmeņus. Varat konfigurēt lietotāju lomas, lai piešķirtu piekļuvi tikai noteiktu hostu skatīšanai un/vai konfigurēšanai. Skatiet arī mazāko privilēģiju principu.

Dublēti vienumi hostos

Ja jums ir vairāki līdzīgi hosti, piemēram, *Network switch 1* un *Network switch 2*, Zabbix nodrošina vairākus veidus, kā ātri atkārtoti izveidot hostu. Jūs varat vienkārši klonēt hostu ar visiem tā mērījumiem, nospiežot pogu Clone, taču šajā gadījumā, lai vēlāk atjauninātu vienumu, tas būs jādara manuāli katrā hostā.

Labākā prakse ir izveidot veidni ar visiem nepieciešamajiem mērījumiem, piemēram, *Network switch template*. Pēc tam sagrupējiet līdzīgos hostus hostu grupā; iepriekš minētajā piemērā tā varētu būt *Network switches*. Tagad sadaļā *Data Collection -> Hosts* varat filtrēt visus hostus pēc hostu grupas un izmantot pogu *Mass update*, lai piesaistītu veidni visiem saviem tīkla komutatoriem.

Atkarīgie vienumi

Lai samazinātu pieprasījumu skaitu mērķa entītijai, Zabbix ļauj izveidot galvenos un atkarīgos vienumus. Šajā gadījumā galvenais vienums vienā pieprasījumā apkopo lielu informācijas kopu. Pēc tam atkarīgos vienumus var konfigurēt tā, lai tie, izmantojot priekšapstrādi, no šīs kopas iegūtu konkrētus datu fragmentus un saglabātu tos kā atsevišķus metriku rādītājus.

Piemēram, galvenais vienums var iegūt JSON vai XML atbildi, kas satur vairākus metriku rādītājus, vai izpildīt datubāzes vaicājumu, kas atgriež vairākas datu kolonnas (piemēram, atvērto savienojumu skaitu, pārtraukto savienojumu skaitu, maksimālo atļauto vienlaicīgo savienojumu skaitu un kopējo kumulatīvo savienojumu skaitu kopš palaišanas), un atkarīgie vienumi analizēs un saglabās katru nepieciešamo vērtību atsevišķi.

Ieteicamā prakse šādai konfigurācijai ir atmest galvenā vienuma vēsturi uzreiz pēc apkopošanas un saglabāt tikai atkarīgo vienumu datus.

Serveri un starpniekserveri

Ja visi hosti atrodas tajā pašā lokālajā tīklā, kur Zabbix serveris, un nav mērogojamības vai veiktspējas problēmu, starpniekserveris var nebūt nepieciešams. Lielākās vai sarežģītākās vidēs ar tiešu hostu uzraudzību, izmantojot Zabbix serveri, var nepietikt. Pievienojot starpniekserveri un piešķirot tam daļu hostu, iespējams nodrošināt vienmērīgāku slodzes sadalījumu.

Labākā prakse ir pievienot Zabbix starpniekserveri, ja:

1. Jūs uzraugāt vairākus hostus, izmantojot dažādas metriku vākšanas metodes aiz ugunsmūra. Starpniekserveris apkopos datus no hostiem un pārsūtīs tos uz Zabbix serveri, samazinot nepieciešamību atvērt vairākus ugunsmūra portus.
2. Jūs uzraugāt attālinātas atrašanās vietas, filiāles un/vai tīklus. Ja starp Zabbix serveri un jūsu attālinātajām atrašanās vietām rodas tīkla pārtraukums, attālinātajās atrašanās vietās izvietotie Zabbix starpniekserveri turpinās datu vākšanu un nosūtīs savāktos datus atpakaļ uz Zabbix serveri, tiklīdz tīkla savienojums tiks atjaunots.
3. Jums ir liela mēroga izvietojums un vēlaties samazināt slodzi uz Zabbix serveri un uzlabot veiktspēju. Liela mēroga izvietojuma definīcija ir ļoti plaša un ir atkarīga ne tikai no hostu skaita, bet arī no sekundē savākto vērtību skaita.

Slepenie makrosi

Jūs varat izmantot slepenos lietotāja makrosus kā slepenu tekstu vai slepenos glabātuves makrosus.

Lai uzlabotu drošību, izmantojot slepenos glabātuves makrosus, ieteicams konfigurēt makrosu vērtību iegūšanu neatkarīgi Zabbix serverim un Zabbix starpniekserveriem. Pēc noklusējuma slepeno makrosu vērtības iegūst Zabbix serveris un tās tiek nodotas Zabbix starpniekserveriem.

18 API

Pārskats Zabbix API ļauj programmatiski izgūt un mainīt Zabbix konfigurāciju, kā arī nodrošina piekļuvi vēsturiskajiem datiem. To plaši izmanto, lai:

- izveidotu jaunas lietojumprogrammas darbam ar Zabbix;
- integrētu Zabbix trešo pušu programmatūrā;
- automatizētu ikdienas uzdevumus.

Zabbix API ir uz HTTP balstīts API, un tas tiek piegādāts kā daļa no tīmekļa lietotāja saskarnes. Tas izmanto JSON-RPC 2.0 protokolu, kas nozīmē divas lietas:

- API sastāv no atsevišķu metožu kopas;
- pieprasījumi un atbildes starp klientiem un API tiek kodēti, izmantojot JSON formātu.

Plašāku informāciju par protokolu un JSON skatiet JSON-RPC 2.0 specifikācijā un JSON formāta mājaslapā.

Plašāku informāciju par Zabbix funkcionalitātes integrēšanu jūsu Python lietojumprogrammās skatiet Python bibliotēka Zabbix.

> **Note:**
> Lietotāja piekļuve Zabbix, tostarp gan konfigurācijai, gan vēsturiskajiem datiem, ir atkarīga no lietotāja tipa, piešķirtās lietotāja lomas un lietotāju grupām.

Struktūra API sastāv no vairākām metodēm, kas nosacīti ir sagrupētas atsevišķos API. Katra metode veic vienu konkrētu uzdevumu. Piemēram, metode `host.create` pieder *host* API un tiek izmantota jaunu hostu izveidei. Vēsturiski API dažkārt tiek dēvēti par "klasēm".

> **Note:**
> Lielākā daļa API satur vismaz četras metodes: `get`, `create`, `update` un `delete`, kas paredzētas attiecīgi datu iegūšanai, izveidei, atjaunināšanai un dzēšanai, taču daži API var nodrošināt pilnīgi atšķirīgu metožu kopu.

Pieprasījumu veikšana Kad esat iestatījis lietotāja saskarne, varat izmantot attālinātus HTTP pieprasījumus, lai izsauktu API. Lai to izdarītu, jums ir jānosūta HTTP POST pieprasījumi uz `api_jsonrpc.php` failu, kas atrodas lietotāja saskarne direktorijā. Piemēram, ja jūsu Zabbix lietotāja saskarne ir instalēta vietnē `https://example.com/zabbix`, HTTP pieprasījums `apiinfo.version` metodes izsaukšanai var izskatīties šādi:

```
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}'
```

Pieprasījumam jābūt ar `Content-Type` galveni, kurai iestatīta viena no šīm vērtībām: `application/json-rpc`, `application/json` vai `application/jsonrequest`.

Pieprasījuma objektam jāsatur šādas īpašības:

- `jsonrpc` - JSON-RPC protokola versija, ko izmanto API (Zabbix API īsteno JSON-RPC 2.0. versiju);
- `method` - izsauktā API metode;
- `params` - parametri, kas tiks nodoti API metodei;
- `id` - patvaļīgs pieprasījuma identifikators (ja tas ir izlaists, API apstrādā pieprasījumu kā paziņojumu).

Ja pieprasījums ir pareizs, API atgrieztajai atbildei vajadzētu izskatīties šādi:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Savukārt atbildes objekts satur šādas īpašības:

- `jsonrpc` - JSON-RPC protokola versija;
- `result` - metodes atgrieztie dati;
- `id` - atbilstošā pieprasījuma identifikators.

Darbplūsmas piemērs Nākamajā sadaļā jūs iepazīstināsim ar dažiem lietošanas piemēriem sīkāk.

Autentifikācija Lai piekļūtu jebkādiem datiem Zabbix, jums ir jāizmanto viens no šiem variantiem:

- izmantot esošu API tokenu (izveidotu Zabbix lietotāja saskarnē vai izmantojot Token API);
- izmantot autentifikācijas tokenu, kas iegūts ar user.login metodi.

Piemēram, ja vēlaties iegūt jaunu autentifikācijas tokenu, piesakoties kā standarta *Admin* lietotājs, tad JSON pieprasījums izskatītos šādi:

```
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}
```

Ja norādījāt akreditācijas datus pareizi, API atgrieztajā atbildē jābūt lietotāja autentifikācijas tokenam:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Autorizācijas metodes Ar "Authorization" galveni

Visiem API pieprasījumiem ir nepieciešama autentifikācija vai API marķieris. Jūs varat norādīt akreditācijas datus, izmantojot pieprasījumā galveni Authorization:

```
curl --request POST \
  --url 'https://example.com/zabbix/api_jsonrpc.php' \
  --header 'Authorization: Bearer 0424bd59b807674191e7d77572075f33'
```

Attention:
Ja rodas autentifikācijas problēmas, skatiet Authorization header forwarding.

Zabbix API pieņem galvenes neatkarīgi no burtu reģistra (piemēram, `authorization`, `Authorization` un `AUTHORIZATION` tiek apstrādātas vienādi).

Authorization galvene tiek atbalstīta starpizcelsmes pieprasījumos (CORS).

Ar Zabbix sīkdatni

Sīkdatne *"zbx_session"* tiek izmantota, lai autorizētu API pieprasījumu no Zabbix lietotāja saskarnes, kas izpildīts, izmantojot JavaScript (no moduļa vai pielāgota logrīka).

Hostu izgūšana Tagad jums ir derīgs lietotāja autentifikācijas marķieris (attēlots kā mainīgais tālākajos piemēros), ko var izmantot, lai piekļūtu datiem Zabbix. Piemēram, varat izmantot metodi host.get, lai izgūtu visu konfigurēto hostu ID, hostu nosaukumus un saskarnes:

Pieprasījums:

```
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` ir fails, kas satur JSON vaicājumu. Faila vietā varat nodot vaicājumu argumentā `--data`.

data.json

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "hostid",
            "host"
        ],
        "selectInterfaces": [
            "interfaceid",
            "ip"
        ]
    },
    "id": 2
}
```

Atbildes objekts saturēs pieprasītos datus par hostiem:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "interfaces": [
                {
                    "interfaceid": "1",
                    "ip": "127.0.0.1"
                }
            ]
        }
    ],
    "id": 2
}
```

Note:
Veiktspējas apsvērumu dēļ vienmēr ir ieteicams norādīt objektu īpašības, kuras vēlaties izgūt. Tādējādi jūs izvairīsieties no visa izgūšanas.

Jauna vienuma izveide Tagad izveidojiet jaunu vienumu hostā "Zabbix serveris", izmantojot datus, ko ieguvāt iepriekšējā `host.get` pieprasījumā. To var izdarīt, izmantojot item.create metodi:

```
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_":
```

Veiksmīgā atbildē būs ietverts jaunizveidotā vienuma ID, ko var izmantot, lai atsauktos uz vienumu turpmākajos pieprasījumos:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
        ]
    },
    "id": 3
}
```

Note:
Metode `item.create`, kā arī citas *izveides metodes*, var pieņemt arī objektu masīvus un ar vienu API izsaukumu izveidot vairākus vienumus.

Vairāku trigeru izveide Tādējādi, ja *izveides metodes* pieņem masīvus, varat pievienot vairākus trigerus, piemēram, šos:

```
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":"Procesora slodze ir pārāk a
```

Veiksmīgā atbilde saturēs jaunizveidoto trigeru ID:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 4
}
```

Vienuma atjaunināšana Iespējojiet vienumu, iestatot tā statusu uz ”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}'
```

Veiksmīgā atbildē būs atjaunināta vienuma ID:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 5
}
```

Note:
Metode `item.update`, kā arī citas *atjaunināšanas metodes*, var pieņemt arī objektu masīvus un atjaunināt vairākus vienumus ar vienu API izsaukumu.

Vairāku trigeru atjaunināšana Iespējojiet vairākus trigerus, iestatot to statusu uz "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
```

Veiksmīgā atbilde saturēs atjaunināto trigeru ID:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938",
            "13939"
        ]
    },
    "id": 6
}
```

Kļūdu apstrāde Līdz šim brīdim viss, ko esat mēģinājis, ir darbojies labi. Bet kas notiktu, ja mēģinātu veikt nepareizu API izsaukumu? Mēģiniet izveidot vēl vienu hostu, izsaucot host.create, bet izlaižot obligāto parametru `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,
```

Tad atbildē būs ietverts kļūdas ziņojums:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32602,
        "message": "Invalid params.",
        "data": "No groups for host \"Linux server\"."
    },
    "id": 7
}
```

Ja ir radusies kļūda, tad īpašības `result` vietā atbildes objekts saturēs īpašību `error` ar šādiem datiem:

- `code` - kļūdas kods;
- `message` - īss kļūdas kopsavilkums;
- `data` - detalizētāks kļūdas ziņojums.

Kļūdas var rasties dažādos gadījumos, piemēram, izmantojot nepareizas ievades vērtības, sesijas noildzes gadījumā vai mēģinot piekļūt neeksistējošiem objektiem. Jūsu lietojumprogrammai jāspēj korekti apstrādāt šāda veida kļūdas.

API versijas Lai vienkāršotu API versiju pārvaldību, начиная ar Zabbix 2.0.4, API versija atbilst pašas Zabbix versijai. Varat izmantot apiinfo.version metodi, lai noskaidrotu API versiju, ar kuru strādājat. Tas var būt noderīgi, lai pielāgotu savu lietotni versijai specifisku iespēju izmantošanai.

Zabbix garantē funkcionalitātes atpakaļsaderību vienas galvenās versijas ietvaros. Veicot atpakaļnesaderīgas izmaiņas starp galvenajiem laidieniem, Zabbix parasti atstāj vecās iespējas kā novecojušas nākamajā laidienā un tās noņem tikai vēlākajā laidienā pēc tā. Reizēm Zabbix var noņemt iespējas starp galvenajiem laidieniem, nenodrošinot nekādu atpakaļsaderību. Ir svarīgi nekad nepaļauties uz novecojušām iespējām un pēc iespējas ātrāk pāriet uz jaunākām alternatīvām.

Note:
Visām API veiktajām izmaiņām varat sekot API izmaiņu žurnālā.

Papildu lasāmviela Tagad jums ir pietiekami daudz zināšanu, lai sāktu strādāt ar Zabbix API, tomēr neapstājieties šeit. Tālākai lasīšanai ieteicams apskatīt pieejamo API sarakstu.

Metožu atsauce

Šajā sadaļā ir sniegts Zabbix API nodrošināto funkciju pārskats, un tā palīdzēs jums orientēties pieejamajās klasēs un metodēs.

Uzraudzība Zabbix API ļauj piekļūt vēsturei un citiem datiem, kas savākti uzraudzības laikā.

Paneļi

Pārvaldiet paneļus un veidojiet uz tiem balstītus ieplānotus pārskatus.

Paneļu API | Veidņu paneļu API | Pārskatu API

Augstas pieejamības klasteris

Iegūstiet servera mezglu sarakstu un to statusu.

Augstas pieejamības klastera API

Vēsture

Iegūstiet vēsturiskās vērtības, ko Zabbix uzraudzības procesi apkopojuši attēlošanai vai tālākai apstrādei.

Vēstures API

Tendences

Iegūstiet tendenču vērtības, ko aprēķinājis Zabbix serveris, attēlošanai vai turpmākai apstrādei.

Tendenču API

Notikumi

Iegūstiet notikumus, ko ģenerējuši trigeri, tīkla atklāšana un citas Zabbix sistēmas, lai nodrošinātu elastīgāku situāciju pārvaldību vai integrāciju ar trešo pušu rīkiem.

Notikumu API

Problēmas

Iegūt problēmas atbilstoši norādītajiem parametriem.

Problem API

Kartes

Konfigurējiet kartes, lai izveidotu detalizētus dinamiskus savas IT infrastruktūras attēlojumus.

Karšu API

Uzdevumi

Mijiedarbojieties ar Zabbix serveris uzdevumu pārvaldnieku, izveidojot uzdevumus un saņemot atbildi.

Uzdevumu API

Pakalpojumi Zabbix API ļauj piekļūt datiem, kas iegūti pakalpojumu uzraudzības laikā.

Pakalpojuma līmeņa vienošanās

Definējiet pakalpojuma līmeņa mērķus (SLO), iegūstiet detalizētu pakalpojuma līmeņa indikatoru (SLI) informāciju par pakalpojuma veiktspēju.

SLA API

Pakalpojumi

Pārvaldiet pakalpojumus pakalpojumu līmeņa uzraudzībai un iegūstiet detalizētu SLA informāciju par jebkuru pakalpojumu.

Pakalpojumu API

Datu vākšana Zabbix API ļauj pārvaldīt jūsu uzraudzības sistēmas konfigurāciju.

Hosti un hostu grupas

Pārvaldiet hostu grupas, hostus un visu ar tiem saistīto, tostarp hostu saskarnes, hostu makrosus un uzturēšanas periodus.

Host API | Host group API | Host interface API | User macro API | Value map API | Maintenance API

Vienumi

Definējiet vienumus uzraudzībai.

Vienumu API

Trigeri

Konfigurējiet trigerus, lai saņemtu paziņojumus par problēmām jūsu sistēmā. Pārvaldiet trigeru atkarības.

Trigeru API

Grafiki

Rediģējiet grafikus vai atsevišķus grafiku vienumus, lai labāk attēlotu savāktos datus.

Grafiku API | Grafiku vienumu API

Veidnes un veidņu grupas

Pārvaldiet veidnes un saistiet tās ar hostiem vai citām veidnēm.

Veidņu API | Veidņu grupu API | Vērtību kartējumu API

Zema līmeņa atklāšana

Konfigurējiet zema līmeņa atklāšanas noteikumus, kā arī vienumu, trigeru un grafiku prototipus, lai uzraudzītu dinamiskas entītijas.

LLD noteikuma API | Vienuma prototipa API | Trigera prototipa API | Grafika prototipa API | Hosta prototipa API

Notikumu korelācija

Izveidojiet pielāgotus notikumu korelācijas noteikumus.

Korelācijas API

Tīkla atklāšana

Pārvaldiet tīkla līmeņa atklāšanas noteikumus, lai automātiski atrastu un uzraudzītu jaunus hostus. Iegūstiet pilnu piekļuvi informācijai par atklātajiem pakalpojumiem un hostiem.

Atklāšanas noteikumu API | Atklāšanas pārbaužu API | Atklātā hosta API | Atklātā pakalpojuma API

Eksportēšana un importēšana

Eksportējiet un importējiet Zabbix konfigurācijas datus konfigurācijas dublējumkopijām, migrācijai vai liela mēroga konfigurācijas atjauninājumiem.

Konfigurācijas API

Tīmekļa uzraudzība

Konfigurējiet tīmekļa scenārijus, lai uzraudzītu savas tīmekļa lietojumprogrammas un pakalpojumus.

Tīmekļa scenārija API

Brīdinājumi Zabbix API ļauj pārvaldīt jūsu uzraudzības sistēmas darbības un brīdinājumus.

Darbības un brīdinājumi

Definējiet darbības un operācijas, lai paziņotu lietotājiem par noteiktiem notikumiem vai automātiski izpildītu attālinātās komandas. Iegūstiet piekļuvi informācijai par ģenerētajiem brīdinājumiem un to saņēmējiem.

Darbību API | Brīdinājumu API

Mediju tipi

Konfigurējiet mediju tipus un vairākus veidus, kā lietotāji saņems brīdinājumus.

Mediju tipu API

Skripti

Konfigurējiet un izpildiet skriptus, lai palīdzētu sev ikdienas uzdevumos.

Skriptu API

Lietotāji Zabbix API ļauj pārvaldīt jūsu uzraudzības sistēmas lietotājus.

Lietotāji un lietotāju grupas

Pievienojiet lietotājus, kuriem būs piekļuve Zabbix, piešķiriet tos lietotāju grupām un piešķiriet atļaujas. Izveidojiet lomas detalizētai lietotāju tiesību pārvaldībai.

Lietotāja API | Lietotāju grupu API | Lietotāju direktorija API | Lietotāju lomu API

API tokeni

Pārvaldiet autorizācijas tokenus.

Token API

Autentifikācija

Mainīt autentifikācijas konfigurācijas opcijas.

Autentifikācijas API

Administrēšana Ar Zabbix API jūs varat mainīt savas uzraudzības sistēmas administrēšanas iestatījumus.

Vispārīgi

Mainiet noteiktas globālās konfigurācijas opcijas.

Autoreģistrācijas API | Ikonu kartes API | Attēlu API | Iestatījumu API | Regulāro izteiksmju API | Moduļu API | Savienotāju API

Audita žurnāls

Sekojiet līdzi konfigurācijas izmaiņām, ko veicis katrs lietotājs.

Audita žurnāla API

Tīrīšana

Konfigurējiet tīrīšanu.

Tīrīšanas API

Starpniekserveri un starpniekserveru grupas

Pārvaldiet starpniekserverus, kas tiek izmantoti jūsu sadalītās uzraudzības iestatījumā.

Starpniekservera API | Starpniekserveru grupas API

Makrosi

Pārvaldiet makrosus.

Lietotāja makro API

API informācija Iegūstiet Zabbix API versiju, lai jūsu lietojumprogramma varētu izmantot konkrētajai versijai raksturīgās iespējas.

API info API

API informācija

Šī klase ir paredzēta API metainformācijas iegūšanai.

Pieejamās metodes:

- apiinfo.version - iegūt Zabbix API versiju

apiinfo.version

Apraksts

`string apiinfo.version(array)`

Šī metode ļauj iegūt Zabbix API versiju.

> **Attention:**
> Šī metode ir pieejama tikai neautentificētiem lietotājiem.

Parametri

`(array)` Metode pieņem tukšu masīvu.

Atgrieztās vērtības

`(string)` Atgriež Zabbix API versiju.

> **Note:**
> Sākot ar Zabbix 2.0.4, API versija atbilst Zabbix versijai.

Piemēri

API versijas iegūšana

Iegūstiet Zabbix API versiju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "apiinfo.version",
    "params": [],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": "8.0.0",
    "id": 1
}
```

Avots

CAPIInfo::version() failā *ui/include/classes/api/services/CAPIInfo.php*.

Atklātais hosts

Šī klase ir paredzēta darbam ar atklātajiem hostiem.

Objektu atsauces:

- Atklātais hosts

Pieejamās metodes:

- dhost.get - izgūt atklātos hostus

Atklātā hosta objekts

Tālāk norādītie objekti ir tieši saistīti ar `dhost` API.

Atklātais hosts

> **Note:**
> Atklātos hostus izveido Zabbix serveris, un tos nevar mainīt, izmantojot API.

Atklātā hosta objekts satur informāciju par hostu, kas atklāts ar tīkla atklāšanas kārtulu. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
dhostid	ID	Atklātā hosta ID.
druleid	ID	Atklāšanas kārtulas ID, kas noteica hostu.
lastdown	timestamp	Laiks, kad atklātais hosts pēdējo reizi kļuva nepieejams.
lastup	timestamp	Laiks, kad atklātais hosts pēdējo reizi kļuva pieejams.
status	integer	Vai atklātais hosts ir pieejams vai nepieejams. Hosts ir pieejams, ja tam ir vismaz viens aktīvs atklātais serviss. Iespējamās vērtības: 0 - hosts pieejams; 1 - hosts nepieejams.

dhost.get

Apraksts

`integer/array dhost.get(object parameters)`

Metode ļauj izgūt atklātos hostus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas, lai iegūtu vairāk informācijas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dhostids	ID/array	Atgriezt tikai atklātos hostus ar norādītajiem ID.
druleids	ID/array	Atgriezt tikai atklātos hostus, kas izveidoti ar norādītajiem atklāšanas noteikumiem.
dserviceids	ID/array	Atgriezt tikai atklātos hostus, kuros darbojas norādītie pakalpojumi.
selectDRules	query	Atgriezt `drules` īpašību ar atklāšanas noteikumu masīvu, kas noteica hostu.
selectDServices	query	Atgriezt `dservices` īpašību ar hostā darbojošos atklāto pakalpojumu masīvu. Atbalsta `count`.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectDServices` — rezultāti tiks kārtoti pēc `dserviceid`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `dhostid`, `druleid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

(integer/array) Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs countOutput.

Piemēri

Atgūt ar atklāšanas kārtulu atrastos hostus

Atgūst visus hostus un to atrastos servisus, kas tajos darbojas un ir noteikti ar atklāšanas kārtulu "4".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dhost.get",
    "params": {
        "output": "extend",
        "selectDServices": "extend",
        "druleids": "4"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Atklātais pakalpojums

- Atklāšanas noteikums

Avots

CDHost::get() failā *ui/include/classes/api/services/CDHost.php*.

Atklātais pakalpojums

Šī klase ir paredzēta darbam ar atklātajiem pakalpojumiem.

Objekta atsauces:

- Atklātais pakalpojums

Pieejamās metodes:

- dservice.get - izgūt atklātos pakalpojumus

Atklātais pakalpojuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `dservice` API.

Atklātais serviss

> **Note:**
> Atklātos servisus izveido Zabbix serveris, un tos nevar modificēt, izmantojot API.

Atklātā servisa objekts satur informāciju par servisu, ko hosts atklāšanas noteikums ir atklājis tīkla atklāšanas kārtulas ietvaros. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
dserviceid	ID	Atklātā servisa ID.
dcheckid	ID	Servisa noteikšanai izmantotās atklāšanas pārbaudes ID.
dhostid	ID	Atklātā hosta, kurā darbojas serviss, ID.
dns	string	Hosta, kurā darbojas serviss, DNS.
ip	string	Hosta, kurā darbojas serviss, IP adrese.
lastdown	timestamp	Laiks, kad atklātais serviss pēdējo reizi kļuva nepieejams.
lastup	timestamp	Laiks, kad atklātais serviss pēdējo reizi kļuva pieejams.
port	integer	Servisa porta numurs.
status	integer	Servisa statuss. Iespējamās vērtības: 0 - serviss pieejams; 1 - serviss nav pieejams.
value	string	Vērtība, ko serviss atgriež, veicot Zabbix aģents, SNMPv1, SNMPv2 vai SNMPv3 atklāšanas pārbaudi.

dservice.get

Apraksts

`integer/array dservice.get(object parameters)`

Metode ļauj izgūt atklātos servisus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dserviceids	ID/array	Atgriezt tikai atklātos servisus ar norādītajiem ID.
dhostids	ID/array	Atgriezt tikai atklātos servisus, kas pieder norādītajiem atklātajiem hostiem.
dcheckids	ID/array	Atgriezt tikai atklātos servisus, kurus ir noteikušas norādītās atklāšanas pārbaudes.
druleids	ID/array	Atgriezt tikai atklātos servisus, kurus ir noteikuši norādītie atklāšanas noteikumi.
selectDRules	query	Atgriezt `drules` īpašību ar atklāšanas noteikumu masīvu, kas noteica servisu.
selectDHosts	query	Atgriezt `dhosts` īpašību ar atklāto hostu masīvu, kuriem serviss pieder.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kuriem ir tāda pati IP adrese un starpniekserveris kā servisam. Atbalsta `count`.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectHosts` - rezultāts tiks kārtots pēc `hostid`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `dserviceid`, `dhostid`, `ip`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Atgūt hostā atklātos servisus

Atgūt visus atklātos servisus, kas noteikti atklātajā hostā ”11”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dservice.get",
    "params": {
        "output": "extend",
        "dhostids": "11"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatīt arī

- Atklātais hosts
- Atklāšanas pārbaude
- Hosts

Avots

CDService::get() failā *ui/include/classes/api/services/CDService.php*.

Atklāšanas kārtula

Šī klase ir paredzēta darbam ar tīkla atklāšanas kārtulām.

> **Note:**
> Šī API ir paredzēta darbam ar tīkla atklāšanas kārtulām. Zema līmeņa atklāšanas kārtulām skatiet LLD kārtulu API.

Objektu atsauces:

- Atklāšanas kārtula

Pieejamās metodes:

- drule.create - izveidot jaunas tīkla atklāšanas kārtulas
- drule.delete - dzēst tīkla atklāšanas kārtulas
- drule.get - izgūt tīkla atklāšanas kārtulas
- drule.update - atjaunināt tīkla atklāšanas kārtulas

Atklāšanas kārtulas objekts

Tālāk norādītie objekti ir tieši saistīti ar `drule` API.

Atklāšanas kārtula

Atklāšanas kārtulas objekts definē tīkla atklāšanas kārtulu. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
druleid	ID	Atklāšanas kārtulas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
iprange	string	Viens vai vairāki pārbaudāmie IP diapazoni, atdalīti ar komatiem. Papildinformāciju par atbalstītajiem IP diapazonu formātiem skatiet sadaļā tīkla atklāšanas konfigurācija. Īpašības darbība: - *obligāts* izveides operācijām
name	string	Atklāšanas kārtulas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
delay	string	Atklāšanas kārtulas izpildes intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d), vai lietotāja makrosu. Noklusējums: 1h.
proxyid	ID	Atklāšanai izmantotā starpniekservera ID.
status	integer	Vai atklāšanas kārtula ir iespējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.
concurrency_max	integer	Maksimālais vienlaicīgo pārbaužu skaits vienai atklāšanas kārtulai. Iespējamās vērtības: 0 - *(noklusējums)* neierobežots pārbaužu skaits; 1 - viena pārbaude; 2-999 - pielāgots pārbaužu skaits.
error	string	Kļūdas teksts, ja atklāšanas kārtulas izpildes laikā radušās problēmas. Īpašības darbība: - *tikai lasāms*

drule.create

Apraksts

`object drule.create(object/array discoveryRules)`

Šī metode ļauj izveidot jaunus atklāšanas noteikumus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atklāšanas kārtulas, ko izveidot.

Papildus standarta atklāšanas kārtulu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
dchecks	array	Atklāšanas pārbaudes, ko izveidot atklāšanas kārtulai. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto atklāšanas noteikumu ID īpašumā `druleids`. Atgriezto ID secība atbilst nodoto atklāšanas noteikumu secībai.

Piemēri

Izveidot atklāšanas kārtulu

Izveidojiet atklāšanas kārtulu, lai atrastu iekārtas, kurās lokālajā tīklā darbojas Zabbix aģents. Kārtulai jāizmanto viena Zabbix aģenta pārbaude portā 10050.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Atklāšanas pārbaude

Avots

CDRule::create() failā *ui/include/classes/api/services/CDRule.php*.

drule.delete

Apraksts

`object drule.delete(array discoveryRuleIds)`

Šī metode ļauj dzēst atklāšanas noteikumus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` atklāšanas noteikumu ID, ko dzēst.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto atklāšanas noteikumu ID zem `druleids` īpašības.

Piemēri

Dzēst vairākus atklāšanas noteikumus

Dzēsiet divus atklāšanas noteikumus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "drule.delete",
    "params": [
        "4",
        "6"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "4",
            "6"
        ]
    },
    "id": 1
}
```

Avots

CDRule::delete() failā *ui/include/classes/api/services/CDRule.php*.

drule.get

Apraksts

`integer/array drule.get(object parameters)`

Metode ļauj izgūt atklāšanas kārtulas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dhostids	ID/array	Atgriezt tikai atklāšanas kārtulas, kas izveidoja norādītos atklātos hostus.

Parametrs	Tips	Apraksts
druleids	ID/array	Atgriezt tikai atklāšanas kārtulas ar norādītajiem ID.
dserviceids	ID/array	Atgriezt tikai atklāšanas kārtulas, kas izveidoja norādītos atklātos servisus.
selectDChecks	query	Atgriezt `dchecks` īpašību ar atklāšanas pārbaudēm, ko izmanto atklāšanas kārtula. Atbalsta `count`.
selectDHosts	query	Atgriezt `dhosts` īpašību ar atklātajiem hostiem, ko izveidoja atklāšanas kārtula. Atbalsta `count`.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectDChecks` - rezultāti tiks kārtoti pēc `dcheckid`; `selectDHosts` - rezultāti tiks kārtoti pēc `dhostsid`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `druleid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt visus atklāšanas noteikumus

Iegūst visus konfigurētos atklāšanas noteikumus un to izmantotās atklāšanas pārbaudes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "drule.get",
    "params": {
        "output": "extend",
        "selectDChecks": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Atklātais hosts
- Atklāšanas pārbaude

Avots

CDRule::get() failā *ui/include/classes/api/services/CDRule.php*.

drule.update

Apraksts

`object drule.update(object/array discoveryRules)`

Šī metode ļauj atjaunināt esošos atklāšanas noteikumus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atklāšanas kārtulu rekvizīti, kas jāatjaunina.

Rekvizītam `druleid` jābūt definētam katrai atklāšanas kārtulai, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta atklāšanas kārtulu rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
dchecks	array	Atklāšanas pārbaudes, kas aizstās esošās pārbaudes.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto atklāšanas noteikumu ID īpašumā `druleids`.

Piemēri

Mainīt atklāšanas kārtulas IP diapazonu

Mainiet atklāšanas kārtulas IP diapazonu uz "192.168.2.1-255".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "drule.update",
    "params": {
        "druleid": "6",
        "iprange": "192.168.2.1-255"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "druleids": [
            "6"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Atklāšanas pārbaude

Avots

CDRule::update() failā *ui/include/classes/api/services/CDRule.php*.

Atklāšanas pārbaude

Šī klase ir paredzēta darbam ar atklāšanas pārbaudēm.

Objektu atsauces:

- Atklāšanas pārbaude

Pieejamās metodes:

- dcheck.get - izgūt atklāšanas pārbaudes

Atklāšanas pārbaudes objekts

Tālāk norādītie objekti ir tieši saistīti ar `dcheck` API.

Atklāšanas pārbaude

Atklāšanas pārbaudes objekts definē konkrētu pārbaudi, ko veic tīkla atklāšanas kārtula. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
dcheckid	ID	Atklāšanas pārbaudes ID.
druleid	ID	Tās atklāšanas kārtulas ID, kurai pieder pārbaude.
key_	string	Vienuma atslēga (ja `type` ir iestatīts uz ”Zabbix aģents”) vai SNMP OID (ja `type` ir iestatīts uz ”SNMPv1 aģents”, ”SNMPv2 aģents” vai ”SNMPv3 aģents”). Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Zabbix aģents”, ”SNMPv1 aģents”, ”SNMPv2 aģents” vai ”SNMPv3 aģents”

Īpašība	Tips	Apraksts
ports	string	Viens vai vairāki pārbaudāmo portu diapazoni, atdalīti ar komatiem. Noklusējums: 0. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SSH” (0), ”LDAP” (1), ”SMTP” (2), ”FTP” (3), ”HTTP” (4), ”POP” (5), ”NNTP” (6), ”IMAP” (7), ”TCP” (8), ”Zabbix aģents” (9), ”SNMPv1 aģents” (10), ”SNMPv2 aģents” (11), ”SNMPv3 aģents” (13), ”HTTPS” (14) vai ”Telnet” (15)
snmp_community	string	SNMP kopiena. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”SNMPv1 aģents” vai ”SNMPv2 aģents”
snmpv3_authpassphrase	string	Autentifikācijas parole. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents” un `snmpv3_securitylevel` ir iestatīts uz ”authNoPriv” vai ”authPriv”
snmpv3_authprotocol	integer	Autentifikācijas protokols. Iespējamās vērtības: 0 - *(noklusējums)* MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents” un `snmpv3_securitylevel` ir iestatīts uz ”authNoPriv” vai ”authPriv”
snmpv3_contextname	string	SNMPv3 konteksta nosaukums. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents”
snmpv3_privpassphrase	string	Privātuma parole. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents” un `snmpv3_securitylevel` ir iestatīts uz ”authPriv”
snmpv3_privprotocol	integer	Privātuma protokols. Iespējamās vērtības: 0 - *(noklusējums)* DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents” un `snmpv3_securitylevel` ir iestatīts uz ”authPriv”
snmpv3_securitylevel	string	Drošības līmenis. Iespējamās vērtības: 0 - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents”

Īpašība	Tips	Apraksts
snmpv3_securityname	string	Drošības nosaukums. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”SNMPv3 aģents”
type	integer	Pārbaudes tips. Iespējamās vērtības: 0 - SSH; 1 - LDAP; 2 - SMTP; 3 - FTP; 4 - HTTP; 5 - POP; 6 - NNTP; 7 - IMAP; 8 - TCP; 9 - Zabbix aģents; 10 - SNMPv1 aģents; 11 - SNMPv2 aģents; 12 - ICMP ping; 13 - SNMPv3 aģents; 14 - HTTPS; 15 - Telnet. Īpašības darbība: - *obligāta*
uniq	integer	Vai izmantot šo pārbaudi kā ierīces unikalitātes kritēriju. Vienai atklāšanas kārtulai var konfigurēt tikai vienu unikālu pārbaudi. Iespējamās vērtības: 0 - *(noklusējums)* neizmantot šo pārbaudi kā unikalitātes kritēriju; 1 - izmantot šo pārbaudi kā unikalitātes kritēriju. Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”Zabbix aģents”, ”SNMPv1 aģents”, ”SNMPv2 aģents” vai ”SNMPv3 aģents”
host_source	integer	Avots hosta nosaukumam. Iespējamās vērtības: 1 - *(noklusējums)* DNS; 2 - IP; 3 - šīs pārbaudes atklāšanas vērtība.
name_source	integer	Avots redzamajam nosaukumam. Iespējamās vērtības: 0 - *(noklusējums)* nav norādīts; 1 - DNS; 2 - IP; 3 - šīs pārbaudes atklāšanas vērtība.
allow_redirect	integer	Atļaut situāciju, kad mērķis, kuram tiek veikts ICMP ping, atbild no citas IP adreses. Iespējamās vērtības: 0 - *(noklusējums)* apstrādāt pāradresētās atbildes tā, it kā mērķa hosts nebūtu pieejams (kļūme); 1 - apstrādāt pāradresētās atbildes tā, it kā mērķa hosts būtu pieejams (veiksme). Īpašības darbība: - *atbalstīta*, ja `type` ir iestatīts uz ”ICMP ping”

dcheck.get

Apraksts

`integer/array dcheck.get(object parameters)`

Metode ļauj izgūt atklāšanas pārbaudes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dcheckids	ID/array	Atgriezt tikai atklāšanas pārbaudes ar norādītajiem ID.
druleids	ID/array	Atgriezt tikai tās atklāšanas pārbaudes, kas pieder norādītajiem atklāšanas noteikumiem.
dserviceids	ID/array	Atgriezt tikai tās atklāšanas pārbaudes, kas ir noteikušas norādītos atklātos servisus.
selectDRules	query	Atgriezt ar atklāšanas pārbaudēm saistītos atklāšanas noteikumus.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `dcheckid`, `druleid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Atgūt atklāšanas pārbaudes atklāšanas kārtulai

Atgūt visas atklāšanas pārbaudes, ko izmanto atklāšanas kārtula "6".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dcheck.get",
    "params": {
        "output": "extend",
        "dcheckids": "6"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Avots

CDCheck::get() failā *ui/include/classes/api/services/CDCheck.php*.

Attēls

Šī klase ir paredzēta darbam ar attēliem.

Objektu atsauces:

- Image

Pieejamās metodes:

- image.create - izveidot jaunus attēlus
- image.delete - dzēst attēlus
- image.get - izgūt attēlus
- image.update - atjaunināt attēlus

Attēla objekts

Tālāk norādītie objekti ir tieši saistīti ar `image` API.

Attēls

Attēla objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
imageid	ID	Attēla ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Attēla nosaukums. Īpašības darbība: - *obligāts* izveides operācijām

Īpašība	Tips	Apraksts
imagetype	integer	Attēla tips. Iespējamās vērtības: 1 - *(noklusējums)* ikona; 2 - fona attēls. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām
image	string	Base64 kodēts attēls. Kodētā attēla maksimālais izmērs ir 1 MB. Maksimālo izmēru var pielāgot, mainot `ZBX_MAX_IMAGE_SIZE` konstantes vērtību. Atbalstītie attēlu formāti: PNG, JPEG, GIF un WebP. Īpašības darbība: - *obligāts* izveides operācijām

image.create

Apraksts

`object image.create(object/array images)`

Šī metode ļauj izveidot jaunus attēlus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Attēli, ko izveidot.

Metode pieņem attēlus ar standarta attēlu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto attēlu ID īpašībā `imageids`. Atgriezto ID secība atbilst nodoto attēlu secībai.

Piemēri

Izveidojiet ikonas attēlu

Izveidojiet mākoņa ikonu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 1,
        "name": "Cloud_(24)",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAAApgB
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188"
        ]
```

```
    },
    "id": 1
}
```

Izveidojiet fona attēlu

Izveidojiet nelielu fona attēlu, ko izmantot kā rakstu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "image.create",
    "params": {
        "imagetype": 2,
        "name": "background_image",
        "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAIAAABvFaqvAAABhWlDQ1BJQ0MgcHJvZmlsZQAAKJF9kb1Lw0AYxp+m
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "189"
        ]
    },
    "id": 1
}
```

Avots

CImage::create() failā *ui/include/classes/api/services/CImage.php*.

image.delete

Apraksts

`object image.delete(array imageIds)`

Šī metode ļauj dzēst attēlus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet User roles.

Parametri

`(array)` dzēšamo attēlu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto attēlu ID zem `imageids` īpašības.

Piemēri

Dzēst vairākus attēlus

Dzēsiet divus attēlus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "image.delete",
    "params": [
        "188",
```

```
        "192"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Avots

CImage::delete() failā *ui/include/classes/api/services/CImage.php*.

image.get

Apraksts

`integer/array image.get(object parameters)`

Šī metode ļauj izgūt attēlus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
imageids	ID/array	Atgriezt tikai attēlus ar norādītajiem ID.
sysmapids	ID/array	Atgriezt attēlus, kas tiek izmantoti norādītajās kartēs.
select_image	flag	Atgriezt `image` īpašību ar Base64 kodētu attēlu.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `imageid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;

- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Attēla izgūšana

Izgūst visus datus attēlam ar ID "2".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "image.get",
    "params": {
        "output": "extend",
        "select_image": true,
        "imageids": "2"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "imageid": "2",
            "imagetype": "1",
            "name": "Cloud_(24)",
            "image": "iVBORw0KGgoAAAANSUhEUgAAABgAAAANCAYAAACzbK7QAAAABHNCSVQICAgIfAhkiAAAAAlwSFlzAAACmAAA
        }
    ],
    "id": 1
}
```

Avots

CImage::get() failā *ui/include/classes/api/services/CImage.php*.

image.update

Apraksts

`object image.update(object/array images)`

Šī metode ļauj atjaunināt esošos attēlus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Atjaunināmo attēlu rekvizīti.

Katram attēlam jābūt definētam rekvizītam `imageid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Metode pieņem attēlus ar standarta attēlu rekvizītiem.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto attēlu ID īpašībā `imageids`.

Piemēri

Pārdēvēt attēlu

Pārdēvējiet attēlu par "Cloud icon".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "image.update",
    "params": {
        "imageid": "2",
        "name": "Cloud icon"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "imageids": [
            "2"
        ]
    },
    "id": 1
}
```

Avots

CImage::update() failā *ui/include/classes/api/services/CImage.php*.

Audita žurnāls

Šī klase ir paredzēta darbam ar audita žurnālu.

Objektu atsauces:

- Audita žurnāls

Pieejamās metodes:

- auditlog.get - izgūt audita žurnāla ierakstus

Audita žurnāla objekts

Tālāk norādītie objekti ir tieši saistīti ar `auditlog` API.

Audita žurnāls

Audita žurnāla objekts satur informāciju par lietotāju darbībām. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
auditid	ID	Audita žurnāla ieraksta ID. Ģenerēts, izmantojot CUID algoritmu.
userid	ID	Audita žurnāla ieraksta autora userid.
username	string	Audita žurnāla ieraksta autora lietotājvārds.
clock	timestamp	Audita žurnāla ieraksta izveides laikspiedols.
ip	string	Audita žurnāla ieraksta autora IP adrese.
action	integer	Audita žurnāla ieraksta darbība. Iespējamās vērtības: 0 - Pievienot; 1 - Atjaunināt; 2 - Dzēst; 4 - Izrakstīties; 7 - Izpildīt; 8 - Pieteikties; 9 - Neveiksmīga pieteikšanās; 10 - Vēstures tīrīšana; 11 - Konfigurācijas atsvaidzināšana; 12 - Nosūtīšana.

Īpašība	Tips	Apraksts
resourcetype	integer	Audita žurnāla ieraksta resursa tips. Iespējamās vērtības: 0 - Lietotājs; 3 - Medija tips; 4 - Hosts; 5 - Darbība; 6 - Grafiks; 11 - Lietotāju grupa; 13 - Trigeris; 14 - Hostu grupa; 15 - Vienums; 16 - Attēls; 17 - Vērtību karte; 18 - Pakalpojums; 19 - Karte; 22 - Tīmekļa scenārijs; 23 - Atklāšanas kārtula; 25 - Skripts; 26 - Starpniekserveris; 27 - Uzturēšana; 28 - Regulārā izteiksme; 29 - Makross; 30 - Veidne; 31 - Trigera prototips; 32 - Ikonu kartēšana; 33 - Panelis; 34 - Notikumu korelācija; 35 - Grafika prototips; 36 - Vienuma prototips; 37 - Hosta prototips; 38 - Automātiskā reģistrācija; 39 - Modulis; 40 - Iestatījumi; 41 - Housekeeping; 42 - Autentifikācija; 43 - Veidnes panelis; 44 - Lietotāja loma; 45 - API marķieris; 46 - Plānotā atskaite; 47 - Augstas pieejamības mezgls; 48 - SLA; 49 - Lietotāju direktorijs; 50 - Veidņu grupa; 51 - Savienotājs; 52 - LLD kārtula; 53 - Vēsture; 54 - Daudzfaktoru autentifikācija; 55 - Starpniekserveru grupa; 56 - LLD kārtulas prototips.
resourceid	ID	Audita žurnāla ieraksta resursa identifikators.
resource_cuid	ID	Audita žurnāla ieraksta resursa unikālais identifikators, kas ģenerēts, izmantojot CUID algoritmu.
resourcename	string	Audita žurnāla ieraksta resursa cilvēklasāms nosaukums.
recordsetid	ID	Audita žurnāla ieraksta ierakstu kopas ID. Audita žurnāla ierakstiem, kas izveidoti vienas un tās pašas operācijas laikā, būs vienāds ierakstu kopas ID. Ģenerēts, izmantojot CUID algoritmu.

Īpašība	Tips	Apraksts
details	text	Audita žurnāla ieraksta detaļas. Detaļas tiek glabātas kā JSON objekts, kur katrs īpašības nosaukums ir ceļš uz īpašību vai ligzdotu objektu, kurā notika izmaiņas, un katra vērtība satur datus (masīva formātā) par izmaiņām šajā īpašībā vai ligzdotajā objektā. Iespējamie vērtību formāti: ["add"] - Ligzdotais objekts ir pievienots; ["add", "<value>"] - Pievienotā objekta īpašība ir vienāda ar <value>; ["update"] - Ligzdotais objekts ir atjaunināts; ["update", "<new value>", "<old value>"] - Atjaunināta objekta īpašība tika mainīta no <old value> uz <new value>; ["delete"] - Ligzdotais objekts ir dzēsts.

auditlog.get

Apraksts

`integer/array auditlog.get(object parameters)`

Šī metode ļauj izgūt audita žurnāla ierakstus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
auditids	ID/array	Atgriezt tikai audita žurnālu ar norādītajiem ID.
userids	ID/array	Atgriezt tikai audita žurnālu, ko izveidojuši norādītie lietotāji.
time_from	timestamp	Atgriež tikai tos audita žurnāla ierakstus, kas izveidoti pēc norādītā laika vai tieši norādītajā laikā.
time_till	timestamp	Atgriež tikai tos audita žurnāla ierakstus, kas izveidoti pirms norādītā laika vai tieši norādītajā laikā.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `auditid`, `userid`, `clock`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Audita žurnāla izgūšana

Izgūstiet divus jaunākos audita žurnāla ierakstus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "auditlog.get",
    "params": {
        "output": "extend",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 2
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Audit log object

Avots

CAuditLog::get() failā *ui/include/classes/api/services/CAuditLog.php*.

Augstas pieejamības mezgls

Šī klase ir paredzēta darbam ar servera mezgliem, kas ir daļa no augstas pieejamības klastera vai atsevišķas servera instances.

Objektu atsauces:

- Augstas pieejamības mezgls

Pieejamās metodes:

- hanode.get - izgūt mezglus

Augstas pieejamības mezgla objekts

Tālāk aprakstītais objekts ir saistīts ar Zabbix serveris augstas pieejamības klastera darbību.

Augstas pieejamības mezgls

> **Note:**
> Mezglus izveido Zabbix serveris, un tos nevar mainīt, izmantojot API.

Augstas pieejamības mezgla objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
ha_nodeid	ID	Mezgla ID.
name	string	Mezglam piešķirtais nosaukums, izmantojot HANodeName konfigurācijas ierakstu failā zabbix_server.conf. Tukšs serverim, kas darbojas savrupā režīmā.
address	string	IP adrese vai DNS nosaukums, no kura mezgls pieslēdzas.
port	integer	Ports, kurā mezgls darbojas.
lastaccess	integer	Sirdspuksta laiks, tas ir, pēdējās atjaunināšanas laiks no mezgla. UTC laikspiedols.
status	integer	Mezgla stāvoklis. Iespējamās vērtības: 0 - gaidstāve; 1 - apturēts manuāli; 2 - nav pieejams; 3 - aktīvs.

hanode.get

Apraksts

`integer/array hanode.get(object parameters)`

Metode ļauj izgūt Augstas pieejamības klastera mezglu sarakstu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
ha_nodeids	ID/array	Atgriezt tikai mezglus ar norādītajiem mezglu ID.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst dotajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Atbalstītās īpašības: `name`, `address`, `status`.

Parametrs	Tips	Apraksts
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `name`, `lastaccess`, `status`.
countOutput	flag	Šie parametri ir aprakstīti atsauces komentārā.
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt pēc statusa sakārtotu mezglu sarakstu

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "preservekeys": true,
        "sortfield": "status",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Atbilde:

```
{
  "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
}
```

Iegūt konkrētu mezglu sarakstu pēc to ID

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "ha_nodeids": ["ckuo7i1nw000e0sajwfttc1mp", "ckuo7i1nv000c0sajz85xcrtt"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Iegūt apturēto mezglu sarakstu

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["ha_nodeid", "address", "port"],
        "filter": {
            "status": 1
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Iegūt gaidstāves mezglu skaitu

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "countOutput": true,
        "filter": {
            "status": 0
        }
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": "3",
    "id": 1
}
```


Pārbaudīt mezglu statusu noteiktās IP adresēs

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "hanode.get",
    "params": {
        "output": ["name", "status"],
        "filter": {
            "address": ["192.168.1.7", "192.168.1.13"]
        }
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
```

```
        {
            "name": "node3",
            "status": "0"
        },
        {
            "name": "node-active",
            "status": "3"
        }
    ],
    "id": 1
}
```

Avots

CHaNode::get() failā *ui/include/classes/api/services/CHaNode.php*.

Autentifikācija

Šī klase ir paredzēta darbam ar autentifikācijas iestatījumiem.

Objektu atsauces:

- Authentication

Pieejamās metodes:

- authentication.get - izgūt autentifikācijas iestatījumus
- authentication.update - atjaunināt autentifikācijas iestatījumus

Autentifikācijas objekts

Tālāk norādītie objekti ir tieši saistīti ar `authentication` API.

Autentifikācija

Autentifikācijas objektam ir šādas īpašības.

Property	Type	Description
authentication_type	integer	Noklusējuma autentifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Iekšējā; 1 - LDAP.
http_auth_enabled	integer	HTTP autentifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Atspējota; 1 - Iespējota. Īpašības darbība: - *atbalstīta*, ja lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*) ir iespējots `$ALLOW_HTTP_AUTH`.
http_login_form	integer	Noklusējuma pieteikšanās forma. Iespējamās vērtības: 0 - *(noklusējums)* Zabbix pieteikšanās forma; 1 - HTTP pieteikšanās forma. Īpašības darbība: - *atbalstīta*, ja lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*) ir iespējots `$ALLOW_HTTP_AUTH`.

Property	Type	Description
http_strip_domains	string	Noņemamais domēna nosaukums. Īpašības darbība: - *atbalstīta*, ja lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*) ir iespējots `$ALLOW_HTTP_AUTH`.
http_case_sensitive	integer	Reģistrjutīga HTTP pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(noklusējums)* Ieslēgta. Īpašības darbība: - *atbalstīta*, ja lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*) ir iespējots `$ALLOW_HTTP_AUTH`.
ldap_auth_enabled	integer	LDAP autentifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Atspējota; 1 - Iespējota.
ldap_case_sensitive	integer	Reģistrjutīga LDAP pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(noklusējums)* Ieslēgta.
ldap_userdirectoryid	ID	LDAP autentifikācijas noklusējuma lietotāju direktorija ID. Izmanto lietotāju grupām, kurām `gui_access` ir iestatīts uz LDAP vai Sistēmas noklusējums. Īpašības darbība: - *obligāta*, ja `ldap_auth_enabled` ir iestatīts uz ”Iespējota”
saml_auth_enabled	integer	SAML autentifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Atspējota; 1 - Iespējota.
saml_case_sensitive	integer	Reģistrjutīga SAML pieteikšanās. Iespējamās vērtības: 0 - Izslēgta; 1 - *(noklusējums)* Ieslēgta.
passwd_min_length	integer	Minimālā paroles garuma prasība. Iespējamās vērtības ir diapazonā no 1 līdz 70. Noklusējums: 8.
passwd_check_rules	integer	Paroles pārbaudes noteikumi. Iespējamās bitkartes vērtības: 0 - Pārbaudīt paroles garumu; 1 - Pārbaudīt, vai parole satur lielos un mazos latīņu burtus; 2 - Pārbaudīt, vai parole satur ciparus; 4 - Pārbaudīt, vai parole satur speciālās rakstzīmes; 8 - *(noklusējums)* Pārbaudīt, vai parole nav bieži lietotu paroļu sarakstā un nesatur vārda ”Zabbix” atvasinājumus vai lietotāja vārdu, uzvārdu vai lietotājvārdu. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 15 visu noteikumu pārbaudei).

Property	Type	Description
ldap_jit_status	integer	LDAP nodrošināšanas statuss. Iespējamās vērtības: 0 - Atspējota konfigurētajiem LDAP IdP; 1 - Iespējota konfigurētajiem LDAP IdP.
saml_jit_status	integer	SAML nodrošināšanas statuss. Iespējamās vērtības: 0 - Atspējota konfigurētajiem SAML IdP; 1 - Iespējota konfigurētajiem SAML IdP.
jit_provision_interval	string	Laika intervāls starp JIT nodrošināšanas pieprasījumiem pieteikušamies lietotājam. Pieņem sekundes un laika vienību ar sufiksu, atbalstot mēnešus un gadus (3600s,60m,1h,1d,1M,1y). Minimālā vērtība: 1h. Noklusējums: 1h. Pieejams tikai LDAP nodrošināšanai.
disabled_usrgrpid	ID	Tās lietotāju grupas ID, kurai piešķirt atnodrošināto lietotāju. Lietotāju grupai jābūt atspējotai, un to nevar iespējot vai dzēst, kamēr tā ir konfigurēta. Īpašības darbība: - *obligāta*, ja `ldap_jit_status` ir iestatīts uz "Iespējota konfigurētajiem LDAP IdP" vai `saml_jit_status` ir iestatīts uz "Iespējota konfigurētajiem SAML IdP"
mfa_status	integer	Daudzfaktoru autentifikācija. Iespējamās vērtības: 0 - Atspējota (visām konfigurētajām MFA metodēm); 1 - Iespējota (visām konfigurētajām MFA metodēm).
mfaid	ID	Noklusējuma MFA metode lietotāju grupām, kurām MFA ir iespējota. Īpašības darbība: - *obligāta*, ja `mfa_status` ir iestatīts uz "Iespējota"

authentication.get

Apraksts

`object authentication.get(object parameters)`

Metode ļauj izgūt autentifikācijas objektu atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta tikai vienu parametru.

Parametrs	Tips	Apraksts
output	query	Šis parametrs ir aprakstīts atsauces komentārā.

Atgrieztās vērtības

`(object)` Atgriež autentifikācijas objektu.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CAuthentication::get() failā *ui/include/classes/api/services/CAuthentication.php*.

authentication.update

Apraksts

`object authentication.update(object authentication)`

Šī metode ļauj atjaunināt esošos autentifikācijas iestatījumus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Autentifikācijas rekvizīti, kas jāatjaunina.

Atgrieztās vērtības

`(array)` Atgriež masīvu ar atjaunināto parametru nosaukumiem.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "authentication.update",
    "params": {
        "http_auth_enabled": 1,
        "http_case_sensitive": 0,
        "http_login_form": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        "http_auth_enabled",
        "http_case_sensitive",
        "http_login_form"
    ],
    "id": 1
}
```

Avots

CAuthentication::update() failā *ui/include/classes/api/services/CAuthentication.php*.

Autoreģistrācija

Šī klase ir paredzēta darbam ar autoreģistrāciju.

Objektu atsauces:

- Autoreģistrācija

Pieejamās metodes:

- autoregistration.get - izgūt autoreģistrāciju
- autoregistration.update - atjaunināt autoreģistrāciju

Autoreģistrācijas objekts

Tālāk norādītie objekti ir tieši saistīti ar `autoregistration` API.

Autoreģistrācija

Autoreģistrācijas objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tls_accept	integer	Atļauto ienākošo savienojumu veids autoreģistrācijai. Iespējamās vērtības: 1 - atļaut nešifrētus savienojumus; 2 - atļaut TLS ar PSK; 3 - atļaut gan nešifrētus savienojumus, gan TLS ar PSK savienojumus.
tls_psk_identity	string	PSK identitāte; tai jābūt sasaistītai tikai ar vienu PSK (starp autoregistration, hosts un proxies). Neiekļaujiet sensitīvu informāciju PSK identitātē, jo tā tiek nosūtīta nešifrētā veidā pa tīklu, lai informētu saņēmēju, kuru PSK izmantot. Īpašības darbība: - *tikai rakstāma*

Īpašība	Type	Apraksts
tls_psk	string	Iepriekš koplietotā atslēga (PSK); tai jābūt vismaz 32 heksadecimālām zīmēm garai. Īpašības darbība: *- tikai rakstāma*

autoregistration.get

Apraksts

`object autoregistration.get(object parameters)`

Šī metode ļauj izgūt automātiskās reģistrācijas objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta tikai vienu parametru.

Parametrs	Type	Apraksts
output	query	Šis parametrs ir aprakstīts atsauces komentārā.

Atgrieztās vērtības

`(object)` Atgriež automātiskās reģistrācijas objektu.

Piemēri

Pieprasījums:

```
{
   "jsonrpc": "2.0",
    "method": "autoregistration.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tls_accept": "3"
    },
    "id": 1
}
```

Avots

CAutoregistration::get() failā *ui/include/classes/api/services/CAutoregistration.php*.

autoregistration.update

Apraksts

`object autoregistration.update(object autoregistration)`

Šī metode ļauj atjaunināt esošu automātisko reģistrāciju.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object)` Autoreģistrācijas īpašības, kas jāatjaunina.

Atgrieztās vērtības

`(boolean )` Veiksmīgas atjaunināšanas gadījumā atgriež boolean true kā rezultātu.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "autoregistration.update",
    "params": {
        "tls_accept": "3",
        "tls_psk_identity": "PSK 001",
        "tls_psk": "11111595725ac58dd977beef14b97461a7c1045b9a1c923453302c5473193478"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Avots

CAutoregistration::update() failā *ui/include/classes/api/services/CAutoregistration.php*.

Brīdinājums

Šī klase ir paredzēta darbam ar brīdinājumiem.

Objekta atsauces:

- Alert

Pieejamās metodes:

- alert.get - izgūt brīdinājumus

Brīdinājuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `alert` API.

Brīdinājums

> **Note:**
> Brīdinājumus izveido Zabbix serveris, un tos nevar modificēt, izmantojot API.

Brīdinājuma objekts satur informāciju par to, vai noteiktas darbības operācijas ir izpildītas veiksmīgi. Tam ir šādas īpašības.

Property	Type	Description
alertid	ID	Brīdinājuma ID.
actionid	ID	Darbības ID, kas ģenerēja brīdinājumu.

Property	Type	Description
alerttype	integer	Brīdinājuma tips. Iespējamās vērtības: 0 - ziņojums; 1 - attālināta komanda.
clock	timestamp	Laiks, kad brīdinājums tika ģenerēts.
error	string	Kļūdas teksts, ja rodas problēmas ar ziņojuma nosūtīšanu vai komandas izpildi.
esc_step	integer	Darbības eskalācijas solis, kura laikā tika ģenerēts brīdinājums.
eventid	ID	Notikuma ID, kas izraisīja darbību.
mediatypeid	ID	Medija tipa ID, kas tika izmantots ziņojuma nosūtīšanai.
message	text	Ziņojuma teksts. Īpašības darbība: - *atbalstīts*, ja `alerttype` ir iestatīts uz ”message”
retries	integer	Reižu skaits, cik Zabbix mēģināja nosūtīt ziņojumu.
sendto	string	Saņēmēja adrese, lietotājvārds vai cits identifikators. Īpašības darbība: - *atbalstīts*, ja `alerttype` ir iestatīts uz ”message”
status	integer	Statuss, kas norāda, vai darbības operācija ir izpildīta veiksmīgi. Iespējamās vērtības, ja `alerttype` ir iestatīts uz ”message”: 0 - ziņojums nav nosūtīts; 1 - ziņojums nosūtīts; 2 - neizdevās pēc vairākiem atkārtotiem mēģinājumiem; 3 - jaunu brīdinājumu vēl nav apstrādājis brīdinājumu pārvaldnieks. Iespējamās vērtības, ja `alerttype` ir iestatīts uz ”remote command”: 0 - komanda nav izpildīta; 1 - komanda izpildīta; 2 - tika mēģināts izpildīt komandu uz Zabbix aģents, bet tas nebija pieejams.
subject	string	Ziņojuma temats. Īpašības darbība: - *atbalstīts*, ja `alerttype` ir iestatīts uz ”message”
userid	ID	Lietotāja ID, kuram tika nosūtīts ziņojums.
p_eventid	ID	Problēmas notikuma ID, kas ģenerēja brīdinājumu.
acknowledgeid	ID	Apstiprinājuma ID, kas ģenerēja brīdinājumu.

alert.get

Apraksts

`integer/array alert.get(object parameters)`

Šī metode ļauj izgūt brīdinājumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
alertids	ID/array	Atgriezt tikai brīdinājumus ar norādītajiem ID.

Parametrs	Tips	Apraksts
actionids	ID/array	Atgriezt tikai brīdinājumus, ko ģenerējušas norādītās darbības.
eventids	ID/array	Atgriezt tikai brīdinājumus, ko ģenerējuši norādītie notikumi.
groupids	ID/array	Atgriezt tikai brīdinājumus, ko ģenerējuši objekti no norādītajām hostu grupām.
hostids	ID/array	Atgriezt tikai brīdinājumus, ko ģenerējuši objekti no norādītajiem hostiem.
mediatypeids	ID/array	Atgriezt tikai ziņojumu brīdinājumus, kuros izmantoti norādītie mediju tipi.
objectids	ID/array	Atgriezt tikai brīdinājumus, ko ģenerējuši norādītie objekti
userids	ID/array	Atgriezt tikai ziņojumu brīdinājumus, kas tika nosūtīti norādītajiem lietotājiem.
eventobject	integer	Atgriezt tikai brīdinājumus, ko ģenerējuši notikumi, kas saistīti ar norādītā tipa objektiem. Atbalstīto objektu tipu sarakstu skatiet notikuma `object` aprakstā. Noklusējums: 0 - trigeris.
eventsource	integer	Atgriezt tikai brīdinājumus, ko ģenerējuši norādītā tipa notikumi. Atbalstīto notikumu tipu sarakstu skatiet notikuma `source` aprakstā. Noklusējums: 0 - trigeru notikumi.
time_from	timestamp	Atgriezt tikai brīdinājumus, kas ģenerēti pēc norādītā laika.
time_till	timestamp	Atgriezt tikai brīdinājumus, kas ģenerēti pirms norādītā laika.
selectHosts	query	Atgriezt `hosts` īpašību ar to hostu datiem, kas izraisīja darbības operāciju.
selectMediatypes	query	Atgriezt `mediatypes` īpašību ar mediju tipu masīvu, kas tika izmantoti ziņojuma brīdinājumam. Skatiet `mediatype.get` lietotāja tipa ierobežojumus.
selectUsers	query	Atgriezt `users` īpašību ar to lietotāju masīvu, kuriem ziņojums bija adresēts. Skatiet `user.get` lietotāja tipa ierobežojumus.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `alertid`, `clock`, `eventid`, `mediatypeid`, `sendto`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Brīdinājumu izgūšana pēc darbības ID

Izgūst visus brīdinājumus, ko ģenerējusi darbība "3".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "alert.get",
    "params": {
        "output": "extend",
        "actionids": "3"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "alertid": "1",
            "actionid": "3",
            "eventid": "21243",
            "userid": "1",
            "clock": "1362128008",
            "mediatypeid": "1",
            "sendto": "support@company.com",
            "subject": "PROBLĒMA: Zabbix aģents Linux serverī nav sasniedzams 5 minūtes: ",
            "message": "Trigeris: Zabbix aģents Linux serverī nav sasniedzams 5 minūtes: \nTrigera stāvokl
            "status": "0",
            "retries": "3",
            "error": "",
            "esc_step": "1",
            "alerttype": "0",
            "p_eventid": "0",
            "acknowledgeid": "0"
        }
    ],
    "id": 1
}
```

Skatiet arī

- Hosts
- Mediju tips
- Lietotājs

Avots

CAlert::get() failā *ui/include/classes/api/services/CAlert.php*.

Darbība

Šī klase ir paredzēta darbam ar darbībām.

Objektu atsauces:

- Darbība
- Darbības operācija
  - Darbības operācijas ziņojums
  - Darbības operācijas nosacījums
- Darbības atkopšanas operācija
- Darbības atjaunināšanas operācija
- Darbības filtrs
  - Darbības filtra nosacījums

Pieejamās metodes:

- action.create - izveidot jaunas darbības
- action.delete - dzēst darbības

- action.get - izgūt darbības
- action.update - atjaunināt darbības

Darbības objekts

Tālāk norādītie objekti ir tieši saistīti ar `action` API.

Darbība

Darbības objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
actionid	ID	Darbības ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
esc_period	string	Noklusējuma operācijas soļa ilgums. Tam jābūt vismaz 60 sekundēm. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Īpašības darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, ”iekšējs notikums” vai ”notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā”
eventsource	integer	Notikumu tips, ko apstrādās darbība. Atbalstīto notikumu tipu sarakstu skatiet notikuma `source` īpašībā. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām
name	string	Darbības nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
status	integer	Vai darbība ir iespējota vai atspējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.
pause_symptoms	integer	Vai apturēt eskalāciju, ja notikums ir simptoma notikums. Iespējamās vērtības: 0 - Neapturēt eskalāciju simptomu problēmām; 1 - *(noklusējums)* Apturēt eskalāciju simptomu problēmām. Īpašības darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”
pause_suppressed	integer	Vai apturēt eskalāciju apkopes periodu laikā vai ne. Iespējamās vērtības: 0 - Neapturēt eskalāciju; 1 - *(noklusējums)* Apturēt eskalāciju. Īpašības darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”

Īpašība	Tips	Apraksts
notify_if_canceled	integer	Vai sūtīt paziņojumu, kad eskalācija tiek atcelta. Iespējamās vērtības: 0 - Nesūtīt paziņojumu, kad eskalācija tiek atcelta; 1 - *(noklusējums)* Sūtīt paziņojumu, kad eskalācija tiek atcelta. Īpašības darbība: - *atbalstīts*, ja `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”

Darbības operācija

Darbības operācijas objekts definē operāciju, kas tiks izpildīta, kad darbība tiek izpildīta. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
operationtype	integer	Operācijas tips. Iespējamās vērtības: 0 - sūtīt ziņojumu; 1 - globālais skripts; 2 - pievienot hostu; 3 - noņemt hostu; 4 - pievienot hostu grupai; 5 - noņemt no hostu grupas; 6 - saistīt veidni; 7 - atsaistīt veidni; 8 - iespējot hostu; 9 - atspējot hostu; 10 - iestatīt hosta inventarizācijas režīmu; 13 - pievienot hosta tagus; 14 - noņemt hosta tagus. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”event created by a trigger” vai ”event created on service status update”: 0 - ”send message”; 1 - ”global script”. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”internal event”: 0 - ”send message”. Īpašības darbība: - *required*
esc_period	string	Eskalācijas soļa ilgums sekundēs. Tam jābūt lielākam par 60 sekundēm. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Ja iestatīts uz 0 vai 0s, tiks izmantots noklusējuma darbības eskalācijas periods. Noklusējums: 0s. Īpašības darbība: - *supported* ja Darbības objekta `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”

Īpašība	Tips	Apraksts
esc_step_from	integer	Solis, no kura sākt eskalāciju. Noklusējums: 1. Īpašības darbība: - *supported* ja Darbības objekta `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”
esc_step_to	integer	Solis, kurā beigt eskalāciju. Noklusējums: 1. Īpašības darbība: - *supported* ja Darbības objekta `eventsource` ir iestatīts uz ”event created by a trigger”, ”internal event” vai ”event created on service status update”
evaltype	integer	Operācijas nosacījuma novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 1 - Un; 2 - Vai.
opcommand	object	Izpildāmais globālais skripts. Globālajam skriptam jābūt definētai `scriptid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”global script”
opcommand_grp	array	Hostu grupas, uz kurām palaist globālos skriptus. Hostu grupām jābūt definētai `groupid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”global script” un `opcommand_hst` nav iestatīts
opcommand_hst	array	Hosts, uz kura palaist globālos skriptus. Hostiem jābūt definētai `hostid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”global script” un `opcommand_grp` nav iestatīts
opconditions	array	Operācijas nosacījumi, kas tiek izmantoti trigeru darbībām. Operācijas nosacījuma objekts ir detalizēti aprakstīts zemāk.
opgroup	array	Hostu grupas, kurām pievienot hostus. Hostu grupām jābūt definētai `groupid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”add to host group” vai ”remove from host group”
opmessage	object	Objekts, kas satur datus par operācijas nosūtīto ziņojumu. Operācijas ziņojuma objekts ir detalizēti aprakstīts zemāk. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”send message”

Īpašība	Tips	Apraksts
opmessage_grp	array	Lietotāju grupas, kurām sūtīt ziņojumus. Lietotāju grupām jābūt definētai `usrgrpid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”send message” un `opmessage_usr` nav iestatīts
opmessage_usr	array	Lietotāji, kuriem sūtīt ziņojumus. Lietotājiem jābūt definētai `userid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”send message” un `opmessage_grp` nav iestatīts
optemplate	array	Veidnes, ko saistīt ar hostiem. Veidnēm jābūt definētai `templateid` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”link template” vai ”unlink template”
opinventory	object	Inventarizācijas režīms, uz kuru iestatīt hostu. Inventarizācijai jābūt definētai `inventory_mode` īpašībai. Īpašības darbība: - *required* ja `operationtype` ir iestatīts uz ”set host inventory mode”
optag	array	Hosta tagi, ko pievienot vai noņemt. Tagiem jābūt definētai `tag` īpašībai. `value` īpašība nav obligāta. Īpašības darbība: - *supported* ja `operationtype` ir iestatīts uz ”add host tags” vai ”remove host tags”.

Darbības operācijas ziņojums

Operācijas ziņojuma objekts satur datus par ziņojumu, kas tiks nosūtīts ar operāciju. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
default_msg	integer	Vai izmantot noklusējuma darbības ziņojuma tekstu un tematu. Iespējamās vērtības: 0 - izmantot datus no operācijas; 1 - *(noklusējums)* izmantot datus no medija tipa.
mediatypeid	ID	Ziņojuma nosūtīšanai izmantotā medija tipa ID. Īpašības darbība: - *atbalstīts*, ja Darbības operācijas objekta, Darbības atkopšanas operācijas objekta vai Darbības atjaunināšanas operācijas objekta `operationtype` ir iestatīts uz ”send message”, vai ja Darbības atjaunināšanas operācijas objekta `operationtype` ir iestatīts uz ”notify all involved”
message	string	Operācijas ziņojuma teksts. Īpašības darbība: - *atbalstīts*, ja `default_msg` ir iestatīts uz ”use the data from the operation”

Īpašība	Tips	Apraksts
subject	string	Operācijas ziņojuma temats. Īpašības darbība: - *atbalstīts*, ja `default_msg` ir iestatīts uz "use the data from the operation"

Darbības operācijas nosacījums

Darbības operācijas nosacījuma objekts definē nosacījumu, kam jābūt izpildītam, lai veiktu pašreizējo operāciju. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
conditiontype	integer	Nosacījuma tips. Iespējamās vērtības: 14 - notikums apstiprināts. Īpašības darbība: - *obligāts*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *obligāts*
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* =

Tālāk ir norādīti operatori un vērtības, kas tiek atbalstītas katram operācijas nosacījuma tipam.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Sagaidāmā vērtība
14	Notikums apstiprināts	=	Vai notikums ir apstiprināts. Iespējamās vērtības: 0 - nav apstiprināts; 1 - apstiprināts.

Darbības atkopšanas operācija

Darbības atkopšanas operācijas objekts definē operāciju, kas tiks izpildīta, kad problēma būs novērsta. Atkopšanas operācijas ir iespējamas **tikai** trigeru, iekšējo un pakalpojumu darbībām. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
operationtype	integer	Operācijas tips. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris” vai ”notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā”: 0 - sūtīt ziņojumu; 1 - globālais skripts; 11 - paziņot visiem iesaistītajiem. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”iekšējais notikums”: 0 - sūtīt ziņojumu; 11 - paziņot visiem iesaistītajiem. Īpašības darbība: - *obligāts*
opcommand	object	Izpildāmais globālais skripts. Globālajam skriptam jābūt definētai īpašībai `scriptid`. Īpašības darbība: - *obligāts*, ja `operationtype` ir iestatīts uz ”globālais skripts”
opcommand_grp	array	Hostu grupas, kurās palaist globālos skriptus. Hostu grupām jābūt definētai īpašībai `groupid`. Īpašības darbība: - *obligāts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, un `operationtype` ir iestatīts uz ”globālais skripts”, un `opcommand_hst` nav iestatīts
opcommand_hst	array	Hosts, kurā palaist globālos skriptus. Hostiem jābūt definētai īpašībai `hostid`. Īpašības darbība: - *obligāts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, un `operationtype` ir iestatīts uz ”globālais skripts”, un `opcommand_grp` nav iestatīts
opmessage	object	Objekts, kas satur datus par ziņojumu, ko nosūta atkopšanas operācija. Operācijas ziņojuma objekts ir detalizēti aprakstīts iepriekš. Īpašības darbība: - *obligāts*, ja `operationtype` ir iestatīts uz ”sūtīt ziņojumu”
opmessage_grp	array	Lietotāju grupas, kurām sūtīt ziņojumus. Lietotāju grupām jābūt definētai īpašībai `usrgrpid`. Īpašības darbība: - *obligāts*, ja `operationtype` ir iestatīts uz ”sūtīt ziņojumu” un `opmessage_usr` nav iestatīts
opmessage_usr	array	Lietotāji, kuriem sūtīt ziņojumus. Lietotājiem jābūt definētai īpašībai `userid`. Īpašības darbība: - *obligāts*, ja `operationtype` ir iestatīts uz ”sūtīt ziņojumu” un `opmessage_grp` nav iestatīts

Darbības atjaunināšanas operācija

Darbības atjaunināšanas operācijas objekts definē operāciju, kas tiks izpildīta, kad problēma tiek atjaunināta (tai pievienots komentārs, tā ir apstiprināta, mainīta tās nopietnība vai tā ir manuāli aizvērta). Atjaunināšanas operācijas ir iespējamas **tikai** trigeru un pakalpojumu darbībām. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
operationtype	integer	Operācijas tips. Iespējamās vērtības: 0 - sūtīt ziņojumu; 1 - globālais skripts; 12 - paziņot visiem iesaistītajiem. Īpašības darbība: - *obligāta*
opcommand	object	Izpildāmais globālais skripts. Globālajam skriptam jābūt definētai `scriptid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”globālais skripts”
opcommand_grp	array	Hostu grupas, kurās palaist globālos skriptus. Hostu grupām jābūt definētai `groupid` īpašībai. Īpašības darbība: - *obligāta*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, `operationtype` ir iestatīts uz ”globālais skripts” un `opcommand_hst` nav iestatīts
opcommand_hst	array	Hosts, kurā palaist globālos skriptus. Hostiem jābūt definētai `hostid` īpašībai. Īpašības darbība: - *obligāta*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, `operationtype` ir iestatīts uz ”globālais skripts” un `opcommand_grp` nav iestatīts
opmessage	object	Objekts, kas satur datus par ziņojumu, ko nosūta atjaunināšanas operācija. Operācijas ziņojuma objekts ir detalizēti aprakstīts iepriekš.
opmessage_grp	array	Lietotāju grupas, kurām sūtīt ziņojumus. Lietotāju grupām jābūt definētai `usrgrpid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”sūtīt ziņojumu” un `opmessage_usr` nav iestatīts
opmessage_usr	array	Lietotāji, kuriem sūtīt ziņojumus. Lietotājiem jābūt definētai `userid` īpašībai. Īpašības darbība: - *obligāta*, ja `operationtype` ir iestatīts uz ”sūtīt ziņojumu” un `opmessage_grp` nav iestatīts

Darbības filtrs

Darbības filtra objekts definē nosacījumu kopu, kam jābūt izpildītai, lai veiktu konfigurētās darbības operācijas. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
conditions	array	Filtra nosacījumu kopa, ko izmantot rezultātu filtrēšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - Un/Vai; 1 - Un; 2 - Vai; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti filtra nosacījumos: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”pielāgota izteiksme”

Darbības filtra nosacījums

Darbības filtra nosacījuma objekts definē konkrētu nosacījumu, kas jāpārbauda pirms darbības operāciju izpildes.

Īpašība	Type	Apraksts
conditiontype	integer	Nosacījuma tips. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”: 0 - hostu grupa; 1 - hosts; 2 - trigeris; 3 - notikuma nosaukums; 4 - trigera nopietnība; 6 - laika periods; 13 - hosta veidne; 16 - problēma ir apspiesta; 25 - notikuma tags; 26 - notikuma taga vērtība. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis atklāšanas noteikums”: 7 - hosta IP; 8 - atklātā servisa tips; 9 - atklātā servisa ports; 10 - atklāšanas statuss; 11 - darbības vai dīkstāves ilgums; 12 - saņemtā vērtība; 18 - atklāšanas noteikums; 19 - atklāšanas pārbaude; 20 - starpniekserveris; 21 - atklāšanas objekts. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojusi aktīva aģenta automātiskā reģistrācija”: 20 - starpniekserveris; 22 - hosta nosaukums; 24 - hosta metadati. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”iekšējais notikums”: 0 - hostu grupa; 1 - hosts; 13 - hosta veidne; 23 - notikuma tips; 25 - notikuma tags; 26 - notikuma taga vērtība. Iespējamās vērtības, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, kas izveidots servisa statusa atjauninašanas laikā”: 25 - notikuma tags; 26 - notikuma taga vērtība; 27 - serviss; 28 - servisa nosaukums. Īpašības darbība: - *obligāts*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *obligāts*

Īpašība	Type	Apraksts
value2	string	Sekundārā vērtība, ar ko salīdzināt. Īpašības darbība: - *obligāts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, `conditiontype` ir iestatīts uz jebkuru iespējamo vērtību trigeru darbībām un nosacījuma tips (skatīt zemāk) ir ”26” - *obligāts*, ja Darbības objekta `eventsource` ir iestatīts uz ”iekšējais notikums”, `conditiontype` ir iestatīts uz jebkuru iespējamo vērtību iekšējām darbībām un nosacījuma tips (skatīt zemāk) ir ”26” - *obligāts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, kas izveidots servisa statusa atjaunināšanas laikā”, `conditiontype` ir iestatīts uz jebkuru iespējamo vērtību servisu darbībām un nosacījuma tips (skatīt zemāk) ir ”26”
formulaid	string	Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Var saturēt tikai lielos burtus. Lietotājam šis ID jādefinē, modificējot filtra nosacījumus, taču pēc tam, kad tie tiek pieprasīti atkārtoti, tas tiks ģenerēts no jauna.
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* vienāds ar; 1 - nav vienāds ar; 2 - satur; 3 - nesatur; 4 - ir iekš; 5 - ir lielāks vai vienāds ar; 6 - ir mazāks vai vienāds ar; 7 - nav iekš; 8 - atbilst; 9 - neatbilst; 10 - Jā; 11 - Nē.

Note:
Lai labāk saprastu, kā izmantot filtrus ar dažādiem izteiksmju tipiem, skatiet piemērus metožu action.get un action.create lapās.

Tālāk norādīti operatori un vērtības, kas tiek atbalstīti katram nosacījuma tipam.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Sagaidāmā vērtība
0	Hostu grupa	vienāds ar, nav vienāds ar	Hostu grupas ID.
1	Hosts	vienāds ar, nav vienāds ar	Hosta ID.
2	Trigeris	vienāds ar, nav vienāds ar	Trigera ID.
3	Notikuma nosaukums	satur, nesatur	Notikuma nosaukums.
4	Trigera nopietnība	vienāds ar, nav vienāds ar, ir lielāks vai vienāds ar, ir mazāks vai vienāds ar	Trigera nopietnība. Atbalstīto trigera nopietnības vērtību sarakstu skatiet trigera `severity` īpašībā.
5	Trigera vērtība	vienāds ar	Trigera vērtība. Atbalstīto trigera vērtību sarakstu skatiet trigera `value` īpašībā.
6	Laika periods	ir iekš, nav iekš	Laiks, kad notikums tika aktivizēts, kā laika periods.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Sagaidāmā vērtība
7	Hosta IP	vienāds ar, nav vienāds ar	Viens vai vairāki pārbaudāmi IP diapazoni, atdalīti ar komatiem. Plašāku informāciju par atbalstītajiem IP diapazonu formātiem skatiet sadaļā tīkla atklāšanas konfigurācija.
8	Atklātā servisa tips	vienāds ar, nav vienāds ar	Atklātā servisa tips. Servisa tips atbilst atklāšanas pārbaudes tipam, kas izmantots servisa noteikšanai. Atbalstīto tipu sarakstu skatiet atklāšanas pārbaudes type īpašībā.
9	Atklātā servisa ports	vienāds ar, nav vienāds ar	Viens vai vairāki portu diapazoni, atdalīti ar komatiem.
10	Atklāšanas statuss	vienāds ar	Atklātā objekta statuss. Iespējamās vērtības: 0 - hosts vai serviss darbojas; 1 - hosts vai serviss nedarbojas; 2 - hosts vai serviss atklāts; 3 - hosts vai serviss zaudēts.
11	Darbības vai dīkstāves ilgums	ir lielāks vai vienāds ar, ir mazāks vai vienāds ar	Laiks sekundēs, kas norāda, cik ilgi atklātais objekts ir bijis pašreizējā statusā.
12	Saņemtās vērtības	vienāds ar, nav vienāds ar, ir lielāks vai vienāds ar, ir mazāks vai vienāds ar, satur, nesatur	Vērtība, kas atgriezta, izpildot Zabbix aģenta, SNMPv1, SNMPv2 vai SNMPv3 atklāšanas pārbaudi.
13	Hosta veidne	vienāds ar, nav vienāds ar	Piesaistītās veidnes ID.
16	Problēma ir apspiesta	Jā, Nē	Vērtība nav nepieciešama: operatora ”Jā” izmantošana nozīmē, ka problēmai jābūt apspiestai, ”Nē” - neapspiestai.
18	Atklāšanas noteikums	vienāds ar, nav vienāds ar	Atklāšanas noteikuma ID.
19	Atklāšanas pārbaude	vienāds ar, nav vienāds ar	Atklāšanas pārbaudes ID.
20	Starpniekserveris	vienāds ar, nav vienāds ar	Starpniekservera ID.
21	Atklāšanas objekts	vienāds ar	Objekta tips, kas izraisīja atklāšanas notikumu. Iespējamās vērtības: 1 - atklāts hosts; 2 - atklāts serviss.
22	Hosta nosaukums	satur, nesatur, atbilst, neatbilst	Hosta nosaukums. Operatoriem *atbilst* un *neatbilst* automātiskās reģistrācijas nosacījumos tiek atbalstīta regulārās izteiksmes izmantošana.
23	Notikuma tips	vienāds ar	Konkrēts iekšējais notikums. Iespējamās vērtības: 0 - vienums stāvoklī ”nav atbalstīts”; 1 - vienums stāvoklī ”normāls”; 2 - LLD noteikums stāvoklī ”nav atbalstīts”; 3 - LLD noteikums stāvoklī ”normāls”; 4 - trigeris stāvoklī ”nezināms”; 5 - trigeris stāvoklī ”normāls”.
24	Hosta metadati	satur, nesatur, atbilst, neatbilst	Automātiski reģistrētā hosta metadati. Operatoriem *atbilst* un *neatbilst* tiek atbalstīta regulārās izteiksmes izmantošana.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Sagaidāmā vērtība
25	Tags	vienāds ar, nav vienāds ar, satur, nesatur	Notikuma tags.
26	Taga vērtība	vienāds ar, nav vienāds ar, satur, nesatur	Notikuma taga vērtība.
27	Serviss	vienāds ar, nav vienāds ar	Servisa ID.
28	Servisa nosaukums	vienāds ar, nav vienāds ar	Servisa nosaukums.

action.create

Apraksts

`object action.create(object/array actions)`

Šī metode ļauj izveidot jaunas darbības.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Darbības, ko izveidot.

Papildus standarta darbības īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
filter	object	Darbības filtra objekts darbībai.
operations	array	Darbības operācijas, ko izveidot darbībai.
recovery_operations	array	Darbības atkopšanas operācijas, ko izveidot darbībai.
update_operations	array	Darbības atjaunināšanas operācijas, ko izveidot darbībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto darbību ID īpašībā `actionids`. Atgriezto ID secība atbilst nodoto darbību secībai.

Piemēri

Izveidot trigera darbību

Izveidojiet trigera darbību, kas sāksies, tiklīdz trigeris (ar vārdu "memory" tā nosaukumā) no hosta "10084" pāries PROBLEM stāvoklī. Darbībai būs konfigurētas 4 operācijas. Pirmā un tūlītējā operācija nosūtīs ziņojumu visiem lietotājiem lietotāju grupā "7", izmantojot medija tipu "1". Ja notikums netiek atrisināts 30 minūšu laikā, tiks izpildīta otrā operācija — skripts "5" (skripts ar tvērumu "Action operation") — visos hostos grupā "2". Ja notikums tiek atrisināts, atkopšanas operācija paziņos visiem lietotājiem, kuri saņēma jebkādus ziņojumus par problēmu. Ja notikums tiek atjaunināts, apstiprināšanas/atjaunināšanas operācija paziņos (ar pielāgotu tēmu un ziņojumu) visiem lietotājiem, kuri saņēma jebkādus ziņojumus par problēmu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17"
        ]
    },
    "id": 1
}
```

Izveidot atklāšanas darbību

Izveidojiet atklāšanas darbību, kas piesaistīs veidni ”10001” atklātajiem hostiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Pielāgotas izteiksmes filtra izmantošana

Izveidojiet trigera darbību, kas darbības nosacījumu novērtēšanai izmanto pielāgotu izteiksmi — ”A and (B or C)”. Kad trigeris ar svarīgumu, kas ir augstāks vai vienāds ar ”Warning”, no hosta ”10084” vai hosta ”10106” pāriet PROBLEM stāvoklī, darbība nosūtīs ziņojumu visiem lietotājiem lietotāju grupā ”7”, izmantojot medija tipu ”1”. Formulu ID ”A”, ”B” un ”C” ir izvēlēti patvaļīgi.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "18"
        ]
    },
    "id": 1
}
```

Izveidot aģenta automātiskās reģistrācijas kārtulu

Izveidojiet automātiskās reģistrācijas darbību, kas pievieno hostu hostu grupai ”2”, ja hosta nosaukums satur ”SRV” vai metadati satur ”AlmaLinux”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            19
```

```
        ]
    },
    "id": 1
}
```

Izveidojiet aģenta automātiskās reģistrācijas kārtulu ar hostu tagiem

Izveidojiet automātiskās reģistrācijas darbību, kas pievieno hostu hostu grupai "2" un pievieno divus hostu tagus.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            20
        ]
    },
    "id": 1
}
```

Skatiet arī

- Darbības filtrs
- Darbības operācija
- Skripts

Avots

CAction::create() failā *ui/include/classes/api/services/CAction.php*.

action.delete

Apraksts

`object action.delete(array actionIds)`

Šī metode ļauj dzēst darbības.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo darbību ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto darbību ID īpašībā `actionids`.

Piemēri

Dzēst vairākas darbības

Dzēst divas darbības.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "action.delete",
    "params": [
        "17",
        "18"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "17",
            "18"
        ]
    },
    "id": 1
}
```

Avots

CAction::delete() failā *ui/include/classes/api/services/CAction.php*.

action.get

Apraksts

`integer/array action.get(object parameters)`

Metode ļauj izgūt darbības atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
actionids	ID/array	Atgriezt tikai darbības ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai darbības, kas darbību nosacījumos izmanto norādītās hostu grupas.
hostids	ID/array	Atgriezt tikai darbības, kas darbību nosacījumos izmanto norādītos hostus.
triggerids	ID/array	Atgriezt tikai darbības, kas darbību nosacījumos izmanto norādītos trigerus.
mediatypeids	ID/array	Atgriezt tikai darbības, kas ziņojumu sūtīšanai izmanto norādītos mediju tipus.
usrgrpids	ID/array	Atgriezt tikai darbības, kas ir konfigurētas ziņojumu sūtīšanai norādītajām lietotāju grupām.
userids	ID/array	Atgriezt tikai darbības, kas ir konfigurētas ziņojumu sūtīšanai norādītajiem lietotājiem.
scriptids	ID/array	Atgriezt tikai darbības, kas ir konfigurētas norādīto skriptu palaišanai.
selectFilter	query	Atgriezt `filter` īpašību ar darbības nosacījumu filtru.
selectOperations	query	Atgriezt `operations` īpašību ar darbības operācijām.
selectRecoveryOperations	query	Atgriezt `recovery_operations` īpašību ar darbības atkopšanas operācijām.
selectUpdateOperations	query	Atgriezt `update_operations` īpašību ar darbības atjaunināšanas operācijām.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `actionid`, `name`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt trigeru darbības

Iegūst visas konfigurētās trigeru darbības kopā ar darbību nosacījumiem un operācijām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectRecoveryOperations": "extend",
        "selectUpdateOperations": "extend",
        "selectFilter": "extend",
        "filter": {
            "eventsource": 0
```

```
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "actionid": "3",
            "name": "Ziņot par problēmām Zabbix administratoriem",
            "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": "Trigeris: {TRIGGER.NAME}\r\nTrigera stāvoklis: {TRIGGER.STATUS}\r\nTri
                        "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": "Atjaunināts: {TRIGGER.NAME}",
                        "message": "{USER.FULLNAME} atjaunināja problēmu {EVENT.UPDATE.DATE} {EVENT.UPDATE
                        "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
}
```

Atgūt atklāšanas darbības

Atgūst visas konfigurētās atklāšanas darbības kopā ar darbību nosacījumiem un operācijām. Filtrs izmanto ”and” novērtēšanas tipu, tāpēc rekvizīts `formula` ir tukšs un `eval_formula` tiek ģenerēts automātiski.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "action.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend",
        "filter": {
```

```
            "eventsource": 1
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatīt arī

- Darbības filtrs
- Darbības operācija

Avots

CAction::get() failā *ui/include/classes/api/services/CAction.php*.

action.update

Apraksts

`object action.update(object/array actions)`

Šī metode ļauj atjaunināt esošās darbības.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās darbības rekvizīti.

Rekvizītam `actionid` jābūt definētam katrai darbībai, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta darbības rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
filter	object	Darbības filtrs objekts, ar kuru aizstāt pašreizējo filtru.
operations	array	Darbības operācijas, ar kurām aizstāt esošās operācijas.
recovery_operations	array	Darbības atkopšanas operācijas, ar kurām aizstāt esošās atkopšanas operācijas. Parametra darbība: - *atbalstīts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris”, ”iekšējs notikums” vai ”notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā”

Parametrs	Type	Apraksts
update_operations	array	Darbības atjaunināšanas operācijas, ar kurām aizstāt esošās atjaunināšanas operācijas. Parametra darbība: - *atbalstīts*, ja Darbības objekta `eventsource` ir iestatīts uz ”notikums, ko izveidojis trigeris” vai ”notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā”

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto darbību ID zem `actionids` īpašības.

Piemēri

Atspējot darbību

Atspējojiet darbību, tas ir, iestatiet tās statusu uz ”1”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "action.update",
    "params": {
        "actionid": "2",
        "status": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "actionids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Darbības filtrs
- Darbības operācija

Avots

CAction::update() failā *ui/include/classes/api/services/CAction.php*.

Grafika prototips

Šī klase ir paredzēta darbam ar grafiku prototipiem.

Objektu atsauces:

- Grafika prototips

Pieejamās metodes:

- graphprototype.create - izveidot jaunus grafiku prototipus
- graphprototype.delete - dzēst grafiku prototipus
- graphprototype.get - izgūt grafiku prototipus
- graphprototype.update - atjaunināt grafiku prototipus

Grafika prototipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `graphprototype` API.

Grafika prototips

Grafika prototipa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
graphid	ID	Grafika prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
height	integer	Grafika prototipa augstums pikseļos. Īpašības darbība: - *obligāts* izveides operācijām
name	string	Grafika prototipa nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
width	integer	Grafika prototipa platums pikseļos. Īpašības darbība: - *obligāts* izveides operācijām
flags	integer	Grafika prototipa izcelsme. Iespējamās vērtības: 2 - grafika prototips; 6 - atklāts grafika prototips Īpašības darbība: - *tikai lasāms*
graphtype	integer	Grafika prototipa izkārtojuma tips. Iespējamās vērtības: 0 - *(noklusējums)* parasts; 1 - sakrauts; 2 - sektoru; 3 - sadalīts.
percent_left	float	Kreisā procentile. Noklusējums: 0.
percent_right	float	Labā procentile. Noklusējums: 0.
show_3d	integer	Vai atklātos sektoru un sadalītos grafikus rādīt 3D režīmā. Iespējamās vērtības: 0 - *(noklusējums)* rādīt 2D režīmā; 1 - rādīt 3D režīmā.
show_legend	integer	Vai atklātajā grafikā rādīt leģendu. Iespējamās vērtības: 0 - slēpt; 1 - *(noklusējums)* rādīt.
show_work_period	integer	Vai atklātajā grafikā rādīt darba laiku. Iespējamās vērtības: 0 - slēpt; 1 - *(noklusējums)* rādīt.

Īpašība	Tips	Apraksts
templateid	ID	Vecākveidnes grafika prototipa ID. Īpašības darbība: - *tikai lasāms*
yaxismax	float	Fiksētā maksimālā vērtība Y asij.
yaxismin	float	Fiksētā minimālā vērtība Y asij.
ymax_itemid	ID	Tā vienuma ID, kas tiek izmantots kā Y ass maksimālā vērtība. Ja lietotājam nav piekļuves norādītajam vienumam, grafiks tiek attēlots tā, it kā `ymax_type` būtu iestatīts uz "calculated".
ymax_type	integer	Y ass maksimālās vērtības aprēķināšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* calculated; 1 - fiksēta; 2 - vienums.
ymin_itemid	ID	Tā vienuma ID, kas tiek izmantots kā Y ass minimālā vērtība. Ja lietotājam nav piekļuves norādītajam vienumam, grafiks tiek attēlots tā, it kā `ymin_type` būtu iestatīts uz "calculated".
ymin_type	integer	Y ass minimālās vērtības aprēķināšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* calculated; 1 - fiksēta; 2 - vienums.
discover	integer	Grafika prototipa atklāšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* tiks atklāti jauni grafiki; 1 - jauni grafiki netiks atklāti, un esošie grafiki tiks atzīmēti kā zaudēti.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto grafiku prototipu sasaistīšanai ar jau esošajiem. Tiek ģenerēts automātiski, ja nav norādīts. Īpašības darbība: - *atbalstīts*, ja grafika prototips pieder veidnei

graphprototype.create

Apraksts

`object graphprototype.create(object/array graphPrototypes)`

Šī metode ļauj izveidot jaunus grafiku prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie grafiku prototipi.

Papildus standarta grafiku prototipu īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
gitems	array	Grafika vienumi, kas jāizveido grafiku prototipiem. Grafika vienumi var atsaukties gan uz vienumiem, gan vienumu prototipiem, taču jābūt norādītam vismaz vienam vienuma prototipam. Parametra darbība: - *required*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto grafiku prototipu ID īpašumā `graphids`. Atgriezto ID secība atbilst nodoto grafiku prototipu secībai.

Piemēri

Grafika prototipa izveide

Izveidojiet grafika prototipu ar diviem vienumiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.create",
    "params": {
        "name": "Diska vietas izmantojums {#FSNAME}",
        "width": 900,
        "height": 200,
        "gitems": [
            {
                "itemid": "22828",
                "color": "00AA00"
            },
            {
                "itemid": "22829",
                "color": "3333FF"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Grafika vienums

Avots

CGraphPrototype::create() failā *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.delete

Apraksts

`object graphprototype.delete(array graphPrototypeIds)`

Šī metode ļauj dzēst grafiku prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo grafiku prototipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto grafiku prototipu ID īpašumā `graphids`.

Piemēri

Vairāku grafiku prototipu dzēšana

Dzēsiet divus grafiku prototipus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Avots

CGraphPrototype::delete() failā *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.get

Apraksts

`integer/array graphprototype.get(object parameters)`

Metode ļauj izgūt grafiku prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
discoveryids	ID/array	Atgriezt tikai tos grafiku prototipus, kas pieder norādītajiem atklāšanas noteikumiem.
graphids	ID/array	Atgriezt tikai grafiku prototipus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai tos grafiku prototipus, kas pieder hostiem vai veidnēm norādītajās hostu grupās vai veidņu grupās.
hostids	ID/array	Atgriezt tikai tos grafiku prototipus, kas pieder norādītajiem hostiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotos grafiku prototipus.
itemids	ID/array	Atgriezt tikai tos grafiku prototipus, kas satur norādītos vienumu prototipus.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai tos grafiku prototipus, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai tos grafiku prototipus, kas pieder norādītajām veidnēm.
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar grafika prototipa atklāšanas objekta datiem. Grafika prototipa atklāšanas objekts sasaista atklātu grafika prototipu ar grafika prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_graphid` - `(ID)` grafika prototipa ID, no kura grafiks ir izveidots; `status` - `(int)` grafika atklāšanas statuss: 0 - *(noklusējums)* grafika prototips ir atklāts, 1 - grafika prototips vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad grafika prototips, kas vairs netiek atklāts, tiks dzēsts.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar LLD noteikumu, kuram pieder grafika prototips.
selectDiscoveryRulePrototype	query	Atgriezt `discoveryRulePrototype` īpašību ar vecāka LLD noteikuma prototipu, kuram pieder grafika prototips.
selectGraphItems	query	Atgriezt `gitems` īpašību ar grafika vienumiem, kas izmantoti grafika prototipā.
selectHostGroups	query	Atgriezt `hostgroups` īpašību ar hostu grupām, kurām pieder grafika prototips.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kuriem pieder grafika prototips.
selectItems	query	Atgriezt `items` īpašību ar vienumiem un vienumu prototipiem, kas izmantoti grafika prototipā.
selectTemplateGroups	query	Atgriezt `templategroups` īpašību ar veidņu grupām, kurām pieder grafika prototips.
selectTemplates	query	Atgriezt `templates` īpašību ar veidnēm, kurām pieder grafika prototips.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram pieder grafika prototips; `hostid` - hosta ID, kuram pieder grafika prototips.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `graphid`, `name`, `graphtype`, `discovered`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parametrs	Tips	Apraksts
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Grafiku prototipu izgūšana no LLD kārtulas

Izgūst visus grafiku prototipus no LLD kārtulas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.get",
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Atklāšanas kārtula
- Grafika vienums
- Vienums
- Hosts

- Hostu grupa
- Veidne
- Veidņu grupa

Avots

CGraphPrototype::get() failā *ui/include/classes/api/services/CGraphPrototype.php*.

graphprototype.update

Apraksts

`object graphprototype.update(object/array graphPrototypes)`

Šī metode ļauj atjaunināt esošos grafiku prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās grafiku prototipu īpašības.

Katram grafika prototipam jābūt definētai `graphid` īpašībai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta grafiku prototipu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
gitems	array	Grafika vienumi, ar kuriem aizstāt esošos grafika vienumus. Ja grafika vienumam ir definēta `gitemid` īpašība, tas tiks atjaunināts; pretējā gadījumā tiks izveidots jauns grafika vienums.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto grafiku prototipu ID īpašībā `graphids`.

Piemēri

Grafika prototipa izmēra maiņa

Mainiet grafika prototipa izmēru uz 1100 x 400 pikseļiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graphprototype.update",
    "params": {
        "graphid": "439",
        "width": 1100,
        "height": 400
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
        ]
    },
    "id": 1
}
```

Avots

CGraphPrototype::update() failā *ui/include/classes/api/services/CGraphPrototype.php*.

Grafika vienums

Šī klase ir paredzēta darbam ar grafika vienumiem.

Objektu atsauces:

- Grafika vienums

Pieejamās metodes:

- graphitem.get - izgūt grafika vienumus

Grafika vienuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `graphitem` API.

Grafika vienums

> **Note:**
> Grafika vienumus var modificēt tikai, izmantojot `graph` API.

Grafika vienuma objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
gitemid	ID	Grafika vienuma ID. Īpašības darbība: - *tikai lasāms*
color	string	Grafika vienuma attēlošanas krāsa heksadecimāla krāsas koda formātā. Īpašības darbība: - *obligāta* izveides darbībām
itemid	ID	Vienuma ID. Īpašības darbība: - *obligāta* izveides darbībām
calc_fnc	integer	Vienuma vērtība, kas tiks attēlota. Iespējamās vērtības: 1 - minimālā vērtība; 2 - *(noklusējums)* vidējā vērtība; 4 - maksimālā vērtība; 7 - visas vērtības; 9 - pēdējā vērtība, tiek izmantota tikai sektoru un sadalītajos grafikos.
drawtype	integer	Grafika vienuma zīmēšanas stils. Iespējamās vērtības: 0 - *(noklusējums)* līnija; 1 - aizpildīts apgabals; 2 - trekna līnija; 3 - punkts; 4 - pārtraukta līnija; 5 - gradienta līnija.
graphid	ID	Grafika ID, kuram pieder grafika vienums.
sortorder	integer	Vienuma pozīcija grafikā. Noklusējums: sākas ar ”0” un ar katru ierakstu palielinās par vienu.

Īpašība	Tips	Apraksts
type	integer	Grafika vienuma tips. Iespējamās vērtības: 0 - *(noklusējums)* vienkāršs; 2 - grafika summa, tiek izmantota tikai sektoru un sadalītajos grafikos.
yaxisside	integer	Grafika puse, kurā tiks attēlota grafika vienuma Y skala. Iespējamās vērtības: 0 - *(noklusējums)* kreisā puse; 1 - labā puse.

graphitem.get

Apraksts

`integer/array graphitem.get(object parameters)`

Metode ļauj izgūt grafiku vienumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
graphids	ID/array	Atgriezt tikai grafiku vienumus, kas pieder norādītajiem grafikiem.
itemids	ID/array	Atgriezt tikai grafiku vienumus ar norādītajiem vienumu ID.
type	integer	Atgriezt tikai grafiku vienumus ar norādīto tipu. Atbalstīto grafiku vienumu tipu sarakstu skatiet grafika vienuma objekta lapā.
selectGraphs	query	Atgriezt `graphs` īpašību ar grafiku masīvu, kuriem pieder vienums.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `gitemid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
limit	integer	
output	query	
preservekeys	boolean	
sortorder	string/array	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Grafika vienumu izgūšana no grafika

Izgūstiet visus grafikā izmantotos grafika vienumus ar papildu informāciju par vienumu un hostu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graphitem.get",
    "params": {
        "output": "extend",
        "graphids": "387"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Grafiks

Avots

CGraphItem::get() failā *ui/include/classes/api/services/CGraphItem.php*.

Grafiks

Šī klase ir paredzēta darbam ar grafikiem.

Objektu atsauces:

- Graph

Pieejamās metodes:

- graph.create - izveidot jaunus grafikus
- graph.delete - dzēst grafikus
- graph.get - izgūt grafikus
- graph.update - atjaunināt grafikus

Grafika objekts

Tālāk norādītie objekti ir tieši saistīti ar `graph` API.

Grafiks

Grafika objektam ir šādas īpašības.

Īpašība	Type	Apraksts
graphid	ID	Grafika ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
height	integer	Grafika augstums pikseļos. Īpašības darbība: - *obligāts* izveides darbībām
name	string	Grafika nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
width	integer	Grafika platums pikseļos. Īpašības darbība: - *obligāts* izveides darbībām
flags	integer	Grafika izcelsme. Iespējamās vērtības: 0 - *(noklusējums)* parasts grafiks; 4 - no prototipa pārveidots grafiks. Īpašības darbība: - *tikai lasāms*
graphtype	integer	Grafika izkārtojuma tips. Iespējamās vērtības: 0 - *(noklusējums)* parasts; 1 - sakrauts; 2 - sektoru; 3 - sadalīts.
percent_left	float	Kreisā procentile. Noklusējums: 0.
percent_right	float	Labā procentile. Noklusējums: 0.
show_3d	integer	Vai sektoru un sadalītos grafikus rādīt 3D skatā. Iespējamās vērtības: 0 - *(noklusējums)* rādīt 2D skatā; 1 - rādīt 3D skatā.

Īpašība	Type	Apraksts
show_legend	integer	Vai grafikā rādīt leģendu. Iespējamās vērtības: 0 - slēpt; 1 - *(noklusējums)* rādīt.
show_work_period	integer	Vai grafikā rādīt darba laiku. Iespējamās vērtības: 0 - slēpt; 1 - *(noklusējums)* rādīt.
show_triggers	integer	Vai grafikā rādīt trigera līniju. Iespējamās vērtības: 0 - slēpt; 1 - *(noklusējums)* rādīt.
templateid	ID	Vecākveidnes grafika ID. Īpašības darbība: - *tikai lasāms*
yaxismax	float	Fiksētā maksimālā vērtība Y asij. Noklusējums: 100.
yaxismin	float	Fiksētā minimālā vērtība Y asij. Noklusējums: 0.
ymax_itemid	ID	Tā vienuma ID, kas tiek izmantots kā Y ass maksimālā vērtība. Ja lietotājam nav piekļuves norādītajam vienumam, grafiks tiek attēlots tā, it kā `ymax_type` būtu iestatīts uz ”calculated”.
ymax_type	integer	Y ass maksimālās vērtības aprēķināšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* calculated; 1 - fixed; 2 - item.
ymin_itemid	ID	Tā vienuma ID, kas tiek izmantots kā Y ass minimālā vērtība. Ja lietotājam nav piekļuves norādītajam vienumam, grafiks tiek attēlots tā, it kā `ymin_type` būtu iestatīts uz ”calculated”.
ymin_type	integer	Y ass minimālās vērtības aprēķināšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* calculated; 1 - fixed; 2 - item.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto grafiku sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja grafiks pieder veidnei

graph.create

Apraksts

`object graph.create(object/array graphs)`

Šī metode ļauj izveidot jaunus grafikus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Veidojamie grafiki.

Papildus standarta grafika īpašībām šī metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
gitems	array	Grafika vienumi, kas jāizveido grafikam. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto grafiku ID īpašībā `graphids`. Atgriezto ID secība atbilst nodoto grafiku secībai.

Piemēri

Grafika izveide

Izveidojiet grafiku ar diviem vienumiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Grafika vienums

Avots

CGraph::create() failā *ui/include/classes/api/services/CGraph.php*.

graph.delete

Apraksts

`object graph.delete(array graphIds)`

Šī metode ļauj dzēst grafikus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo grafiku ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto grafiku ID īpašībā `graphids`.

Piemēri

Vairāku grafiku dzēšana

Dzēsiet divus grafikus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graph.delete",
    "params": [
        "652",
        "653"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "652",
            "653"
        ]
    },
    "id": 1
}
```

Avots

CGraph::delete() failā *ui/include/classes/api/services/CGraph.php*.

graph.get

Apraksts

`integer/array graph.get(object parameters)`

Šī metode ļauj izgūt grafikus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
graphids	ID/array	Atgriezt tikai grafikus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai grafikus, kas pieder hostiem vai veidnēm norādītajās hostu grupās vai veidņu grupās.
templateids	ID/array	Atgriezt tikai grafikus, kas pieder norādītajām veidnēm.
hostids	ID/array	Atgriezt tikai grafikus, kas pieder norādītajiem hostiem.
itemids	ID/array	Atgriezt tikai grafikus, kas satur norādītos vienumus.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai grafikus, kas pieder veidnēm.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai grafikus, kas mantoti no veidnes.
expandName	flag	Izvērst makrosus grafika nosaukumā.
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar grafika atklāšanas objekta datiem. Grafika atklāšanas objekts sasaista atklātu grafiku ar grafika prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_graphid` - `(ID)` grafika prototipa ID, no kura grafiks ir izveidots; `status` - `(int)` grafika atklāšanas statuss: 0 - *(noklusējums)* grafiks ir atklāts, 1 - grafiks vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad grafiks, kas vairs netiek atklāts, tiks dzēsts.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar zema līmeņa atklāšanas kārtulu, kas izveidoja grafiku.
selectHostGroups	query	Atgriezt `hostgroups` īpašību ar hostu grupām, kurām grafiks pieder.
selectTemplateGroups	query	Atgriezt `templategroups` īpašību ar veidņu grupām, kurām grafiks pieder.
selectTemplates	query	Atgriezt `templates` īpašību ar veidnēm, kurām grafiks pieder.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kuriem grafiks pieder.
selectItems	query	Atgriezt `items` īpašību ar grafikā izmantotajiem vienumiem.
selectGraphItems	query	Atgriezt `gitems` īpašību ar grafikā izmantotajiem vienumiem.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram grafiks pieder; `hostid` - hosta ID, kuram grafiks pieder.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `graphid`, `name`, `graphtype`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Parametrs	Tips	Apraksts
selectGraphDiscovery	query	Atgriezt `graphDiscovery` īpašību ar grafika atklāšanas objektu. Grafika atklāšanas objekti sasaista grafiku ar grafika prototipu, no kura tas tika izveidots. Šis vaicājums ir **novecojis**, lūdzu, izmantojiet `selectDiscoveryData` vietā.

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Grafiku izgūšana no hostiem

Izgūstiet visus grafikus no hosta "10107" un sakārtojiet tos pēc nosaukuma.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graph.get",
    "params": {
        "output": "extend",
        "hostids": 10107,
        "sortfield": "name"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Atklāšanas noteikums
- Grafika vienums
- Vienums
- Hosts
- Hostu grupa
- Veidne
- Veidņu grupa

Avots

CGraph::get() failā *ui/include/classes/api/services/CGraph.php*.

graph.update

Apraksts

`object graph.update(object/array graphs)`

Šī metode ļauj atjaunināt esošos grafikus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmā grafika īpašības.

Īpašībai `graphid` jābūt definētai katram grafikam, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta grafika īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
gitems	array	Grafika vienumi, ar kuriem aizstāt esošos grafika vienumus. Ja grafika vienumam ir definēta īpašība `gitemid`, tas tiks atjaunināts, pretējā gadījumā tiks izveidots jauns grafika vienums.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto grafiku ID īpašībā `graphids`.

Piemēri

Y skalas maksimālās vērtības iestatīšana

Iestatiet Y skalas maksimālo vērtību uz fiksētu vērtību 100.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "graph.update",
    "params": {
        "graphid": "439",
        "ymax_type": 1,
        "yaxismax": 100
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "graphids": [
            "439"
```

```
        ]
    },
    "id": 1
}
```

Avots

CGraph::update() failā *ui/include/classes/api/services/CGraph.php*.

Hosta prototips

Šī klase ir paredzēta darbam ar hostu prototipiem.

Objektu atsauces:

- Hosta prototips
- Grupas saite
- Grupas prototips
- Hosta prototipa tags
- Pielāgota saskarne
    - Pielāgotas saskarnes informācija

Pieejamās metodes:

- hostprototype.create - izveidot jaunus hostu prototipus
- hostprototype.delete - dzēst hostu prototipus
- hostprototype.get - izgūt hostu prototipus
- hostprototype.update - atjaunināt hostu prototipus

Hosta prototipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `hostprototype` API.

Hosta prototips

Hosta prototipa objektam ir šādas īpašības.

Īpašība	Type	Apraksts
hostid	ID	Hosta prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
host	string	Hosta prototipa tehniskais nosaukums. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
name	string	Hosta prototipa redzamais nosaukums. Noklusējums: `host` īpašības vērtība. Īpašības darbība: - *tikai lasāms* mantotajiem objektiem
status	integer	Hosta prototipa statuss. Iespējamās vērtības: 0 - *(noklusējums)* uzraudzīts hosts; 1 - neuzraudzīts hosts.

Īpašība	Type	Apraksts
flags	integer	Hosta prototipa izcelsme. Iespējamās vērtības: 2 - hosta prototips; 6 - atklāts hosta prototips Īpašības darbība: - *tikai lasāms*
inventory_mode	integer	Hosta inventāra aizpildīšanas režīms. Iespējamās vērtības: -1 - *(noklusējums)* atspējots; 0 - manuāls; 1 - automātisks.
templateid	ID	Vecākveidnes hosta prototipa ID. Īpašības darbība: - *tikai lasāms*
discover	integer	Hosta prototipa atklāšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* tiks atklāti jauni hosti; 1 - jauni hosti netiks atklāti, un esošie hosti tiks atzīmēti kā zaudēti.
custom_interfaces	integer	Pielāgoto saskarņu avots hostiem, kas izveidoti no hosta prototipa. Iespējamās vērtības: 0 - *(noklusējums)* mantot saskarnes no vecāka hosta; 1 - izmantot hosta prototipu pielāgotās saskarnes. Īpašības darbība: - *tikai lasāms* mantotajiem objektiem
uuid	string	Universāls unikāls identifikators, ko izmanto importēto hostu prototipu sasaistīšanai ar jau esošajiem. Tiek ģenerēts automātiski, ja nav norādīts. Īpašības darbība: - *atbalstīts*, ja hosta prototips pieder veidnei

Grupas saite

Grupas saites objekts sasaista hosta prototipu ar hostu grupu. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
groupid	ID	Hostu grupas ID. Īpašības darbība: - *obligāts*

Grupas prototips

Grupas prototipa objekts definē grupu, kas tiks izveidota atklātam hostam. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
group_prototypeid	ID	Grupas prototipa ID. Īpašības darbība: - *tikai lasāms*

Īpašība	Type	Apraksts
name	string	Grupas prototipa nosaukums. Īpašības darbība: - *obligāts* izveides darbībām

Hosta prototipa tags

Hosta prototipa taga objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Hosta prototipa taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Hosta prototipa taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

Pielāgota saskarne

Pielāgotas saskarnes tiek atbalstītas, ja Host prototype object parametrs `custom_interfaces` ir iestatīts uz ”izmantot hostu prototipu pielāgotās saskarnes”. Pielāgotās saskarnes objektam ir šādas īpašības.

Īpašība	Type	Apraksts
type	integer	Saskarnes tips. Iespējamās vērtības: 1 - Aģents; 2 - SNMP; 3 - IPMI; 4 - JMX. Property behavior: - *required*
useip	integer	Vai savienojums jāveido, izmantojot IP. Iespējamās vērtības: 0 - savienoties, izmantojot hosta DNS nosaukumu; 1 - savienoties, izmantojot hosta IP adresi. Property behavior: - *required*

Īpašība	Type	Apraksts
ip	string	Saskarnes izmantotā IP adrese. Var saturēt makrosus. Property behavior: - *required* ja `useip` ir iestatīts uz ”savienoties, izmantojot hosta IP adresi”
dns	string	Saskarnes izmantotais DNS nosaukums. Var saturēt makrosus. Property behavior: - *required* ja `useip` ir iestatīts uz ”savienoties, izmantojot hosta DNS nosaukumu”
port	string	Saskarnes izmantotais porta numurs. Var saturēt lietotāja un LLD makrosus. Property behavior: - *required*
main	integer	Vai saskarne tiek izmantota kā noklusējuma saskarne hostā. Hostā tikai viena noteikta tipa saskarne var būt iestatīta kā noklusējuma saskarne. Iespējamās vērtības: 0 - nav noklusējuma; 1 - noklusējuma. Property behavior: - *required*
details	object	Papildu objekts pielāgotās saskarnes informācijai. Property behavior: - *required* ja `type` ir iestatīts uz ”SNMP”

Pielāgotās saskarnes informācija

Objektam details ir šādas īpašības.

Īpašība	Tips	Apraksts
version	integer	SNMP saskarnes versija. Iespējamās vērtības: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Īpašības darbība: - *obligāta*
bulk	integer	Vai izmantot SNMP lielapjoma pieprasījumus. Iespējamās vērtības: 0 - neizmantot lielapjoma pieprasījumus; 1 - *(noklusējums)* - izmantot lielapjoma pieprasījumus.
community	string	SNMP kopiena. Īpašības darbība: - *obligāta*, ja `version` ir iestatīts uz ”SNMPv1” vai ”SNMPv2c”
max_repetitions	integer	Maksimālā atkārtojumu vērtība vietējiem SNMP lielapjoma pieprasījumiem (GetBulkRequest-PDUs). Izmanto tikai `discovery[]` un `walk[]` vienumiem SNMPv2 un v3. Noklusējums: 10.

Īpašība	Tips	Apraksts
securityname	string	SNMPv3 drošības nosaukums. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3”
securitylevel	integer	SNMPv3 drošības līmenis. Iespējamās vērtības: 0 - *(noklusējums)* - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3”
authpassphrase	string	SNMPv3 autentifikācijas parole. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3” un `securitylevel` ir iestatīts uz ”authNoPriv” vai ”authPriv”
privpassphrase	string	SNMPv3 privātuma parole. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3” un `securitylevel` ir iestatīts uz ”authPriv”
authprotocol	integer	SNMPv3 autentifikācijas protokols. Iespējamās vērtības: 0 - *(noklusējums)* - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3” un `securitylevel` ir iestatīts uz ”authNoPriv” vai ”authPriv”
privprotocol	integer	SNMPv3 privātuma protokols. Izmanto tikai SNMPv3 saskarnes. Iespējamās vērtības: 0 - *(noklusējums)* - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3” un `securitylevel` ir iestatīts uz ”authPriv”
contextname	string	SNMPv3 konteksta nosaukums. Īpašības darbība: - *atbalstīta*, ja `version` ir iestatīts uz ”SNMPv3”

hostprototype.create

Apraksts

`object hostprototype.create(object/array hostPrototypes)`

Šī metode ļauj izveidot jaunus hostu prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie hostu prototipi.

Papildus standarta hosta prototipa īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groupLinks	array	Grupu saites, kas tiks izveidotas hosta prototipam. Parametra darbība: - *obligāts*
ruleid	ID	LLD kārtulas ID, kurai pieder hosta prototips. Parametra darbība: - *obligāts*
groupPrototypes	array	Grupu prototipi, kas tiks izveidoti hosta prototipam.
macros	object/array	Lietotāja makrosi, kas tiks izveidoti hosta prototipam.
tags	object/array	Hosta prototipa birkas.
interfaces	object/array	Hosta prototipa pielāgotās saskarnes.
templates	object/array	Veidnes, kas tiks piesaistītas hosta prototipam. Veidnēm jābūt definētai tikai `templateid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hostu prototipu ID īpašībā `hostids`. Atgriezto ID secība atbilst nodoto hostu prototipu secībai.

Piemēri

Hosts prototipa izveide

Izveidojiet hosts prototipu "{#VM.NAME}" LLD kārtulai "23542" ar grupas prototipu "{#HV.NAME}", birkas pāri "datacenter": "{#DATACENTER.NAME}" un pielāgotu SNMPv2 saskarni 127.0.0.1:161 ar kopienu {$SNMP_COMMUNITY}. Saistiet to ar hostu grupu "2".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Grupas saite
- Grupas prototips
- Hosta prototipa tags
- Pielāgota saskarne
- Lietotāja makross

Avots

CHostPrototype::create() failā *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.delete

Apraksts

`object hostprototype.delete(array hostPrototypeIds)`

Šī metode ļauj dzēst hostu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Dzēšamo hostu prototipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hostu prototipu ID zem `hostids` īpašības.

Piemēri

Vairāku hostu prototipu dzēšana

Dzēsiet divus hostu prototipus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.delete",
    "params": [
        "10103",
        "10105"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10103",
            "10105"
        ]
    },
    "id": 1
}
```

Avots

CHostPrototype::delete() failā *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.get

Apraksts

`integer/array hostprototype.get(object parameters)`

Metode ļauj izgūt hostu prototipus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
hostids	ID/array	Atgriezt tikai hostu prototipus ar norādītajiem ID.
discoveryids	ID/array	Atgriezt tikai hostu prototipus, kas pieder norādītajiem LLD noteikumiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai vienumus, kas mantoti no veidnes.

Parametrs	Tips	Apraksts
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar hosta prototipa atklāšanas objekta datiem. Hosta prototipa atklāšanas objekts sasaista atklātu hosta prototipu ar hosta prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `host` - `(string)` hosta ID; `parent_hostid` - `(string)` hosta prototipa ID, no kura hosta prototips tika izveidots; `status` - `(int)` hosta prototipa atklāšanas statuss: 0 - *(noklusējums)* hosta prototips ir atklāts, 1 - hosta prototips vairs netiek atklāts; `ts_delete` - `(timestamp)` laiks, kad hosta prototips, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad hosta prototips, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` indikators, vai hosta prototips tika atspējots ar LLD noteikumu vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD noteikumu.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar LLD noteikumu, kuram pieder hosta prototips.
selectDiscoveryRulePrototype	query	Atgriezt `discoveryRulePrototype` īpašību ar vecāka LLD noteikuma prototipu, kuram pieder hosta prototips.
selectInterfaces	query	Atgriezt `interfaces` īpašību ar hosta prototipa pielāgotajām saskarnēm.
selectGroupLinks	query	Atgriezt `groupLinks` īpašību ar hosta prototipa grupu saitēm.
selectGroupPrototypes	query	Atgriezt `groupPrototypes` īpašību ar hosta prototipa grupu prototipiem.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas atrodas piesaistītajās veidnēs.
selectMacros	query	Atgriezt `macros` īpašību ar hosta prototipa makrosiem.
selectParentHost	query	Atgriezt `parentHost` īpašību ar hostu, kuram pieder hosta prototips.
selectTags	query	Atgriezt `tags` īpašību ar hosta prototipa birkām.
selectTemplates	query	Atgriezt `templates` īpašību ar veidnēm, kas piesaistītas hosta prototipam. Atbalsta `count`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `hostid`, `host`, `name`, `status`, `discovered`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Hostu prototipu izgūšana no LLD kārtulas

Izgūst visus hostu prototipus, to grupu saites, grupu prototipus un birkas no LLD kārtulas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.get",
    "params": {
        "output": "extend",
        "selectInterfaces": "extend",
        "selectGroupLinks": "extend",
        "selectGroupPrototypes": "extend",
        "selectTags": "extend",
        "discoveryids": "23554"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Grupas saite
- Grupas prototips
- Lietotāja makross

Avots

CHostPrototype::get() failā *ui/include/classes/api/services/CHostPrototype.php*.

hostprototype.update

Apraksts

`object hostprototype.update(object/array hostPrototypes)`

Šī metode ļauj atjaunināt esošos hostu prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjauninámo hostu prototipu īpašības.

Katram hosta prototipam jābūt definētai `hostid` īpašībai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta hosta prototipa īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groupLinks	array	Grupu saites, ar kurām aizstāt pašreizējās grupu saites hosta prototipā. Parametra darbība: - *tikai lasāms* mantotiem objektiem
groupPrototypes	array	Grupu prototipi, ar kuriem aizstāt esošos grupu prototipus hosta prototipā. Visi grupu prototipi, kas pieprasījumā nav norādīti, tiks noņemti. Parametra darbība: - *tikai lasāms* mantotiem objektiem
macros	object/array	Lietotāja makrosi, ar kuriem aizstāt pašreizējos lietotāja makrosus. Visi makrosi, kas pieprasījumā nav norādīti, tiks noņemti.

Parametrs	Tips	Apraksts
tags	object/array	Hosta prototipa birkas, ar kurām aizstāt pašreizējās birkas. Visas birkas, kas pieprasījumā nav norādītas, tiks noņemtas. Parametra darbība: - *tikai lasāms* mantotiem objektiem
interfaces	object/array	Hosta prototipa pielāgotās saskarnes, ar kurām aizstāt pašreizējās saskarnes. Pielāgotās saskarnes objektam jāietver visi tā parametri. Visas saskarnes, kas pieprasījumā nav norādītas, tiks noņemtas. Parametra darbība: - *atbalstīts*, ja Hosta prototipa objekta `custom_interfaces` ir iestatīts uz "use host prototypes custom interfaces" - *tikai lasāms* mantotiem objektiem
templates	object/array	Veidnes, ar kurām aizstāt pašlaik piesaistītās veidnes. Veidnēm jābūt definētai tikai `templateid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu prototipu ID zem `hostids` īpašības.

Piemēri

Hosta prototipa atspējošana

Atspējojiet hosta prototipu, tas ir, iestatiet tā statusu uz "1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "status": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Hosta prototipa tagu atjaunināšana

Aizstājiet hosta prototipa tagus ar jauniem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostprototype.update",
    "params": {
        "hostid": "10092",
        "tags": [
            {
                "tag": "datacenter",
```

```
                "value": "{#DATACENTER.NAME}"
            },
            {
                "tag": "instance-type",
                "value": "{#INSTANCE_TYPE}"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Hosta prototipa pielāgoto saskarņu atjaunināšana

Aizstājiet mantotās saskarnes ar hosta prototipa pielāgotajām saskarnēm.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10092"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Grupas saite
- Grupas prototips
- Hosta prototipa tags
- Pielāgota saskarne
- Lietotāja makross

Avots

CHostPrototype::update() failā *ui/include/classes/api/services/CHostPrototype.php*.

Hosta saskarne

Šī klase ir paredzēta darbam ar hostu saskarnēm.

Objektu atsauces:

- Hosta saskarne
  - Detaļas

Pieejamās metodes:

- hostinterface.create - izveidot jaunas hostu saskarnes
- hostinterface.delete - dzēst hostu saskarnes
- hostinterface.get - izgūt hostu saskarnes
- hostinterface.massadd - pievienot hostu saskarnes hostiem
- hostinterface.massremove - noņemt hostu saskarnes no hostiem
- hostinterface.update - atjaunināt hostu saskarnes

Hosta saskarnes objekts

Tālāk norādītie objekti ir tieši saistīti ar `hostinterface` API.

Hosta saskarne

Hosta saskarnes objektam ir šādas īpašības.

Attention:
Ņemiet vērā, ka gan `ip`, gan `dns` īpašības ir *obligātas* izveides darbībām. Ja nevēlaties izmantot DNS, iestatiet to kā tukšu virkni.

Īpašība	Type	Apraksts
interfaceid	ID	Saskarnes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
available	integer	Hosta saskarnes pieejamība. Iespējamās vērtības: 0 - *(noklusējums)* nezināma; 1 - pieejama; 2 - nepieejama. Īpašības darbība: - *tikai lasāms*
hostid	ID	Hosta, kuram pieder saskarne, ID. Īpašības darbība: - *konstants* - *obligāts* izveides darbībām

Īpašība	Type	Apraksts
type	integer	Saskarnes tips. Iespējamās vērtības: 1 - Aģents; 2 - SNMP; 3 - IPMI; 4 - JMX. Īpašības darbība: - *obligāts* izveides darbībām
ip	string	Saskarnes izmantotā IP adrese. Var būt tukša, ja savienojums tiek veidots, izmantojot DNS. Īpašības darbība: - *obligāts* izveides darbībām
dns	string	Saskarnes izmantotais DNS nosaukums. Var būt tukšs, ja savienojums tiek veidots, izmantojot IP. Īpašības darbība: - *obligāts* izveides darbībām
port	string	Saskarnes izmantotais porta numurs. Var saturēt lietotāja makrosus. Īpašības darbība: - *obligāts* izveides darbībām
useip	integer	Vai savienojums jāveido, izmantojot IP. Iespējamās vērtības: 0 - savienoties, izmantojot hosta DNS nosaukumu; 1 - savienoties, izmantojot hosta IP adresi. Īpašības darbība: - *obligāts* izveides darbībām
main	integer	Vai saskarne tiek izmantota kā noklusējuma saskarne hostā. Tikai viena noteikta tipa saskarne hostā var būt iestatīta kā noklusējuma saskarne. Iespējamās vērtības: 0 - nav noklusējuma; 1 - noklusējuma. Īpašības darbība: - *obligāts* izveides darbībām
details	object	Papildu saskarnes details objekts. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP”
disable_until	timestamp	Nākamais nepieejama hosta saskarnes aptaujas laiks. Īpašības darbība: - *tikai lasāms*
error	string	Kļūdas teksts, ja hosta saskarne nav pieejama. Īpašības darbība: - *tikai lasāms*
errors_from	timestamp	Laiks, kad hosta saskarne kļuva nepieejama. Īpašības darbība: - *tikai lasāms*

Detalizēta informācija

Objektam details ir šādas īpašības.

Īpašība	Tips	Apraksts
version	integer	SNMP saskarnes versija. Iespējamās vērtības: 1 - SNMPv1; 2 - SNMPv2c; 3 - SNMPv3. Īpašības darbība: - *obligāta*
bulk	integer	Vai izmantot SNMP lielapjoma pieprasījumus. Iespējamās vērtības: 0 - neizmantot lielapjoma pieprasījumus; 1 - (noklusējums) - izmantot lielapjoma pieprasījumus.
community	string	SNMP kopiena. Tiek izmantota tikai SNMPv1 un SNMPv2 saskarnēs. Īpašības darbība: - *obligāta*, ja `version` ir iestatīts uz ”SNMPv1” vai ”SNMPv2c”
max_repetitions	integer	Maksimālā atkārtojumu vērtība vietējiem SNMP lielapjoma pieprasījumiem (GetBulkRequest-PDUs). Tiek izmantota tikai `discovery[]` un `walk[]` vienumiem SNMPv2 un v3. Noklusējums: 10.
securityname	string	SNMPv3 drošības nosaukums. Tiek izmantots tikai SNMPv3 saskarnēs.
securitylevel	integer	SNMPv3 drošības līmenis. Tiek izmantots tikai SNMPv3 saskarnēs. Iespējamās vērtības: 0 - (noklusējums) - noAuthNoPriv; 1 - authNoPriv; 2 - authPriv.
authpassphrase	string	SNMPv3 autentifikācijas parole. Tiek izmantota tikai SNMPv3 saskarnēs.
privpassphrase	string	SNMPv3 privātuma parole. Tiek izmantota tikai SNMPv3 saskarnēs.
authprotocol	integer	SNMPv3 autentifikācijas protokols. Tiek izmantots tikai SNMPv3 saskarnēs. Iespējamās vērtības: 0 - (noklusējums) - MD5; 1 - SHA1; 2 - SHA224; 3 - SHA256; 4 - SHA384; 5 - SHA512.
privprotocol	integer	SNMPv3 privātuma protokols. Tiek izmantots tikai SNMPv3 saskarnēs. Iespējamās vērtības: 0 - (noklusējums) - DES; 1 - AES128; 2 - AES192; 3 - AES256; 4 - AES192C; 5 - AES256C.
contextname	string	SNMPv3 konteksta nosaukums. Tiek izmantots tikai SNMPv3 saskarnēs.

hostinterface.create

Apraksts

`object hostinterface.create(object/array hostInterfaces)`

Šī metode ļauj izveidot jaunas hostu saskarnes.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Hosta saskarnes, ko izveidot.

Metode pieņem hosta saskarnes ar standarta hosta saskarnes īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hostu saskarņu ID īpašībā `interfaceids`. Atgriezto ID secība atbilst nodoto hostu saskarņu secībai.

Piemēri

Izveidot jaunu saskarni

Izveidojiet sekundāru IP aģenta saskarni hostam "30052".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Izveidojiet saskarni ar SNMP informāciju

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30063"
        ]
    },
    "id": 1
}
```

Skatiet arī

- hostinterface.massadd
- host.massadd

Avots

CHostInterface::create() failā *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.delete

Apraksts

`object hostinterface.delete(array hostInterfaceIds)`

Šī metode ļauj dzēst hostu saskarnes.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo hostu saskarņu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hostu saskarņu ID zem `interfaceids` īpašības.

Piemēri

Dzēst hosta saskarni

Dzēsiet hosta saskarni ar ID 30062.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.delete",
    "params": [
        "30062"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30062"
        ]
    },
    "id": 1
}
```

Skatiet arī

- hostinterface.massremove
- host.massremove

Avots

CHostInterface::delete() failā *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.get

Apraksts

`integer/array hostinterface.get(object parameters)`

Metode ļauj izgūt hostu saskarnes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
hostids	ID/array	Atgriezt tikai hostu saskarnes, ko izmanto norādītie hosti.
interfaceids	ID/array	Atgriezt tikai hostu saskarnes ar norādītajiem ID.
itemids	ID/array	Atgriezt tikai hostu saskarnes, ko izmanto norādītie vienumi.
triggerids	ID/array	Atgriezt tikai hostu saskarnes, ko izmanto vienumi norādītajos trigeros.
selectItems	query	Atgriezt `items` īpašību ar vienumiem, kas izmanto saskarni. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` īpašību ar hostu masīvu, kas izmanto saskarni.
limitSelects	integer	Ierobežo apakšatlases vaicājumu atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectItems`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `interfaceid`, `dns`, `ip`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parametrs	Tips	Apraksts
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt hosta saskarnes

Iegūstiet visus datus par saskarnēm, ko izmanto hosts "30057."

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.get",
    "params": {
        "output": "extend",
        "hostids": "30057"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Hosts
- Vienums

Avots

CHostInterface::get() failā *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massadd

Apraksts

`object hostinterface.massadd(object parameters)`

Šī metode ļauj vienlaikus pievienot hostu saskarnes vairākiem hostiem.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur izveidojamos hostu saskarnes norādītajos hostos.

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
interfaces	object/array	Hostu saskarnes, kas jāizveido norādītajos hostos. Parametra darbība: - *obligāts*
hosts	object/array	Hosti, kas jāatjaunina. Hostiem jābūt definētam tikai rekvizītam `hostid`. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hosta saskarņu ID zem `interfaceids` īpašības.

Piemēri

Saskarņu izveide

Izveidojiet saskarni divos hostos.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Skatiet arī

- hostinterface.create
- host.massadd
- Hosts

Avots

CHostInterface::massAdd() failā *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.massremove

Apraksts

`object hostinterface.massremove(object parameters)`

Šī metode ļauj noņemt hostu saskarnes no norādītajiem hostiem.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjaunināmo hostu ID un noņemamās saskarnes.

Parametrs	Tips	Apraksts
interfaces	object/array	Hostu saskarnes, kas jānoņem no norādītajiem hostiem. Hosta saskarnes objektam jābūt definētām tikai `ip`, `dns` un `port` īpašībām. Parametra darbība: - *obligāts*
hostids	ID/array	Atjaunināmo hostu ID. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hostu saskarņu ID īpašībā `interfaceids`.

Piemēri

Saskarņu noņemšana

Noņemiet SNMP saskarni "127.0.0.1" no diviem hostiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.massremove",
    "params": {
        "hostids": [
            "30050",
            "30052"
        ],
        "interfaces": {
            "dns": "",
            "ip": "127.0.0.1",
            "port": "161"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30069",
            "30070"
        ]
    },
    "id": 1
}
```

Skatīt arī

- hostinterface.delete
- host.massremove

Avots

CHostInterface::massRemove() failā *ui/include/classes/api/services/CHostInterface.php*.

hostinterface.update

Apraksts

`object hostinterface.update(object/array hostInterfaces)`

Šī metode ļauj atjaunināt esošās hostu saskarnes.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Host interface properties, kas jāatjaunina.

Katram hosta interfeisam jābūt norādītam rekvizītam `interfaceid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai norādītie rekvizīti, visi pārējie paliks nemainīti.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu saskarņu ID zem `interfaceids` īpašības.

Piemēri

Hosta saskarnes porta maiņa

Mainiet hosta saskarnes portu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": {
        "interfaceid": "30048",
        "port": "10055"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30048"
        ]
    },
    "id": 1
}
```

Vairāku hostu saskarņu portu maiņa

Mainiet vairāku hostu saskarņu portu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostinterface.update",
    "params": [
        {
            "interfaceid": "30049",
            "port": "10055"
        },
        {
            "interfaceid": "30050",
            "port": "10055"
        },
        {
            "interfaceid": "30051",
            "port": "10055"
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "interfaceids": [
            "30049",
            "30050",
            "30051"
```

```
        ]
    },
    "id": 1
}
```

Avots

CHostInterface::update() failā *ui/include/classes/api/services/CHostInterface.php*.

Hosts

Šī klase ir paredzēta darbam ar hostiem.

Objektu atsauces:

- Hosts
- Hosta inventārs
- Hosta tags

Pieejamās metodes:

- host.create - izveidot jaunus hostus
- host.delete - dzēst hostus
- host.get - iegūt hostus
- host.massadd - pievienot saistītos objektus hostiem
- host.massremove - noņemt saistītos objektus no hostiem
- host.update - atjaunināt hostus

Hosta objekts

Tālāk norādītie objekti ir tieši saistīti ar hosta API.

Hosts

Hosta objektam ir šādas īpašības.

Property	Type	Description
hostid	ID	Hosta ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
host	string	Hosta tehniskais nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
description	text	Hosta apraksts.
flags	integer	Hosta izcelsme. Iespējamās vērtības: 0 - parasts hosts; 4 - hosts, kas pārveidots no prototipa. Īpašības darbība: - *tikai lasāms*
inventory_mode	integer	Hosta inventarizācijas aizpildīšanas režīms. Iespējamās vērtības: -1 - *(noklusējums)* atspējots; 0 - manuāls; 1 - automātisks.

Property	Type	Description
ipmi_authtype	integer	IPMI autentifikācijas algoritms. Iespējamās vērtības: -1 - *(noklusējums)* noklusētais; 0 - nav; 1 - MD2; 2 - MD5 4 - straight; 5 - OEM; 6 - RMCP+.
ipmi_password	string	IPMI parole.
ipmi_privilege	integer	IPMI privilēģiju līmenis. Iespējamās vērtības: 1 - callback; 2 - *(noklusējums)* lietotājs; 3 - operators; 4 - administrators; 5 - OEM.
ipmi_username	string	IPMI lietotājvārds.
maintenance_from	timestamp	Spēkā esošās uzturēšanas sākuma laiks. Īpašības darbība: - *tikai lasāms*
maintenance_status	integer	Spēkā esošās uzturēšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* nav uzturēšanas; 1 - uzturēšana ir spēkā. Īpašības darbība: - *tikai lasāms*
maintenance_type	integer	Spēkā esošās uzturēšanas tips. Iespējamās vērtības: 0 - *(noklusējums)* uzturēšana ar datu vākšanu; 1 - uzturēšana bez datu vākšanas. Īpašības darbība: - *tikai lasāms*
maintenanceid	ID	Tās uzturēšanas ID, kas pašlaik ir spēkā hostam. Īpašības darbība: - *tikai lasāms*
name	string	Hosta redzamais nosaukums. Noklusējums: `host` īpašības vērtība.
monitored_by	integer	Avots, kas tiek izmantots hosta uzraudzībai. Iespējamās vērtības: 0 - *(noklusējums)* Zabbix serveris; 1 - starpniekserveris; 2 - starpniekserveru grupa.
proxyid	ID	Tā starpniekservera ID, kas tiek izmantots hosta uzraudzībai. Īpašības darbība: - *obligāts*, ja `monitored_by` ir iestatīts uz ”Proxy”
proxy_groupid	ID	Tās starpniekserveru grupas ID, kas tiek izmantota hosta uzraudzībai. Īpašības darbība: - *obligāts*, ja `monitored_by` ir iestatīts uz ”Proxy group”

Property	Type	Description
status	integer	Hosta statuss un funkcija. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
tls_connect	integer	Savienojumi ar hostu. Iespējamās vērtības: 1 - *(noklusējums)* bez šifrēšanas; 2 - PSK; 4 - sertifikāts.
tls_accept	integer	Savienojumi no hosta. Iespējamās bitkartes vērtības: 1 - *(noklusējums)* bez šifrēšanas; 2 - PSK; 4 - sertifikāts. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 6 PSK un sertifikātam).
tls_issuer	string	Sertifikāta izdevējs.
tls_subject	string	Sertifikāta subjekts.
tls_psk_identity	string	PSK identitāte; tai jābūt sasaistītai tikai ar vienu PSK (starp autoregistration, hosts un proxies). Neiekļaujiet sensitīvu informāciju PSK identitātē, jo tā tiek nosūtīta nešifrēta pa tīklu, lai informētu saņēmēju, kuru PSK izmantot. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur ”PSK” bitu
tls_psk	string	Iepriekš koplietotā atslēga (PSK); tai jābūt vismaz 32 heksadecimālām zīmēm garai. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur ”PSK” bitu
active_available	integer	Hosta aktīvās saskarnes pieejamības statuss. Iespējamās vērtības: 0 - saskarnes statuss nav zināms; 1 - saskarne ir pieejama; 2 - saskarne nav pieejama. Īpašības darbība: - *tikai lasāms*
assigned_proxyid	ID	Zabbix servera piešķirtā starpniekservera ID, ja hosts tiek uzraudzīts ar starpniekserveru grupu. Īpašības darbība: - *tikai lasāms*

Hosta inventārs

Hosta inventāra objektam ir šādas īpašības.

Note:
Katrai īpašībai ir savs unikāls ID numurs, kas tiek izmantots, lai saistītu hosta inventāra laukus ar vienumiem.

ID	Īpašība	Type	Apraksts	Maksimālais garums
4	alias	string	Aizstājvārds.	128 rakstzīmes
11	asset_tag	string	Aktīva birka.	64 rakstzīmes
28	chassis	string	Šasija.	64 rakstzīmes
23	contact	string	Kontaktpersona.	65535 rakstzīmes SQL datubāzēm
32	contract_number	string	Līguma numurs.	64 rakstzīmes
47	date_hw_decomm	string	Aparatūras ekspluatācijas pārtraukšanas datums.	64 rakstzīmes
46	date_hw_expiry	string	Aparatūras uzturēšanas termiņa beigu datums.	64 rakstzīmes
45	date_hw_install	string	Aparatūras uzstādīšanas datums.	64 rakstzīmes
44	date_hw_purchase	string	Aparatūras iegādes datums.	64 rakstzīmes
34	deployment_status	string	Izvietošanas statuss.	64 rakstzīmes
14	hardware	string	Aparatūra.	255 rakstzīmes
15	hardware_full	string	Detalizēta aparatūra.	65535 rakstzīmes SQL datubāzēm
39	host_netmask	string	Hosta apakštīkla maska.	39 rakstzīmes
38	host_networks	string	Hosta tīkli.	65535 rakstzīmes SQL datubāzēm
40	host_router	string	Hosta maršrutētājs.	39 rakstzīmes
30	hw_arch	string	Aparatūras arhitektūra.	32 rakstzīmes
33	installer_name	string	Uzstādītāja vārds.	64 rakstzīmes
24	location	string	Atrašanās vieta.	65535 rakstzīmes SQL datubāzēm
25	location_lat	string	Atrašanās vietas platums.	16 rakstzīmes
26	location_lon	string	Atrašanās vietas garums.	16 rakstzīmes
12	macaddress_a	string	MAC adrese A.	64 rakstzīmes
13	macaddress_b	string	MAC adrese B.	64 rakstzīmes
29	model	string	Modelis.	64 rakstzīmes
3	name	string	Nosaukums.	128 rakstzīmes
27	notes	string	Piezīmes.	65535 rakstzīmes SQL datubāzēm
41	oob_ip	string	OOB IP adrese.	39 rakstzīmes
42	oob_netmask	string	OOB hosta apakštīkla maska.	39 rakstzīmes
43	oob_router	string	OOB maršrutētājs.	39 rakstzīmes
5	os	string	OS nosaukums.	128 rakstzīmes
6	os_full	string	Detalizēts OS nosaukums.	255 rakstzīmes
7	os_short	string	Saīsināts OS nosaukums.	128 rakstzīmes
61	poc_1_cell	string	Primārās POC mobilā tālruņa numurs.	64 rakstzīmes
58	poc_1_email	string	Primārā e-pasta adrese.	128 rakstzīmes
57	poc_1_name	string	Primārās POC vārds.	128 rakstzīmes
63	poc_1_notes	string	Primārās POC piezīmes.	65535 rakstzīmes SQL datubāzēm
59	poc_1_phone_a	string	Primārās POC tālrunis A.	64 rakstzīmes
60	poc_1_phone_b	string	Primārās POC tālrunis B.	64 rakstzīmes
62	poc_1_screen	string	Primārās POC ekrānvārds.	64 rakstzīmes
68	poc_2_cell	string	Sekundārās POC mobilā tālruņa numurs.	64 rakstzīmes
65	poc_2_email	string	Sekundārā e-pasta adrese.	128 rakstzīmes
64	poc_2_name	string	Sekundārās POC vārds.	128 rakstzīmes
70	poc_2_notes	string	Sekundārās POC piezīmes.	65535 rakstzīmes SQL datubāzēm
66	poc_2_phone_a	string	Sekundārās POC tālrunis A.	64 rakstzīmes
67	poc_2_phone_b	string	Sekundārās POC tālrunis B.	64 rakstzīmes
69	poc_2_screen	string	Sekundārās POC ekrānvārds.	64 rakstzīmes
8	serialno_a	string	Sērijas numurs A.	64 rakstzīmes
9	serialno_b	string	Sērijas numurs B.	64 rakstzīmes
48	site_address_a	string	Vietnes adrese A.	128 rakstzīmes
49	site_address_b	string	Vietnes adrese B.	128 rakstzīmes
50	site_address_c	string	Vietnes adrese C.	128 rakstzīmes
51	site_city	string	Vietnes pilsēta.	128 rakstzīmes
53	site_country	string	Vietnes valsts.	64 rakstzīmes
56	site_notes	string	Vietnes piezīmes.	65535 rakstzīmes SQL datubāzēm
55	site_rack	string	Vietnes statņa atrašanās vieta.	128 rakstzīmes
52	site_state	string	Vietnes štats/reģions.	64 rakstzīmes
54	site_zip	string	Vietnes ZIP/pasta indekss.	64 rakstzīmes
16	software	string	Programmatūra.	255 rakstzīmes
18	software_app_a	string	Programmatūras lietotne A.	64 rakstzīmes

ID	Īpašība	Type	Apraksts	Maksimālais garums
19	software_app_b	string	Programmatūras lietotne B.	64 rakstzīmes
20	software_app_c	string	Programmatūras lietotne C.	64 rakstzīmes
21	software_app_d	string	Programmatūras lietotne D.	64 rakstzīmes
22	software_app_e	string	Programmatūras lietotne E.	64 rakstzīmes
17	software_full	string	Programmatūras informācija.	65535 rakstzīmes SQL datubāzēm
10	tag	string	Birka.	64 rakstzīmes
1	type	string	Tips.	64 rakstzīmes
2	type_full	string	Tipa informācija.	64 rakstzīmes
35	url_a	string	URL A.	2048 rakstzīmes
36	url_b	string	URL B.	2048 rakstzīmes
37	url_c	string	URL C.	2048 rakstzīmes
31	vendor	string	Piegādātājs.	64 rakstzīmes

Hosta tags

Hosta taga objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Hosta taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Hosta taga vērtība. Īpašības darbība: - *tikai lasāms*.
automatic	integer	Hosta taga tips. Iespējamās vērtības: 0 - *(noklusējums)* manuāls (tags izveidots lietotāja); 1 - automātisks (tags izveidots ar zema līmeņa atklāšanu) Īpašības darbība: - *tikai lasāms*
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

host.create

Apraksts

`object host.create(object/array hosts)`

Šī metode ļauj izveidot jaunus hostus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Hosti, ko izveidot.

Papildus standarta hosta īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Hostu grupas, kurām pievienot hostu. Hostu grupām jābūt definētai tikai `groupid` īpašībai. Parametra darbība: - *obligāts*
interfaces	object/array	Saskarnes, kas tiks izveidotas hostam.
tags	object/array	Hosta birkas.
templates	object/array	Veidnes, kas tiks piesaistītas hostam. Veidnēm jābūt definētai tikai `templateid` īpašībai.
macros	object/array	Lietotāja makrosi, kas tiks izveidoti hostam.
inventory	object	Hosta inventarizācijas īpašības.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hostu ID īpašībā `hostids`. Atgriezto ID secība atbilst nodoto hostu secībai.

Piemēri

Hosta izveide

Izveidojiet hostu ar nosaukumu "Linux server" ar IP saskarni un tagiem, pievienojiet to grupai, piesaistiet tam veidni un iestatiet MAC adreses hosta inventārā.

Pieprasījums:

```
{
    "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": "platuma, garuma un augstuma koordinātas"
            }
        ],
        "inventory_mode": 0,
        "inventory": {
            "macaddress_a": "01234",
            "macaddress_b": "56768"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "107819"
        ]
    },
    "id": 1
}
```

Hosta izveide ar SNMP saskarni

Izveidojiet hostu ar nosaukumu ”SNMP host” ar SNMPv3 saskarni un detalizētu informāciju.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10658"
        ]
    },
    "id": 1
}
```

Hosta izveide ar PSK šifrēšanu

Izveidojiet hostu ar nosaukumu "PSK host", kuram ir konfigurēta PSK šifrēšana. Ņemiet vērā, ka hostam jābūt iepriekš konfigurētam PSK izmantošanai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Hosta, ko uzrauga starpniekserveris, izveide

Izveidojiet hostu, ko uzraudzīs starpniekserveris ar ID "1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
```

```
        "host": "Host monitored by proxy",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 1,
        "proxyid": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10591"
        ]
    },
    "id": 1
}
```

Hosta, ko uzrauga starpniekserveru grupa, izveide

Izveidojiet hostu, ko uzraudzīs starpniekserveru grupa ar ID ”1”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.create",
    "params": {
        "host": "Host monitored by proxy group",
        "groups": [
            {
                "groupid": "2"
            }
        ],
        "monitored_by": 2,
        "proxy_groupid": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10592"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Hostu grupa
- Veidne
- Lietotāja makross
- Hosta saskarne
- Hosta inventārs
- Hosta birka
- Starpniekserveris

- Starpniekserveru grupa

Avots

CHost::create() failā *ui/include/classes/api/services/CHost.php*.

host.delete

Apraksts

`object host.delete(array hosts)`

Šī metode ļauj dzēst hostus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo hostu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hostu ID zem `hostids` īpašības.

Piemēri

Vairāku hostu dzēšana

Dzēsiet divus hostus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Avots

CHost::delete() failā *ui/include/classes/api/services/CHost.php*.

host.get

Apraksts

`integer/array host.get(object parameters)`

Šī metode ļauj iegūt hostus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai hostus, kas pieder norādītajām grupām.
dserviceids	ID/array	Atgriezt tikai hostus, kas ir saistīti ar norādītajiem atklātajiem servisiem.
graphids	ID/array	Atgriezt tikai hostus, kuriem ir norādītie grafiki.
hostids	ID/array	Atgriezt tikai hostus ar norādītajiem hostu ID.
httptestids	ID/array	Atgriezt tikai hostus, kuriem ir norādītās tīmekļa pārbaudes.
interfaceids	ID/array	Atgriezt tikai hostus, kas izmanto norādītās saskarnes.
itemids	ID/array	Atgriezt tikai hostus, kuriem ir norādītie vienumi.
maintenanceids	ID/array	Atgriezt tikai hostus, kurus ietekmē norādītās apkopes.
monitored_hosts	flag	Atgriezt tikai uzraudzītos hostus.
proxyids	ID/array	Atgriezt tikai hostus, kurus uzrauga norādītie starpniekserveri.
proxy_groupids	ID/array	Atgriezt tikai hostus, kurus uzrauga norādītās starpniekserveru grupas.
templated_hosts	flag	Atgriezt gan hostus, gan veidnes.
templateids	ID/array	Atgriezt tikai hostus, kas ir sasaistīti ar norādītajām veidnēm.
triggerids	ID/array	Atgriezt tikai hostus, kuriem ir norādītie trigeri.
with_items	flag	Atgriezt tikai hostus, kuriem ir vienumi. Pārraksta parametrus `with_monitored_items` un `with_simple_graph_items`.
with_item_prototypes	flag	Atgriezt tikai hostus, kuriem ir vienumu prototipi. Pārraksta parametru `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	flag	Atgriezt tikai hostus, kuriem ir vienumu prototipi, kas ir iespējoti izveidei un kuriem ir skaitlisks informācijas tips.
with_graphs	flag	Atgriezt tikai hostus, kuriem ir grafiki.
with_graph_prototypes	flag	Atgriezt tikai hostus, kuriem ir grafiku prototipi.
with_httptests	flag	Atgriezt tikai hostus, kuriem ir tīmekļa pārbaudes. Pārraksta parametru `with_monitored_httptests`.
with_monitored_httptests	flag	Atgriezt tikai hostus, kuriem ir iespējotas tīmekļa pārbaudes.
with_monitored_items	flag	Atgriezt tikai hostus, kuriem ir iespējoti vienumi. Pārraksta parametru `with_simple_graph_items`.
with_monitored_triggers	flag	Atgriezt tikai hostus, kuriem ir iespējoti trigeri. Visiem vienumiem, kas tiek izmantoti trigerī, arī jābūt iespējotiem.
with_simple_graph_items	flag	Atgriezt tikai hostus, kuriem ir vienumi ar skaitlisku informācijas tipu.
with_triggers	flag	Atgriezt tikai hostus, kuriem ir trigeri. Pārraksta parametru `with_monitored_triggers`.
withProblemsSuppressed	boolean	Ja iestatīts uz `true`, atgriezt tikai hostus ar apspiestām problēmām.
evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
severities	integer/array	Atgriezt hostus, kuriem ir tikai problēmas ar norādītajām nopietnības pakāpēm. Piemērojams tikai tad, ja problēmas objekts ir trigeris.

Parametrs	Tips	Apraksts
tags	object/array	Atgriezt tikai hostus ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus hostus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
inheritedTags	boolean	Atgriezt hostus, kuriem norādītās `tags` ir arī visās ar tiem sasaistītajās veidnēs. Iespējamās vērtības: `true` - sasaistītajām veidnēm arī jābūt ar norādītajām `tags`; `false` - *(noklusējums)* sasaistīto veidņu birkas tiek ignorētas.
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar hosta atklāšanas objekta datiem. Hosta atklāšanas objekts sasaista atklātu hostu ar hosta prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `host` - `(string)` hosta ID; `parent_hostid` - `(string)` hosta prototipa ID, no kura hosts tika izveidots; `status` - `(int)` hosta atklāšanas statuss: 0 - *(noklusējums)* hosts ir atklāts, 1 - hosts vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad hosts, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad hosts, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` indikators, vai hosts tika atspējots ar LLD kārtulu vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD kārtulu.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar zema līmeņa atklāšanas kārtulu, kas izveidoja hostu (no hosta prototipa VMware uzraudzībā).
selectDiscoveryRules	query	Atgriezt `discoveryRules` īpašību ar hosta LLD kārtulām. Atbalsta `count`.
selectGraphs	query	Atgriezt `graphs` īpašību ar hosta grafikiem. Atbalsta `count`.
selectHostGroups	query	Atgriezt `hostgroups` īpašību ar to hostu grupu datiem, kurām hosts pieder.
selectHttpTests	query	Atgriezt `httpTests` īpašību ar hosta tīmekļa scenārijiem. Atbalsta `count`.
selectInterfaces	query	Atgriezt `interfaces` īpašību ar hosta saskarnēm. Atbalsta `count`.
selectInventory	query	Atgriezt `inventory` īpašību ar hosta inventāra datiem.
selectItems	query	Atgriezt `items` īpašību ar hosta vienumiem. Atbalsta `count`.
selectMacros	query	Atgriezt `macros` īpašību ar hosta makrosiem.

Parametrs	Tips	Apraksts
selectParentTemplates	query	Atgriezt `parentTemplates` īpašību ar veidnēm, ar kurām hosts ir sasaistīts. Papildus Veidnes objekta laukiem tā satur `link_type` - `(integer)` veidu, kādā veidne ir sasaistīta ar hostu. Iespējamās vērtības: 0 - *(noklusējums)* sasaistīta manuāli; 1 - automātiski sasaistīta ar LLD. Atbalsta `count`.
selectDashboards	query	Atgriezt `dashboards` īpašību. Atbalsta `count`.
selectTags	query	Atgriezt `tags` īpašību ar hosta birkām.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas ir visās veidnēs, kuras ir sasaistītas ar hostu.
selectTriggers	query	Atgriezt `triggers` īpašību ar hosta trigeriem. Atbalsta `count`.
selectValueMaps	query	Atgriezt `valuemaps` īpašību ar hosta vērtību kartēm.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: Host interface īpašības.
limitSelects	integer	Ierobežo apakšatlases vaicājumu atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectParentTemplates` - rezultāti tiks kārtoti pēc `host`; `selectInterfaces`; `selectItems` - kārtoti pēc `name`; `selectDiscoveryRules` - kārtoti pēc `name`; `selectTriggers` - kārtoti pēc `description`; `selectGraphs` - kārtoti pēc `name`; `selectDashboards` - kārtoti pēc `name`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir virknes, ko meklēt. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalsta tikai `string` un `text` datu tipa īpašības. Atbalsta papildu īpašības: Host interface īpašības.
searchInventory	object	Atgriezt hostus, kuriem ir inventāra dati, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir virknes, ko meklēt. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalsta tikai `string` un `text` datu tipa īpašības.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	

Parametrs	Tips	Apraksts
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectDiscoveries	query	Atgriezt `discoveries` īpašību ar hosta zema līmeņa atklāšanas kārtulām. Atbalsta `count`. Šis vaicājums ir **novecojis**, lūdzu, izmantojiet `selectDiscoveryRules`.
selectHostDiscovery	query	Atgriezt `hostDiscovery` īpašību ar hosta atklāšanas objekta datiem. Šis vaicājums ir **novecojis**, lūdzu, izmantojiet `selectDiscoveryData`.

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Datu izgūšana pēc nosaukuma

Izgūstiet visus datus par diviem hostiem ar nosaukumiem ”Zabbix server” un ”Linux server”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "filter": {
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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": "Zabbix uzraudzības serveris.",
            "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
}
```


Hostu grupu izgūšana

Izgūstiet hostu grupas, kuru dalībnieks ir hosts ”Zabbix server”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectHostGroups": "extend",
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Atbilde:


```
{
    "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
}
```


Saistīto veidņu izgūšana

Izgūstiet hostam ”10084” saistīto veidņu ID un nosaukumus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectParentTemplates": [
            "templateid",
            "name"
        ],
        "hostids": "10084"
    },
    "id": 1
}
```

Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "parentTemplates": [
                {
                    "name": "Linux",
                    "templateid": "10001"
                },
                {
                    "name": "Zabbix Server",
                    "templateid": "10047"
                }
            ]
        }
    ],
    "id": 1
```

```
}
```

Hostu izgūšana pēc veidnes

Izgūstiet hostus, kuriem ir piesaistīta veidne ”10001” (*Linux by Zabbix aģents*).

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid", "name"],
        "templateids": "10001"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Meklēšana pēc hosta inventāra datiem

Iegūstiet hostus, kuru hosta inventāra lauks ”OS” satur ”Linux”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": [
            "host"
        ],
        "selectInventory": [
            "os"
        ],
        "searchInventory": {
            "os": "Linux"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10084",
            "host": "Zabbix server",
            "inventory": {
                "os": "Linux Ubuntu"
            }
        },
        {
            "hostid": "10107",
            "host": "Linux server",
            "inventory": {
                "os": "Linux Mint"
            }
        }
    ],
    "id": 1
}
```

Meklēšana pēc hostu tagiem

Iegūstiet hostus, kuriem tags ”host-name” ir vienāds ar ”linux-server”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "linux-server",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10085",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "linux-server",
                    "automatic": "0"
                },
                {
                    "tag": "os",
                    "value": "rhel-7",
                    "automatic": "0"
                }
            ]
```

```
        }
    ],
    "id": 1
}
```

Iegūstiet hostus, kuriem šie tagi ir ne tikai hosta līmenī, bet arī to piesaistītajās vecākveidnēs.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "tags": [
            {
                "tag": "os",
                "value": "rhel-7",
                "operator": 1
            }
        ],
        "inheritedTags": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10623",
            "name": "PC room 1"
        },
        {
            "hostid": "10601",
            "name": "Office"
        }
    ],
    "id": 1
}
```

Iegūstiet hostus, izslēdzot tos, kuriem ir tags ”component”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "component",
                "value": "",
                "operator": 5
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Hosta meklēšana ar birkām un veidnes birkām

Iegūstiet hostu ar birkām un visām birkām, kas ir saistītas ar vecākveidnēm.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "hostids": 10502,
        "selectTags": ["tag", "value"],
        "selectInheritedTags": ["tag", "value"]
    },
    "id": 1
}
```

Atbilde:


```
{
    "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
}
```

Hostu meklēšana pēc problēmas nopietnības pakāpes

Iegūstiet hostus, kuriem ir ”Disaster” problēmas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": 5
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "10160",
            "name": "Zabbix server"
        }
    ],
    "id": 1
}
```

Iegūstiet hostus, kuriem ir ”Average” un ”High” problēmas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["name"],
        "severities": [3, 4]
    },
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostid": "20170",
            "name": "Database"
        },
        {
            "hostid": "20183",
            "name": "workstation"
        }
    ],
    "id": 1
}
```

Hosta vienumu izgūšana

Izgūstiet vienumus hostam "Zabbix server". Pieprasījums ir ierobežots ar vienuma īpašībām `itemid`, `name` un `status`. Lielā atbildes apjoma dēļ piemērā ir parādīta tikai daļa no vienumiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "output": ["hostid"],
        "selectItems": ["itemid","name","status"],
        "filter": {
            "host": [
                "Zabbix server"
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
  "jsonrpc": "2.0",
  "result": [
    {
      "hostid": "10084",
      "items": [
        {
          "itemid": "42227",
          "name": "Darbojošā Zabbix aģenta hosta nosaukums",
          "status": "0"
        },
        {
          "itemid": "42237",
          "name": "Zabbix aģenta ping",
          "status": "0"
        },
        {
          "itemid": "42250",
          "name": "Darbojošā Zabbix aģenta versija",
          "status": "0"
        },
        {
```

```
          "itemid": "42251",
          "name": "Maksimālais atvērto failu deskriptoru skaits",
          "status": "0"
        },
        {
          "itemid": "42252",
          "name": "Maksimālais procesu skaits",
          "status": "0"
        },
        {
          "itemid": "69869",
          "name": "Saskarne enp0s3: atmestās ienākošās paketes",
          "status": "0"
        },
        {
          "itemid": "69870",
          "name": "Saskarne enp0s3: ienākošās paketes ar kļūdām",
          "status": "0"
        },
        {
          "itemid": "69871",
          "name": "Saskarne enp0s3: saņemtie biti",
          "status": "0"
        },
        {
          "itemid": "69872",
          "name": "Saskarne enp0s3: atmestās izejošās paketes",
          "status": "0"
        },
        {
          "itemid": "69873",
          "name": "Saskarne enp0s3: izejošās paketes ar kļūdām",
          "status": "0"
        },
        {
          "itemid": "69874",
          "name": "Saskarne enp0s3: nosūtītie biti",
          "status": "0"
        },
        {
          "itemid": "42253",
          "name": "Procesu skaits",
          "status": "0"
        },
        {
          "itemid": "42254",
          "name": "Darbojošos procesu skaits",
          "status": "0"
        },
        {
          "itemid": "42255",
          "name": "Sistēmas sāknēšanas laiks",
          "status": "0"
        }
      ]
    }
  ],
  "id": 1
}
```


Skatiet arī

- Hostu grupa

- Veidne
- Lietotāja makross
- Hosta saskarne
- Starpniekserveris
- Starpniekserveru grupa

Avots

CHost::get() failā *ui/include/classes/api/services/CHost.php*.

host.massadd

Apraksts

`object host.massadd(object parameters)`

Šī metode ļauj vienlaikus pievienot vairākus saistītus objektus visiem norādītajiem hostiem.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo hostu ID un objektus, ko pievienot visiem hostiem.

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hosts	object/array	Hosti, kas jāatjaunina. Hostiem jābūt definētai tikai `hostid` īpašībai. Parametra darbība: - *obligāts*
groups	object/array	Hostu grupas, ko pievienot norādītajiem hostiem. Hostu grupām jābūt definētai tikai `groupid` īpašībai.
interfaces	object/array	Hostu saskarnes, kas jāizveido norādītajiem hostiem.
macros	object/array	Lietotāja makrosi, kas jāizveido norādītajiem hostiem.
templates	object/array	Veidnes, ko piesaistīt norādītajiem hostiem. Veidnēm jābūt definētai tikai `templateid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu ID zem `hostids` īpašības.

Piemēri

Makrosu pievienošana

Pievienojiet divus jaunus makrosus diviem hostiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.massadd",
    "params": {
        "hosts": [
            {
                "hostid": "10160"
            },
            {
                "hostid": "10167"
```

```
            }
        ],
        "macros": [
            {
                "macro": "{$TEST1}",
                "value": "MACROTEST1"
            },
            {
                "macro": "{$TEST2}",
                "value": "MACROTEST2",
                "description": "Testa apraksts"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10160",
            "10167"
        ]
    },
    "id": 1
}
```

Skatiet arī

- host.update
- Hostu grupa
- Veidne
- Lietotāja makross
- Hosta saskarne

Avots

CHost::massAdd() failā *ui/include/classes/api/services/CHost.php*.

host.massremove

Apraksts

`object host.massremove(object parameters)`

Šī metode ļauj noņemt saistītos objektus no vairākiem hostiem.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo hostu ID un objektus, kas jānoņem.

Parametrs	Type	Apraksts
hostids	ID/array	Atjauninámo hostu ID. Parametra darbība: - *obligāts*
groupids	ID/array	To hostu grupu ID, no kurām jānoņem norādītie hosti.

Parametrs	Type	Apraksts
interfaces	object/array	Hostu saskarnes, kas jānoņem no norādītajiem hostiem. Hosta saskarnes objektam jābūt definētām tikai `ip`, `dns` un `port` īpašībām.
macros	string/array	Lietotāja makrosi, kas jādzēš no norādītajiem hostiem.
templateids	ID/array	To veidņu ID, kuras jāatsaista no norādītajiem hostiem.
templateids_clear	ID/array	To veidņu ID, kuras jāatsaista un jānotīra no norādītajiem hostiem.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu ID zem `hostids` īpašības.

Piemēri

Veidņu atsaistīšana

Atsaistiet veidni no diviem hostiem un dzēsiet visas no veidnes mantotās entītijas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.massremove",
    "params": {
        "hostids": ["69665", "69666"],
        "templateids_clear": "325"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "69665",
            "69666"
        ]
    },
    "id": 1
}
```

Skatiet arī

- host.update
- Lietotāja makross
- Hosta saskarne

Avots

CHost::massRemove() failā *ui/include/classes/api/services/CHost.php*.

host.update

Apraksts

`object host.update(object/array hosts)`

Šī metode ļauj atjaunināt esošos hostus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

(object/array) Atjaunināmie hosta rekvizīti.

Katram hostam jābūt definētam rekvizītam hostid, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai norādītie rekvizīti, visi pārējie paliks nemainīti.

Tomēr ņemiet vērā, ka, atjauninot hosta tehnisko nosaukumu, tiks atjaunināts arī hosta redzamais nosaukums (ja tas nav norādīts atsevišķi), izmantojot hosta tehniskā nosaukuma vērtību.

Papildus standarta hosta rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Hostu grupas, ar kurām aizstāt pašreizējās hostu grupas, kurām hosts pieder. Visu hostu grupu saites, kas pieprasījumā nav norādītas, tiks noņemtas. Hostu grupām jābūt definētam tikai rekvizītam groupid.
interfaces	object/array	Hostu saskarnes, ar kurām aizstāt pašreizējās hostu saskarnes. Visas saskarnes, kas pieprasījumā nav norādītas, tiks noņemtas.
tags	object/array	Hostu birkas, ar kurām aizstāt pašreizējās hostu birkas. Visas birkas, kas pieprasījumā nav norādītas, tiks noņemtas.
inventory	object	Hosta inventarizācijas rekvizīti.
macros	object/array	Lietotāja makrosi, ar kuriem aizstāt pašreizējos lietotāja makrosus. Visi makrosi, kas pieprasījumā nav norādīti, tiks noņemti.
templates	object/array	Veidnes, ar kurām aizstāt pašlaik piesaistītās veidnes. Visām veidnēm, kas pieprasījumā nav norādītas, tiks tikai noņemta saite. Veidnēm jābūt definētam tikai rekvizītam templateid.
templates_clear	object/array	Veidnes, kurām noņemt saiti un notīrīt no hosta. Veidnēm jābūt definētam tikai rekvizītam templateid.

Atgrieztās vērtības

(object) Atgriež objektu, kas satur atjaunināto hostu ID zem hostids īpašības.

Piemēri

Hosta iespējošana

Iespējojiet hosta uzraudzību, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Vairāku hostu iespējošana

Iespējojiet divu hostu uzraudzību, tas ir, iestatiet to statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": [
        {
            "hostid": "10127",
            "status": 0
        },
        {
            "hostid": "10128",
            "status": 0
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10127",
            "10128"
        ]
    },
    "id": 1
}
```

Veidņu atsaistīšana

Atsaistīt un notīrīt divas veidnes no hosta.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "templates_clear": [
            {
                "templateid": "10124"
            },
            {
                "templateid": "10125"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Hosta makrosu atjaunināšana

Aizstājiet visus hosta makrosus ar diviem jauniem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10126",
        "macros": [
            {
                "macro": "{$PASS}",
                "value": "password"
            },
            {
                "macro": "{$DISC}",
                "value": "sda",
                "description": "Atjaunināts apraksts"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10126"
        ]
    },
    "id": 1
}
```

Hosta inventāra atjaunināšana

Mainiet inventāra režīmu un pievienojiet atrašanās vietu

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "inventory_mode": 0,
        "inventory": {
            "location": "Latvia, Riga"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Hosta birku atjaunināšana

Aizstājiet visas hosta birkas ar jaunu birku.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "tags": {
            "tag": "os",
            "value": "rhel-7"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Atklāto hostu makrosu atjaunināšana

Pārveidojiet atklāšanas kārtulas izveidoto ”automatic” makrosu par ”manual” un nomainiet tā vērtību uz ”new-value”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.update",
    "params": {
        "hostid": "10387",
        "macros": {
            "hostmacroid": "5541",
            "value": "new-value",
            "automatic": "0"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10387"
        ]
    },
    "id": 1
}
```

Hosta šifrēšanas atjaunināšana

Atjauniniet hostu ”10590”, lai tas izmantotu tikai PSK šifrēšanu savienojumiem no hosta uz Zabbix serveri, un nomainiet PSK identitāti un PSK atslēgu. Ņemiet vērā, ka hostam jābūt iepriekš konfigurētam PSK izmantošanai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostids": [
            "10590"
        ]
    },
    "id": 1
}
```

Skatiet arī

- host.massadd
- host.massremove
- Hostu grupa
- Veidne
- Lietotāja makross
- Hosta saskarne
- Hosta inventārs
- Hosta tags
- Starpniekserveris
- Starpniekserveru grupa

Avots

CHost::update() failā *ui/include/classes/api/services/CHost.php*.

Hostu grupa

Šī klase ir paredzēta darbam ar hostu grupām.

Objektu atsauces:

- Hostu grupa

Pieejamās metodes:

- hostgroup.create - izveidot jaunas hostu grupas
- hostgroup.delete - dzēst hostu grupas
- hostgroup.get - izgūt hostu grupas
- hostgroup.massadd - pievienot saistītos objektus hostu grupām
- hostgroup.massremove - noņemt saistītos objektus no hostu grupām
- hostgroup.propagate - izplatīt atļaujas un birku filtrus hostu grupu apakšgrupām
- hostgroup.update - atjaunināt hostu grupas

Hostu grupas objekts

Tālāk norādītie objekti ir tieši saistīti ar `hostgroup` API.

Hostu grupa

Hostu grupas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
groupid	ID	Hostu grupas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Hostu grupas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
flags	integer	Hostu grupas izcelsme. Iespējamās vērtības: 0 - parasta hostu grupa; 4 - atklāta hostu grupa. Īpašības darbība: - *tikai lasāms*
uuid	string	Universāli unikāls identifikators, ko izmanto, lai sasaistītu importētās hostu grupas ar jau esošajām. Tiek ģenerēts automātiski, ja nav norādīts.

hostgroup.create

Apraksts

`object hostgroup.create(object/array hostGroups)`

Šī metode ļauj izveidot jaunas hostu grupas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās hostu grupas.

Metode pieņem hostu grupas ar standarta hostu grupu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hostu grupu ID īpašībā `groupids`. Atgriezto ID secība atbilst nodoto hostu grupu secībai.

Piemēri

Hostu grupas izveide

Izveidojiet hostu grupu ar nosaukumu "Linux servers".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.create",
    "params": {
        "name": "Linux servers"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "groupids": [
            "107819"
        ]
    },
    "id": 1
}
```

Avots

CHostGroup::create() failā *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.delete

Apraksts

`object hostgroup.delete(array hostGroupIds)`

Šī metode ļauj dzēst hostu grupas.

Hostu grupu nevar dzēst, ja:

- tā satur hostus, kas pieder tikai šai grupai;
- tā ir atzīmēta kā iekšēja;
- tā tiek izmantota hosta prototipā;
- tā tiek izmantota globālajā skriptā;
- tā tiek izmantota korelācijas nosacījumā.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Papildinformāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo hostu grupu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hostu grupu ID īpašībā `groupids`.

Piemēri

Vairāku hostu grupu dzēšana

Dzēsiet divas hostu grupas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.delete",
    "params": [
        "107824",
        "107825"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107824",
            "107825"
        ]
    },
    "id": 1
}
```

Avots

CHostGroup::delete() failā *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.get

Apraksts

`integer/array hostgroup.get(object parameters)`

Metode ļauj izgūt hostu grupas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
graphids	ID/array	Atgriezt tikai tās hostu grupas, kurās ir hosti ar norādītajiem grafikiem.
groupids	ID/array	Atgriezt tikai hostu grupas ar norādītajiem hostu grupu ID.
hostids	ID/array	Atgriezt tikai tās hostu grupas, kurās ir norādītie hosti.
maintenanceids	ID/array	Atgriezt tikai tās hostu grupas, kuras ietekmē norādītās apkopes.
triggerids	ID/array	Atgriezt tikai tās hostu grupas, kurās ir hosti ar norādītajiem trigeriem.
with_graphs	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar grafikiem.
with_graph_prototypes	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar grafiku prototipiem.
with_hosts	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti.
with_httptests	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar tīmekļa pārbaudēm. Aizstāj parametru `with_monitored_httptests`.
with_items	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar vienumiem. Aizstāj parametrus `with_monitored_items` un `with_simple_graph_items`.
with_item_prototypes	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar vienumu prototipiem. Aizstāj parametru `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar vienumu prototipiem, kas ir iespējoti izveidei un kuriem ir skaitlisks informācijas tips.
with_monitored_httptests	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar iespējotām tīmekļa pārbaudēm.
with_monitored_hosts	boolean	Atgriezt tikai tās hostu grupas, kurās ir uzraudzīti hosti.
with_monitored_items	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar iespējotiem vienumiem. Aizstāj parametru `with_simple_graph_items`.
with_monitored_triggers	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar iespējotiem trigeriem. Visiem trigerī izmantotajiem vienumiem arī jābūt iespējotiem.
with_simple_graph_items	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar skaitliskiem vienumiem.
with_triggers	boolean	Atgriezt tikai tās hostu grupas, kurās ir hosti ar trigeriem. Aizstāj parametru `with_monitored_triggers`.
selectDiscoveryRules	query	Atgriezt īpašību `discoveryRules` ar LLD kārtulām, kas atklāja hostu grupu.

Parametrs	Tips	Apraksts
selectDiscoveryData	query	Atgriezt īpašību `discoveryData` ar hostu grupu atklāšanas objektiem. Katrs hostu grupas atklāšanas objekts ir hostu grupas prototips, kas saistīts ar atklāto hostu grupu, un tam ir šādas īpašības: `parent_group_prototypeid` - `(ID)` hostu grupas prototipa ID, no kura tika atklāta hostu grupa; `name` - `(string)` hostu grupas prototipa nosaukums; `ts_delete` - `(timestamp)` laiks, kad hostu grupa, kas vairs netiek atklāta, tiks dzēsta; `status` - `(int)` hostu grupas atklāšanas statuss: 0 - *(noklusējums)* hostu grupa ir atklāta, 1 - hostu grupa vairs netiek atklāta.
selectHostPrototypes	query	Atgriezt īpašību `hostPrototypes` ar hostu prototipiem, kas atklāja šo hostu grupu.
selectHosts	query	Atgriezt īpašību `hosts` ar hostiem, kas pieder hostu grupai. Atbalsta `count`.
limitSelects	integer	Ierobežo apakšatlases vaicājumu atgriezto ierakstu skaitu. Attiecas uz šādiem apakšatlases vaicājumiem: `selectHosts` - rezultāti tiks kārtoti pēc `host`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `groupid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectGroupDiscoveries	query	Atgriezt īpašību `groupDiscoveries` ar hostu grupu atklāšanas objektiem. Katrs hostu grupas atklāšanas objekts ir hostu grupas prototips, kas saistīts ar atklāto hostu grupu. Šis vaicājums ir **novecojis**, lūdzu, tā vietā izmantojiet `selectDiscoveryData`.

Atgrieztās vērtības

`(integer/array)` Atgriež:

- objektu masīvu
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`

Piemēri

Datu izgūšana pēc nosaukuma

Izgūstiet visus datus par divām hostu grupām ar nosaukumiem ”Zabbix servers” un ”Linux servers”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.get",
    "params": {
```

```
        "output": "extend",
        "filter": {
            "name": [
                "Zabbix servers",
                "Linux servers"
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "2",
            "name": "Linux servers",
            "flags": "0",
            "uuid": "dc579cd7a1a34222933f24f52a68bcd8"
        },
        {
            "groupid": "4",
            "name": "Zabbix servers",
            "flags": "0",
            "uuid": "6f6799aa69e844b4b3918f779f2abf08"
        }
    ],
    "id": 1
}
```

Skatiet arī

- Hosts

Avots

CHostGroup::get() failā *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massadd

Apraksts

`object hostgroup.massadd(object parameters)`

Šī metode ļauj vienlaikus pievienot vairākus saistītos objektus visām norādītajām hostu grupām.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjaunināmo hostu grupu ID un objektus, ko pievienot visām hostu grupām.

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Hostu grupas, kas tiks atjauninātas. Hostu grupām jābūt definētai tikai `groupid` īpašībai. Parametra darbība: - *obligāts*

Parametrs	Tips	Apraksts
hosts	object/array	Hosti, ko pievienot visām hostu grupām. Hostiem jābūt definētai tikai `hostid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu grupu ID īpašībā `groupids`.

Piemēri

Hostu pievienošana hostu grupām

Pievienojiet divus hostus hostu grupām ar ID 5 un 6.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "5"
            },
            {
                "groupid": "6"
            }
        ],
        "hosts": [
            {
                "hostid": "30050"
            },
            {
                "hostid": "30001"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Hosts

Avots

CHostGroup::massAdd() failā *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.massremove

Apraksts

`object hostgroup.massremove(object parameters)`

Šī metode ļauj noņemt saistītos objektus no vairākām hostu grupām.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo hostu grupu ID un objektus, kas jānoņem.

Parametrs	Type	Apraksts
groupids	ID/array	Atjauninámo hostu grupu ID. Parametra darbība: - *obligāts*
hostids	ID/array	To hostu ID, kas jānoņem no visām hostu grupām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu grupu ID īpašībā `groupids`.

Piemēri

Hostu noņemšana no hostu grupām

Noņemiet divus hostus no norādītajām hostu grupām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.massremove",
    "params": {
        "groupids": [
            "5",
            "6"
        ],
        "hostids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Avots

CHostGroup::massRemove() failā *ui/include/classes/api/services/CHostGroup.php*.

hostgroup.propagate

Apraksts

`object hostgroup.propagate(object parameters)`

Šī metode ļauj piemērot atļaujas un birku filtrus visām hostu grupas apakšgrupām.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem.
> Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos.
> Plašāku informāciju skatiet [Lietotāju
> lomas](/manual/web_interface/frontend_sections/users/user_roles).

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

|Parametrs|Tips|Apraksts|

|--|--|------|

|groups|object/array|Hostu grupas, ko izplatīt.

Hostu grupām jābūt definētai `groupid` īpašībai.

Parametra darbība:
- *obligāts*|

|permissions|boolean|Iestatiet uz ”true”, lai izplatītu atļaujas.

Parametra darbība:
- *obligāts*, ja `tag_filters` nav iestatīts|

|tag_filters|boolean|Iestatiet uz ”true”, lai izplatītu birku filtrus.

Parametra darbība:
- *obligāts*, ja `permissions` nav iestatīts|

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur propagēto hostu

grupu ID īpašībā `groupids`.

Piemēri

Hostu grupas atļauju un birku filtru izplatīšana uz tās apakšgrupām.

Izplatiet hostu grupas atļaujas un birku filtrus uz tās apakšgrupām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "6"
            }
        ],
        "permissions": true,
        "tag_filters": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "6",
        ]
    },
    "id": 1
}
```

Skatiet arī

- hostgroup.update
- hostgroup.massadd
- Hosts

Avots

CHostGroup::propagate() failā

ui/include/classes/api/services/CHostGroup.php.

hostgroup.update

Apraksts

`object hostgroup.update(object/array hostGroups)`

Šī metode ļauj atjaunināt esošās hostu grupas.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Hostu grupu īpašības, kas jāatjaunina.

Īpašībai `groupid` jābūt definētai katrai hostu grupai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai norādītās īpašības, visas pārējās paliks nemainītas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hostu grupu ID īpašībā `groupids`.

Piemēri

Hostu grupas pārdēvēšana

Pārdēvējiet hostu grupu par "Linux hosts".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": {
        "groupid": "7",
        "name": "Linux hosts"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Vairāku hostu grupu pārdēvēšana

Pārdēvējiet divas hostu grupas, lai to nosaukumos iekļautu reģiona informāciju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "hostgroup.update",
    "params": [
        {
            "groupid": "8",
            "name": "Europe/Linux hosts"
        },
        {
            "groupid": "9",
            "name": "Europe/Windows hosts"
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "8",
            "9"
        ]
    },
    "id": 1
}
```

Avots

CHostGroup::update() failā *ui/include/classes/api/services/CHostGroup.php*.

Housekeeping

Šī klase ir paredzēta darbam ar housekeeping.

Objektu atsauces:

- Housekeeping

Pieejamās metodes:

- housekeeping.get - izgūt housekeeping
- housekeeping.update - atjaunināt housekeeping

Housekeeping objekts

Tālāk norādītie objekti ir tieši saistīti ar `housekeeping` API.

Housekeeping

Iestatījumu objektam ir šādas īpašības.

Īpašība	Type	Apraksts
hk_events_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu notikumiem un brīdinājumiem. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_events_trigger	string	Trigeru datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 365d.

Īpašība	Type	Apraksts
hk_events_service	string	Pakalpojumu datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 1d.
hk_events_internal	string	Iekšējo datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 1d.
hk_events_discovery	string	Tīkla atklāšanas datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 1d.
hk_events_autoreg	string	Automātiskās reģistrācijas datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 1d.
hk_services_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu pakalpojumiem. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_services	string	Pakalpojumu datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 365d.
hk_audit_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu auditam. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_audit	string	Audita datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 31d.
hk_sessions_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu sesijām. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_sessions	string	Sesiju datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 31d.
hk_history_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu vēsturei. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
hk_history_global	integer	Pārrakstīt vienuma vēstures periodu. Iespējamās vērtības: 0 - Nepārrakstīt; 1 - *(noklusējums)* Pārrakstīt.
hk_history	string	Vēstures datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 31d.
hk_trends_mode	integer	Iespējot iekšējo novecojušo datu tīrīšanu tendencēm. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.

Īpašība	Type	Apraksts
hk_trends_global	integer	Pārrakstīt vienuma tendenču periodu. Iespējamās vērtības: 0 - Nepārrakstīt; 1 - *(noklusējums)* Pārrakstīt.
hk_trends	string	Tendenču datu glabāšanas periods. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 365d.
db_extension	string	DB paplašinājuma konfigurācijas karodziņš. Ja šis karodziņš ir iestatīts uz ”timescaledb”, serveris maina savu darbību attiecībā uz novecojušo datu tīrīšanu un vienumu dzēšanu. Īpašības darbība: - *tikai lasāms*
compression_availability	integer	Vai datu saspiešanu atbalsta datubāze (vai tās paplašinājums). Iespējamās vērtības: 0 - Nav pieejams; 1 - Pieejams. Īpašības darbība: - *tikai lasāms*
compression_status	integer	Iespējot TimescaleDB saspiešanu vēsturei un tendencēm. Iespējamās vērtības: 0 - *(noklusējums)* Izslēgts; 1 - Ieslēgts.
compress_older	string	Saspiest vēstures un tendenču ierakstus, kas ir vecāki par norādīto periodu. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 7d.

housekeeping.get

Apraksts

`object housekeeping.get(object parameters)`

Metode ļauj izgūt uzturēšanas objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta tikai vienu parametru.

Parametrs	Type	Apraksts
output	query	Šis parametrs ir aprakstīts atsauces komentārā.

Atgrieztās vērtības

`(object)` Atgriež housekeeping objektu.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "housekeeping.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CHousekeeping ::get() failā *ui/include/classes/api/services/CHousekeeping.php*.

housekeeping.update

Apraksts

`object housekeeping.update(object housekeeping)`

Šī metode ļauj atjaunināt esošos housekeeper iestatījumus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object)` Uzturēšanas īpašības, kas jāatjaunina.

Atgrieztās vērtības

`(array)` Atgriež masīvu ar atjaunināto parametru nosaukumiem.

Piemēri

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        "hk_events_mode",
        "hk_events_trigger",
        "hk_events_internal",
        "hk_events_discovery"
    ],
    "id": 1
}
```

Avots

CHousekeeping::update() failā *ui/include/classes/api/services/CHousekeeping.php*.

Iestatījumi

Šī klase ir paredzēta darbam ar vispārējiem administrēšanas iestatījumiem.

Objektu atsauces:

- Settings

Pieejamās metodes:

- settings.get - izgūt iestatījumus
- settings.update - atjaunināt iestatījumus

Iestatījumu objekts

Tālāk norādītie objekti ir tieši saistīti ar `settings` API.

Iestatījumi

Iestatījumu objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
default_lang	string	Sistēmas noklusējuma valoda. Noklusējums: `en_US`.
default_timezone	string	Sistēmas noklusējuma laika josla. Noklusējums: `system` - sistēmas noklusējuma vērtība. Pilnu atbalstīto laika joslu sarakstu skatiet PHP dokumentācijā.

Īpašība	Tips	Apraksts
default_theme	string	Noklusējuma tēma. Iespējamās vērtības: `blue-theme` - *(noklusējums)* Blue; `blue-classic-theme` - Blue (classic); `dark-theme` - Dark; `dark-classic-theme` - Dark (classic); `hc-light` - High-contrast light; `hc-dark` - High-contrast dark.
search_limit	integer	Meklēšanas un filtrēšanas rezultātu ierobežojums. Noklusējums: 1000.
max_overview_table_size	integer	Maksimālais kolonnu un rindu skaits paneļa Data overview un Trigger overview logrīkos. Noklusējums: 50.
max_in_table	integer	Maksimālais elementu skaits, ko rādīt tabulas šūnā. Noklusējums: 50.
server_check_interval	integer	Rādīt brīdinājumu, ja Zabbix serveris nav pieejams. Iespējamās vērtības: 0 - Nerādīt brīdinājumu; 10 - *(noklusējums)* Rādīt brīdinājumu.
work_period	string	Darba laiks. Noklusējums: 1-5,09:00-18:00.
show_technical_errors	integer	Rādīt tehniskās kļūdas (PHP/SQL) lietotājiem, kuri nav Super admin, un lietotājiem, kuri nav lietotāju grupās ar iespējotu atkļūdošanas režīmu. Iespējamās vērtības: 0 - *(noklusējums)* Nerādīt tehniskās kļūdas; 1 - Rādīt tehniskās kļūdas.
history_period	string	Maksimālais periods, kurā attēlot vēstures datus paneļa logrīkos Latest data, Web un Data overview. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 24h.
period_default	string	Laika filtra noklusējuma periods. Pieņem sekundes un laika vienību ar sufiksu, atbalstot arī mēnešus un gadus (30s, 1m, 2h, 1d, 1M, 1y). Noklusējums: 1h.
max_period	string	Maksimālais periods laika filtram. Pieņem sekundes un laika vienību ar sufiksu, atbalstot arī mēnešus un gadus (30s, 1m, 2h, 1d, 1M, 1y). Noklusējums: 2y.
severity_color_0	string	Krāsa nopietnības līmenim "Not classified" heksadecimāla krāsas koda formātā. Noklusējums: 97AAB3.
severity_color_1	string	Krāsa nopietnības līmenim "Information" heksadecimāla krāsas koda formātā. Noklusējums: 7499FF.
severity_color_2	string	Krāsa nopietnības līmenim "Warning" heksadecimāla krāsas koda formātā. Noklusējums: FFC859.

Īpašība	Tips	Apraksts
severity_color_3	string	Krāsa nopietnības līmenim ”Average” heksadecimāla krāsas koda formātā. Noklusējums: FFA059.
severity_color_4	string	Krāsa nopietnības līmenim ”High” heksadecimāla krāsas koda formātā. Noklusējums: E97659.
severity_color_5	string	Krāsa nopietnības līmenim ”Disaster” heksadecimāla krāsas koda formātā. Noklusējums: E45959.
severity_name_0	string	Nosaukums nopietnības līmenim ”Not classified”. Noklusējums: Not classified.
severity_name_1	string	Nosaukums nopietnības līmenim ”Information”. Noklusējums: Information.
severity_name_2	string	Nosaukums nopietnības līmenim ”Warning”. Noklusējums: Warning.
severity_name_3	string	Nosaukums nopietnības līmenim ”Average”. Noklusējums: Average.
severity_name_4	string	Nosaukums nopietnības līmenim ”High”. Noklusējums: High.
severity_name_5	string	Nosaukums nopietnības līmenim ”Disaster”. Noklusējums: Disaster.
custom_color	integer	Izmantot pielāgotas notikumu statusu krāsas. Iespējamās vērtības: 0 - *(noklusējums)* Neizmantot pielāgotas notikumu statusu krāsas; 1 - Izmantot pielāgotas notikumu statusu krāsas.
ok_period	string	Attēlot OK trigeru periodu. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 5m.
blink_period	string	Mirgošanas periods, mainoties trigeris stāvoklim. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 2m.
problem_unack_color	string	Krāsa neapstiprinātiem PROBLEM notikumiem heksadecimāla krāsas koda formātā. Noklusējums: CC0000.
problem_ack_color	string	Krāsa apstiprinātiem PROBLEM notikumiem heksadecimāla krāsas koda formātā. Noklusējums: CC0000.
ok_unack_color	string	Krāsa neapstiprinātiem RESOLVED notikumiem heksadecimāla krāsas koda formātā. Noklusējums: 009900.
ok_ack_color	string	Krāsa apstiprinātiem RESOLVED notikumiem heksadecimāla krāsas koda formātā. Noklusējums: 009900.

Īpašība	Tips	Apraksts
problem_unack_style	integer	Mirgošana neapstiprinātiem PROBLEM notikumiem. Iespējamās vērtības: 0 - Nerādīt mirgošanu; 1 - *(noklusējums)* Rādīt mirgošanu.
problem_ack_style	integer	Mirgošana apstiprinātiem PROBLEM notikumiem. Iespējamās vērtības: 0 - Nerādīt mirgošanu; 1 - *(noklusējums)* Rādīt mirgošanu.
ok_unack_style	integer	Mirgošana neapstiprinātiem RESOLVED notikumiem. Iespējamās vērtības: 0 - Nerādīt mirgošanu; 1 - *(noklusējums)* Rādīt mirgošanu.
ok_ack_style	integer	Mirgošana apstiprinātiem RESOLVED notikumiem. Iespējamās vērtības: 0 - Nerādīt mirgošanu; 1 - *(noklusējums)* Rādīt mirgošanu.
url	string	Lietotāja saskarnes URL.
discovery_groupid	ID	Tās hostu grupas ID, kurā automātiski tiks ievietoti atklātie hosti.
default_inventory_mode	integer	Noklusējuma hosta inventarizācijas režīms. Iespējamās vērtības: -1 - *(noklusējums)* Atspējots; 0 - Manuāls; 1 - Automātisks.
alert_usrgrpid	ID	Tās lietotāju grupas ID, kurai tiks sūtīts datubāzes nepieejamības trauksmes ziņojums. Ja iestatīts uz ”0”, trauksmes ziņojums netiks nosūtīts.
snmptrap_logging	integer	Reģistrēt neatbilstošus SNMP trapus. Iespējamās vērtības: 0 - Nereģistrēt neatbilstošus SNMP trapus; 1 - *(noklusējums)* Reģistrēt neatbilstošus SNMP trapus.
login_attempts	integer	Neveiksmīgu pieteikšanās mēģinājumu skaits, pēc kura pieteikšanās forma tiks bloķēta. Noklusējums: 5.
login_block	string	Laika intervāls, kurā pieteikšanās forma tiks bloķēta, ja neveiksmīgu pieteikšanās mēģinājumu skaits pārsniedz laukā login_attempts definēto vērtību. Pieņem sekundes un laika vienību ar sufiksu. Noklusējums: 30s.
validate_uri_schemes	integer	Validēt URI shēmas. Iespējamās vērtības: 0 - Nevalidēt; 1 - *(noklusējums)* Validēt.
uri_valid_schemes	string	Derīgās URI shēmas. Noklusējums: http,https,ftp,file,mailto,tel,ssh.
x_frame_options	string	X-Frame-Options HTTP galvene. Noklusējums: SAMEORIGIN.

Īpašība	Tips	Apraksts
iframe_sandboxing_enabled	integer	Izmantot iframe smilškastes režīmu. Iespējamās vērtības: 0 - Neizmantot; 1 - *(noklusējums)* Izmantot.
iframe_sandboxing_exceptions	string	Iframe smilškastes režīma izņēmumi.
connect_timeout	string	Savienojuma noildze ar Zabbix serveri. Iespējamo vērtību diapazons: 1-30s. Noklusējums: 3s. Īpašības darbība: - *required*
socket_timeout	string	Tīkla noklusējuma noildze. Iespējamo vērtību diapazons: 1-300s. Noklusējums: 3s. Īpašības darbība: - *required*
media_type_test_timeout	string	Tīkla noildze medija tipa testam. Iespējamo vērtību diapazons: 1-300s. Noklusējums: 65s. Īpašības darbība: - *required*
item_test_timeout	string	Tīkla noildze vienuma testiem. Iespējamo vērtību diapazons: 1-600s. Noklusējums: 60s. Īpašības darbība: - *required*
script_timeout	string	Tīkla noildze skripta izpildei. Iespējamo vērtību diapazons: 1-300s. Noklusējums: 60s. Īpašības darbība: - *required*
report_test_timeout	string	Tīkla noildze ieplānotā pārskata testam. Iespējamo vērtību diapazons: 1-300s. Noklusējums: 60s. Īpašības darbība: - *required*
auditlog_enabled	integer	Vai iespējot audita žurnalēšanu. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.

Īpašība	Tips	Apraksts
auditlog_mode	integer	Vai iespējot servera (System user) veikto zemā līmeņa atklāšanas, tīkla atklāšanas un automātiskās reģistrācijas darbību audita žurnalēšanu. Iespējamās vērtības: 0 - Atspējot; 1 - *(noklusējums)* Iespējot.
ha_failover_delay	string	Pārslēgšanās aizkave sekundēs. Noklusējums: 1m. Īpašības darbība: - *read-only*
geomaps_tile_provider	string	Ģeokartes flīžu nodrošinātājs. Iespējamās vērtības: `OpenStreetMap.Mapnik` - *(noklusējums)* OpenStreetMap Mapnik; `OpenTopoMap` - OpenTopoMap; `Stamen.TonerLite` - Stamen Toner Lite; `Stamen.Terrain` - Stamen Terrain; `USGS.USTopo` - USGS US Topo; `USGS.USImagery` - USGS US Imagery. Atbalsta tukšu virkni, lai norādītu pielāgotas `geomaps_tile_url`, `geomaps_max_zoom` un `geomaps_attribution` vērtības.
geomaps_tile_url	string	Ģeokartes flīžu URL. Īpašības darbība: - *supported* if `geomaps_tile_provider` is set to empty string
geomaps_max_zoom	integer	Ģeokartes maksimālais tālummaiņas līmenis. Iespējamās vērtības: 0-30. Īpašības darbība: - *supported* if `geomaps_tile_provider` is set to empty string
geomaps_attribution	string	Ģeokartes atsauces teksts. Īpašības darbība: - *supported* if `geomaps_tile_provider` is set to empty string
vault_provider	integer	Vault nodrošinātājs. Iespējamās vērtības: 0 - *(noklusējums)* HashiCorp Vault; 1 - CyberArk Vault.
proxy_secrets_provider	integer	Atrisināt slepeno makrosu vērtības, izmantojot: Iespējamās vērtības: 0 - *(noklusējums)* tikai serveris; 1 - serveris un starpniekserveri neatkarīgi.
timeout_zabbix_agent	string	Apstrādei netērēt vairāk par `timeout_*` sekundēm. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: 3s. Noklusējums `timeout_browser`: 60s. Īpašības darbība: - *required*
timeout_simple_check		
timeout_snmp_agent		

Īpašība	Tips	Apraksts
timeout_external_check		
timeout_db_monitor		
timeout_http_agent		
timeout_ssh_agent		
timeout_telnet_agent		
timeout_script		
timeout_browser		

settings.get

Apraksts

`object settings.get(object parameters)`

Metode ļauj izgūt iestatījumu objektu atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta tikai vienu parametru.

Parametrs	Tips	Apraksts
output	query	Šis parametrs ir aprakstīts atsauces komentārā.

Atgrieztās vērtības

`(object)` Atgriež iestatījumu objektu.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "settings.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Avots

CSettings::get() failā *ui/include/classes/api/services/CSettings.php*.

settings.update

Apraksts

`object settings.update(object settings)`

Šī metode ļauj atjaunināt esošos kopīgos iestatījumus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildu informācijai.

Parametri

`(object)` Iestatījumu īpašības, kas jāatjaunina.

Atgrieztās vērtības

`(array)` Atgriež masīvu ar atjaunināto parametru nosaukumiem.

Piemēri

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "settings.update",
    "params": {
        "login_attempts": "1",
        "login_block": "1m"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        "login_attempts",
        "login_block"
    ],
    "id": 1
}
```

Avots

CSettings::update() failā *ui/include/classes/api/services/CSettings.php*.

Ikonu karte

Šī klase ir paredzēta darbam ar ikonu kartēm.

Objektu atsauces:

- Ikonu karte
- Ikonu kartēšana

Pieejamās metodes:

- iconmap.create - izveidot jaunas ikonu kartes
- iconmap.delete - dzēst ikonu kartes
- iconmap.get - izgūt ikonu kartes
- iconmap.update - atjaunināt ikonu kartes

Ikonu kartes objekts

Tālāk norādītie objekti ir tieši saistīti ar `iconmap` API.

Ikonu karte

Ikonu kartes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
iconmapid	ID	Ikonu kartes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
default_iconid	ID	Noklusējuma ikonas ID. Īpašības darbība: - *obligāts* izveides operācijām
name	string	Ikonu kartes nosaukums. Īpašības darbība: - *obligāts* izveides operācijām

Ikonu kartēšana

Ikonu kartēšanas objekts definē konkrētu ikonu, kas tiks izmantota hostiem ar noteiktu inventāra lauka vērtību. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
iconid	ID	Ikonu kartēšanā izmantotās ikonas ID. Īpašības darbība: - *obligāta*
expression	string	Izteiksme, ar kuru salīdzināt inventāra lauku. Īpašības darbība: - *obligāta*
inventory_link	integer	Hosta inventāra lauka ID. Atbalstīto inventāra lauku sarakstu skatiet hosta inventāra objektā. Īpašības darbība: - *obligāta*
sortorder	integer	Ikonu kartēšanas pozīcija ikonu kartē. Īpašības darbība: - *tikai lasāma*

iconmap.create

Apraksts

`object iconmap.create(object/array iconMaps)`

Šī metode ļauj izveidot jaunas ikonu kartes.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

(`object/array`) Izveidojamās ikonu kartes.

Papildus standarta ikonu kartes īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
mappings	array	Ikonu kartējumi, kas jāizveido ikonu kartei. Parametra darbība: - *obligāts*

Atgrieztās vērtības

(`object`) Atgriež objektu, kas satur izveidoto ikonu karšu ID īpašībā `iconmapids`. Atgriezto ID secība atbilst nodoto ikonu karšu secībai.

Piemēri

Izveidot ikonu karti

Izveidojiet ikonu karti, lai attēlotu dažādu tipu hostus.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Ikonu kartēšana

Avots

CIconMap::create() failā *ui/include/classes/api/services/CIconMap.php*.

iconmap.delete

Apraksts

`object iconmap.delete(array iconMapIds)`

Šī metode ļauj dzēst ikonu kartes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` dzēšamo ikonu karšu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto ikonu karšu ID zem `iconmapids` īpašības.

Piemēri

Dzēst vairākas ikonu kartes

Dzēsiet divas ikonu kartes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.delete",
    "params": [
        "2",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "2",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CIconMap::delete() failā *ui/include/classes/api/services/CIconMap.php*.

iconmap.get

Apraksts

`integer/array iconmap.get(object parameters)`

Metode ļauj izgūt ikonu kartes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
iconmapids	ID/array	Atgriezt tikai ikonu kartes ar norādītajiem ID.
sysmapids	ID/array	Atgriezt tikai ikonu kartes, kas tiek izmantotas norādītajās kartēs.
selectMappings	query	Atgriezt `mappings` īpašību ar izmantotajām ikonu atbilstībām.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `iconmapid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Ikonu kartes izgūšana

Izgūst visus datus par ikonu karti "3".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.get",
    "params": {
        "iconmapids": "3",
        "output": "extend",
        "selectMappings": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mappings": [
                {
                    "iconmappingid": "3",
                    "iconmapid": "3",
                    "iconid": "6",
                    "inventory_link": "1",
                    "expression": "server",
                    "sortorder": "0"
                },
                {
                    "iconmappingid": "4",
                    "iconmapid": "3",
                    "iconid": "10",
                    "inventory_link": "1",
```

```
                    "expression": "switch",
                    "sortorder": "1"
                }
            ],
            "iconmapid": "3",
            "name": "Host type icons",
            "default_iconid": "2"
        }
    ],
    "id": 1
}
```

Skatīt arī

- Ikonu kartēšana

Avots

CIconMap::get() failā *ui/include/classes/api/services/CIconMap.php*.

iconmap.update

Apraksts

`object iconmap.update(object/array iconMaps)`

Šī metode ļauj atjaunināt esošās ikonu kartes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo ikonu karšu īpašības.

Īpašībai `iconmapid` jābūt definētai katrai ikonu kartei, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta ikonu karšu īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
mappings	array	Ikonu kartējumi, ar kuriem aizstāt esošos ikonu kartējumus.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto ikonu karšu ID zem `iconmapids` īpašības.

Piemēri

Pārdēvēt ikonu karti

Pārdēvējiet ikonu karti par ”OS icons”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "iconmap.update",
    "params": {
        "iconmapid": "1",
        "name": "OS icons"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "iconmapids": [
            "1"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Ikonu kartēšana

Avots

CIconMap::update() failā *ui/include/classes/api/services/CIconMap.php*.

Karte

Šī klase ir paredzēta darbam ar kartēm.

Objektu atsauces:

- Karte
- Kartes elements
  - Kartes elements Hosts
  - Kartes elements Hostu grupa
  - Kartes elements Karte
  - Kartes elements Trigeris
  - Kartes elements Tags
  - Kartes elements URL
- Kartes saite
  - Kartes saites trigeris
  - Kartes saites indikatori
- Kartes URL
- Kartes lietotājs
- Kartes lietotāju grupa
- Kartes formas
- Kartes līnijas

Pieejamās metodes:

- map.create - izveidot jaunas kartes
- map.delete - dzēst kartes
- map.get - izgūt kartes
- map.update - atjaunināt kartes

Kartes objekts

Tālāk norādītie objekti ir tieši saistīti ar `map` API.

Karte

Kartes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
sysmapid	ID	Kartes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
height	integer	Kartes augstums pikseļos. Īpašības darbība: - *obligāts* izveides darbībām

Īpašība	Tips	Apraksts
name	string	Kartes nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
width	integer	Kartes platums pikseļos. Īpašības darbība: - *obligāts* izveides darbībām
backgroundid	ID	Attēla ID, kas tiek izmantots kā kartes fons.
background_scale	integer	Vai iespējot fona attēla proporcionālu mērogošanu. Iespējamās vērtības: 0 - atspējot mērogošanu; 1 - *(noklusējums)* iespējot attēla mērogošanu.
expand_macros	integer	Vai, konfigurējot karti, izvērst makrosus etiķetēs. Iespējamās vērtības: 0 - *(noklusējums)* neizvērst makrosus; 1 - izvērst makrosus.
expandproblem	integer	Vai elementiem ar vienu problēmu tiks parādīts problēmas trigeris. Iespējamās vērtības: 0 - vienmēr rādīt problēmu skaitu; 1 - *(noklusējums)* rādīt problēmas trigeri, ja ir tikai viena problēma.
grid_align	integer	Vai iespējot līdzināšanu pie režģa. Iespējamās vērtības: 0 - atspējot līdzināšanu pie režģa; 1 - *(noklusējums)* iespējot līdzināšanu pie režģa.
grid_show	integer	Vai rādīt režģi kartē. Iespējamās vērtības: 0 - nerādīt režģi; 1 - *(noklusējums)* rādīt režģi.
grid_size	integer	Kartes režģa izmērs pikseļos. Atbalstītās vērtības: 20, 40, 50, 75 un 100. Noklusējums: 50.
highlight	integer	Vai ikonu izcelšana ir iespējota. Iespējamās vērtības: 0 - izcelšana atspējota; 1 - *(noklusējums)* izcelšana iespējota.
iconmapid	ID	Kartē izmantotās ikonu kartes ID.
label_format	integer	Vai iespējot paplašinātās etiķetes. Iespējamās vērtības: 0 - *(noklusējums)* atspējot paplašinātās etiķetes; 1 - iespējot paplašinātās etiķetes.
label_location	integer	Kartes elementa etiķetes atrašanās vieta. Iespējamās vērtības: 0 - *(noklusējums)* apakšā; 1 - pa kreisi; 2 - pa labi; 3 - augšā.
label_string_host	string	Pielāgota etiķete hosta elementiem. Īpašības darbība: - *obligāts*, ja `label_type_host` ir iestatīts uz ”custom”

Īpašība	Tips	Apraksts
label_string_hostgroup	string	Pielāgota etiķete hostu grupu elementiem. Īpašības darbība: - *obligāts*, ja `label_type_hostgroup` ir iestatīts uz ”custom”
label_string_image	string	Pielāgota etiķete attēlu elementiem. Īpašības darbība: - *obligāts*, ja `label_type_image` ir iestatīts uz ”custom”
label_string_map	string	Pielāgota etiķete kartes elementiem. Īpašības darbība: - *obligāts*, ja `label_type_map` ir iestatīts uz ”custom”
label_string_trigger	string	Pielāgota etiķete trigeru elementiem. Īpašības darbība: - *obligāts*, ja `label_type_trigger` ir iestatīts uz ”custom”
label_type	integer	Kartes elementa etiķetes tips. Iespējamās vērtības: 0 - etiķete; 1 - IP adrese; 2 - *(noklusējums)* elementa nosaukums; 3 - tikai statuss; 4 - nekas.
label_type_host	integer	Etiķetes tips hosta elementiem. Iespējamās vērtības: 0 - etiķete; 1 - IP adrese; 2 - *(noklusējums)* elementa nosaukums; 3 - tikai statuss; 4 - nekas; 5 - pielāgots.
label_type_hostgroup	integer	Etiķetes tips hostu grupu elementiem. Iespējamās vērtības: 0 - etiķete; 2 - *(noklusējums)* elementa nosaukums; 3 - tikai statuss; 4 - nekas; 5 - pielāgots.
label_type_image	integer	Etiķetes tips hostu grupu elementiem. Iespējamās vērtības: 0 - etiķete; 2 - *(noklusējums)* elementa nosaukums; 4 - nekas; 5 - pielāgots.
label_type_map	integer	Etiķetes tips kartes elementiem. Iespējamās vērtības: 0 - etiķete; 2 - *(noklusējums)* elementa nosaukums; 3 - tikai statuss; 4 - nekas; 5 - pielāgots.

Īpašība	Tips	Apraksts
label_type_trigger	integer	Etiķetes tips trigeru elementiem. Iespējamās vērtības: 0 - etiķete; 2 - *(noklusējums)* elementa nosaukums; 3 - tikai statuss; 4 - nekas; 5 - pielāgots.
markelements	integer	Vai izcelt kartes elementus, kuriem nesen mainījies statuss. Iespējamās vērtības: 0 - *(noklusējums)* neizcelt elementus; 1 - izcelt elementus.
severity_min	integer	Minimālā trigeru nopietnība, kas tiks parādīta kartē. Atbalstīto trigeru nopietnības līmeņu sarakstu skatiet trigera `severity` īpašībā.
show_element_label	integer	Kā pēc noklusējuma parādīt elementu etiķetes. Iespējamās vērtības: 0 - vienmēr rādīt; 1 - *(noklusējums)* automātiski paslēpt.
show_link_label	integer	Kā pēc noklusējuma parādīt saišu etiķetes. Iespējamās vērtības: 0 - vienmēr rādīt; 1 - *(noklusējums)* automātiski paslēpt.
show_unack	integer	Kā jāparāda problēmas. Iespējamās vērtības: 0 - *(noklusējums)* rādīt visu problēmu skaitu; 1 - rādīt tikai neapstiprināto problēmu skaitu; 2 - rādīt apstiprināto un neapstiprināto problēmu skaitu atsevišķi.
userid	ID	Lietotāja ID, kurš ir kartes īpašnieks.
private	integer	Kartes koplietošanas tips. Iespējamās vērtības: 0 - publiska karte; 1 - *(noklusējums)* privāta karte.
show_suppressed	integer	Vai tiek rādītas apspiestās problēmas. Iespējamās vērtības: 0 - *(noklusējums)* slēpt apspiestās problēmas; 1 - rādīt apspiestās problēmas.

Kartes elements

Kartes elementa objekts definē objektu, kas tiek attēlots kartē. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
selementid	ID	Kartes elementa ID. Īpašības darbība: - *tikai lasāms*
elements	array	Elementa datu objekts. Īpašības darbība: - *obligāts*, ja `elementtype` ir iestatīts uz ”host”, ”map”, ”trigger” vai ”host group”

Īpašība	Tips	Apraksts
elementtype	integer	Kartes elementa tips. Iespējamās vērtības: 0 - hosts; 1 - karte; 2 - trigeris; 3 - hostu grupa; 4 - attēls. Īpašības darbība: - *obligāts*
iconid_off	ID	Attēla ID, kas tiek izmantots elementa attēlošanai noklusējuma stāvoklī. Īpašības darbība: - *obligāts*
areatype	integer	Kā jāattēlo atsevišķi hostu grupas hosti. Iespējamās vērtības: 0 - *(noklusējums)* hostu grupas elements aizņems visu karti; 1 - hostu grupas elementam būs fiksēts izmērs.
elementsubtype	integer	Kā hostu grupas elements jāattēlo kartē. Iespējamās vērtības: 0 - *(noklusējums)* attēlot hostu grupu kā vienu elementu; 1 - attēlot katru grupas hostu atsevišķi.
evaltype	integer	Kartes elementa birku filtrēšanas nosacījuma novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
height	integer	Fiksēta izmēra hostu grupas elementa augstums pikseļos. Noklusējums: 200.
iconid_disabled	ID	Attēla ID, kas tiek izmantots atspējotu kartes elementu attēlošanai. Īpašības darbība: - *atbalstīts*, ja `elementtype` ir iestatīts uz ”host”, ”map”, ”trigger” vai ”host group”
iconid_maintenance	ID	Attēla ID, kas tiek izmantots kartes elementu attēlošanai uzturēšanas režīmā. Īpašības darbība: - *atbalstīts*, ja `elementtype` ir iestatīts uz ”host”, ”map”, ”trigger” vai ”host group”
iconid_on	ID	Attēla ID, kas tiek izmantots kartes elementu attēlošanai ar problēmām. Īpašības darbība: - *atbalstīts*, ja `elementtype` ir iestatīts uz ”host”, ”map”, ”trigger” vai ”host group”
label	string	Elementa etiķete.
label_location	integer	Kartes elementa etiķetes novietojums. Iespējamās vērtības: -1 - *(noklusējums)* noklusējuma novietojums; 0 - apakšā; 1 - pa kreisi; 2 - pa labi; 3 - augšā.

Īpašība	Tips	Apraksts
permission	integer	Atļauju līmeņa tips. Iespējamās vērtības: -1 - nav; 2 - tikai lasīšana; 3 - lasīšana-rakstīšana.
show_label	integer	Kā attēlot elementa etiķeti. Iespējamās vērtības: -1 - *(noklusējums)* kartes noklusējums; 0 - vienmēr attēlot; 1 - automātiski paslēpt.
sysmapid	ID	Kartes ID, kurai elements pieder. Īpašības darbība: - *tikai lasāms*
urls	array	Kartes elementa URL. Kartes elementa URL objekts ir detalizēti aprakstīts zemāk.
use_iconmap	integer	Vai hostu elementiem jāizmanto ikonu kartēšana. Iespējamās vērtības: 0 - neizmantot ikonu kartēšanu; 1 - *(noklusējums)* izmantot ikonu kartēšanu.
viewtype	integer	Hostu grupas elementa izvietošanas algoritms. Iespējamās vērtības: 0 - *(noklusējums)* režģis.
width	integer	Fiksēta izmēra hostu grupas elementa platums pikseļos. Noklusējums: 200.
x	integer	Elementa X koordinātas pikseļos. Noklusējums: 0.
y	integer	Elementa Y koordinātas pikseļos. Noklusējums: 0.
zindex	integer	Vērtība, ko izmanto kartes elementu secības noteikšanai (z-index). Noklusējums: 0.

Kartes elements Hosts

Kartes elementa Hosts objekts definē vienu hosta elementu.

Īpašība	Type	Apraksts
hostid	ID	Hosta ID.

Kartes elements Host group

Kartes elementa Host group objekts definē vienu hostu grupas elementu.

Īpašība	Type	Apraksts
groupid	ID	Hostu grupas ID.

Kartes elements Karte

Kartes elementa Karte objekts definē vienu kartes elementu.

Īpašība	Type	Apraksts
sysmapid	ID	Kartes ID.

Kartes elementa trigeris

Kartes elementa trigeris objekts definē vienu vai vairākus trigeru elementus.

Īpašība	Type	Apraksts
triggerid	ID	Trigera ID.

Kartes elementa tags

Kartes elementa taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Kartes elementa taga nosaukums. Īpašības darbība: - *obligāts*
operator	integer	Kartes elementa taga nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
value	string	Kartes elementa taga vērtība.

Kartes elementa URL

Kartes elementa URL objekts definē klikšķināmu saiti, kas būs pieejama konkrētam kartes elementam. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
sysmapelementurlid	ID	Kartes elementa URL ID. Īpašības darbība: - *tikai lasāms*
name	string	Saites virsraksts. Īpašības darbība: - *obligāts*
url	string	Saites URL. Īpašības darbība: - *obligāts*
selementid	ID	Kartes elementa ID, kuram pieder URL.

Kartes saite

Kartes saites objekts definē saiti starp diviem kartes elementiem. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
linkid	ID	Kartes saites ID. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
sysmapid	ID	Kartes ID, kurai saite pieder.
selementid1	ID	Pirmā kartes elementa ID, kas ir saistīts vienā galā. Īpašības darbība: - *obligāts*
selementid2	ID	Pirmā kartes elementa ID, kas ir saistīts otrā galā. Īpašības darbība: - *obligāts*
drawtype	integer	Saites līnijas zīmēšanas stils. Iespējamās vērtības: 0 - *(noklusējums)* līnija; 2 - trekna līnija; 3 - punktota līnija; 4 - pārtraukta līnija.
color	string	Līnijas krāsa heksadecimāla krāsas koda formā. Noklusējums: 000000.
label	string	Saites etiķete.
show_label	integer	Kā attēlot saites etiķeti. Iespējamās vērtības: -1 - *(noklusējums)* kartes noklusējums; 0 - vienmēr attēlot; 1 - automātiski paslēpt.
indicator_type	integer	Atlasiet saites indikatora tipu. Iespējamās vērtības: 0 - *(noklusējums)* statiska saite; 1 - trigeris; 2 - vienuma vērtība.
linktriggers	array	Kartes saites trigeri, ko izmantot kā saites statusa indikatorus. Kartes saites trigera objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāts*, ja `indicator_type` ir iestatīts uz ”trigger” (1).
itemid	ID	Vienuma ID. Īpašības darbība: - *obligāts*, ja `indicator_type` ir iestatīts uz ”item value” (2).
highlights	array	Kartes saites izcēlumi, ko izmantot kā saites statusa indikatorus. Kartes saites indikatoru objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāts*, ja `indicator_type` ir iestatīts uz ”item value” (2).
thresholds	array	Kartes saites sliekšņi, ko izmantot kā saites statusa indikatorus. Kartes saites indikatoru objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāts*, ja `indicator_type` ir iestatīts uz ”item value” (2).
permission	integer	Atļaujas līmeņa tips. Iespējamās vērtības: -1 - nav; 2 - tikai lasīšana; 3 - lasīšana-rakstīšana.

Kartes saites trigeris

Kartes saites trigera objekts definē kartes saites statusa indikatoru, pamatojoties uz trigera stāvokli. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
triggerid	ID	Kā saites indikators izmantotā trigera ID. Īpašības darbība: - *obligāts*
color	string	Indikatora krāsa kā heksadecimāls krāsas kods. Noklusējums: DD0000.
drawtype	integer	Indikatora zīmēšanas stils. Iespējamās vērtības: 0 - *(noklusējums)* līnija; 2 - trekna līnija; 3 - punktota līnija; 4 - pārtraukta līnija.

Kartes saites indikatori

Kartes saites indikatoru objekts definē kartes saites statusa indikatoru, pamatojoties uz vienuma vērtību. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
pattern	string	Regulārā izteiksme salīdzināšanai. Pieejama tikai izcelšanai.
sortorder	integer	Izmanto, lai definētu izcelšanas kārtošanas secību. Pieejama tikai izcelšanai.
threshold	string	Skaitliska vērtība salīdzināšanai. Pieejama tikai sliekšņiem.
drawtype	integer	Indikatora zīmēšanas stils. Iespējamās vērtības: 0 - *(noklusējums)* līnija; 2 - trekna līnija; 3 - punktota līnija; 4 - pārtraukta līnija.
color	string	Indikatora krāsa kā heksadecimāls krāsas kods. Noklusējums: DD0000. Īpašības darbība: - *required*

Kartes URL

Kartes URL objekts definē klikšķināmu saiti, kas kartē būs pieejama visiem noteikta tipa elementiem. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
sysmapurlid	ID	Kartes URL ID. Īpašības darbība: - *tikai lasāms*
name	string	Saites virsraksts. Īpašības darbība: - *obligāts*
url	string	Saites URL. Īpašības darbība: - *obligāts*

Īpašība	Tips	Apraksts
elementtype	integer	Kartes elementa tips, kuram URL būs pieejams. Atbalstīto tipu sarakstu skatiet kartes elementa type īpašībā. Noklusējums: 0.
sysmapid	ID	Kartes ID, kurai pieder URL.

Kartes lietotājs

Kartes atļauju saraksts, pamatojoties uz lietotājiem. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
sysmapuserid	ID	Kartes lietotāja ID. Property behavior: - *read-only*
userid	ID	Lietotāja ID. Property behavior: - *required*
permission	integer	Atļaujas līmeņa tips. Iespējamās vērtības: 2 - tikai lasīšanai; 3 - lasīšana-rakstīšana. Property behavior: - *required*

Kartes lietotāju grupa

Kartes atļauju saraksts, pamatojoties uz lietotāju grupām. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
sysmapusrgrpid	ID	Kartes lietotāju grupas ID. Īpašības darbība: - *tikai lasāms*
usrgrpid	ID	Lietotāju grupas ID. Īpašības darbība: - *obligāts*
permission	integer	Atļaujas līmeņa tips. Iespējamās vērtības: 2 - tikai lasīšana; 3 - lasīšana-rakstīšana. Īpašības darbība: - *obligāts*

Kartes figūras

Kartes figūras objekts definē ģeometrisku figūru (ar tekstu vai bez tā), kas tiek attēlota kartē. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
sysmap_shapeid	ID	Kartes figūras elementa ID. Īpašības darbība: - *tikai lasāms*
type	integer	Kartes figūras elementa tips. Iespējamās vērtības: 0 - taisnstūris; 1 - elipse. Īpašība ir obligāta, veidojot jaunas figūras. Īpašības darbība: - *obligāts*
x	integer	Figūras X koordinātas pikseļos. Noklusējums: 0.
y	integer	Figūras Y koordinātas pikseļos. Noklusējums: 0.
width	integer	Figūras platums pikseļos. Noklusējums: 200.
height	integer	Figūras augstums pikseļos. Noklusējums: 200.
text	string	Figūras teksts.
font	integer	Teksta fonts figūras iekšpusē. Iespējamās vērtības: 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 Noklusējums: 9.
font_size	integer	Fonta lielums pikseļos. Noklusējums: 11.
font_color	string	Fonta krāsa. Noklusējums: 000000.
text_halign	integer	Teksta horizontālā līdzināšana. Iespējamās vērtības: 0 - centrēts; 1 - pa kreisi; 2 - pa labi. Noklusējums: 0.

Īpašība	Tips	Apraksts
text_valign	integer	Teksta vertikālā līdzināšana. Iespējamās vērtības: 0 - vidū; 1 - augšā; 2 - apakšā. Noklusējums: 0.
border_type	integer	Apmales tips. Iespējamās vērtības: 0 - nav; 1 - ————; 2 - ┅; 3 - – – –. Noklusējums: 0.
border_width	integer	Apmales platums pikseļos. Noklusējums: 0.
border_color	string	Apmales krāsa. Noklusējums: 000000.
background_color	string	Fona krāsa (aizpildījuma krāsa). Noklusējums: `(tukšs)`.
zindex	integer	Vērtība, ko izmanto visu figūru un līniju secības noteikšanai (z-index). Noklusējums: 0.

Kartes līnijas

Kartes līnijas objekts definē līniju, kas tiek attēlota kartē. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
sysmap_shapeid	ID	Kartes formas elementa ID. Īpašības darbība: - *tikai lasāms*
x1	integer	Līnijas 1. punkta X koordinātas pikseļos. Noklusējums: 0.
y1	integer	Līnijas 1. punkta Y koordinātas pikseļos. Noklusējums: 0.
x2	integer	Līnijas 2. punkta X koordinātas pikseļos. Noklusējums: 200.
y2	integer	Līnijas 2. punkta Y koordinātas pikseļos. Noklusējums: 200.
line_type	integer	Līniju tips. Iespējamās vērtības: 0 - nav; 1 - ————; 2 - ┅; 3 - – – –. Noklusējums: 0.

Īpašība	Tips	Apraksts
line_width	integer	Līniju platums pikseļos. Noklusējums: 0.
line_color	string	Līnijas krāsa. Noklusējums: 000000.
zindex	integer	Vērtība, ko izmanto visu formu un līniju secības noteikšanai (z-index). Noklusējums: 0.

map.create

Apraksts

`object map.create(object/array maps)`

Šī metode ļauj izveidot jaunas kartes.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Kartes, ko izveidot.

Papildus standarta kartes īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
links	array	Kartes saites, kas jāizveido kartē.
selements	array	Kartes elementi, kas jāizveido kartē.
urls	array	Kartes URL, kas jāizveido kartē.
users	array	Kartes lietotāju koplietošanas ieraksti, kas jāizveido kartē.
userGroups	array	Kartes lietotāju grupu koplietošanas ieraksti, kas jāizveido kartē.
shapes	array	Kartes formas, kas jāizveido kartē.
lines	array	Kartes līnijas, kas jāizveido kartē.

> **Note:**
> Lai izveidotu kartes saites, jums būs jāiestata kartes elementa `selementid` uz patvaļīgu vērtību un pēc tam jāizmanto šī vērtība, lai atsauktos uz šo elementu saišu īpašībās `selementid1` vai `selementid2`. Kad elements tiks izveidots, šī vērtība tiks aizstāta ar pareizo ID, ko ģenerējis Zabbix. Skatīt piemēru.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto karšu ID īpašībā `sysmapids`. Atgriezto ID secība atbilst nodoto karšu secībai.

Piemēri

Izveidot tukšu karti

Izveidojiet karti bez elementiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "map.create",
    "params": {
        "name": "Map",
        "width": 600,
        "height": 600
    },
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
        ]
    },
    "id": 1
}
```

Izveidot hostu karti

Izveidojiet karti ar diviem hosta elementiem un saiti starp tiem. Ņemiet vērā pagaidu "selementid1" un "selementid2" vērtību izmantošanu kartes saites objektā, lai atsauktos uz kartes elementiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
```

```
    },
    "id": 1
}
```

Izveidot trigeru karti

Izveidojiet karti ar trigeru elementu, kurā ir divi trigeri.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Kartes kopīgošana

Izveidojiet karti ar diviem kopīgošanas veidiem (lietotājs un lietotāju grupa).

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Kartes formas

Izveidojiet karti ar kartes nosaukumu virsrakstā.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "10"
        ]
    },
    "id": 1
}
```

Kartes līnijas

Izveidojiet kartes līniju.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "11"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Kartes elements
- Kartes saite
- Kartes URL
- Kartes lietotājs
- Kartes lietotāju grupa
- Kartes forma
- Kartes līnija

Avots

CMap::create() failā *ui/include/classes/api/services/CMap.php*.

map.delete

Apraksts

`object map.delete(array mapIds)`

Šī metode ļauj dzēst kartes.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo karšu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto karšu ID zem `sysmapids` īpašības.

Piemēri

Dzēst vairākas kartes

Dzēsiet divas kartes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "map.delete",
    "params": [
        "12",
        "34"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "12",
            "34"
        ]
    },
    "id": 1
}
```

Avots

CMap::delete() failā *ui/include/classes/api/services/CMap.php*.

map.get

Apraksts

`integer/array map.get(object parameters)`

Šī metode ļauj izgūt kartes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
sysmapids	ID/array	Atgriež tikai kartes ar norādītajiem ID.
userids	ID/array	Atgriež tikai kartes, kas pieder norādītajiem lietotāju ID.
expandUrls	flag	Pievieno globālos karšu URL atbilstošajiem kartes elementiem un izvērš makrosus visos kartes elementu URL.
selectIconMap	query	Atgriež `iconmap` īpašību ar kartē izmantoto ikonu karti.
selectLinks	query	Atgriež `links` īpašību ar kartes saitēm starp elementiem.
selectSelements	query	Atgriež `selements` īpašību ar kartes elementiem.
selectUrls	query	Atgriež `urls` īpašību ar kartes URL.
selectUsers	query	Atgriež `users` īpašību ar lietotājiem, kuriem karte ir koplietota.
selectUserGroups	query	Atgriež `userGroups` īpašību ar lietotāju grupām, ar kurām karte ir koplietota.
selectShapes	query	Atgriež `shapes` īpašību ar kartes figūrām.
selectLines	query	Atgriež `lines` īpašību ar kartes līnijām.
sortfield	string/array	Kārto rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `name`, `width`, `height`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	

Parametrs	Tips	Apraksts
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt karti

Iegūt visus datus par karti "3".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "selements": [
                {
                    "selementid": "10",
                    "sysmapid": "3",
                    "elementtype": "4",
                    "evaltype": "0",
                    "iconid_off": "1",
                    "iconid_on": "0",
                    "label": "Zabbix serveris",
                    "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": "Tīmekļa serveris",
                "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": "Lokālais tīkls",
            "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
}
```


Skatiet arī

- Ikonu karte
- Kartes elements
- Kartes saite
- Kartes URL
- Kartes lietotājs
- Kartes lietotāju grupa
- Kartes formas
- Kartes līnijas

Avots

CMap::get() failā *ui/include/classes/api/services/CMap.php*.

map.update

Apraksts

`object map.update(object/array maps)`

Šī metode ļauj atjaunināt esošās kartes.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Kartes rekvizīti, kas jāatjaunina.

Rekvizītam `mapid` jābūt definētam katrai kartei, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta kartes rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
links	array	Kartes saites, ar kurām aizstāt esošās saites.
selements	array	Kartes elementi, ar kuriem aizstāt esošos elementus.
urls	array	Kartes URL, ar kuriem aizstāt esošos URL.
users	array	Kartes lietotāja koplietojumi, ar kuriem aizstāt esošos elementus.
userGroups	array	Kartes lietotāju grupas koplietojumi, ar kuriem aizstāt esošos elementus.
shapes	array	Kartes formas, ar kurām aizstāt esošās formas.
lines	array	Kartes līnijas, ar kurām aizstāt esošās līnijas.

> **Note:**
> Lai izveidotu kartes saites starp jauniem kartes elementiem, jums būs jāiestata elementa `selementid` uz patvaļīgu vērtību un pēc tam jāizmanto šī vērtība, lai atsauktos uz šo elementu saišu rekvizītos `selementid1` vai `selementid2`. Kad elements tiks izveidots, šī vērtība tiks aizstāta ar pareizo ID, ko ģenerējis Zabbix. Skatiet piemēru map.create.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto karšu ID zem `sysmapids` īpašības.

Piemēri

Mainīt kartes izmēru

Mainiet kartes izmēru uz 1200x1200 pikseļiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "8",
        "width": 1200,
        "height": 1200
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "8"
```

```
        ]
    },
    "id": 1
}
```

Mainīt kartes īpašnieku

Pieejams tikai administratoriem un super administratoriem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "map.update",
    "params": {
        "sysmapid": "9",
        "userid": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "sysmapids": [
            "9"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Kartes elements
- Kartes saite
- Kartes URL
- Kartes lietotājs
- Kartes lietotāju grupa
- Kartes formas
- Kartes līnijas

Avots

CMap::update() failā *ui/include/classes/api/services/CMap.php*.

Konfigurācija

Šī klase ir paredzēta Zabbix konfigurācijas datu eksportēšanai un importēšanai.

Pieejamās metodes:

- configuration.export - eksportēt konfigurācijas datus
- configuration.import - importēt konfigurācijas datus
- configuration.importcompare - salīdzināt importēšanas failu ar pašreizējiem sistēmas elementiem

configuration.export

Apraksts

`string configuration.export(object parameters)`

Šī metode ļauj eksportēt konfigurācijas datus kā serializētu virkni.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka eksportējamos objektus un izmantojamo formātu.

Parametrs	Type	Apraksts
format	string	Formāts, kādā dati jāeksportē. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON; `raw` - neapstrādāts PHP masīvs. Parametra darbība: - *required*
prettyprint	boolean	Padara izvadi cilvēkam lasāmāku, pievienojot atkāpes. Iespējamās vērtības: `true` - pievienot atkāpes; `false` - *(noklusējums)* nepievienot atkāpes.
options	object	Eksportējamie objekti. Objektam `options` ir šādi parametri: `host_groups` - `(array)` eksportējamo hostu grupu ID; `hosts` - `(array)` eksportējamo hostu ID; `images` - `(array)` eksportējamo attēlu ID; `maps` - `(array)` eksportējamo karšu ID; `mediaTypes` - `(array)` eksportējamo mediju tipu ID; `template_groups` - `(array)` eksportējamo veidņu grupu ID; `templates` - `(array)` eksportējamo veidņu ID. *Admin* un *User* tipa lietotāji drīkst eksportēt tikai tos objektus, kuriem viņiem ir *read-only* vai *read-write* atļauja, kā arī attēlus, bet ne mediju tipus. Parametra darbība: - *required*

Atgrieztās vērtības

`(string)` Atgriež serializētu virkni, kas satur pieprasītos konfigurācijas datus.

Piemēri

Veidnes eksportēšana

Eksportējiet veidnes ”10571” konfigurāciju kā XML virkni.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "configuration.export",
    "params": {
        "options": {
            "templates": [
                "10571"
            ]
        },
        "format": "xml"
```

```
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n<zabbix_export><version>8.0</version><template_
    "id": 1
}
```

Avots

CConfiguration::export() failā *ui/include/classes/api/services/CConfiguration.php*.

configuration.import

Apraksts

`boolean configuration.import(object parameters)`

Šī metode ļauj importēt konfigurācijas datus no serializētas virknes.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur importējamos datus un noteikumus, kā šie dati jāapstrādā.

Parametrs	Tips	Apraksts
format	string	Serializētās virknes formāts. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON. Parametra darbība: - *required*
source	string	Serializēta virkne, kas satur konfigurācijas datus. Parametra darbība: - *required*
rules	object	Noteikumi, kā importēt jaunus un esošus objektus. *Admin* tipa lietotāji drīkst importēt tikai tos objektus, kuriem viņiem ir *read-write* atļauja, kā arī kartes. Piemēram, hosts un tā entītijas (vienumi, trigeri, grafiki utt.) var tikt importēti tikai tad, ja lietotāja lietotāju grupai ir atļauja uz hostu grupu, kurai importētais hosts piederēs. Attēlus un mediju tipus *Admin* tipa lietotāji importēt nevar. Parametrs `rules` ir detalizēti aprakstīts zemāk esošajā tabulā. Parametra darbība: - *required*

Attention:
Ja nav norādīti nekādi noteikumi, konfigurācija netiks atjaunināta.

Objekts `rules` atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
discoveryRules	object	Noteikumi, kā importēt LLD noteikumus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni LLD noteikumi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie LLD noteikumi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, LLD noteikumi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
graphs	object	Noteikumi, kā importēt grafikus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni grafiki; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie grafiki tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, grafiki, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
host_groups	object	Noteikumi, kā importēt hostu grupas. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas hostu grupas; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās hostu grupas tiks atjauninātas; noklusējums: `false`.
template_groups	object	Noteikumi, kā importēt veidņu grupas. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas veidņu grupas; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās veidņu grupas tiks atjauninātas; noklusējums: `false`.
hosts	object	Noteikumi, kā importēt hostus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni hosti; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie hosti tiks atjaunināti; noklusējums: `false`.
httptests	object	Noteikumi, kā importēt tīmekļa scenārijus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni tīmekļa scenāriji; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie tīmekļa scenāriji tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, tīmekļa scenāriji, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
images	object	Noteikumi, kā importēt attēlus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni attēli; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie attēli tiks atjaunināti; noklusējums: `false`.

Parametrs	Tips	Apraksts
items	object	Noteikumi, kā importēt vienumus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni vienumi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie vienumi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, vienumi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
maps	object	Noteikumi, kā importēt kartes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas kartes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās kartes tiks atjauninātas; noklusējums: `false`.
mediaTypes	object	Noteikumi, kā importēt mediju tipus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni mediju tipi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie mediju tipi tiks atjaunināti; noklusējums: `false`.
templateLinkage	object	Noteikumi, kā importēt veidņu saites. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas nav piesaistītas importējamajam hostam vai veidnei, bet ir iekļautas importētajos datos, tiks piesaistītas; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas ir piesaistītas importējamajam hostam vai veidnei, bet nav iekļautas importētajos datos, tiks atsaistītas, nenoņemot entītijas (vienumus, trigerus utt.), kas mantotas no atsaistītajām veidnēm; noklusējums: `false`.
templates	object	Noteikumi, kā importēt veidnes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas veidnes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās veidnes tiks atjauninātas; noklusējums: `false`.
templateDashboards	object	Noteikumi, kā importēt veidņu paneļus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni veidņu paneļi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie veidņu paneļi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidņu paneļi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
triggers	object	Noteikumi, kā importēt trigerus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni trigeri; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie trigeri tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, trigeri, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.

Parametrs	Tips	Apraksts
valueMaps	object	Noteikumi, kā importēt hosta vai veidnes vērtību kartes. Atbalstītie parametri: `createMissing` - (`boolean`) ja iestatīts uz `true`, tiks izveidotas jaunas vērtību kartes; noklusējums: `false`; `updateExisting` - (`boolean`) ja iestatīts uz `true`, esošās vērtību kartes tiks atjauninātas; noklusējums: `false`; `deleteMissing` - (`boolean`) ja iestatīts uz `true`, vērtību kartes, kas nav iekļautas importētajos datos, tiks dzēstas no datubāzes; noklusējums: `false`.

Atgrieztās vērtības

(`boolean`) Atgriež `true`, ja importēšana ir bijusi veiksmīga.

Piemēri

Veidnes importēšana

Importējiet veidnes konfigurāciju, kas ietverta XML virknē. Ja XML virknē trūkst kādu vienumu vai trigeru, tie tiks dzēsti no datubāzes, bet viss pārējais paliks nemainīts.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Avots

CConfiguration::import() failā *ui/include/classes/api/services/CConfiguration.php*.

configuration.importcompare

Apraksts

`array configuration.importcompare(object parameters)`

Šī metode ļauj salīdzināt importēšanas failu ar pašreizējiem sistēmas elementiem un parāda, kas tiks mainīts, ja šis importēšanas fails tiks importēts.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur iespējamos importējamos datus un noteikumus, kā šie dati jāapstrādā.

Parametrs	Type	Apraksts
format	string	Serializētās virknes formāts. Iespējamās vērtības: `yaml` - YAML; `xml` - XML; `json` - JSON. Parametra darbība: - *required*
source	string	Serializēta virkne, kas satur konfigurācijas datus. Parametra darbība: - *required*
rules	object	Noteikumi, kā jāsalīdzina jauni un esoši objekti. *Admin* un *User* tipa lietotāji var salīdzināt jaunus objektus tikai ar esošajiem objektiem, kuriem viņiem ir *read-only* vai *read-write* atļauja. Parametrs `rules` ir detalizēti aprakstīts zemāk esošajā tabulā. Parametra darbība: - *required*

Attention:
Ja nav norādīti nekādi noteikumi, nebūs nekā, ko atjaunināt, un rezultāts būs tukšs.

Note:
Salīdzināšana tiks veikta tikai hostu grupām un veidnēm. Trigeri un grafiki tiks salīdzināti tikai importētajām veidnēm, visi pārējie tiks uzskatīti par "jauniem".

Objekts `rules` atbalsta šādus parametrus.

Parametrs	Type	Apraksts
discoveryRules	object	Noteikumi, kā importēt LLD noteikumus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni LLD noteikumi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie LLD noteikumi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, LLD noteikumi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
graphs	object	Noteikumi, kā importēt grafikus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni grafiki; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie grafiki tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, grafiki, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
host_groups	object	Noteikumi, kā importēt hostu grupas. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas hostu grupas; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās hostu grupas tiks atjauninātas; noklusējums: `false`.
template_groups	object	Noteikumi, kā importēt veidņu grupas. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas veidņu grupas; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās veidņu grupas tiks atjauninātas; noklusējums: `false`.
hosts	object	Noteikumi, kā importēt hostus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni hosti; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie hosti tiks atjaunināti; noklusējums: `false`. Šis parametrs neietekmēs izvadi. Tas ir atļauts tikai saskaņotības nodrošināšanai ar `configuration.import`.
httptests	object	Noteikumi, kā importēt tīmekļa scenārijus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni tīmekļa scenāriji; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie tīmekļa scenāriji tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, tīmekļa scenāriji, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.

Parametrs	Type	Apraksts
images	object	Noteikumi, kā importēt attēlus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni attēli; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie attēli tiks atjaunināti; noklusējums: `false`. Šis parametrs neietekmēs izvadi. Tas ir atļauts tikai saskaņotības nodrošināšanai ar `configuration.import`.
items	object	Noteikumi, kā importēt vienumus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni vienumi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie vienumi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, vienumi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
maps	object	Noteikumi, kā importēt kartes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas kartes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās kartes tiks atjauninātas; noklusējums: `false`. Šis parametrs neietekmēs izvadi. Tas ir atļauts tikai saskaņotības nodrošināšanai ar `configuration.import`.
mediaTypes	object	Noteikumi, kā importēt mediju tipus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni mediju tipi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie mediju tipi tiks atjaunināti; noklusējums: `false`. Šis parametrs neietekmēs izvadi. Tas ir atļauts tikai saskaņotības nodrošināšanai ar `configuration.import`.
templateLinkage	object	Noteikumi, kā importēt veidņu saites. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas nav piesaistītas importējamajam hostam vai veidnei, bet ir iekļautas importētajos datos, tiks piesaistītas; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidnes, kas ir piesaistītas importējamajam hostam vai veidnei, bet nav iekļautas importētajos datos, tiks atsaistītas, nenoņemot entītijas (vienumus, trigerus utt.), kas mantotas no atsaistītajām veidnēm; noklusējums: `false`.
templates	object	Noteikumi, kā importēt veidnes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas veidnes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās veidnes tiks atjauninātas; noklusējums: `false`.

Parametrs	Type	Apraksts
templateDashboards	object	Noteikumi, kā importēt veidņu paneļus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni veidņu paneļi; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie veidņu paneļi tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, veidņu paneļi, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
triggers	object	Noteikumi, kā importēt trigerus. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidoti jauni trigeri; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošie trigeri tiks atjaunināti; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, trigeri, kas nav iekļauti importētajos datos, tiks dzēsti no datubāzes; noklusējums: `false`.
valueMaps	object	Noteikumi, kā importēt hosta vai veidnes vērtību kartes. Atbalstītie parametri: `createMissing` - `(boolean)` ja iestatīts uz `true`, tiks izveidotas jaunas vērtību kartes; noklusējums: `false`; `updateExisting` - `(boolean)` ja iestatīts uz `true`, esošās vērtību kartes tiks atjauninātas; noklusējums: `false`; `deleteMissing` - `(boolean)` ja iestatīts uz `true`, vērtību kartes, kas nav iekļautas importētajos datos, tiks dzēstas no datubāzes; noklusējums: `false`.

Atgrieztās vērtības

`(array)` Atgriež masīvu ar izmaiņām konfigurācijā, kas tiks veiktas.

Piemēri

Veidnes importēšanas salīdzināšana

Salīdziniet XML virknē ietverto veidni ar pašreizējiem sistēmas elementiem un parādiet, kas tiks mainīts, ja šī veidne tiks importēta.

Pieprasījums:

```
{
    "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
}
```


Atbilde:


```
{
    "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
}
```

Avots

CConfiguration::importcompare() failā *ui/include/classes/api/services/CConfiguration.php*.

Korelācija

Šī klase ir paredzēta darbam ar korelācijām.

Objektu atsauces:

- Korelācija
- Korelācijas darbība
- Korelācijas filtrs
  - Korelācijas filtra nosacījums

Pieejamās metodes:

- correlation.create - izveidot jaunas korelācijas
- correlation.delete - dzēst korelācijas
- correlation.get - izgūt korelācijas
- correlation.update - atjaunināt korelācijas

Korelācijas objekts

Tālāk norādītie objekti ir tieši saistīti ar `correlation` API.

Korelācija

Korelācijas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
correlationid	ID	Korelācijas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Korelācijas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
description	string	Korelācijas apraksts.
status	integer	Vai korelācija ir iespējota vai atspējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.

Korelācijas operācija

Korelācijas operācijas objekts definē operāciju, kas tiks izpildīta, kad tiks veikta korelācija. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
type	integer	Operācijas tips. Iespējamās vērtības: 0 - aizvērt vecos notikumus; 1 - aizvērt jauno notikumu. Īpašības darbība: - *obligāts*

Korelācijas filtrs

Korelācijas filtra objekts definē nosacījumu kopu, kam jābūt izpildītai, lai veiktu konfigurētās korelācijas darbības. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
conditions	array	Filtra nosacījumu kopa, ko izmantot rezultātu filtrēšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - Un/Vai; 1 - Un; 2 - Vai; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti filtra nosacījumos: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”pielāgota izteiksme”

Korelācijas filtra nosacījums

Korelācijas filtra nosacījuma objekts definē konkrētu nosacījumu, kas jāpārbauda pirms korelācijas operāciju izpildes.

Īpašība	Tips	Apraksts
type	integer	Nosacījuma tips. Iespējamās vērtības: 0 - vecā notikuma tags; 1 - jaunā notikuma tags; 2 - jaunā notikuma hostu grupa; 3 - notikuma tagu pāris; 4 - vecā notikuma taga vērtība; 5 - jaunā notikuma taga vērtība. Īpašības darbība: - *obligāts*
tag	string	Notikuma tags (vecais vai jaunais). Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”vecā notikuma tags”, ”jaunā notikuma tags”, ”vecā notikuma taga vērtība” vai ”jaunā notikuma taga vērtība”
groupid	ID	Hostu grupas ID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”jaunā notikuma hostu grupa”
oldtag	string	Vecā notikuma tags. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”notikuma tagu pāris”
newtag	string	Vecā notikuma tags. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”notikuma tagu pāris”
value	string	Notikuma taga (vecā vai jaunā) vērtība. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”vecā notikuma taga vērtība” vai ”jaunā notikuma taga vērtība”
formulaid	string	Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Tas var saturēt tikai lielos burtus. Lietotājam šis ID jādefinē, modificējot filtra nosacījumus, taču pēc tam, kad tie tiek pieprasīti atkārtoti, tas tiks ģenerēts no jauna.
operator	integer	Nosacījuma operators. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”jaunā notikuma hostu grupa”, ”vecā notikuma taga vērtība” vai ”jaunā notikuma taga vērtība”

Note:
Lai labāk saprastu, kā izmantot filtrus ar dažādiem izteiksmju tipiem, skatiet piemērus correlation.get un correlation.create metožu lapās.

Tālāk ir norādīti operatori un vērtības, kas tiek atbalstīti katram nosacījuma tipam.

Nosacījums	Nosacījuma nosaukums	Atbalstītie operatori	Sagaidāmā vērtība
2	Hostu grupa	=, <>	Hostu grupas ID.
4	Vecā notikuma taga vērtība	=, <>, like, not like	string
5	Jaunā notikuma taga vērtība	=, <>, like, not like	string

correlation.create

Apraksts

`object correlation.create(object/array correlations)`

Šī metode ļauj izveidot jaunas korelācijas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās korelācijas.

Papildus standarta korelācijas īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
operations	array	Izveidojamās korelācijas operācijas. Parametra darbība: - *obligāts*
filter	object	Korelācijas filtra objekts korelācijai. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto korelāciju ID īpašībā `correlationids`. Atgriezto ID secība atbilst nodoto korelāciju secībai.

Piemēri

Izveidot jaunu notikumu birku korelāciju

Izveidojiet korelāciju, izmantojot novērtēšanas metodi `AND/OR` ar vienu nosacījumu un vienu darbību. Pēc noklusējuma korelācija būs iespējota.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
```

```
            "1"
        ]
    },
    "id": 1
}
```

Pielāgotas izteiksmes filtra izmantošana

Izveidojiet korelāciju, kas izmantos pielāgotu filtra nosacījumu. Formulas ID "A" vai "B" ir izvēlēti patvaļīgi. Nosacījuma tips būs "Host group" ar operatoru "<>".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Korelācijas filtrs
- Korelācijas darbība

Avots

CCorrelation::create() failā *ui/include/classes/api/services/CCorrelation.php*.

correlation.delete

Apraksts

`object correlation.delete(array correlationids)`

Šī metode ļauj dzēst korelācijas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` dzēšamo korelāciju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto korelāciju ID zem `correlationids` īpašības.

Piemērs

Dzēst vairākas korelācijas

Dzēsiet divas korelācijas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Avots

CCorrelation::delete() failā *ui/include/classes/api/services/CCorrelation.php*.

correlation.get

Apraksts

`integer/array correlation.get(object parameters)`

Metode ļauj izgūt korelācijas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

(object) Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
correlationids	ID/array	Atgriezt tikai korelācijas ar norādītajiem ID.
selectFilter	query	Atgriezt `filter` īpašību ar korelācijas nosacījumiem.
selectOperations	query	Atgriezt `operations` īpašību ar korelācijas operācijām.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `correlationid`, `name`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

(integer/array) Atgriež:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt korelācijas

Iegūt visas konfigurētās korelācijas kopā ar korelāciju nosacījumiem un operācijām. Filtrs izmanto ”and/or” novērtēšanas tipu, tāpēc `formula` īpašība ir tukša un `eval_formula` tiek ģenerēta automātiski.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.get",
    "params": {
        "output": "extend",
        "selectOperations": "extend",
        "selectFilter": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Korelācijas filtrs
- Korelācijas operācija

Avots

CCorrelation::get() failā *ui/include/classes/api/services/CCorrelation.php*.

correlation.update

Apraksts

`object correlation.update(object/array correlations)`

Šī metode ļauj atjaunināt esošās korelācijas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Korelācijas rekvizīti, kas jāatjaunina.

Katram korelācijas objektam jābūt definētam rekvizītam `correlationid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta korelācijas rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
filter	object	Korelācijas filtrs objekts, ar ko aizstāt pašreizējo filtru.
operations	array	Korelācijas operācijas, ar ko aizstāt esošās operācijas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto korelāciju ID īpašībā `correlationids`.

Piemēri

Atspējot korelāciju

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
```

```
        "status": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Aizstāt nosacījumus, bet saglabāt novērtēšanas metodi

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "correlation.update",
    "params": {
        "correlationid": "1",
        "filter": {
            "conditions": [
                {
                    "type": 3,
                    "oldtag": "error",
                    "newtag": "ok"
                }
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "correlationids": [
            "1"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Korelācijas filtrs
- Korelācijas darbība

Avots

CCorrelation::update() failā *ui/include/classes/api/services/CCorrelation.php*.

Lietotāja makross

Šī klase ir paredzēta darbam ar hosta līmeņa un globālajiem lietotāja makrosiem.

Objektu atsauces:

- Globālais makross
- Hosta makross

- Makrosa konfigurācija

Pieejamās metodes:

- usermacro.create - izveidot jaunus hosta makrosus
- usermacro.createglobal - izveidot jaunus globālos makrosus
- usermacro.delete - dzēst hosta makrosus
- usermacro.deleteglobal - dzēst globālos makrosus
- usermacro.get - izgūt hosta un globālos makrosus
- usermacro.update - atjaunināt hosta makrosus
- usermacro.updateglobal - atjaunināt globālos makrosus

Lietotāja makro objekts

Tālāk norādītie objekti ir tieši saistīti ar `usermacro` API.

Globālais makross

Globālā makrosa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
globalmacroid	ID	Globālā makrosa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
description	string	Makrosa apraksts.
macro	string	Makrosa virkne. Īpašības darbība: - *obligāts* izveides operācijām
type	integer	Makrosa tips. Iespējamās vērtības: 0 - *(noklusējums)* Teksta makross; 1 - Slepenais makross; 2 - Vault noslēpums.
value	string	Makrosa vērtība. Īpašības darbība: - *tikai rakstāms*, ja `type` ir iestatīts uz "Secret macro" - *obligāts* izveides operācijām

Hosta makross

Hosta makrosa objekts definē makrosu, kas ir pieejams hostā, hosta prototipā vai veidnē. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
hostmacroid	ID	Makrosa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
automatic	integer	Nosaka, vai makrosu kontrolē atklāšanas kārtula. Iespējamās vērtības: 0 - *(noklusējums)* Makrosu pārvalda lietotājs; 1 - Makrosu pārvalda atklāšanas kārtula.
config	object/array	Lietotājam nav atļauts izveidot automātisku makrosu. Lai atjauninātu automātisku makrosu, tas ir jāpārveido par manuālu. Makrosa konfigurācija, kas nosaka, kā makross tiek attēlots Host Wizard.

Īpašība	Type	Apraksts
description	string	Makrosa apraksts.
hostid	ID	Hosta, hosta prototipa vai veidnes ID, kuram pieder makross. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām
macro	string	Makrosa virkne. Īpašības darbība: - *obligāts* izveides operācijām
type	integer	Makrosa tips. Iespējamās vērtības: 0 - *(noklusējums)* Teksta makross; 1 - Slepens makross; 2 - Vault noslēpums.
value	string	Makrosa vērtība. Īpašības darbība: - *tikai rakstāms*, ja `type` ir iestatīts uz ”Secret macro” - *obligāts* izveides operācijām

Makro konfigurācija

Makro konfigurācijas objekts nosaka, kā makro tiek attēlots Host Wizard.

Īpašība	Tips	Apraksts
type	integer	Makro ievades lauka tips. Iespējamās vērtības: 0 - Makro netiek izmantots Host Wizard; 1 - Teksta lauks; 2 - Saraksts; 3 - Izvēles rūtiņa. Īpašības darbība: - *obligāts*
label	string	Makro ievades lauka etiķete. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Textbox”, ”List” vai ”Checkbox”
description	text	Palīdzības teksts, kas tiek parādīts blakus makro ievades laukam. Atbalsta Markdown formatējumu.
priority	integer	Makro ievades lauka pozīcija makro sarakstā. Ja `priority` nav iestatīts, makro tiek pievienots negrupēto makro saraksta beigās.
required	integer	Norāda, ka makro ir obligāts. Iespējamās vērtības: 0 - Nav obligāts; 1 - Obligāts. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Textbox” vai ”List”
regex	string	Regulārā izteiksme lietotāja ievades pārbaudei teksta laukā. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Textbox”

Īpašība	Tips	Apraksts
section_name	string	Sakļaujamās sadaļas etiķete, kurā makro tiek grupēts. Ja `section_name` nav iestatīts, makro netiek grupēts. Negrupētie makro tiek parādīti vispirms; grupētie makro tiek parādīti zemāk, sakārtoti pēc to `priority` katrā sakļaujamajā sadaļā.
options	text	JSON virkne, kas definē saraksta vienumus vai izvēles rūtiņas vērtības. Sarakstiem: objektu masīvs ar `value` un `text` īpašībām. Piemērs: `[{"value": "http", "text": "HTTP"}, {"value": "https", "text": "HTTPS"}]` Izvēles rūtiņām: objekts ar `checked` un `unchecked` īpašībām. Piemērs: `{"checked": true, "unchecked": false}` Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”List” vai ”Checkbox”

usermacro.create

Apraksts

`object usermacro.create(object/array hostMacros)`

Šī metode ļauj izveidot jaunus hosta makrosus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Hosta makrosi, ko izveidot.

Metode pieņem hosta makrosus ar standarta hosta makrosu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto hosta makrosu ID īpašībā `hostmacroids`. Atgriezto ID secība atbilst nodoto hosta makrosu secībai.

Piemēri

Hosta makro izveide

Izveidojiet hosta makro ”{$SNMP_COMMUNITY}” ar vērtību ”public” hostam ”10198”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.create",
    "params": {
        "hostid": "10198",
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "11"
```

```
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::create() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.createglobal

Apraksts

`object usermacro.createglobal(object/array globalMacros)`

Šī metode ļauj izveidot jaunus globālos makrosus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Izveidojamie globālie makrosi.

Metode pieņem globālos makrosus ar standarta globālo makrosu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto globālo makro ID īpašībā `globalmacroids`. Atgriezto ID secība atbilst nodoto globālo makro secībai.

Piemēri

Globālas makrokomandas izveide

Izveidojiet globālu makrokomandu "{$SNMP_COMMUNITY}" ar vērtību "public".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.createglobal",
    "params":  {
        "macro": "{$SNMP_COMMUNITY}",
        "value": "public"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "6"
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::createGlobal() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.delete

Apraksts

`object usermacro.delete(array hostMacroIds)`

Šī metode ļauj dzēst hostu makrosus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo hosta makrosu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto hosta makro ID zem `hostmacroids` īpašības.

Piemēri

Vairāku hosta makrosu dzēšana

Dzēsiet divus hosta makrosus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.delete",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::delete() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.deleteglobal

Apraksts

`object usermacro.deleteglobal(array globalMacroIds)`

Šī metode ļauj dzēst globālos makrosus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet User roles papildinformācijai.

Parametri

`(array)` dzēšamo globālo makrosu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto globālo makro ID zem `globalmacroids` īpašības.

Piemēri

Vairāku globālo makrosu dzēšana

Dzēsiet divus globālos makrosus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.deleteglobal",
    "params": [
        "32",
        "11"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "32",
            "11"
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::deleteGlobal() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.get

Apraksts

`integer/array usermacro.get(object parameters)`

Metode ļauj izgūt hosta un globālos makrosus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
globalmacro	flag	Atgriezt globālos makrosus hosts makrosu vietā.
globalmacroids	ID/array	Atgriezt tikai globālos makrosus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai hosts makrosus, kas pieder hostiem vai veidnēm no norādītajām hostu grupām vai veidņu grupām.
hostids	ID/array	Atgriezt tikai makrosus, kas pieder norādītajiem hostiem vai veidnēm.
hostmacroids	ID/array	Atgriezt tikai hosts makrosus ar norādītajiem ID.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai hosts prototipa lietotāja makrosus, kas mantoti no veidnes.
selectHostGroups	query	Atgriezt hostu grupas, kurām hosts makross pieder, īpašībā `hostgroups`. Izmanto tikai, izgūstot hosts makrosus.

Parametrs	Tips	Apraksts
selectHosts	query	Atgriezt hostus, kuriem hosts makross pieder, īpašībā `hosts`.
selectTemplateGroups	query	Izmanto tikai, izgūstot hosts makrosus. Atgriezt veidņu grupas, kurām veidnes makross pieder, īpašībā `templategroups`.
selectTemplates	query	Izmanto tikai, izgūstot veidņu makrosus. Atgriezt veidnes, kurām hosts makross pieder, īpašībā `templates`.
sortfield	string/array	Izmanto tikai, izgūstot hosts makrosus. Kārtot rezultātu pēc norādītajām īpašībām.
countOutput	boolean	Iespējamās vērtības: `macro`. Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Hosta makrosu izgūšana hostam

Izgūstiet visus hosta makrosus, kas definēti hostam ”10198”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10198"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Hosta makrosu izgūšana veidnei

Izgūstiet hostam definētos makrosus veidnei ”10265”, kuru makrosa nosaukumā ir ”STATUS”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "hostids": "10265",
        "search": {
            "macro": "STATUS"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "hostmacroid": "6709",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.HOST}",
            "value": "",
            "description": "Apache statusa lapas hosta resursdatora nosaukums vai IP adrese.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "1",
                "section_name": "",
                "label": "Apache statusa hosts",
                "description": "Apache statusa lapas hosta resursdatora nosaukums vai IP adrese.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "814",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PATH}",
            "value": "server-status?auto",
            "description": "Apache statusa lapas URL ceļš.",
            "type": "0",
            "automatic": "0",
            "config": {
```

```
                "type": "1",
                "priority": "3",
                "section_name": "",
                "label": "Apache statusa lapas ceļš",
                "description": "Apache statusa lapas URL ceļš.",
                "required": "1",
                "regex": "",
                "options": []
            }
        },
        {
            "hostmacroid": "815",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.PORT}",
            "value": "80",
            "description": "Apache statusa lapas ports.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "1",
                "priority": "2",
                "section_name": "",
                "label": "Apache statusa lapas ports",
                "description": "Diapazonā no 1 līdz 65535 ieskaitot.",
                "required": "1",
                "regex": "^-?([0-9]+|(([0-9]+)\\.([0-9]+)))$",
                "options": []
            }
        },
        {
            "hostmacroid": "816",
            "hostid": "10265",
            "macro": "{$APACHE.STATUS.SCHEME}",
            "value": "http",
            "description": "Pieprasījuma shēma, kas var būt HTTP vai HTTPS.",
            "type": "0",
            "automatic": "0",
            "config": {
                "type": "2",
                "priority": "4",
                "section_name": "",
                "label": "Pieprasījuma shēma",
                "description": "Pieprasījuma shēma, kas var būt HTTP vai HTTPS.",
                "required": "0",
                "regex": "",
                "options": [
                    {
                        "value": "http",
                        "text": "HTTP"
                    },
                    {
                        "value": "https",
                        "text": "HTTPS"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```


Globālo makrosu izgūšana

Izgūstiet visus globālos makrosus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.get",
    "params": {
        "output": "extend",
        "globalmacro": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "globalmacroid": "6",
            "macro": "{$SNMP_COMMUNITY}",
            "value": "public",
            "description": "",
            "type": "0"
        }
    ],
    "id": 1
}
```

Avots

CUserMacro::get() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.update

Apraksts

`object usermacro.update(object/array hostMacros)`

Šī metode ļauj atjaunināt esošos hostu makrosus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` hosta makro īpašības, kas tiks atjauninātas.

Īpašībai `hostmacroid` jābūt definētai katram hosta makro, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto hosta makrosu ID īpašībā `hostmacroids`.

Piemēri

Hosta makro vērtības maiņa

Mainiet hosta makro vērtību uz ”public”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
```

```
        "value": "public"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Mainīt makro vērtību, ko izveidoja atklāšanas kārtula

Pārveidot ar atklāšanas kārtulu izveidotu ”automatic” makro par ”manual” un mainīt tā vērtību uz ”new-value”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.update",
    "params": {
        "hostmacroid": "1",
        "value": "new-value",
        "automatic": "0"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "hostmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::update() failā *ui/include/classes/api/services/CUserMacro.php*.

usermacro.updateglobal

Apraksts

`object usermacro.updateglobal(object/array globalMacros)`

Šī metode ļauj atjaunināt esošos globālos makrosus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Globālā makro īpašības, kas tiks atjauninātas.

Īpašībai `globalmacroid` jābūt definētai katram globālajam makro, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto globālo makro ID zem `globalmacroids` īpašības.

Piemēri

Globālā makro vērtības maiņa

Mainiet globālā makro vērtību uz ”public”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usermacro.updateglobal",
    "params": {
        "globalmacroid": "1",
        "value": "public"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "globalmacroids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CUserMacro::updateGlobal() failā *ui/include/classes/api/services/CUserMacro.php*.

Lietotājs

Šī klase ir paredzēta darbam ar lietotājiem.

Objektu atsauces:

- Lietotājs
- Medijs

Pieejamās metodes:

- user.checkauthentication - pārbaudīt un pagarināt lietotāja sesijas
- user.create - izveidot jaunus lietotājus
- user.delete - dzēst lietotājus
- user.get - izgūt lietotājus
- user.login - pieteikties API
- user.logout - izrakstīties no API
- user.provision - nodrošināt LDAP lietotājus
- user.resettotp - atiestatīt lietotāja TOTP noslēpumus
- user.unblock - atbloķēt lietotājus
- user.update - atjaunināt lietotājus

Lietotāja objekts

Tālāk norādītie objekti ir tieši saistīti ar `user` API.

Lietotājs

Lietotāja objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
userid	ID	Lietotāja ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
username	string	Lietotāja vārds. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* nodrošinātiem lietotājiem, ja lietotājs ir saistīts ar lietotāju direktoriju (`userdirectoryid` ir iestatīts uz derīgu vērtību, kas nav "0"), un lietotāju direktorijas nodrošināšanas statuss ir iespējots (`provision_status` no Lietotāju direktorijas objekta ir iestatīts uz "1"), un visu LDAP vai SAML nodrošināšanas autentifikācijas statuss ir iespējots (`ldap_jit_status` no Autentifikācijas objekta ir iestatīts uz "Enabled for configured LDAP IdPs" vai `saml_jit_status` no Autentifikācijas objekta ir iestatīts uz "Enabled for configured SAML IdPs")
passwd	string	Lietotāja parole. Šī parametra vērtība var būt tukša virkne, ja lietotājs ir saistīts ar lietotāju direktoriju. Īpašības darbība: - *tikai rakstāms*
roleid	ID	Lietotāja lomas ID. Ņemiet vērā, ka lietotāji bez lomas var pieteikties Zabbix tikai, izmantojot LDAP vai SAML autentifikāciju, ja viņu LDAP/SAML informācija atbilst Zabbix konfigurētajām lietotāju grupu kartēšanām.
attempt_clock	timestamp	Pēdējā neveiksmīgā pieteikšanās mēģinājuma laiks. Īpašības darbība: - *tikai lasāms*
attempt_failed	integer	Pēdējo neveiksmīgo pieteikšanās mēģinājumu skaits. Īpašības darbība: - *tikai lasāms*
attempt_ip	string	IP adrese, no kuras tika veikts pēdējais neveiksmīgais pieteikšanās mēģinājums. Īpašības darbība: - *tikai lasāms*
autologin	integer	Vai iespējot automātisko pieteikšanos. Iespējamās vērtības: 0 - *(noklusējums)* automātiskā pieteikšanās atspējota; 1 - automātiskā pieteikšanās iespējota.
autologout	string	Lietotāja sesijas darbības laiks. Pieņem sekundes un laika vienību ar sufiksu. Ja iestatīts uz 0s, sesijai nekad nebeigsies derīguma termiņš. Noklusējums: 15m.
lang	string	Lietotāja valodas valodas kods, piemēram, `en_US`. Noklusējums: `default` - sistēmas noklusējums.
name	string	Lietotāja vārds.

Īpašība	Tips	Apraksts
provisioned	integer	Vai lietotājs ir nodrošināts. Iespējamās vērtības: 0 - nav nodrošināts; 1 - nodrošināts. Īpašības darbība: - *tikai lasāms*
refresh	string	Automātiskās atsvaidzināšanas periods. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 90s, 1m, 1h). Noklusējums: 30s.
rows_per_page	integer	Objekta rindu skaits, ko rādīt vienā lapā. Noklusējums: 50.
surname	string	Lietotāja uzvārds.
theme	string	Lietotāja tēma. Iespējamās vērtības: `default` - *(noklusējums)* sistēmas noklusējums; `blue-theme` - Zila; `blue-classic-theme` - Zila (klasiskā); `dark-theme` - Tumša; `dark-classic-theme` - Tumša (klasiskā); `hc-light` - Augsta kontrasta gaišā; `hc-dark` - Augsta kontrasta tumšā.
ts_provisioned	timestamp	Laiks, kad tika veikta jaunākā nodrošināšanas operācija. Īpašības darbība: - *tikai lasāms* - *atbalstīts Super admin* tipa lietotājiem
url	string	Lapas URL, uz kuru novirzīt lietotāju pēc pieteikšanās.
userdirectoryid	ID	Tās lietotāju direktorijas ID, ar kuru lietotājs ir saistīts. Izmanto nodrošināšanai (izveidei vai atjaunināšanai), kā arī tāda lietotāja pieteikšanai, kurš ir saistīts ar lietotāju direktoriju. Pieteikšanās operācijām šīs īpašības vērtībai būs prioritāte pār lietotāju grupu `userdirectoryid` īpašību, kurām lietotājs pieder. Noklusējums: 0. Īpašības darbība: - *tikai lasāms* - *atbalstīts Super admin* tipa lietotājiem
timezone	string	Lietotāja laika josla, piemēram, `Europe/London`, `UTC`. Noklusējums: `default` - sistēmas noklusējums. Pilnam atbalstīto laika joslu sarakstam skatiet PHP dokumentāciju.

Medijs

Medija objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
mediaid	ID	Lietotāja medija ID. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
mediatypeid	ID	Lietotāja medijā izmantotā medija tipa ID. Īpašības darbība: - *obligāts*
sendto	string/array	Saņēmēja adrese, lietotājvārds vai cits identifikators. Ja Medija tipa `type` ir iestatīts uz "Email", vērtības tiek attēlotas kā masīvs. Citiem Mediju tipu tipiem vērtība tiek attēlota kā virkne. Īpašības darbība: - *obligāts*
active	integer	Vai medijs ir iespējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
severity	integer	Trigeru smaguma pakāpes, par kurām sūtīt paziņojumus. Iespējamās bitkartes vērtības: 1 - Nav klasificēts; 2 - Informācija; 4 - Brīdinājums; 8 - Vidējs; 16 - Augsts; 32 - Katastrofa. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 48 vērtībām Vidējs, Augsts un Katastrofa). Noklusējums: 63.
period	string	Laiks, kad var sūtīt paziņojumus, kā laika periods vai lietotāja makrosi, kas atdalīti ar semikolu. Noklusējums: 1-7,00:00-24:00.
provisioned	integer	Vai lietotājs ir provisioned. Iespējamās vērtības: 0 - nav provisioned; 1 - provisioned.
userdirectory_mediaid	ID	Lietotāju direktorija medija kartējuma ID provisioned medijam. Īpašības darbība: - *tikai lasāms* - *atbalstīts Super admin* tipa lietotājiem

user.checkAuthentication

Apraksts

`object user.checkAuthentication`

Šī metode pārbauda un pagarina lietotāja sesiju.

> **Attention:**
> Izsaucot metodi `user.checkAuthentication`, izmantojot parametru `sessionid`, lietotāja sesija pēc noklusējuma tiek pagarināta.

Parametri

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
extend	boolean	Vai pagarināt lietotāja sesiju. Noklusējuma vērtība: ”true”. Ja vērtība ir iestatīta uz ”false”, ir iespējams pārbaudīt lietotāja sesiju, to nepagarinot.
sessionid	string	Parametra darbība: - *atbalstīts*, ja ir iestatīts `sessionid` Lietotāja autentifikācijas marķieris.
token	string	Parametra darbība: - *obligāts*, ja `token` nav iestatīts Lietotāja API marķieris.
		Parametra darbība: - *obligāts*, ja `sessionid` nav iestatīts

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur informāciju par lietotāju.

Papildus standarta lietotāja īpašībām tiek atgriezta šāda informācija.

Īpašība	Tips	Apraksts
auth_type	integer	Noklusējuma autentifikācija lietotājam. Iespējamo vērtību sarakstu skatiet Authentication object īpašībā `authentication_type`.
debug_mode	integer	Vai lietotājam ir iespējots vai atspējots atkļūdošanas režīms. Iespējamo vērtību sarakstu skatiet User group object īpašībā `debug_mode`.
deprovisioned	boolean	Vai lietotājs pieder deprovisioned users group.
gui_access	string	Lietotāja autentifikācijas metode lietotāja saskarnei. Iespējamo vērtību sarakstu skatiet User group object īpašībā `gui_access`.
secret	string	Nejauša 32 rakstzīmju virkne. Tiek ģenerēta lietotāja pieteikšanās laikā. Īpašība `secret` netiek atgriezta, ja lietotāja sesija tiek pārbaudīta, izmantojot API tokenu.
sessionid	string	Autentifikācijas tokens, kas jāizmanto turpmākajos API pieprasījumos. Īpašība `sessionid` netiek atgriezta, ja lietotāja sesija tiek pārbaudīta, izmantojot API tokenu.
type	integer	Lietotāja tips. Iespējamo vērtību sarakstu skatiet Role object īpašībā `type`.
userip	string	Lietotāja IP adrese.

Piemēri

Pārbaudīt autentifikāciju, izmantojot autentifikācijas marķieri

Pārbaudiet un pagariniet lietotāja sesiju, izmantojot lietotāja autentifikācijas marķieri, un atgrieziet papildu informāciju par lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
```

```
    "params": {
        "sessionid": "673b8ba11562a35da902c66cf5c23fa2"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Pārbaudīt autentifikāciju, izmantojot API pilnvaru

Pārbaudiet lietotāja sesiju, izmantojot lietotāja API pilnvaru, un atgrieziet papildu informāciju par lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.checkAuthentication",
    "params": {
        "token": "00aff470e07c12d707e50d98cfe39edef9e6ec349c14728dbdfbc8ddc5ea3eae"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Avots

CUser::checkAuthentication() failā *ui/include/classes/api/services/CUser.php*.

user.create

Apraksts

`object user.create(object/array users)`

Šī metode ļauj izveidot jaunus lietotājus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Note:
Lietotāja paroles stiprums tiek validēts saskaņā ar paroļu politikas noteikumiem, ko definē Authentication API. Skatiet Authentication API, lai iegūtu vairāk informācijas.

Parametri

`(object/array)` Lietotāji, ko izveidot.

Papildus standarta lietotāja īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
usrgrps	array	Lietotāju grupas, kurām pievienot lietotāju. Lietotāju grupām jābūt definētai tikai `usrgrpid` īpašībai.
medias	array	Lietotāja datu nesēji, kas tiks izveidoti.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto lietotāju ID īpašībā `userids`. Atgriezto ID secība atbilst nodoto lietotāju secībai.

Piemēri

Lietotāja izveide

Izveidojiet jaunu lietotāju, pievienojiet viņu lietotāju grupai un izveidojiet viņam jaunu mediju.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Autentifikācija
- Medijs
- Lietotāju grupa
- Loma

Avots

CUser::create() failā *ui/include/classes/api/services/CUser.php*.

user.delete

Apraksts

`object user.delete(array users)`

Šī metode ļauj dzēst lietotājus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo lietotāju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto lietotāju ID īpašībā `userids`.

Piemēri

Vairāku lietotāju dzēšana

Dzēsiet divus lietotājus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.delete",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CUser::delete() failā *ui/include/classes/api/services/CUser.php*.

user.get

Apraksts

`integer/array user.get(object parameters)`

Metode ļauj izgūt lietotājus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

> **Note:**
> Pieprasot lietotāja medijus, atļaujas vai lomas informāciju, *Admin* un *User* tipa lietotāji var iegūt datus tikai par savu lietotāju. Piemēru skatiet sadaļā Lietotāju izgūšana kā *Admin*.

Parametrs	Tips	Apraksts
mediaids	ID/array	Atgriezt tikai tos lietotājus, kuri izmanto norādīto mediju.
mediatypeids	ID/array	Atgriezt tikai tos lietotājus, kuri izmanto norādītos mediju tipus.
userids	ID/array	Atgriezt tikai lietotājus ar norādītajiem ID.
usrgrpids	ID/array	Atgriezt tikai tos lietotājus, kuri pieder norādītajām lietotāju grupām.

Parametrs	Tips	Apraksts
getAccess	flag	Pievieno papildu informāciju par lietotāja atļaujām. Katram lietotājam pievieno šādas īpašības: `gui_access` - `(integer)` lietotāja lietotāja saskarne autentifikācijas metode. Iespējamo vērtību sarakstu skatiet lietotāju grupas objekta īpašībā `gui_access`. `debug_mode` - `(integer)` norāda, vai lietotājam ir iespējots atkļūdošanas režīms. Iespējamās vērtības: 0 - atkļūdošana atspējota, 1 - atkļūdošana iespējota. `users_status` - `(integer)` norāda, vai lietotājs ir atspējots. Iespējamās vērtības: 0 - lietotājs iespējots, 1 - lietotājs atspējots.
selectMedias	query	Atgriezt lietotāja izmantotos medijus īpašībā `medias`.
selectMediatypes	query	Atgriezt lietotāja izmantotos mediju tipus īpašībā `mediatypes`. Skatiet `mediatype.get`, lai uzzinātu par ierobežojumiem atkarībā no lietotāja tipa.
selectUsrgrps	query	Atgriezt lietotāju grupas, kurām lietotājs pieder, īpašībā `usrgrps`. Skatiet `usergroup.get`, lai uzzinātu par ierobežojumiem atkarībā no lietotāja tipa.
selectRole	query	Atgriezt lietotāja lomu īpašībā `role`.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Iespējamās lietotāja objekta īpašības *Admin* un *User* tipa lietotājiem, pieprasot datus par lietotājiem savā lietotāju grupā: `userid`, `name`, `surname`, `username`.
output	query	Atgriežamās lietotāja objekta īpašības. *Admin* un *User* tipa lietotāji var iegūt tikai šādas īpašības: - Savam lietotājam: `userid`, `attempt_clock`, `attempt_failed`, `attempt_ip`, `autologin`, `autologout`, `lang`, `name`, `provisioned`, `refresh`, `roleid`, `rows_per_page`, `surname`, `theme`, `timezone`, `url`, `username`. - Lietotājiem savā lietotāju grupā: `userid`, `name`, `surname`, `username`. Noklusējums: `extend`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalsta tikai `string` un `text` datu tipa īpašības. Iespējamās lietotāja objekta īpašības *Admin* un *User* tipa lietotājiem, pieprasot datus par lietotājiem savā lietotāju grupā: `name`, `surname`, `username`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `userid`, `username`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	

Parametrs	Tips	Apraksts
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Lietotāju izgūšana

Izgūst visus konfigurētos lietotājus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Lietotāju izgūšana kā *Admin*

Kā *Admin* tipa lietotājs izgūstiet detalizētus datus par savu lietotāju un ierobežotus datus par lietotājiem savā lietotāju grupā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": "extend",
        "getAccess": true,
        "selectMedias": "extend",
        "selectMediatypes": "extend",
        "selectUsrgrps": "extend",
        "selectRole": "extend"
    },
    "id": 1
}
```

Atbilde:


```
{
    "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
}
```

Lietotāja datu iegūšana

Iegūstiet lietotāja ar ID ”12” datus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.get",
    "params": {
        "output": ["userid", "username"],
        "selectRole": "extend",
        "userids": "12"
    },
    "id": 1
}
```

Atbilde:

```
{
  "jsonrpc": "2.0",
  "result": [
      {
          "userid": "12",
          "username": "John",
          "role": {
              "roleid": "5",
              "name": "Operator",
              "type": "1",
              "readonly": "0"
          }
      }
  ],
  "id": 1
}
```

Skatiet arī

- Medijs
- Medija tips
- Lietotāju grupa
- Loma

Avots

CUser::get() failā *ui/include/classes/api/services/CUser.php*.

user.login

Apraksts

`string/object user.login(object parameters)`

Šī metode ļauj pieteikties API un ģenerēt autentifikācijas marķieri.

Warning:
Izmantojot šo metodi, jums ir jāizsauc arī user.logout, lai novērstu liela skaita atvērtu sesiju ierakstu ģenerēšanu.

Attention:
Šī metode ir pieejama tikai neautentificētiem lietotājiem, kuri nepieder nevienai lietotāju grupai, kurai ir iespējota daudzfaktoru autentifikācija.

Parametri

`(object)` Parametri, kas satur lietotāja vārdu un paroli.

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
password	string	Lietotāja parole. Parametra darbība: - *obligāts*
username	string	Lietotāja vārds. Parametra darbība: - *obligāts*
userData	flag	Atgriezt informāciju par autentificēto lietotāju.

Atgrieztās vērtības

`(string/object)` Ja tiek izmantots parametrs `userData`, tiek atgriezts objekts, kas satur informāciju par autentificēto lietotāju.

Papildus standarta lietotāja īpašībām tiek atgriezta šāda informācija:

Īpašība	Tips	Apraksts
auth_type	integer	Noklusējuma autentifikācija lietotājam. Iespējamo vērtību sarakstu skatiet Authentication object īpašībā `authentication_type`.
debug_mode	integer	Vai lietotājam ir iespējots vai atspējots atkļūdošanas režīms. Iespējamo vērtību sarakstu skatiet User group object īpašībā `debug_mode`.
deprovisioned	boolean	Vai lietotājs pieder deprovisioned users group.
gui_access	string	Lietotāja autentifikācijas metode lietotāja saskarnei. Iespējamo vērtību sarakstu skatiet User group object īpašībā `gui_access`.
mfaid	integer	MFA method ID, kas jāizmanto lietotājam pieteikšanās laikā. Atgriež "0", ja MFA ir globāli atspējota vai atspējota visām lietotāju grupām, kurām lietotājs pieder.
secret	string	Nejauša 32 rakstzīmju virkne. Tiek ģenerēta lietotāja pieteikšanās laikā.
sessionid	string	Autentifikācijas marķieris, kas jāizmanto turpmākajos API pieprasījumos.
type	integer	Lietotāja tips. Iespējamo vērtību sarakstu skatiet Role object īpašībā `type`.
userip	string	Lietotāja IP adrese.

Note:
Ja lietotājs ir veiksmīgi autentificēts pēc viena vai vairākiem neveiksmīgiem mēģinājumiem, metode atgriezīs pašreizējās vērtības īpašībām `attempt_clock`, `attempt_failed` un `attempt_ip`, un pēc tam tās atiestatīs.

Ja parametrs `userData` netiek izmantots, metode atgriež autentifikācijas marķieri, kas ir nepieciešams autentifikācijai.

Piemēri

Lietotāja autentificēšana

Autentificējiet lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": "0424bd59b807674191e7d77572075f33",
    "id": 1
}
```

Autentificētā lietotāja informācijas pieprasīšana

Autentificējiet un atgrieziet papildu informāciju par lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.login",
    "params": {
        "username": "Admin",
        "password": "zabbix",
        "userData": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatīt arī

- user.logout

Avots

CUser::login() failā *ui/include/classes/api/services/CUser.php*.

user.logout

Apraksts

`string/object user.logout(array)`

Šī metode ļauj izrakstīties no API un padara pašreizējo autentifikācijas marķieri nederīgu.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Metode pieņem tukšu masīvu.

Atgrieztās vērtības

`(boolean)` Atgriež `true`, ja lietotājs ir veiksmīgi atteicies no sistēmas.

Piemēri

Izrakstīšanās

Izrakstīties no API.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "user.logout",
    "params": [],
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```


Skatiet arī

- user.login

Avots

CUser::login() failā *ui/include/classes/api/services/CUser.php*.

user.provision

Apraksts

`object user.provision(object/array users)`

Šī metode ļauj nodrošināt LDAP lietotājus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` nodrošināmo lietotāju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur nodrošināto lietotāju ID zem `userids` īpašības.

Piemēri

Vairāku lietotāju nodrošināšana

Nodrošiniet divus lietotājus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.provision",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CUser::provision() failā *ui/include/classes/api/services/CUser.php*.

user.resettotp

Apraksts

`object user.resettotp(object/array users)`

Šī metode ļauj atiestatīt lietotāja TOTP noslēpumus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Lietotāju ID, kuriem atiestatīt TOTP noslēpumus.

> **Note:**
> Norādīto lietotāju sesijas arī tiks dzēstas (izņemot lietotāju, kurš nosūta pieprasījumu).

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur to lietotāju ID, kuriem ir atiestatīti TOTP noslēpumi, īpašībā `userids`.

Piemēri

TOTP noslēpumu atiestatīšana vairākiem lietotājiem

Atiestatiet TOTP noslēpumus diviem lietotājiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.resettotp",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Skatiet arī

- MFA objekts

Avots

CUser::resettotp() failā *ui/include/classes/api/services/CUser.php*.

user.unblock

Apraksts

`object user.unblock(array userids)`

Šī metode ļauj atbloķēt lietotājus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` atbloķējamo lietotāju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atbloķēto lietotāju ID īpašībā `userids`.

Piemēri

Vairāku lietotāju atbloķēšana

Atbloķējiet divus lietotājus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.unblock",
    "params": [
        "1",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CUser::unblock() failā *ui/include/classes/api/services/CUser.php*.

user.update

Apraksts

`object user.update(object/array users)`

Šī metode ļauj atjaunināt esošos lietotājus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Note:
Lietotāja paroles stiprums tiek validēts saskaņā ar paroles politikas noteikumiem, ko definē Authentication API. Plašāku informāciju skatiet Authentication API.

Parametri

`(object/array)` Atjaunināmie lietotāja rekvizīti.

Katram lietotājam ir jābūt norādītam rekvizītam `userid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta lietotāja rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
current_passwd	string	Lietotāja pašreizējā parole. Šī parametra vērtība var būt tukša virkne, ja lietotājs ir saistīts ar lietotāju direktoriju. Parametra darbība: - *tikai rakstīšanai* - *obligāts*, ja User object rekvizīts `passwd` ir iestatīts un lietotājs maina savu paroli

Parametrs	Tips	Apraksts
usrgrps	array	Lietotāju grupas, ar kurām aizstāt esošās lietotāju grupas.
medias	array	Lietotāju grupām jābūt definētam tikai rekvizītam `usrgrpid`. Lietotāja datu nesēji, ar kuriem aizstāt esošos, neprovisionētos datu nesējus. Provisionētie datu nesēji var tikt izlaisti, atjauninot datu nesējus.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto lietotāju ID īpašībā `userids`.

Piemēri

Lietotāja pārdēvēšana

Pārdēvējiet lietotāju par John Doe.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "1",
        "name": "John",
        "surname": "Doe"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "1"
        ]
    },
    "id": 1
}
```

Lietotāja lomas maiņa

Mainiet lietotāja lomu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "user.update",
    "params": {
        "userid": "12",
        "roleid": "6"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userids": [
            "12"
        ]
    },
```

```
    "id": 1
}
```

Skatiet arī

- Autentifikācija

Avots

CUser::update() failā *ui/include/classes/api/services/CUser.php*.

Lietotāju direktorijs

Šī klase ir paredzēta darbam ar lietotāju direktorijiem.

Objektu atsauces:

- Lietotāju direktorijs
  - Mediju tipu kartējumi
  - Nodrošināšanas grupu kartējumi

Pieejamās metodes:

- userdirectory.create - izveidot jaunu lietotāju direktoriju
- userdirectory.delete - dzēst lietotāju direktoriju
- userdirectory.get - izgūt lietotāju direktoriju
- userdirectory.update - atjaunināt lietotāju direktoriju
- userdirectory.test - pārbaudīt lietotāju direktorija savienojumu

Lietotāju direktorija objekts

Tālāk norādītie objekti ir tieši saistīti ar `userdirectory` API.

Lietotāju direktorijs

Lietotāju direktorija objektam ir šādas īpašības.

Property	Type	Description
userdirectoryid	ID	Lietotāju direktorija ID. Ja lietotāju direktorijs tiek dzēsts, User object īpašības `userdirectoryid` vērtība tiek iestatīta uz ”0” visiem lietotājiem, kas ir saistīti ar dzēsto lietotāju direktoriju. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
idp_type	integer	Autentifikācijas protokola tips, ko identitātes nodrošinātājs izmanto lietotāju direktorijam. Ņemiet vērā, ka var pastāvēt tikai viens SAML tipa lietotāju direktorijs. Iespējamās vērtības: 1 - LDAP tipa lietotāju direktorijs; 2 - SAML tipa lietotāju direktorijs. Īpašības darbība: - *obligāts* izveides darbībām

Property	Type	Description
group_name	string	LDAP/SAML lietotāju direktorija atribūts, kas satur grupas nosaukumu, ko izmanto grupu kartēšanai starp LDAP/SAML lietotāju direktoriju un Zabbix. Piemērs: *cn* Īpašības darbība: - *obligāts*, ja `provision_status` ir iestatīts uz "Enabled" un Authentication object `saml_jit_status` ir iestatīts uz "Enabled for configured SAML IdPs"
user_username	string	LDAP/SAML lietotāju direktorija atribūts (arī SCIM atribūts, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"), kas satur lietotāja vārdu un tiek izmantots kā User object īpašības `name` vērtība, kad lietotājs tiek provisionēts. Piemēri: *cn*, *commonName*, *displayName*, *name*
user_lastname	string	LDAP/SAML lietotāju direktorija atribūts (arī SCIM atribūts, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"), kas satur lietotāja uzvārdu un tiek izmantots kā User object īpašības `surname` vērtība, kad lietotājs tiek provisionēts. Piemēri: *sn*, *surname*, *lastName*
provision_status	integer	Lietotāju direktorija provisionēšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* Disabled (šī lietotāju direktorija izveidoto lietotāju provisionēšana ir atspējota); 1 - Enabled (šī lietotāju direktorija izveidoto lietotāju provisionēšana ir iespējota; papildus tam jābūt iespējotam LDAP vai SAML provisionēšanas statusam (`ldap_jit_status` vai `saml_jit_status` no Authentication object)).
provision_groups	array	Provisionēšanas grupu kartējumu objektu masīvs LDAP/SAML lietotāju grupu šablona kartēšanai uz Zabbix lietotāju grupu un lietotāja lomu. Īpašības darbība: - *obligāts*, ja `provision_status` ir iestatīts uz "Enabled"
provision_media	array	Mediju tipu kartējumu objektu masīvs lietotāja LDAP/SAML mediju atribūtu (piemēram, e-pasta) kartēšanai uz Zabbix lietotāja medijiem paziņojumu sūtīšanai.
LDAP-specifiskas īpašības:		
name	string	Unikāls lietotāju direktorija nosaukums. Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
host	string	LDAP servera resursdatora nosaukums, IP vai URI. URI jāietver shēma (`ldap://` vai `ldaps://`), hosts un ports (neobligāti). Piemēri: *host.example.com* *127.0.0.1* *ldap://ldap.example.com:389* Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
port	integer	LDAP servera ports. Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"

Property	Type	Description
base_dn	string	LDAP lietotāju direktorija bāzes ceļš uz lietotāju kontiem. Piemēri: *ou=Users,dc=example,dc=org* *ou=Users,ou=system* (OpenLDAP gadījumā) *DC=company,DC=com* (Microsoft Active Directory gadījumā) *uid=%{user},dc=example,dc=com* (tiešai lietotāja piesaistei; vietturis "*%{user}*" ir obligāts) Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
search_attribute	string	LDAP lietotāju direktorija atribūts, pēc kura identificēt lietotāja kontu no pieteikšanās pieprasījumā sniegtās informācijas. Piemēri: *uid* (OpenLDAP gadījumā) *sAMAccountName* (Microsoft Active Directory gadījumā) Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
bind_dn	string	LDAP servera konts piesaistei un meklēšanai LDAP serverī. Tiešas lietotāja piesaistes un anonīmas piesaistes gadījumā `bind_dn` jābūt tukšam. Piemēri: *uid=ldap_search,ou=system* (OpenLDAP gadījumā) *CN=ldap_search,OU=user_group,DC=company,DC=com* (Microsoft Active Directory gadījumā) *CN=Admin,OU=Users,OU=Zabbix,DC=zbx,DC=local* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
bind_password	string	LDAP parole kontam, ko izmanto piesaistei un meklēšanai LDAP serverī. Tiešas lietotāja piesaistes un anonīmas piesaistes gadījumā `bind_password` jābūt tukšam. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
description	string	Lietotāju direktorija apraksts. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
group_basedn	string	LDAP lietotāju direktorija bāzes ceļš uz grupām; tiek izmantots, lai konfigurētu lietotāja dalības pārbaudi LDAP lietotāju direktorijā. Tiek ignorēts, provisionējot lietotāju, ja ir iestatīts `group_membership`. Piemērs: *ou=Groups,dc=example,dc=com* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"

Property	Type	Description
group_filter	string	Filtra virkne LDAP lietotāju direktorija grupu iegūšanai, kuru dalībnieks ir lietotājs; tiek izmantota, lai konfigurētu lietotāja dalības pārbaudi LDAP lietotāju direktorijā. Tiek ignorēts, provisionējot lietotāju, ja ir iestatīts `group_membership`. Atbalstītie `group_filter` vietturi: *%{attr}* - meklēšanas atribūts (aizstāts ar īpašības `search_attribute` vērtību); *%{groupattr}* - grupas atribūts (aizstāts ar īpašības `group_member` vērtību); *%{host}* - LDAP servera resursdatora nosaukums, IP vai URI (aizstāts ar īpašības `host` vērtību); *%{user}* - Zabbix lietotāja lietotājvārds. Noklusējums: *(%{groupattr}=%{user})* Piemēri: - *(member=uid=%{ref},ou=Users,dc=example,dc=com)* atbildīs "User1", ja LDAP grupas objekts satur atribūtu "*member*" ar vērtību "*uid=User1,ou=Users,dc=example,dc=com*", un atgriezīs grupu, kuras dalībnieks ir "User1"; - *(%{groupattr}=cn=%{ref},ou=Users,ou=Zabbix,DC=example,DC=com)* atbildīs "User1", ja LDAP grupas objekts satur atribūtu, kas norādīts īpašībā `group_member`, ar vērtību "*cn=User1,ou=Users,ou=Zabbix,DC=example,DC=com*", un atgriezīs grupu, kuras dalībnieks ir "User1".
group_member	string	Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" LDAP lietotāju direktorija atribūts, kas satur informāciju par grupas dalībniekiem; tiek izmantots, lai konfigurētu lietotāja dalības pārbaudi LDAP lietotāju direktorijā. Tiek ignorēts, provisionējot lietotāju, ja ir iestatīts `group_membership`.
group_membership	string	Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" LDAP lietotāju direktorija atribūts, kas satur informāciju par grupām, kurām lietotājs pieder. Piemērs: *memberOf*
search_filter	string	Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" Pielāgota filtra virkne, ko izmanto lietotāja atrašanai un autentificēšanai LDAP lietotāju direktorijā, pamatojoties uz pieteikšanās pieprasījumā sniegto informāciju. Atbalstītie `search_filter` vietturi: *%{attr}* - meklēšanas atribūta nosaukums (piemēram, *uid*, *sAMAccountName*); *%{user}* - Zabbix lietotāja lietotājvārds. Noklusējums: *(%{attr}=%{user})* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"

Property	Type	Description
start_tls	integer	LDAP servera konfigurācijas opcija, kas ļauj saziņu ar LDAP serveri aizsargāt, izmantojot Transport Layer Security (TLS). Ņemiet vērā, ka `start_tls` jābūt iestatītam uz "Disabled" hosts, kas izmanto `ldaps://` protokolu. Iespējamās vērtības: 0 - *(noklusējums)* Disabled; 1 - Enabled.
user_ref_attr	string	Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP" LDAP lietotāju direktorija atribūts, ko izmanto atsaucei uz lietotāja objektu. `user_ref_attr` vērtība tiek izmantota, lai iegūtu vērtības no norādītā atribūta lietotāju direktorijā un ievietotu tās *%{ref}* viettura vietā `group_filter` virknē. Piemēri: *cn*, *uid*, *member*, *uniqueMember* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type LDAP"
SAML-specifiskas īpašības:		
idp_entityid	string	URI, kas identificē identitātes nodrošinātāju un tiek izmantots saziņai ar identitātes nodrošinātāju SAML ziņojumos. Piemērs: *https://idp.example.com/idp* Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sp_entityid	string	URL vai jebkura virkne, kas identificē identitātes nodrošinātāja pakalpojuma nodrošinātāju. Piemēri: *https://idp.example.com/sp* *zabbix* Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
username_attribute	string	SAML lietotāju direktorija atribūts (arī SCIM atribūts, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"), kas satur lietotāja lietotājvārdu un autentificēšanas laikā tiek salīdzināts ar User object īpašības `username` vērtību. Piemēri: *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* Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sso_url	string	Identitātes nodrošinātāja SAML vienotās pierakstīšanās pakalpojuma URL, uz kuru Zabbix sūtīs SAML autentifikācijas pieprasījumus. Piemērs: *http://idp.example.com/idp/sso/saml* Īpašības darbība: - *obligāts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"

Property	Type	Description
slo_url	string	Identitātes nodrošinātāja SAML vienotās atteikšanās pakalpojuma URL, uz kuru Zabbix sūtīs SAML atteikšanās pieprasījumus. Piemērs: *https://idp.example.com/idp/slo/saml* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
encrypt_nameid	integer	Vai SAML name ID ir jāšifrē. Iespējamās vērtības: 0 - *(noklusējums)* Nešifrēt name ID; 1 - Šifrēt name ID. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
encrypt_assertions	integer	Vai SAML assertions ir jāšifrē. Iespējamās vērtības: 0 - *(noklusējums)* Nešifrēt assertions; 1 - Šifrēt assertions. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
nameid_format	string	SAML identitātes nodrošinātāja pakalpojuma nodrošinātāja Name ID formāts. Piemēri: *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* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
scim_status	integer	Vai SAML SCIM provisionēšana ir iespējota vai atspējota. Iespējamās vērtības: 0 - *(noklusējums)* SCIM provisionēšana ir atspējota; 1 - SCIM provisionēšana ir iespējota. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_assertions	integer	Vai SAML assertions ir jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - *(noklusējums)* Neparakstīt assertions; 1 - Parakstīt assertions. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_authn_requests	integer	Vai SAML AuthN pieprasījumi ir jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - *(noklusējums)* Neparakstīt AuthN pieprasījumus; 1 - Parakstīt AuthN pieprasījumus. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"

Property	Type	Description
sign_messages	integer	Vai SAML ziņojumi ir jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - *(noklusējums)* Neparakstīt ziņojumus; 1 - Parakstīt ziņojumus. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_logout_requests	integer	Vai SAML atteikšanās pieprasījumi ir jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - *(noklusējums)* Neparakstīt atteikšanās pieprasījumus; 1 - Parakstīt atteikšanās pieprasījumus. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
sign_logout_responses	integer	Vai SAML atteikšanās atbildes ir jāparaksta ar SAML parakstu. Iespējamās vērtības: 0 - *(noklusējums)* Neparakstīt atteikšanās atbildes; 1 - Parakstīt atteikšanās atbildes. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML"
idp_certificate	string	Service Provider (SP) sertifikāta saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai. Piemēri: *-----BEGIN CERTIFICATE----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai rakstāms*
sp_private_key	string	Service Provider (SP) privātās atslēgas saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai. Nodrošina drošu autentifikāciju un datu apmaiņu ar Identity Provider (IdP). Piemēri: *-----BEGIN CERTIFICATE----- MIIEvQIBADANBgkqhkiG9w0BA... ...more encoded data... -----END CERTIFICATE-----* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai rakstāms*

Property	Type	Description
sp_certificate	string	Service Provider (SP) sertifikāta saturs SAML vienotās pierakstīšanās (SSO) pakalpojuma iestatīšanai. Piemēri: *-----BEGIN CERTIFICATE-----* * MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA7... ...more encoded data... -----END CERTIFICATE-----* Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai rakstāms*
idp_certificate_hash	string	`idp_certificate` vērtības md5 jaucējkods. Tukšam `idp_certificate` tiks atgriezta tukša virkne. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai lasāms*
sp_private_key_hash	string	`sp_private_key` vērtības md5 jaucējkods. Tukšam `sp_private_key` tiks atgriezta tukša virkne. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai lasāms*
sp_certificate_hash	string	`sp_certificate` vērtības md5 jaucējkods. Tukšam `sp_certificate` tiks atgriezta tukša virkne. Īpašības darbība: - *atbalstīts*, ja `idp_type` ir iestatīts uz "User directory of type SAML" un `$SSO['CERT_STORAGE']` ir iestatīts uz `database` failā zabbix.conf.php - *tikai lasāms*

Mediju tipu kartējumi

Mediju tipu kartējumu objektam ir šādas īpašības.

Īpašība	Type	Apraksts
userdirectory_mediaid	ID	Mediju tipu kartējuma ID. Īpašības darbība: - *tikai lasāms*
name	string	Sarakstā redzamais mediju tipu kartējuma nosaukums. Īpašības darbība: - *obligāts*
mediatypeid	ID	Izveidojamā mediju tipa ID; tiek izmantots kā Media object īpašības `mediatypeid` vērtība. Īpašības darbība: - *obligāts*

Īpašība	Type	Apraksts
attribute	string	LDAP/SAML lietotāju direktorija atribūts (arī SCIM atribūts, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"), kas satur lietotāja mediju (piemēram, *user@example.com*), un tas tiek izmantots kā Media object īpašības `sendto` vērtība. Ja tas ir ietverts no LDAP/SAML identitātes nodrošinātāja saņemtajos datos un vērtība nav tukša, tas izraisīs medija izveidi nodrošinātajam lietotājam. Īpašības darbība: - *obligāts*
active	integer	Lietotāja medija īpašības `active` vērtība, kad medijs tiek izveidots nodrošinātajam lietotājam. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
severity	integer	Lietotāja medija īpašības `severity` vērtība, kad medijs tiek izveidots nodrošinātajam lietotājam. Noklusējums: 63.
period	string	Lietotāja medija īpašības `period` vērtība, kad medijs tiek izveidots nodrošinātajam lietotājam. Noklusējums: 1-7,00:00-24:00.

Provisionēšanas grupu kartējumi

Provisionēšanas grupu kartējumiem ir šādas īpašības.

Īpašība	Type	Apraksts
name	string	Pilns grupas nosaukums (piemēram, *Zabbix administrators*) LDAP/SAML lietotāju direktorijā (arī SCIM, ja `scim_status` ir iestatīts uz "SCIM provisioning is enabled"). Atbalsta aizstājējzīmi "*". Unikāls visos provisionēšanas grupu kartējumos. Īpašības darbība: - *required*
roleid	ID	Lietotājam piešķiramās lietotāja lomas ID. Ja atbilst vairāki provisionēšanas grupu kartējumi, lietotājam tiek piešķirta loma ar augstāko lietotāja tipu (*User*, *Admin* vai *Super admin*). Ja ir vairākas lomas ar vienādu lietotāja tipu, lietotājam tiek piešķirta pirmā loma (sakārtota alfabētiskā secībā). Īpašības darbība: - *required*
user_groups	array	Zabbix lietotāju grupu ID objektu masīvs. Katram objektam ir šādas īpašības: `usrgrpid` - (ID) lietotājam piešķiramās Zabbix lietotāju grupas ID. Ja atbilst vairāki provisionēšanas grupu kartējumi, lietotājam tiek piešķirtas visu atbilstošo kartējumu Zabbix lietotāju grupas. Īpašības darbība: - *required*

userdirectory.create

Apraksts

`object userdirectory.create(object/array userDirectory)`

Šī metode ļauj izveidot jaunus lietotāju direktorijus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam.

Parametri

`(object/array)` Izveidojamie lietotāju direktoriji.

Metode pieņem lietotāju direktorijus ar standarta lietotāju direktoriju īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto lietotāju direktoriju ID īpašībā `userdirectoryids`. Atgriezto ID secība atbilst nodoto lietotāju direktoriju secībai.

Piemēri

Lietotāju direktorija izveide

Izveidojiet lietotāju direktoriju, lai autentificētu lietotājus ar StartTLS, izmantojot LDAP. Ņemiet vērā, ka, lai autentificētu lietotājus, izmantojot LDAP, jābūt iespējotai LDAP autentifikācijai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Lietotāju direktorija izveide (JIT nodrošināšana iespējota)

Izveidojiet lietotāju direktoriju, lai autentificētu lietotājus, izmantojot LDAP (ar iespējotu JIT nodrošināšanu). Ņemiet vērā, ka, lai autentificētu lietotājus, izmantojot LDAP, jābūt iespējotai LDAP autentifikācijai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
        ]
    },
    "id": 1
}
```

Avots

CUserDirectory::create() failā *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.delete

Apraksts

`object userdirectory.delete(array userDirectoryIds)`

Šī metode ļauj dzēst lietotāju direktorijus. Lietotāju direktoriju nevar dzēst, ja tas tiek tieši izmantots vismaz vienai lietotāju grupai.
 Noklusējuma LDAP lietotāju direktoriju nevar dzēst, ja `authentication.ldap_configured` ir iestatīts uz 1 vai ja ir palikuši citi lietotāju direktoriji.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam.

Parametri

`(array)` dzēšamo lietotāju direktoriju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto lietotāju direktoriju ID īpašībā `userdirectoryids`.

Piemēri

Vairāku lietotāju direktoriju dzēšana

Dzēsiet divas lietotāju direktorijas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.delete",
    "params": [
        "2",
        "12"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2",
            "12"
        ]
    },
    "id": 1
}
```

Avots

CUserDirectory::delete() failā *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.get

Apraksts

`integer/array userdirectory.get(object parameters)`

Metode ļauj izgūt lietotāju direktorijus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
userdirectoryids	ID/array	Atgriezt tikai lietotāju direktorijus ar norādītajiem ID.
selectUsrgrps	query	Atgriezt `usrgrps` īpašību ar lietotāju grupām, kas saistītas ar lietotāju direktoriju. Atbalsta `count`.
selectProvisionMedia	query	Atgriezt `provision_media` īpašību ar mediju tipu kartējumiem, kas saistīti ar lietotāju direktoriju.
selectProvisionGroups	query	Atgriezt `provision_groups` īpašību ar nodrošināšanas grupu kartējumiem, kas saistīti ar lietotāju direktoriju.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `name`.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs. Atbalstītās īpašības: `userdirectoryid`, `idp_type`, `provision_status`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalstītās īpašības: `name`, `description`. SAML tipa lietotāju direktorijam gan `name`, gan `description` īpašībām būs tukša vērtība. Abas īpašības var mainīt ar `userdirectory.update` operāciju.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`

Piemēri

Lietotāju direktoriju izgūšana

Izgūstiet visas lietotāju direktorijas ar papildu īpašībām, kas parāda multivides tipu kartējumus un nodrošināšanas grupu kartējumus, kuri saistīti ar katru lietotāju direktoriju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.get",
    "params": {
        "output": "extend",
```

```
        "selectProvisionMedia": "extend",
        "selectProvisionGroups": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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 serveris",
            "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 administratori",
                    "roleid": "2",
                    "user_groups": [
                        {
                            "usrgrpid": "7"
                        },
                        {
                            "usrgrpid": "8"
                        }
                    ]
                }
            ]
        },
        {
            "userdirectoryid": "3",
            "idp_type": "1",
            "name": "LDAP API serveris #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
}
```

Skatiet arī

- Lietotāju grupa

Avots

CUserDirectory::get() failā *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.test

Apraksts

`object userdirectory.test(array userDirectory)`

Šī metode ļauj pārbaudīt lietotāju direktorija savienojuma iestatījumus.

Note:
Šī metode arī ļauj pārbaudīt, kā konfigurētie dati atbilst lietotāju direktorija iestatījumiem lietotāju nodrošināšanai (piemēram, kāda lietotāja loma, lietotāju grupas, lietotāja mediji tiks piešķirti lietotājam). Šāda veida pārbaudei API pieprasījums jāveic lietotāju direktorijam, kuram `provision_status` ir iestatīts kā enabled.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam.

Parametri

`(object)` Lietotāju direktorija rekvizīti.

Tā kā `userdirectory.get` API neatgriež lauku `bind_password`, ir jānorāda `userdirectoryid` un/vai `bind_password`.
Papildus standarta lietotāju direktorija rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
test_username	string	Lietotājvārds testēšanai lietotāju direktorijā.
test_password	string	Ar lietotājvārdu saistītā parole testēšanai lietotāju direktorijā.

Atgrieztās vērtības

`(bool)` Veiksmes gadījumā atgriež true.

Piemēri

Pārbaudīt lietotāju direktoriju esošam lietotājam

Pārbaudiet lietotāju direktoriju ”3” lietotājam ”user1”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": true,
    "id": 1
}
```

Testēt lietotāju direktoriju neesošam lietotājam

Testēt lietotāju direktoriju ”3” neesošam lietotājam ”user2”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "error": {
        "code": -32500,
        "message": "Application error.",
        "data": "Incorrect user name or password or account is temporarily blocked."
    },
    "id": 1
}
```

Pārbaudīt lietotāju direktoriju lietotāju nodrošināšanai

Pārbaudīt lietotāju direktoriju ”3”, lai noskaidrotu, kādi konfigurētie dati atbilst lietotāju direktorija iestatījumiem lietotāja ”user3” nodrošināšanai (piemēram, kāda lietotāja loma, lietotāju grupas, lietotāja mediji tiks piešķirti lietotājam).

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "username": "user3",
        "name": "John",
        "surname": "Doe",
        "medias": [],
        "usrgrps": [
            {
                "usrgrpid": "8"
            },
            {
                "usrgrpid": "7"
            }
        ],
        "roleid": "2",
        "userdirectoryid": "2"
    },
    "id": 1
}
```

Avots

CUserDirectory::test() failā *ui/include/classes/api/services/CUserDirectory.php*.

userdirectory.update

Apraksts

`object userdirectory.update(object/array userDirectory)`

Šī metode ļauj atjaunināt esošos lietotāju direktorijus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam.

Parametri

`(object/array)` Lietotāju direktorija rekvizīti, kas jāatjaunina.

Rekvizītam `userdirectoryid` jābūt definētam katrai lietotāju direktorijai, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto lietotāju direktoriju ID zem `userdirectoryids` īpašības.

Piemēri

Atjaunināt saistīšanas paroli lietotāju direktorijam

Iestatiet jaunu saistīšanas paroli lietotāju direktorijam.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "userdirectory.update",
    "params": {
        "userdirectoryid": "3",
        "bind_password": "newldappassword"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "3"
        ]
    },
    "id": 1
}
```

Atjaunināt lietotāju direktorija kartējumus

Atjauniniet nodrošināšanas grupu kartējumus un mediju tipu kartējumus lietotāju direktorijam ”2”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "userdirectoryids": [
            "2"
```

```
        ]
    },
    "id": 1
}
```

Avots

CUserDirectory::update() failā *ui/include/classes/api/services/CUserDirectory.php*.

Lietotāju grupa

Šī klase ir paredzēta darbam ar lietotāju grupām.

Objektu atsauces:

- Lietotāju grupa
- Atļauja
- Uz birkām balstīta atļauja

Pieejamās metodes:

- usergroup.create - izveidot jaunas lietotāju grupas
- usergroup.delete - dzēst lietotāju grupas
- usergroup.get - izgūt lietotāju grupas
- usergroup.update - atjaunināt lietotāju grupas

Lietotāju grupas objekts

Tālāk norādītie objekti ir tieši saistīti ar `usergroup` API.

Lietotāju grupa

Lietotāju grupas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
usrgrpid	ID	Lietotāju grupas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Lietotāju grupas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
debug_mode	integer	Vai atkļūdošanas režīms ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* atspējots; 1 - iespējots.
gui_access	integer	Lietotāja saskarnes autentifikācijas metode grupas lietotājiem. Iespējamās vērtības: 0 - *(noklusējums)* izmantot sistēmas noklusēto autentifikācijas metodi; 1 - izmantot iekšējo autentifikāciju; 2 - izmantot LDAP autentifikāciju; 3 - atspējot piekļuvi lietotāja saskarnei.
mfa_status	integer	Vai MFA ir iespējota vai atspējota grupas lietotājiem. Iespējamās vērtības: 0 - atspējota (visām konfigurētajām MFA metodēm); 1 - iespējota (visām konfigurētajām MFA metodēm).

Īpašība	Tips	Apraksts
mfaid	ID	MFA metode, kas tiek izmantota grupas lietotājiem. Īpašības darbība: - *atbalstīts Super admin* tipa lietotājiem - *atbalstīts*, ja Authentication object `mfa_status` ir iestatīts uz "Enabled"
users_status	integer	Vai lietotāju grupa ir iespējota vai atspējota. Atatdalītiem lietotājiem lietotāju grupa nevar būt iespējota. Iespējamās vērtības: 0 - *(noklusējums)* iespējota; 1 - atspējota.
userdirectoryid	ID	Autentifikācijai izmantotā lietotāju direktorija ID. Īpašības darbība: - *atbalstīts Super admin* tipa lietotājiem - *atbalstīts*, ja `gui_access` ir iestatīts uz "use the system default authentication method" vai "use LDAP authentication"

Atļauja

Atļaujas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
id	ID	Hostu grupas vai veidņu grupas ID, kurai pievienot atļauju. Īpašības darbība: - *obligāta*
permission	integer	Piekļuves līmenis hostu grupai vai veidņu grupai. Iespējamās vērtības: 0 - piekļuve liegta; 2 - tikai lasāma piekļuve; 3 - lasīšanas un rakstīšanas piekļuve. Īpašības darbība: - *obligāta* izveides darbībām

Uz birkām balstīta atļauja

Uz birkām balstītam atļaujas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
groupid	ID	Hostu grupas ID, kurai pievienot atļauju. Īpašības darbība: - *obligāts*
tag	string	Birkas nosaukums.
value	string	Birkas vērtība.

usergroup.create

Apraksts

`object usergroup.create(object/array userGroups)`

Šī metode ļauj izveidot jaunas lietotāju grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās lietotāju grupas.

Papildus standarta lietotāju grupu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hostgroup_rights	object/array	Hostu grupu atļaujas, ko piešķirt lietotāju grupai.
templategroup_rights	object/array	Veidņu grupu atļaujas, ko piešķirt lietotāju grupai.
tag_filters	array	Uz birkām balstītas atļaujas, ko piešķirt lietotāju grupai.
users	object/array	Lietotāji, ko pievienot lietotāju grupai. Lietotājam jābūt definētai tikai `userid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto lietotāju grupu ID īpašībā `usrgrpids`. Atgriezto ID secība atbilst nodoto lietotāju grupu secībai.

Piemēri

Lietotāju grupas izveide

Izveidojiet lietotāju grupu *Operation managers* ar liegtu piekļuvi hostu grupai "2" un pievienojiet tai lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.create",
    "params": {
        "name": "Operation managers",
        "hostgroup_rights": {
            "id": "2",
            "permission": 0
        },
        "users": [
            {
                "userid": "12"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Atļauja

Avots

CUserGroup::create() failā *ui/include/classes/api/services/CUserGroup.php*.

usergroup.delete

Apraksts

`object usergroup.delete(array userGroupIds)`

Šī metode ļauj dzēst lietotāju grupas.

Attention:
Atatdalīto lietotāju grupa (lietotāju grupa, kas norādīta `disabled_usrgrpid` laukam sadaļā Autentifikācija) nevar tikt dzēsta.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo lietotāju grupu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto lietotāju grupu ID īpašībā `usrgrpids`.

Piemēri

Vairāku lietotāju grupu dzēšana

Dzēsiet divas lietotāju grupas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.delete",
    "params": [
        "20",
        "21"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "20",
            "21"
        ]
    },
    "id": 1
}
```

Avots

CUserGroup::delete() failā *ui/include/classes/api/services/CUserGroup.php*.

usergroup.get

Apraksts

`integer/array usergroup.get(object parameters)`

Metode ļauj iegūt lietotāju grupas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
mfaids	ID/array	Atgriezt tikai lietotāju grupas ar norādītajām MFA metodēm. Parametra darbība: - *atbalstīts Super admin* tipa lietotājiem
mfa_status	integer	Atgriezt tikai lietotāju grupas ar norādīto MFA statusu. Atbalstīto statusu sarakstu skatiet lietotāju grupas lapā.
status	integer	Atgriezt tikai lietotāju grupas ar norādīto statusu. Atbalstīto statusu sarakstu skatiet lietotāju grupas lapā.
userids	ID/array	Atgriezt tikai tās lietotāju grupas, kurās ir norādītie lietotāji.
usrgrpids	ID/array	Atgriezt tikai lietotāju grupas ar norādītajiem ID.
selectTagFilters	query	Atgriezt lietotāju grupas uz birkām balstītās atļaujas īpašībā `tag_filters`.
selectUsers	query	Atgriezt lietotājus no lietotāju grupas īpašībā `users`. Skatiet `user.get`, lai uzzinātu par ierobežojumiem atkarībā no lietotāja tipa.
selectHostGroupRights	query	Atgriezt lietotāju grupas hostu grupu atļaujas īpašībā `hostgroup_rights`. Hostu grupu piekļuves līmeņu sarakstu skatiet lietotāju grupas lapā.
selectTemplateGroupRights	query	Atgriezt lietotāju grupas veidņu grupu atļaujas īpašībā `templategroup_rights`. Veidņu grupu piekļuves līmeņu sarakstu skatiet lietotāju grupas lapā.
limitSelects	integer	Ierobežo apakšatlases vaicājumu atgriezto ierakstu skaitu.
output	query	Atgriežamās lietotāju grupas objekta īpašības. *Admin* un *User* tipa lietotāji var iegūt tikai šādas īpašības: `usrgrpid`, `name`, `gui_access`, `users_status`, `debug_mode`, `mfa_status`. Noklusējums: `extend`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `usrgrpid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iespējoto lietotāju grupu izgūšana

Izgūst visas iespējotās lietotāju grupas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.get",
    "params": {
        "output": "extend",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Lietotājs

Avots

CUserGroup::get() failā *ui/include/classes/api/services/CUserGroup.php*.

usergroup.update

Apraksts

`object usergroup.update(object/array userGroups)`

Šī metode ļauj atjaunināt esošās lietotāju grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Atjaunināmās lietotāju grupas īpašības.

Īpašībai `usrgrpid` jābūt definētai katrai lietotāju grupai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta lietotāju grupas īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hostgroup_rights	object/array	Hostu grupu atļaujas, kas aizstās lietotāju grupai pašlaik piešķirtās atļaujas.
templategroup_rights	object/array	Veidņu grupu atļaujas, kas aizstās lietotāju grupai pašlaik piešķirtās atļaujas.
tag_filters	array	Uz birkām balstītas atļaujas, kas aizstās lietotāju grupai pašlaik piešķirtās atļaujas.
users	object/array	Lietotāji, kas aizstās lietotāju grupai pašlaik piešķirtos lietotājus. Lietotājam jābūt definētai tikai īpašībai `userid`.

Atgrieztās vērtības

(object) Atgriež objektu, kas satur atjaunināto lietotāju grupu ID īpašībā usrgrpids.

Piemēri

Lietotāju grupas iespējošana un atļauju atjaunināšana

Iespējojiet lietotāju grupu un piešķiriet tai lasīšanas-rakstīšanas piekļuvi hostu grupām ”2” un ”4”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "usergroup.update",
    "params": {
        "usrgrpid": "17",
        "users_status": "0",
        "hostgroup_rights": [
            {
                "id": "2",
                "permission": 3
            },
            {
                "id": "4",
                "permission": 3
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "usrgrpids": [
            "17"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Atļauja

Avots

CUserGroup::update() failā *ui/include/classes/api/services/CUserGroup.php*.

LLD noteikuma prototips

Šī klase ir paredzēta darbam ar zema līmeņa atklāšanas noteikumu prototipiem.

Objektu atsauces:

- LLD noteikuma prototips
  - HTTP galvene
  - HTTP vaicājuma lauks
- LLD noteikuma prototipa filtrs
  - LLD noteikuma prototipa filtra nosacījums
- LLD prototipa makrosa ceļš
- LLD noteikuma prototipa priekšapstrāde
- LLD noteikuma prototipa ignorēšanas
  - LLD noteikuma prototipa ignorēšanas filtrs
    - LLD noteikuma prototipa ignorēšanas filtra nosacījums
  - LLD noteikuma prototipa ignorēšanas operācija
    - LLD noteikuma prototipa ignorēšanas operācijas statuss
    - LLD noteikuma prototipa ignorēšanas operācijas atklāšana

    * LLD noteikuma prototipa ignorēšanas operācijas periods
    * LLD noteikuma prototipa ignorēšanas operācijas vēsture
    * LLD noteikuma prototipa ignorēšanas operācijas tendences
    * LLD noteikuma prototipa ignorēšanas operācijas nopietnība
    * LLD noteikuma prototipa ignorēšanas operācijas tags
    * LLD noteikuma prototipa ignorēšanas operācijas veidne
    * LLD noteikuma prototipa ignorēšanas operācijas inventārs

Pieejamās metodes:

- discoveryruleprototype.create - izveidot jaunus LLD noteikumu prototipus
- discoveryruleprototype.delete - dzēst LLD noteikumu prototipus
- discoveryruleprototype.get - izgūt LLD noteikumu prototipus
- discoveryruleprototype.update - atjaunināt LLD noteikumu prototipus

LLD noteikuma prototipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `discoveryruleprototype` API.

LLD kārtulas prototips

Zema līmeņa atklāšanas kārtulas prototipa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
itemid	ID	LLD kārtulas prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
ruleid	ID	Vecākās LLD kārtulas/LLD kārtulas prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* izveides operācijām
delay	string	LLD kārtulas prototipa atjaunināšanas intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”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), vai ja `type` ir iestatīts uz ”Zabbix agent (active)” (7) un `key_` nesatur ”mqtt.get”
hostid	ID	Hosta ID, kuram pieder LLD kārtulas prototips. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām

Īpašība	Tips	Apraksts
flags	integer	LLD kārtulas prototipa izcelsme. Iespējamās vērtības: 3 - zema līmeņa atklāšanas kārtulas prototips; 7 - atklāts zema līmeņa atklāšanas kārtulas prototips. Īpašības darbība: - *tikai lasāms*
interfaceid	ID	LLD kārtulas prototipa hosta saskarnes ID. Īpašības darbība: - *obligāts*, ja LLD kārtulas prototips pieder hostam un `type` ir iestatīts uz ”Zabbix agent”, ”IPMI agent”, ”JMX agent” vai ”SNMP agent” - *atbalstīts*, ja LLD kārtulas prototips pieder hostam un `type` ir iestatīts uz ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent”
key_	string	LLD kārtulas prototipa atslēga. Nepieciešams vismaz viens LLD makross. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
name	string	LLD kārtulas prototipa nosaukums. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
type	integer	LLD kārtulas prototipa tips. Iespējamās vērtības: 0 - Zabbix aģents; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix aģents (active); 10 - External check; 11 - Database monitor; 12 - IPMI aģents; 13 - SSH aģents; 14 - TELNET aģents; 16 - JMX aģents; 18 - atkarīgais vienums; 19 - HTTP aģents; 20 - SNMP aģents; 21 - Script; 22 - Browser; 23 - Nested. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
url	string	URL virkne. Atbalsta lietotāja makrosus, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
allow_traps	integer	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. Iespējamās vērtības: 0 - *(noklusējums)* Neatļaut pieņemt ienākošos datus; 1 - Atļaut pieņemt ienākošos datus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent”
authtype	integer	Autentifikācijas metode. Iespējamās vērtības, ja `type` ir iestatīts uz ”SSH agent”: 0 - *(noklusējums)* parole; 1 - publiskā atslēga. Iespējamās vērtības, ja `type` ir iestatīts uz ”HTTP agent”: 0 - *(noklusējums)* nav; 1 - basic; 2 - NTLM. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
description	string	LLD kārtulas prototipa apraksts.
error	string	Kļūdas teksts, ja ir problēmas ar LLD kārtulas prototipa vērtības atjaunināšanu. Īpašības darbība: - *tikai lasāms*
follow_redirects	integer	Sekot atbildes pāradresācijām datu aptaujas laikā. Iespējamās vērtības: 0 - Nesekot pāradresācijām; 1 - *(noklusējums)* Sekot pāradresācijām. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ipmi_sensor	string	IPMI sensors. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”IPMI agent” un `key_` nav iestatīts uz ”ipmi.get” - *atbalstīts*, ja `type` ir iestatīts uz ”IPMI agent” - *tikai lasāms* mantotajiem objektiem
jmx_endpoint	string	JMX aģenta pielāgota savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent”

Īpašība	Tips	Apraksts
lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks dzēsti. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Noklusējums: 7d.
lifetime_type	integer	Scenārijs zaudēto LLD resursu dzēšanai. Iespējamās vērtības: 0 - *(noklusējums)* Dzēst pēc lifetime sliekšņa sasniegšanas; 1 - Nedzēst; 2 - Dzēst nekavējoties.
enabled_lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks atspējoti. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Noklusējums: 0.
enabled_lifetime_type	integer	Scenārijs zaudēto LLD resursu atspējošanai. Iespējamās vērtības: 0 - Atspējot pēc lifetime sliekšņa sasniegšanas; 1 - Neatspējot; 2 - *(noklusējums)* Atspējot nekavējoties.
master_itemid	ID	Galvenā vienuma ID. Atklāšanas kārtula nevar būt galvenais vienums citai atklāšanas kārtulai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotajiem objektiem
output_format	integer	Vai atbilde jākonvertē uz JSON. Iespējamās vērtības: 0 - *(noklusējums)* Saglabāt neapstrādātu; 1 - Konvertēt uz JSON. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
params	string	Papildu parametri atkarībā no LLD kārtulas prototipa tipa: - izpildāmais skripts SSH un Telnet LLD kārtulām; - SQL vaicājums datubāzes monitora LLD kārtulām; - formula aprēķinātajām LLD kārtulām; - skripts Script un Browser LLD kārtulām. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”Script” vai ”Browser”)
parameters	object/array	Papildu parametri, ja `type` ir iestatīts uz ”Script” vai ”Browser”. Objektu masīvs ar `name` un `value` īpašībām, kur `name` jābūt unikālam. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem
password	string	Parole autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”JMX agent” un `username` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)

Īpašība	Tips	Apraksts
post_type	integer	`posts` īpašībā saglabātā POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* Neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "HTTP agent" un `post_type` ir iestatīts uz "JSON data" vai "XML data" - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" un `post_type` ir iestatīts uz "Raw data" - *tikai lasāms* mantotajiem objektiem
privatekey	string	Privātās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent" un `authtype` ir iestatīts uz "public key"
publickey	string	Publiskās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent" un `authtype` ir iestatīts uz "public key"
query_fields	array	Vaicājuma lauku masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem
retrieve_mode	integer	Kura atbildes daļa jāsaglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz "GET", "POST" vai "PUT": 0 - *(noklusējums)* Pamatteksts; 1 - Galvenes; 2 - Tiks saglabāts gan pamatteksts, gan galvenes. Iespējamās vērtības, ja `request_method` ir iestatīts uz "HEAD": 1 - Galvenes. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
snmp_oid	string	SNMP OID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotajiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
state	integer	LLD kārtulas prototipa stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - neatbalstīts. Īpašības darbība: - *tikai lasāms*
status	integer	LLD kārtulas prototipa statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots LLD kārtulas prototips; 1 - atspējots LLD kārtulas prototips.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalsta arī lietotāja makrosus kā daļu no ar komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
templateid	ID	Vecākās veidnes LLD kārtulas prototipa ID. Īpašības darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: ”” - izmantot starpniekservera/globālos iestatījumus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Zabbix agent” (0), ”Simple check” (3) un `key_` nesākas ar ”`vmware.`” un ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) un `snmp_oid` sākas ar ”`walk[`” vai ”`get[`”, ”Script” (21), ”Browser” (22) - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
trapper_hosts	string	Atļautie hosti. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Zabbix trapper”, vai ja `type` ir iestatīts uz ”HTTP agent” un `allow_traps` ir iestatīts uz ”Allow to accept incoming data”
username	string	Lietotājvārds autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent”, ”TELNET agent”, vai ja `type` ir iestatīts uz ”JMX agent” un `password` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”Database monitor” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
uuid	string	Universāls unikāls identifikators, ko izmanto importēto LLD kārtulu prototipu sasaistīšanai ar jau esošajiem. Tiek ģenerēts automātiski, ja nav norādīts. Īpašības darbība: - *atbalstīts*, ja LLD kārtulas prototips pieder veidnei
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

HTTP galvene

Galvenes objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	HTTP galvenes nosaukums. Īpašības darbība: - *obligāta*
value	string	Galvenes vērtība. Īpašības darbība: - *obligāta*

HTTP vaicājuma lauks

Vaicājuma lauka objekts definē nosaukumu un vērtību, ko izmanto URL parametra norādīšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāts*
value	string	Parametra vērtība. Īpašības darbība: - *obligāts*

LLD noteikuma prototipa filtrs

LLD noteikuma prototipa filtra objekts definē nosacījumu kopu, ko var izmantot atklāto objektu filtrēšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
conditions	object/array	Filtra nosacījumu kopa, ko izmantot rezultātu filtrēšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - Un/Vai; 1 - Un; 2 - Vai; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti filtra nosacījumos: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”custom expression”

LLD noteikuma prototipa filtra nosacījums

LLD noteikuma prototipa filtra nosacījuma objekts definē atsevišķu pārbaudi, kas jāveic LLD makro vērtībai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
macro	string	LLD makro, kuram jāveic pārbaude. Īpašības darbība: - *obligāta*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *obligāta*, ja `operator` ir iestatīts uz ”atbilst regulārajai izteiksmei” vai ”neatbilst regulārajai izteiksmei”

Īpašība	Tips	Apraksts
formulaid	string	Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Tas var saturēt tikai lielos burtus. ID lietotājam jānorāda, modificējot filtra nosacījumus, taču pēc tam, pieprasot tos atkārtoti, tas tiks ģenerēts no jauna. Īpašības darbība: - *obligāta*, ja LLD noteikuma prototipa filtra objekta `evaltype` ir iestatīts uz ”pielāgota izteiksme”
operator	integer	Nosacījuma operators. Iespējamās vērtības: 8 - *(noklusējums)* atbilst regulārajai izteiksmei; 9 - neatbilst regulārajai izteiksmei; 12 - eksistē; 13 - neeksistē.

Note:
Lai labāk saprastu, kā izmantot filtrus ar dažādiem izteiksmju tipiem, skatiet piemērus metožu discoveryruleprototype.get un discoveryruleprototype.create lapās.

LLD makro ceļš

LLD makro ceļam ir šādas īpašības:

Īpašība	Type	Apraksts
lld_macro	string	LLD makro. Īpašības darbība: - *required*
path	string	Selektors vērtībai, kas tiks piešķirta atbilstošajam makro. Īpašības darbība: - *required*

LLD kārtulas prototipa priekšapstrāde

LLD kārtulas prototipa priekšapstrādes objektam ir šādas īpašības.

Īpašība	Type	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 5 - Regulārā izteiksme; 11 - XML XPath; 12 - JSONPath; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - Pārbaudīt kļūdu JSON; 17 - Pārbaudīt kļūdu XML; 20 - Atmest nemainītu ar sirdspukstu; 21 - JavaScript; 23 - Prometheus uz JSON; 24 - CSV uz JSON; 25 - Aizstāt; 27 - XML uz JSON; 28 - SNMP walk vērtība; 29 - SNMP walk uz JSON; 30 - SNMP get vērtība. Īpašības darbība: - *obligāta*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri tiek atdalīti ar jaunās rindas (`\n`) rakstzīmi. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Atmest nemainītu ar sirdspukstu” (20), ”JavaScript” (21), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Aizstāt” (25), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)
error_handler	integer	Darbības tips, ko izmanto priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”XML uz JSON” (27), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādātāja parametri. Īpašības darbība: - *obligāta*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Tālāk norādīti parametri un kļūdu apstrādātāji, kas tiek atbalstīti katram priekšapstrādes tipam.

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
5	Regulārā izteiksme	raksts[1]	izvade[2]		0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
11	XML XPath	ceļš[3]			0, 1, 2, 3
12	JSONPath	ceļš[3]			0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	raksts[1]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	raksts[1]			0, 1, 2, 3
16	Pārbaudīt kļūdu JSON	ceļš[3]			0, 1, 2, 3
17	Pārbaudīt kļūdu XML	ceļš[3]			0, 1, 2, 3
20	Atmest nemainītu ar sirdspukstu	sekundes^4, 5^			
21	JavaScript	skripts[2]			
23	Prometheus uz JSON	raksts^5, 6^			0, 1, 2, 3
24	CSV uz JSON	rakstzīme[2]	rakstzīme[2]	0,1	0, 1, 2, 3
25	Aizstāt	meklējamā virkne[2]	aizstāšana[2]		
27	XML uz JSON				0, 1, 2, 3
28	SNMP walk vērtība	OID[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[7]	Lauka nosaukums[2]	OID prefikss[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS			0, 1, 2, 3

[1] regulārā izteiksme
 [2] virkne
 [3] JSONPath vai XML XPath
 [4] pozitīvs vesels skaitlis (ar laika sufiksu atbalstu, piemēram, 30s, 1m, 2h, 1d)
 [5] lietotāja makross
 [6] Prometheus raksts, kas atbilst sintaksei: `<metric name>{<label name>="<label value>", ...} == <value>`. Katrs Prometheus raksta komponents (metrika, etiķetes nosaukums, etiķetes vērtība un metrikas vērtība) var būt lietotāja makross.
 [7] Atbalsta vairākus "Lauka nosaukums,OID prefikss,Formāts ieraksti" ierakstus, kas atdalīti ar jaunās rindas rakstzīmi.

LLD noteikuma prototipa pārrakstīšanas

LLD noteikuma prototipa pārrakstīšanas objekts definē noteikumu kopu (filtrus, nosacījumus un operācijas), kas tiek izmantoti, lai pārrakstītu dažādu prototipu objektu īpašības. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
name	string	Unikāls pārrakstīšanas nosaukums. Īpašības darbība: - *obligāta*
step	integer	Unikāls pārrakstīšanas secības numurs. Īpašības darbība: - *obligāta*
stop	integer	Pārtraukt nākamo pārrakstīšanu apstrādi, ja ir atbilstība. Iespējamās vērtības: 0 - *(noklusējums)* nepārtraukt pārrakstīšanu apstrādi; 1 - pārtraukt pārrakstīšanu apstrādi, ja filtrs atbilst.
filter	object	Pārrakstīšanas filtrs.
operations	object/array	Pārrakstīšanas operācijas.

LLD noteikuma prototipa ignorēšanas filtrs

LLD noteikuma prototipa ignorēšanas filtra objekts definē nosacījumu kopu, kuru atbilstības gadījumā atklātajam objektam tiek piemērota ignorēšana. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
conditions	object/array	Ignorēšanas filtra nosacījumu kopa, ko izmantot atklāto objektu atbilstības noteikšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Ignorēšanas filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - And/Or; 1 - And; 2 - Or; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota ignorēšanas filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem ignorēšanas filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*

Īpašība	Tips	Apraksts
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota, lai novērtētu ignorēšanas filtru nosacījumus ar pielāgotu izteiksmi. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem ignorēšanas filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti ignorēšanas filtra nosacījumos: nevienu nosacījumu nedrīkst atstāt neizmantotu vai izlaistu. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”custom expression”

LLD noteikuma prototipa ignorēšanas filtra nosacījums

LLD noteikuma prototipa ignorēšanas filtra nosacījuma objekts definē atsevišķu pārbaudi, kas jāveic LLD makro vērtībai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
macro	string	LLD makro, kuram jāveic pārbaude. Īpašības darbība: - *obligāta*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *obligāta*, ja `operator` ir iestatīts uz ”atbilst regulārajai izteiksmei” vai ”neatbilst regulārajai izteiksmei”
formulaid	string	Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Tas drīkst saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču pēc tam, pieprasot tos atkārtoti, tas tiks ģenerēts no jauna. Īpašības darbība: - *obligāta*, ja LLD noteikuma prototipa ignorēšanas filtra objekta `evaltype` ir iestatīts uz ”pielāgota izteiksme”
operator	integer	Nosacījuma operators. Iespējamās vērtības: 8 - *(noklusējums)* atbilst regulārajai izteiksmei; 9 - neatbilst regulārajai izteiksmei; 12 - eksistē; 13 - neeksistē.

LLD noteikuma prototipa pārrakstīšanas operācija

LLD noteikuma prototipa pārrakstīšanas operācija ir nosacījumu un darbību kombinācija, ko izpilda ar prototipa objektu. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
operationobject	integer	Atklātā objekta tips, kuram izpildīt darbību. Iespējamās vērtības: 0 - vienuma prototips; 1 - trigera prototips; 2 - grafika prototips; 3 - hosta prototips. Īpašības darbība: - *obligāts*

Īpašība	Type	Apraksts
operator	integer	Pārrakstīšanas nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* vienāds ar; 1 - nav vienāds ar; 2 - satur; 3 - nesatur; 8 - atbilst; 9 - neatbilst.
value	string	Meklējamais šablons, lai atbilstu vienuma, trigera, grafika vai hosta prototipa nosaukumam atkarībā no izvēlētā objekta.
opstatus	object	Pārrakstīšanas operācijas statusa objekts vienuma, trigera un hosta prototipa objektiem.
opdiscover	object	Pārrakstīšanas operācijas atklāšanas statusa objekts (visiem objektu tipiem).
opperiod	object	Pārrakstīšanas operācijas perioda (atjaunināšanas intervāla) objekts vienuma prototipa objektam.
ophistory	object	Pārrakstīšanas operācijas vēstures objekts vienuma prototipa objektam.
optrends	object	Pārrakstīšanas operācijas tendenču objekts vienuma prototipa objektam.
opseverity	object	Pārrakstīšanas operācijas nopietnības objekts trigera prototipa objektam.
optag	object/array	Pārrakstīšanas operācijas taga objekts trigera un hosta prototipa objektiem.
optemplate	object/array	Pārrakstīšanas operācijas veidnes objekts hosta prototipa objektam.
opinventory	object	Pārrakstīšanas operācijas inventāra objekts hosta prototipa objektam.

LLD noteikuma prototipa ignorēšanas operācijas statuss

LLD noteikuma prototipa ignorēšanas operācijas statuss, kas ir iestatīts atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
status	integer	Ignorēt statusu atlasītajam objektam. Iespējamās vērtības: 0 - Izveidot iespējotu; 1 - Izveidot atspējotu. Īpašības darbība: - *obligāts*

LLD kārtulas prototipa ignorēšanas operācijas atklāšanas statuss

LLD kārtulas prototipa ignorēšanas operācijas atklāšanas statuss, kas ir iestatīts atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
discover	integer	Ignorēt atlasītā objekta atklāšanas statusu. Iespējamās vērtības: 0 - Jā, turpināt objektu atklāšanu; 1 - Nē, jauni objekti netiks atklāti un esošie tiks atzīmēti kā zaudēti. Īpašības darbība: - *obligāts*

LLD noteikuma prototipa ignorēšanas darbības periods

LLD noteikuma prototipa ignorēšanas darbības periods ir atjaunināšanas intervāla vērtība, kas tiek iestatīta atklātajam vienumam. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
delay	string	Ignorēt vienuma prototipa atjaunināšanas intervālu. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus vai LLD makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *required*

LLD noteikuma prototipa ignorēšanas operācijas vēsture

LLD noteikuma prototipa ignorēšanas operācijas vēstures vērtība, kas ir iestatīta atklātajam vienumam. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
history	string	Ignorēt vienuma prototipa vēsturi, kas ir laika vienība, cik ilgi jāglabā vēstures dati. Pieņem arī lietotāja makrosu un LLD makrosu. Īpašības darbība: - *required*

LLD noteikuma prototipa ignorēšanas operācijas tendenču iestatījumi

LLD noteikuma prototipa ignorēšanas operācijas tendenču vērtība, kas tiek iestatīta atklātajam vienumam. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
trends	string	Ignorē vienuma prototipa tendenču iestatījumu, kas ir laika vienība, cik ilgi tendenču dati jāglabā. Pieņem arī lietotāja makrosu un LLD makrosu. Īpašības darbība: - *obligāts*

LLD kārtulas prototipa ignorēšanas operācijas trigera nopietnība

LLD kārtulas prototipa ignorēšanas operācijas trigera nopietnības vērtība, kas tiek iestatīta atklātajam trigerim. Tai ir šādas īpašības:

Īpašība	Tips	Apraksts
severity	integer	Ignorēt trigera prototipa nopietnību. Iespējamās vērtības: 0 - *(noklusējums)* nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa. Īpašības darbība: - *obligāts*

LLD noteikuma prototipa ignorēšanas operācijas tags

LLD noteikuma prototipa ignorēšanas operācijas taga objekts satur taga nosaukumu un vērtību, kas tiek iestatīti atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
tag	string	Jaunais taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Jaunā taga vērtība.

LLD noteikuma prototipa ignorēšanas darbības veidne

LLD noteikuma prototipa ignorēšanas darbības veidnes objekts, kas ir saistīts ar atklāto hostu. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
templateid	ID	Ignorē hosta prototipa saistīto veidņu veidni. Īpašības darbība: - *obligāts*

LLD noteikuma prototipa ignorēšanas darbības inventarizācija

LLD noteikuma prototipa ignorēšanas darbības inventarizācijas režīma vērtība, kas tiek iestatīta atklātajam hostam. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
inventory_mode	integer	Ignorēt hosta prototipa inventarizācijas režīmu. Iespējamās vērtības: -1 - atspējots; 0 - *(noklusējums)* manuāls; 1 - automātisks. Īpašības darbība: - *obligāts*

discoveryruleprototype.create

Apraksts

`object discoveryruleprototype.create(object/array lldRules)`

Šī metode ļauj izveidot jaunus LLD noteikumu prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie LLD noteikumu prototipi.

Papildus standarta LLD noteikumu prototipu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
filter	object	LLD noteikuma prototipa filtrs LLD noteikumam.
preprocessing	object/array	LLD noteikuma prototipa priekšapstrādes opcijas.
lld_macro_paths	object/array	LLD noteikuma prototipa lld_macro_path opcijas.
overrides	object/array	LLD noteikuma prototipa aizstāšanas opcijas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto LLD noteikumu prototipu ID īpašībā `itemids`. Atgriezto ID secība atbilst nodoto LLD noteikumu prototipu secībai.

Piemēri

LLD noteikuma prototipa izveide

Izveidojiet LLD noteikuma prototipu (Tips: Nested), lai atklātu tablespace datubāzes instancē.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47252"
        ]
    },
    "id": 1
}
```

Skatiet arī

- LLD noteikuma prototipa filtrs
- LLD makro ceļi
- LLD noteikuma prototipa priekšapstrāde

Avots

CDiscoveryRulePrototype::create() failā *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.delete

Apraksts

`object discoveryruleprototype.delete(array lldRuleIds)`

Šī metode ļauj dzēst LLD noteikumu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo LLD noteikumu prototipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto LLD noteikumu prototipu ID īpašībā `ruleids`.

Piemēri

Vairāku LLD kārtulu prototipu dzēšana

Dzēsiet divus LLD kārtulu prototipus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.delete",
    "params": [
        "47252",
        "47253"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "47252",
            "47253"
        ]
    },
    "id": 1
}
```

Avots

CDiscoveryRulePrototype::delete() failā *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.get

Apraksts

`integer/array discoveryruleprototype.get(object parameters)`

Metode ļauj izgūt LLD noteikumu prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
discoveryids	ID/array	Atgriezt tikai tos LLD noteikumu prototipus, kas pieder norādītajiem LLD noteikumiem vai LLD noteikumu prototipiem.
itemids	ID/array	Atgriezt tikai LLD noteikumu prototipus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai tos LLD noteikumu prototipus, kas pieder hostiem no norādītajām grupām.
hostids	ID/array	Atgriezt tikai tos LLD noteikumu prototipus, kas pieder norādītajiem hostiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotos LLD noteikumu prototipus.
interfaceids	ID/array	Atgriezt tikai tos LLD noteikumu prototipus, kas izmanto norādītās hosta saskarnes.
monitored	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotos LLD noteikumu prototipus, kas pieder uzraudzītiem hostiem.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai tos LLD noteikumu prototipus, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai tos LLD noteikumu prototipus, kas pieder norādītajām veidnēm.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar LLD noteikumu prototipa vecākobjektu — LLD noteikumu. Tiks atgriezta tukša atbilde, ja vecākobjekts ir atklāšanas noteikuma prototips.
selectDiscoveryRulePrototype	query	Atgriezt `discoveryRulePrototype` īpašību ar LLD noteikumu prototipa vecākobjektu — LLD noteikumu prototipu. Tiks atgriezta tukša atbilde, ja vecākobjekts ir atklāšanas noteikums (nevis prototips).
selectDiscoveryRulePrototypes	query	Atgriezt `discoveryRulePrototypes` īpašību ar bērnobjektu LLD noteikumu prototipiem, kas pieder LLD noteikumu prototipam. Atbalsta `count`.
selectFilter	query	Atgriezt `filter` īpašību ar datiem par filtru, ko izmanto LLD noteikumu prototips.
selectGraphs	query	Atgriež `graphs` īpašību ar grafiku prototipiem, kas pieder LLD noteikumu prototipam. Atbalsta `count`.
selectHostPrototypes	query	Atgriezt `hostPrototypes` īpašību ar hostu prototipiem, kas pieder LLD noteikumu prototipam. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` īpašību ar hostu masīvu, kuriem pieder LLD noteikumu prototips.
selectItems	query	Atgriezt `items` īpašību ar vienumu prototipiem, kas pieder LLD noteikumu prototipam. Atbalsta `count`.
selectTriggers	query	Atgriezt `triggers` īpašību ar trigeru prototipiem, kas pieder LLD noteikumu prototipam. Atbalsta `count`.
selectLLDMacroPaths	query	Atgriezt `lld_macro_paths` īpašību ar LLD makrosu sarakstu un ceļiem uz vērtībām, kas piešķirtas katram atbilstošajam makrosam.
selectPreprocessing	query	Atgriezt `preprocessing` īpašību ar LLD noteikumu prototipa priekšapstrādes opcijām.
selectOverrides	query	Atgriezt `lld_rule_overrides` īpašību ar pārrakstīšanas filtru, nosacījumu un darbību sarakstu, kas tiek izpildītas ar prototipu objektiem.

Parametrs	Tips	Apraksts
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs salīdzināšanai. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram pieder LLD noteikumu prototips.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādiem apakšatlases vaicājumiem: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Examples

Atklāšanas kārtulu prototipu izgūšana no hosta

Izgūstiet visus atklāšanas kārtulu prototipus konkrētam hosta ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.get",
    "params": {
        "hostids": "10084"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "47253",
            "type": "23",
            "snmp_oid": "",
            "hostid": "10084",
            "name": "Atklāt tabulvietas priekš {#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
}
```


Skatiet arī

- Grafika prototips
- Hosts
- Vienuma prototips
- LLD noteikuma prototipa filtrs
- Trigera prototips

Avots

CDiscoveryRulePrototype::get() failā *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

discoveryruleprototype.update

Apraksts

`object discoveryruleprototype.update(object/array lldRules)`

Šī metode ļauj atjaunināt esošos LLD noteikumu prototipus.

Ņemiet vērā, ka jau atklāto prototipu atjaunināšana ir ierobežota.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo LLD noteikumu prototipu rekvizīti.

Katram LLD noteikuma prototipam jābūt definētam rekvizītam `itemid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta LLD noteikumu prototipu rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
filter	object	LLD noteikuma prototipa filtrs, ar ko aizstāt esošo filtru.
preprocessing	object/array	LLD noteikuma prototipa priekšapstrādes opcijas, ar ko aizstāt esošās priekšapstrādes opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem
lld_macro_paths	object/array	LLD noteikuma prototipa lld_macro_path opcijas, ar ko aizstāt esošās lld_macro_path opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem
overrides	object/array	LLD noteikuma prototipa ignorēšanas opcijas, ar ko aizstāt esošās ignorēšanas opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto LLD kārtulu prototipu ID zem `itemids` īpašības.

Piemēri

LLD noteikuma prototipa priekšapstrādes opciju atjaunināšana

Atjauniniet LLD noteikuma prototipu ar JSONPath priekšapstrādes noteikumu. Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryruleprototype.update",
    "params": {
        "itemid": "47253",
        "preprocessing": [
            {
                "type": 12,
                "params": "$.tablespaces",
                "error_handler": 1
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "47253"
        ]
    },
    "id": 1
}
```

Avots

CDiscoveryRulePrototype::update() failā *ui/include/classes/api/services/CDiscoveryRulePrototype.php*.

LLD noteikums

Šī klase ir paredzēta darbam ar zema līmeņa atklāšanas noteikumiem.

Objektu atsauces:

- LLD noteikums
  - HTTP galvene
  - HTTP vaicājuma lauks
- LLD noteikuma filtrs
  - LLD noteikuma filtra nosacījums
- LLD makro ceļš
- LLD noteikuma priekšapstrāde
- LLD noteikuma ignorēšanas
  - LLD noteikuma ignorēšanas filtrs
    - LLD noteikuma ignorēšanas filtra nosacījums
  - LLD noteikuma ignorēšanas operācija
    - LLD noteikuma ignorēšanas operācijas statuss
    - LLD noteikuma ignorēšanas operācijas atklāšana
    - LLD noteikuma ignorēšanas operācijas periods
    - LLD noteikuma ignorēšanas operācijas vēsture
    - LLD noteikuma ignorēšanas operācijas tendences
    - LLD noteikuma ignorēšanas operācijas nopietnība
    - LLD noteikuma ignorēšanas operācijas birka
    - LLD noteikuma ignorēšanas operācijas veidne
    - LLD noteikuma ignorēšanas operācijas inventārs

Pieejamās metodes:

- discoveryrule.create - izveidot jaunus LLD noteikumus
- discoveryrule.delete - dzēst LLD noteikumus
- discoveryrule.get - izgūt LLD noteikumus
- discoveryrule.update - atjaunināt LLD noteikumus

LLD noteikuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `discoveryrule` API.

LLD kārtula

Zema līmeņa atklāšanas kārtulas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
itemid	ID	LLD kārtulas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām

Īpašība	Tips	Apraksts
delay	string	LLD kārtulas atjaunināšanas intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”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), vai ja `type` ir iestatīts uz ”Zabbix agent (active)” (7) un `key_` nesatur ”mqtt.get”
hostid	ID	Hosta ID, kuram pieder LLD kārtula. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām
flags	integer	Atklāšanas kārtulas izcelsme. Iespējamās vērtības: 1 - zema līmeņa atklāšanas kārtula; 5 - zema līmeņa atklāšanas kārtula, kas pārveidota no prototipa. Īpašības darbība: - *tikai lasāms*
interfaceid	ID	LLD kārtulas hosta saskarnes ID. Īpašības darbība: - *obligāts*, ja LLD kārtula pieder hostam un `type` ir iestatīts uz ”Zabbix agent”, ”IPMI agent”, ”JMX agent” vai ”SNMP agent” - *atbalstīts*, ja LLD kārtula pieder hostam un `type` ir iestatīts uz ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent”
key_	string	LLD kārtulas atslēga. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
name	string	LLD kārtulas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
type	integer	LLD kārtulas tips. Iespējamās vērtības: 0 - Zabbix aģents; 2 - Zabbix trapper; 3 - Simple check; 5 - Zabbix internal; 7 - Zabbix aģents (aktīvs); 10 - External check; 11 - Database monitor; 12 - IPMI aģents; 13 - SSH aģents; 14 - TELNET aģents; 16 - JMX aģents; 18 - atkarīgais vienums; 19 - HTTP aģents; 20 - SNMP aģents; 21 - Script; 22 - Browser; 23 - ligzdots. Šis tips ir atļauts tikai tad, ja kārtulas vecāks ir veidne (pieņemot, ka tā atklāšanas laikā ir sasaistīta ar hosta prototipu) vai atklāts hosts (kuram saknē ir LLD kārtula). Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
url	string	URL virkne. Atbalsta lietotāja makrosus, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
allow_traps	integer	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. Iespējamās vērtības: 0 - *(noklusējums)* Neatļaut pieņemt ienākošos datus; 1 - Atļaut pieņemt ienākošos datus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent”
authtype	integer	Autentifikācijas metode. Iespējamās vērtības, ja `type` ir iestatīts uz ”SSH agent”: 0 - *(noklusējums)* parole; 1 - publiskā atslēga. Iespējamās vērtības, ja `type` ir iestatīts uz ”HTTP agent”: 0 - *(noklusējums)* nav; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
description	string	LLD kārtulas apraksts.

Īpašība	Tips	Apraksts
error	string	Kļūdas teksts, ja rodas problēmas ar LLD kārtulas vērtības atjaunināšanu. Īpašības darbība: - *tikai lasāms*
follow_redirects	integer	Sekot atbildes pāradresācijām datu aptaujas laikā. Iespējamās vērtības: 0 - Nesekot pāradresācijām; 1 - *(noklusējums)* Sekot pāradresācijām. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ipmi_sensor	string	IPMI sensors. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”IPMI agent” un `key_` nav iestatīts uz ”ipmi.get” - *atbalstīts*, ja `type` ir iestatīts uz ”IPMI agent” - *tikai lasāms* mantotajiem objektiem
jmx_endpoint	string	JMX aģenta pielāgota savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent”
lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks dzēsti. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Noklusējums: `7d`.
lifetime_type	integer	Scenārijs zaudēto LLD resursu dzēšanai. Iespējamās vērtības: 0 - *(noklusējums)* Dzēst pēc lifetime sliekšņa sasniegšanas; 1 - Nedzēst; 2 - Dzēst nekavējoties.
enabled_lifetime	string	Laika periods, pēc kura vienumi, kas vairs netiek atklāti, tiks atspējoti. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makrosu. Noklusējums: `0`.
enabled_lifetime_type	integer	Scenārijs zaudēto LLD resursu atspējošanai. Iespējamās vērtības: 0 - Atspējot pēc lifetime sliekšņa sasniegšanas; 1 - Neatspējot; 2 - *(noklusējums)* Atspējot nekavējoties.

Īpašība	Tips	Apraksts
master_itemid	ID	Galvenā vienuma ID. Atklāšanas kārtula nevar būt galvenais vienums citai atklāšanas kārtulai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotajiem objektiem
output_format	integer	Vai atbilde jākonvertē uz JSON. Iespējamās vērtības: 0 - *(noklusējums)* Saglabāt neapstrādātu; 1 - Konvertēt uz JSON. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
params	string	Papildu parametri atkarībā no LLD kārtulas tipa: - izpildāmais skripts SSH un Telnet LLD kārtulām; - SQL vaicājums datubāzes uzrauga LLD kārtulām; - formula aprēķinātajām LLD kārtulām; - skripts Script un Browser LLD kārtulām. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”Script” vai ”Browser”)
parameters	object/array	Papildu parametri, ja `type` ir iestatīts uz ”Script” vai ”Browser”. Objektu masīvs ar `name` un `value` īpašībām, kur `name` jābūt unikālam. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem
password	string	Parole autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”JMX agent” un `username` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”Database monitor”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
post_type	integer	`posts` īpašībā saglabātā POST datu ķermeņa tips. Iespējamās vērtības: 0 - *(noklusējums)* Neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
posts	string	HTTP(S) pieprasījuma ķermeņa dati. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”JSON data” vai ”XML data” - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”Raw data” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
privatekey	string	Privātās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
publickey	string	Publiskās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
query_fields	array	Vaicājuma lauku masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
retrieve_mode	integer	Kura atbildes daļa jāsaglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”GET”, ”POST” vai ”PUT”: 0 - *(noklusējums)* Ķermenis; 1 - Galvenes; 2 - Tiks saglabāts gan ķermenis, gan galvenes. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”HEAD”: 1 - Galvenes. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
snmp_oid	string	SNMP OID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotajiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
state	integer	LLD kārtulas stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - neatbalstīts. Īpašības darbība: - *tikai lasāms*
status	integer	LLD kārtulas statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējota LLD kārtula; 1 - atspējota LLD kārtula.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalsta arī lietotāja makrosus kā daļu no ar komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem
templateid	ID	Vecākveidnes LLD kārtulas ID. Īpašības darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera/globālos iestatījumus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix agent" (0), "Simple check" (3) un `key_` nesākas ar "`vmware.`" un "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) un `snmp_oid` sākas ar "`walk[`" vai "`get[`", "Script" (21), "Browser" (22) - *tikai lasāms* mantotajiem objektiem
trapper_hosts	string	Atļautie hosti. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix trapper", vai ja `type` ir iestatīts uz "HTTP agent" un `allow_traps` ir iestatīts uz "Allow to accept incoming data"
username	string	Lietotājvārds autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent", "TELNET agent", vai ja `type` ir iestatīts uz "JMX agent" un `password` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz "Simple check", "Database monitor" vai "HTTP agent" - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz "HTTP agent")

Īpašība	Tips	Apraksts
uuid	string	Universāls unikāls identifikators, ko izmanto importēto LLD kārtulu sasaistīšanai ar jau esošajām. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja LLD kārtula pieder veidnei
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

HTTP galvene

Galvenes objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	HTTP galvenes nosaukums. Īpašības darbība: - *obligāta*
value	string	Galvenes vērtība. Īpašības darbība: - *obligāta*

HTTP vaicājuma lauks

Vaicājuma lauka objekts definē nosaukumu un vērtību, ko izmanto URL parametra norādīšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāts*
value	string	Parametra vērtība. Īpašības darbība: - *obligāts*

LLD noteikuma filtrs

LLD noteikuma filtra objekts definē nosacījumu kopu, ko var izmantot atklāto objektu filtrēšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
conditions	object/array	Filtra nosacījumu kopa, ko izmantot rezultātu filtrēšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - Un/Vai; 1 - Un; 2 - Vai; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota filtru ar pielāgotu izteiksmi nosacījumu novērtēšanai. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti filtra nosacījumos: neviens nosacījums nedrīkst palikt neizmantots vai izlaists. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”custom expression”

LLD noteikuma filtra nosacījums

LLD noteikuma filtra nosacījuma objekts definē atsevišķu pārbaudi, kas jāveic LLD makro vērtībai. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
macro	string	LLD makro, kuram jāveic pārbaude. Īpašības darbība: - *required*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *required* ja `operator` ir iestatīts uz ”atbilst regulārajai izteiksmei” vai ”neatbilst regulārajai izteiksmei”
formulaid	string	Patvaļīgs unikāls ID, ko izmanto, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču pēc tam, kad tie tiek pieprasīti atkārtoti, tas tiks ģenerēts no jauna. Īpašības darbība: - *required* ja LLD noteikuma filtra objekta `evaltype` ir iestatīts uz ”pielāgota izteiksme”
operator	integer	Nosacījuma operators. Iespējamās vērtības: 8 - *(noklusējums)* atbilst regulārajai izteiksmei; 9 - neatbilst regulārajai izteiksmei; 12 - eksistē; 13 - neeksistē.

Note:
Lai labāk saprastu, kā izmantot filtrus ar dažādiem izteiksmju tipiem, skatiet piemērus metožu lapās discoveryrule.get un discoveryrule.create.

LLD makro ceļš

LLD makro ceļam ir šādas īpašības:

Īpašība	Type	Apraksts
lld_macro	string	LLD makro. Īpašības darbība: - *obligāts*
path	string	Atlasītājs vērtībai, kas tiks piešķirta atbilstošajam makro. Īpašības darbība: - *obligāts*

LLD kārtulas priekšapstrāde

LLD kārtulas priekšapstrādes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 5 - Regulārā izteiksme; 11 - XML XPath; 12 - JSONPath; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - Pārbaudīt kļūdu JSON; 17 - Pārbaudīt kļūdu XML; 20 - Atmest nemainītu ar sirdspukstu; 21 - JavaScript; 23 - Prometheus uz JSON; 24 - CSV uz JSON; 25 - Aizstāt; 27 - XML uz JSON; 28 - SNMP walk vērtība; 29 - SNMP walk uz JSON; 30 - SNMP get vērtība. Īpašības darbība: - *obligāta*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri tiek atdalīti ar jaunās rindas (\n) rakstzīmi. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Atmest nemainītu ar sirdspukstu” (20), ”JavaScript” (21), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Aizstāt” (25), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)

Īpašība	Tips	Apraksts
error_handler	integer	Darbības tips, ko izmanto priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”XML uz JSON” (27), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādātāja parametri. Īpašības darbība: - *obligāta*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Tālāk norādīti parametri un kļūdu apstrādātāji, kas tiek atbalstīti katram priekšapstrādes tipam.

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
5	Regulārā izteiksme	raksts[1]	izvade[2]		0, 1, 2, 3
11	XML XPath	ceļš[3]			0, 1, 2, 3
12	JSONPath	ceļš[3]			0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	raksts[1]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	raksts[1]			0, 1, 2, 3
16	Pārbaudīt kļūdu JSON	ceļš[3]			0, 1, 2, 3
17	Pārbaudīt kļūdu XML	ceļš[3]			0, 1, 2, 3
20	Atmest ne-mainītu ar sird-spuk-stu	sekundes^4, 5^			
21	JavaScript	skripts[2]			
23	Prometheus uz JSON	raksts^5, 6^			0, 1, 2, 3
24	CSV uz JSON	rakstzīme[2]	rakstzīme[2]	0,1	0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
25	Aizstāt	meklējamā virkne[2]	aizvietojums[2]		
27	XML uz JSON				0, 1, 2, 3
28	SNMP walk vērtība	OID[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[7]	Lauka nosaukums[2]	OID prefikss[2]	Formāts: 0 - Nemainīts 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS			0, 1, 2, 3

[1] regulārā izteiksme
 [2] virkne
 [3] JSONPath vai XML XPath
 [4] pozitīvs vesels skaitlis (ar laika sufiksu atbalstu, piemēram, 30s, 1m, 2h, 1d)
 [5] lietotāja makross
 [6] Prometheus raksts, kas atbilst sintaksei: `<metric name>{<label name>="<label value>", ...} == <value>`. Katrs Prometheus raksta komponents (metrika, etiķetes nosaukums, etiķetes vērtība un metrikas vērtība) var būt lietotāja makross.
 [7] Atbalsta vairākus "Lauka nosaukums,OID prefikss,Formāts ierakstus", kas atdalīti ar jaunās rindas rakstzīmi.

LLD noteikuma pārrakstīšanas

LLD noteikuma pārrakstīšanas objekts definē noteikumu kopu (filtrus, nosacījumus un operācijas), kas tiek izmantoti, lai pārrakstītu dažādu prototipu objektu īpašības. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Unikāls pārrakstīšanas nosaukums. Īpašības darbība: - *obligāta*
step	integer	Unikāls pārrakstīšanas secības numurs. Īpašības darbība: - *obligāta*
stop	integer	Pārtraukt nākamo pārrakstīšanu apstrādi, ja ir atbilstība. Iespējamās vērtības: 0 - *(noklusējums)* nepārtraukt pārrakstīšanu apstrādi; 1 - pārtraukt pārrakstīšanu apstrādi, ja filtrs atbilst.
filter	object	Pārrakstīšanas filtrs.
operations	object/array	Pārrakstīšanas operācijas.

LLD noteikuma ignorēšanas filtrs

LLD noteikuma ignorēšanas filtra objekts definē nosacījumu kopu, un, ja tie atbilst atklātajam objektam, tiek piemērota ignorēšana. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
conditions	object/array	Ignorēšanas filtra nosacījumu kopa, ko izmantot atklāto objektu atbilstības noteikšanai. Nosacījumi tiks sakārtoti to izvietojuma secībā formulā. Īpašības darbība: - *obligāta*
evaltype	integer	Ignorēšanas filtra nosacījumu novērtēšanas metode. Iespējamās vērtības: 0 - Un/Vai; 1 - Un; 2 - Vai; 3 - Pielāgota izteiksme. Īpašības darbība: - *obligāta*
eval_formula	string	Ģenerēta izteiksme, kas tiks izmantota ignorēšanas filtra nosacījumu novērtēšanai. Izteiksme satur ID, kas atsaucas uz konkrētiem ignorēšanas filtra nosacījumiem pēc to `formulaid`. `eval_formula` vērtība ir vienāda ar `formula` vērtību filtriem ar pielāgotu izteiksmi. Īpašības darbība: - *tikai lasāma*
formula	string	Lietotāja definēta izteiksme, kas tiks izmantota, lai novērtētu ignorēšanas filtru nosacījumus ar pielāgotu izteiksmi. Izteiksmei jāsatur ID, kas atsaucas uz konkrētiem ignorēšanas filtra nosacījumiem pēc to `formulaid`. Izteiksmē izmantotajiem ID precīzi jāsakrīt ar tiem, kas definēti ignorēšanas filtra nosacījumos: nevienu nosacījumu nedrīkst atstāt neizmantotu vai izlaistu. Īpašības darbība: - *obligāta*, ja `evaltype` ir iestatīts uz ”custom expression”

LLD noteikuma ignorēšanas filtra nosacījums

LLD noteikuma ignorēšanas filtra nosacījuma objekts definē atsevišķu pārbaudi, kas jāveic ar LLD makro vērtību. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
macro	string	LLD makro, kuram jāveic pārbaude. Īpašības darbība: - *required*
value	string	Vērtība, ar ko salīdzināt. Īpašības darbība: - *required* ja `operator` ir iestatīts uz ”atbilst regulārajai izteiksmei” vai ”neatbilst regulārajai izteiksmei”
formulaid	string	Patvaļīgs unikāls ID, kas tiek izmantots, lai atsauktos uz nosacījumu no pielāgotas izteiksmes. Tas var saturēt tikai lielos burtus. ID lietotājam jādefinē, modificējot filtra nosacījumus, taču pēc tam, kad tie tiek pieprasīti atkārtoti, tas tiks ģenerēts no jauna. Īpašības darbība: - *required* ja LLD noteikuma ignorēšanas filtra objekta `evaltype` ir iestatīts uz ”pielāgota izteiksme”

Īpašība	Type	Apraksts
operator	integer	Nosacījuma operators. Iespējamās vērtības: 8 - *(noklusējums)* atbilst regulārajai izteiksmei; 9 - neatbilst regulārajai izteiksmei; 12 - eksistē; 13 - neeksistē.

LLD noteikuma ignorēšanas operācija

LLD noteikuma ignorēšanas operācija ir nosacījumu un darbību kombinācija, kas jāveic ar prototipa objektu. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
operationobject	integer	Atklātā objekta tips, kuram jāveic darbība. Iespējamās vērtības: 0 - vienuma prototips; 1 - trigera prototips; 2 - grafika prototips; 3 - hosta prototips; 4 - atklāšanas prototips. Īpašības darbība: - *obligāts*
operator	integer	Ignorēšanas nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* vienāds ar; 1 - nav vienāds ar; 2 - satur; 3 - nesatur; 8 - atbilst; 9 - neatbilst.
value	string	Meklējamais šablons, kas atbilst vienuma, trigera, grafika vai hosta prototipa nosaukumam atkarībā no izvēlētā objekta.
opstatus	object	Ignorēšanas operācijas statusa objekts vienuma, trigera un hosta prototipa objektiem.
opdiscover	object	Ignorēšanas operācijas atklāšanas statusa objekts (visiem objektu tipiem).
opperiod	object	Ignorēšanas operācijas perioda (atjauninašanas intervāla) objekts vienuma prototipa objektam.
ophistory	object	Ignorēšanas operācijas vēstures objekts vienuma prototipa objektam.
optrends	object	Ignorēšanas operācijas tendenču objekts vienuma prototipa objektam.
opseverity	object	Ignorēšanas operācijas smaguma pakāpes objekts trigera prototipa objektam.
optag	object/array	Ignorēšanas operācijas taga objekts trigera un hosta prototipa objektiem.
optemplate	object/array	Ignorēšanas operācijas veidnes objekts hosta prototipa objektam.
opinventory	object	Ignorēšanas operācijas inventāra objekts hosta prototipa objektam.

LLD noteikuma ignorēšanas operācijas statuss

LLD noteikuma ignorēšanas operācijas statuss, kas ir iestatīts atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
status	integer	Ignorēt statusu atlasītajam objektam. Iespējamās vērtības: 0 - Izveidot iespējotu; 1 - Izveidot atspējotu. Īpašības darbība: - *required*

LLD noteikuma ignorēšanas operācijas atklāšana

LLD noteikuma ignorēšanas operācijas atklāšanas statuss, kas ir iestatīts atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
discover	integer	Ignorēt atklāšanas statusu atlasītajam objektam. Iespējamās vērtības: 0 - Jā, turpināt objektu atklāšanu; 1 - Nē, jauni objekti netiks atklāti, un esošie tiks atzīmēti kā zaudēti. Īpašības darbība: - *obligāts*

LLD kārtulas ignorēšanas darbības periods

LLD kārtulas ignorēšanas darbības periods ir atjauninašanas intervāla vērtība, kas tiek iestatīta atklātajam vienumam. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
delay	string	Ignorē vienuma prototipa atjauninašanas intervālu. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotos intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus vai LLD makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *required*

LLD noteikuma ignorēšanas darbības vēsture

LLD noteikuma ignorēšanas darbības vēstures vērtība, kas ir iestatīta atklātajam vienumam. Tai ir šādas īpašības:

Īpašība	Type	Apraksts
history	string	Ignorē vienuma prototipa vēsturi, kas ir laika vienība, cik ilgi vēstures dati ir jāglabā. Pieņem arī lietotāja makrosu un LLD makrosu. Īpašības darbība: - *obligāts*

LLD noteikuma ignorēšanas operācijas tendenču iestatījums

LLD noteikuma ignorēšanas operācijas tendenču vērtība, kas tiek iestatīta atklātajam vienumam. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
trends	string	Ignorēt vienuma prototipa tendenču iestatījumu, kas ir laika vienība, cik ilgi tendenču dati ir jāglabā. Pieņem arī lietotāja makrosu un LLD makrosu. Īpašības darbība: - *required*

LLD noteikuma ignorēšanas operācijas trigera nopietnība

LLD noteikuma ignorēšanas operācijas trigera nopietnības vērtība, kas tiek iestatīta atklātajam trigerim. Tai ir šādas īpašības:

Īpašība	Tips	Apraksts
severity	integer	Ignorēt trigera prototipa nopietnību. Iespējamās vērtības: 0 - *(noklusējums)* nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa. Īpašības darbība: - *obligāts*

LLD noteikuma ignorēšanas operācijas tags

LLD noteikuma ignorēšanas operācijas taga objekts satur taga nosaukumu un vērtību, kas tiek iestatīti atklātajam objektam. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
tag	string	Jauns taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Jauna taga vērtība.

LLD noteikuma ignorēšanas darbības veidne

LLD noteikuma ignorēšanas darbības veidnes objekts, kas ir saistīts ar atklāto hostu. Tam ir šādas īpašības:

Īpašība	Type	Apraksts
templateid	ID	Ignorē hosta prototipa saistīto veidņu veidni. Īpašības darbība: - *obligāts*

LLD noteikuma ignorēšanas darbības inventarizācija

LLD noteikuma ignorēšanas darbības inventarizācijas režīma vērtība, kas ir iestatīta atklātajam hostam. Tai ir šādas īpašības:

Īpašība	Tips	Apraksts
inventory_mode	integer	Ignorēt hosta prototipa inventarizācijas režīmu. Iespējamās vērtības: -1 - atspējots; 0 - *(noklusējums)* manuāls; 1 - automātisks. Īpašības darbība: - *obligāts*

discoveryrule.create

Apraksts

`object discoveryrule.create(object/array lldRules)`

Šī metode ļauj izveidot jaunus LLD noteikumus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie LLD noteikumi.

Papildus standarta LLD noteikumu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
filter	object	LLD noteikuma filtrs LLD noteikumam.
preprocessing	object/array	LLD noteikuma priekšapstrādes opcijas.
lld_macro_paths	object/array	LLD noteikuma lld_macro_path opcijas.
overrides	object/array	LLD noteikuma ignorēšanas opcijas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto LLD noteikumu ID īpašībā `itemids`. Atgriezto ID secība atbilst nodoto LLD noteikumu secībai.

Piemēri

LLD kārtulas izveide

Izveidojiet Zabbix aģenta LLD kārtulu, lai atklātu piemontētās failu sistēmas. Atklātie vienumi tiks atjaunināti ik pēc 30 sekundēm.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Filtra izmantošana

Izveidojiet LLD kārtulu ar nosacījumu kopu, pēc kuras filtrēt rezultātus. Nosacījumi tiks grupēti kopā, izmantojot loģisko operatoru ”and”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
```

```
}
```

LLD kārtulas izveide ar makro ceļiem

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "LLD kārtula ar LLD makro ceļiem",
        "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Pielāgotas izteiksmes filtra izmantošana

Izveidojiet LLD kārtulu ar filtru, kas nosacījumu novērtēšanai izmantos pielāgotu izteiksmi. LLD kārtulai jāatklāj tikai tie objekti, kuru ”{#MACRO1}” makro vērtība atbilst gan regulārajai izteiksmei ”regex1”, gan ”regex2”, un ”{#MACRO2}” vērtība atbilst vai nu ”regex3”, vai ”regex4”. Formulu ID ”A”, ”B”, ”C” un ”D” ir izvēlēti patvaļīgi.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27665"
        ]
    },
    "id": 1
}
```

Pielāgotu vaicājuma lauku un galveņu izmantošana

Izveidojiet LLD kārtulu ar pielāgotiem vaicājuma laukiem un galvenēm.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "hostid": "10257",
        "interfaceid": "5",
        "type": 19,
        "name": "API HTTP aģents",
        "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

LLD kārtulas izveide ar priekšapstrādi

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Atklāšanas kārtula ar priekšapstrādi",
        "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

LLD kārtulas izveide ar ignorēšanas noteikumiem

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.create",
    "params": {
        "name": "Atklāt datubāzes hostu",
        "key_": "lld.with.overrides",
        "hostid": "10001",
        "type": 0,
        "delay": "60s",
        "interfaceid": "1155",
        "overrides": [
            {
                "name": "Atklāt MySQL hostu",
                "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": "Datubāzes hosts",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10170"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "mysql"
                            }
                        ]
                    }
                ]
            },
            {
                "name": "Atklāt PostgreSQL hostu",
                "step": "2",
                "stop": "1",
                "filter": {
                    "evaltype": "0",
                    "conditions": [
                        {
                            "macro": "{#UNIT.NAME}",
```

```
                            "operator": "8",
                            "value": "^postgresql\\.service$"
                        }
                    ]
                },
                "operations": [
                    {
                        "operationobject": "3",
                        "operator": "2",
                        "value": "Datubāzes hosts",
                        "opstatus": {
                            "status": "0"
                        },
                        "optemplate": [
                            {
                                "templateid": "10263"
                            }
                        ],
                        "optag": [
                            {
                                "tag": "database",
                                "value": "postgresql"
                            }
                        ]
                    }
                ]
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "30980"
        ]
    },
    "id": 1
}
```

Izveidot skripta LLD kārtulu

Izveidojiet vienkāršu datu vākšanu, izmantojot skripta LLD kārtulu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Izveidot LLD kārtulu ar norādītu atspējošanas laika periodu un bez dzēšanas

Izveidojiet LLD kārtulu ar pielāgotu laika periodu entītijas atspējošanai pēc tam, kad tā vairs netiek atklāta, ar iestatījumu, ka tā nekad netiks dzēsta.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Skatiet arī

- LLD kārtulas filtrs
- LLD makro ceļi
- LLD kārtulas priekšapstrāde

Avots

CDiscoveryRule::create() failā *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.delete

Apraksts

`object discoveryrule.delete(array lldRuleIds)`

Šī metode ļauj dzēst LLD kārtulas.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo LLD noteikumu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto LLD noteikumu ID zem `ruleids` īpašības.

Piemēri

Vairāku LLD noteikumu dzēšana

Dzēsiet divus LLD noteikumus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.delete",
    "params": [
        "27665",
        "27668"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "ruleids": [
            "27665",
            "27668"
        ]
    },
    "id": 1
}
```

Avots

CDiscoveryRule::delete() failā *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.get

Apraksts

`integer/array discoveryrule.get(object parameters)`

Šī metode ļauj izgūt LLD kārtulas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
itemids	ID/array	Atgriezt tikai LLD kārtulas ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai LLD kārtulas, kas pieder hostiem no norādītajām grupām.

Parametrs	Tips	Apraksts
hostids	ID/array	Atgriezt tikai LLD kārtulas, kas pieder norādītajiem hostiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotās LLD kārtulas.
interfaceids	ID/array	Atgriezt tikai LLD kārtulas, kas izmanto norādītās hosta saskarnes.
monitored	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotās LLD kārtulas, kas pieder uzraudzītiem hostiem.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai LLD kārtulas, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai LLD kārtulas, kas pieder norādītajām veidnēm.
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar LLD kārtulas atklāšanas objekta datiem. LLD kārtulas atklāšanas objekts sasaista atklātu LLD kārtulu ar LLD kārtulas prototipu, no kura tā tika atklāta.
selectDiscoveryRulePrototypes	query	Atgriezt `discoveryRulePrototypes` īpašību ar LLD kārtulu prototipiem, kas pieder LLD kārtulai. Atbalsta `count`.
selectFilter	query	Atgriezt `filter` īpašību ar LLD kārtulas izmantotā filtra datiem.
selectGraphs	query	Atgriež `graphs` īpašību ar grafiku prototipiem, kas pieder LLD kārtulai. Atbalsta `count`.
selectHostPrototypes	query	Atgriezt `hostPrototypes` īpašību ar hostu prototipiem, kas pieder LLD kārtulai. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` īpašību ar hostu masīvu, kuriem pieder LLD kārtula.
selectItems	query	Atgriezt `items` īpašību ar vienumu prototipiem, kas pieder LLD kārtulai. Atbalsta `count`.
selectTriggers	query	Atgriezt `triggers` īpašību ar trigeru prototipiem, kas pieder LLD kārtulai. Atbalsta `count`.
selectLLDMacroPaths	query	Atgriezt `lld_macro_paths` īpašību ar LLD makrosu sarakstu un ceļiem uz vērtībām, kas piešķirtas katram atbilstošajam makrosam.
selectPreprocessing	query	Atgriezt `preprocessing` īpašību ar LLD kārtulas priekšapstrādes opcijām.
selectOverrides	query	Atgriezt `lld_rule_overrides` īpašību ar ignorēšanas filtru, nosacījumu un darbību sarakstu, kas tiek izpildītas ar prototipu objektiem.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs salīdzināšanai. Neatbalsta `text` datu tipa īpašības.
limitSelects	integer	Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram pieder LLD kārtula. Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādiem apakšatlases vaicājumiem: `selectItems`, `selectGraphs`, `selectTriggers`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `name`, `key_`, `delay`, `type`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parametrs	Tips	Apraksts
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Atklāšanas kārtulu izgūšana no hosta

Izgūst visas atklāšanas kārtulas norādītajam hosta ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": "extend",
        "hostids": "10202"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "27425",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10202",
            "name": "Tīkla saskarņu atklāšana",
            "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": "Tīkla saskarņu atklāšana, kā noteikts globālajā regulārajā izteiksmē \"Network
            "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": "Piemontēto failu sistēmu atklāšana",
            "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": "Dažādu tipu failu sistēmu atklāšana, kā noteikts globālajā regulārajā izteiksm
            "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
}
```

Filtra nosacījumu izgūšana

Izgūstiet LLD kārtulas "24681" nosaukumu un tās filtra nosacījumus. Filtrs izmanto "and" novērtēšanas tipu, tāpēc `formula` īpašība ir tukša un `eval_formula` tiek ģenerēta automātiski.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "selectFilter": "extend",
        "itemids": ["24681"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Iegūt LLD kārtulu pēc URL

Iegūt hosta LLD kārtulu pēc kārtulas URL lauka vērtības. Tiek atbalstīta tikai precīza LLD kārtulai definētās URL virknes atbilstība.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "hostids": "10257",
        "filter": {
            "type": 19,
            "url": "http://127.0.0.1/discoverer.php"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Iegūt LLD kārtulu ar ignorēšanas iestatījumiem

Iegūt vienu LLD kārtulu, kurai ir dažādi ignorēšanas iestatījumi.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.get",
    "params": {
        "output": ["name"],
        "itemids": "30980",
        "selectOverrides": ["name", "step", "stop", "filter", "operations"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Grafika prototips
- Hosts
- Vienuma prototips
- LLD kārtulas filtrs
- Trigera prototips

Avots

CDiscoveryRule::get() failā *ui/include/classes/api/services/CDiscoveryRule.php*.

discoveryrule.update

Apraksts

`object discoveryrule.update(object/array lldRules)`

Šī metode ļauj atjaunināt esošos LLD noteikumus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjauninámo LLD noteikumu īpašības.

Īpašībai `itemid` jābūt definētai katram LLD noteikumam, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta LLD noteikumu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
filter	object	LLD noteikuma filtrs, ar kuru aizstāt esošo filtru.

Parametrs	Tips	Apraksts
preprocessing	object/array	LLD noteikuma priekšapstrādes opcijas, ar kurām aizstāt esošās priekšapstrādes opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem
lld_macro_paths	object/array	LLD noteikuma lld_macro_path opcijas, ar kurām aizstāt esošās lld_macro_path opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem
overrides	object/array	LLD noteikuma aizstāšanas opcijas, ar kurām aizstāt esošās overrides opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto LLD noteikumu ID zem `itemids` īpašības.

Piemēri

Filtra pievienošana LLD kārtulai

Pievienojiet filtru, lai makrosa *{#FSTYPE}* saturs atbilstu regulārajai izteiksmei *@File systems for discovery*.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "22450",
        "filter": {
            "evaltype": 1,
            "conditions": [
                {
                    "macro": "{#FSTYPE}",
                    "value": "@File systems for discovery"
                }
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

LLD makro ceļu pievienošana

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
```

```
        "itemid": "22450",
        "lld_macro_paths": [
            {
                "lld_macro": "{#MACRO1}",
                "path": "$.json.path"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22450"
        ]
    },
    "id": 1
}
```

Atspējot slazdošanu

Atspējot LLD slazdošanu atklāšanas kārtulai.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "28336",
        "allow_traps": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28336"
        ]
    },
    "id": 1
}
```

LLD kārtulas priekšapstrādes opciju atjaunināšana

Atjauniniet LLD kārtulu ar priekšapstrādes kārtulu ”JSONPath”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

LLD noteikuma skripta atjaunināšana

Atjauniniet LLD noteikuma skriptu ar citu skriptu un noņemiet nevajadzīgos parametrus, kas tika izmantoti iepriekšējā skriptā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

LLD noteikuma darbības laika atjaunināšana

Atjauniniet LLD noteikumu, lai atspējotu vairs neatklātu entītiju pēc 12 stundām un dzēstu to pēc 7 dienām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "discoveryrule.update",
    "params": {
        "itemid": "46864",
        "lifetime_type": 0,
        "lifetime": "7d",
        "enabled_lifetime_type": 0,
        "enabled_lifetime": "12h"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "46864"
        ]
    },
    "id": 1
}
```

Avots

CDiscoveryRule::update() failā *ui/include/classes/api/services/CDiscoveryRule.php*.

Loma

Šī klase ir paredzēta darbam ar lietotāju lomām.

Objektu atsauces:

- Loma
- Lomu noteikumi
  - Lietotāja saskarnes elements
  - Pakalpojums
  - Pakalpojuma tags
  - Modulis
  - Darbība

Pieejamās metodes:

- role.create - izveidot jaunas lietotāju lomas
- role.delete - dzēst lietotāju lomas
- role.get - izgūt lietotāju lomas
- role.update - atjaunināt lietotāju lomas

Lomas objekts

Tālāk norādītie objekti ir tieši saistīti ar `role` API.

Loma

Lomas objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
roleid	ID	Lomas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Lomas nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
type	integer	Lietotāja tips. Iespējamās vērtības: 1 - *(noklusējums)* Lietotājs; 2 - Administrators; 3 - Super administrators. Īpašības darbība: - *obligāts* izveides darbībām

Īpašība	Tips	Apraksts
readonly	integer	Vai loma ir tikai lasāma. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā. Īpašības darbība: - *tikai lasāms*

Lomas noteikumi

Lomas noteikumu objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
ui	array	Lietotāja saskarnes elementa objektu masīvs.
ui.default_access	integer	Vai piekļuve jauniem lietotāja saskarnes elementiem ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
services.read.mode	integer	Tikai lasāma piekļuve pakalpojumiem. Iespējamās vērtības: 0 - Tikai lasāma piekļuve pakalpojumiem, kas norādīti īpašībā `services.read.list` vai atbilst īpašībā `services.read.tag` norādītajām birkām; 1 - *(noklusējums)* Tikai lasāma piekļuve visiem pakalpojumiem.
services.read.list	array	Pakalpojuma objektu masīvs. Norādītajiem pakalpojumiem, ieskaitot apakšpakalpojumus, lietotāja lomai tiks piešķirta tikai lasāma piekļuve. Tikai lasāma piekļuve neaizstās lasīšanas-rakstīšanas piekļuvi pakalpojumiem. Īpašības darbība: - *atbalstīta*, ja `services.read.mode` ir iestatīts uz ”0”
services.read.tag	object	Pakalpojuma birkas objektu masīvs. Pakalpojumiem, kas atbilst birkām, ieskaitot apakšpakalpojumus, lietotāja lomai tiks piešķirta tikai lasāma piekļuve. Tikai lasāma piekļuve neaizstās lasīšanas-rakstīšanas piekļuvi pakalpojumiem. Īpašības darbība: - *atbalstīta*, ja `services.read.mode` ir iestatīts uz ”0”
services.write.mode	integer	Lasīšanas-rakstīšanas piekļuve pakalpojumiem. Iespējamās vērtības: 0 - *(noklusējums)* Lasīšanas-rakstīšanas piekļuve pakalpojumiem, kas norādīti īpašībā `services.write.list` vai atbilst īpašībā `services.write.tag` norādītajām birkām; 1 - Lasīšanas-rakstīšanas piekļuve visiem pakalpojumiem.
services.write.list	array	Pakalpojuma objektu masīvs. Norādītajiem pakalpojumiem, ieskaitot apakšpakalpojumus, lietotāja lomai tiks piešķirta lasīšanas-rakstīšanas piekļuve. Lasīšanas-rakstīšanas piekļuve aizstās tikai lasāmu piekļuvi pakalpojumiem. Īpašības darbība: - *atbalstīta*, ja `services.write.mode` ir iestatīts uz ”0”

Īpašība	Tips	Apraksts
services.write.tag	object	Pakalpojuma birkas objektu masīvs. Pakalpojumiem, kas atbilst birkām, ieskaitot apakšpakalpojumus, lietotāja lomai tiks piešķirta lasīšanas-rakstīšanas piekļuve. Lasīšanas-rakstīšanas piekļuve aizstās tikai lasāmu piekļuvi pakalpojumiem. Īpašības darbība: - *atbalstīta*, ja `services.write.mode` ir iestatīts uz "0"
modules	array	Moduļa objektu masīvs.
modules.default_access	integer	Vai piekļuve jauniem moduļiem ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
api.access	integer	Vai piekļuve API ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
api.mode	integer	Režīms API metodēm, kas uzskaitītas īpašībā `api`. Iespējamās vērtības: 0 - *(noklusējums)* Aizliegto metožu saraksts; 1 - Atļauto metožu saraksts.
api	array	API metožu masīvs.
actions	array	Darbības objektu masīvs.
actions.default_access	integer	Vai piekļuve jaunām darbībām ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.

Lietotāja saskarnes elements

Lietotāja saskarnes elementa objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Lietotāja saskarnes elementa nosaukums. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User”, ”Admin” vai ”Super admin”: `monitoring.dashboard` - *Paneļi*; `monitoring.problems` - *Uzraudzība → Problēmas*; `monitoring.hosts` - *Uzraudzība → Hosti*; `monitoring.latest_data` - *Uzraudzība → Jaunākie dati*; `monitoring.maps` - *Uzraudzība → Kartes*; `services.services` - *Pakalpojumi → Pakalpojumi*; `services.sla_report` - *Pakalpojumi → SLA atskaite*; `inventory.overview` - *Inventārs → Pārskats*; `inventory.hosts` - *Inventārs → Hosti*; `reports.availability_report` - *Atskaites → Pieejamības atskaite*; `reports.top_triggers` - *Atskaites → 100 populārākie trigeri*. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Admin” vai ”Super admin”: `monitoring.discovery` - *Uzraudzība → Atklāšana*; `services.sla` - *Pakalpojumi → SLA*; `reports.scheduled_reports` - *Atskaites → Plānotās atskaites*; `reports.notifications` - *Atskaites → Paziņojumi*; `configuration.template_groups` - *Datu ievākšana → Veidņu grupas*; `configuration.host_groups` - *Datu ievākšana → Hostu grupas*; `configuration.templates` - *Datu ievākšana → Veidnes*; `configuration.hosts` - *Datu ievākšana → Hosti*; `configuration.maintenance` - *Datu ievākšana → Uzturēšana*; `configuration.discovery` - *Datu ievākšana → Atklāšana*; `configuration.trigger_actions` - *Brīdinājumi → Darbības → Trigeru darbības*; `configuration.service_actions` - *Brīdinājumi → Darbības → Pakalpojumu darbības*; `configuration.discovery_actions` - *Brīdinājumi → Darbības → Atklāšanas darbības*; `configuration.autoregistration_actions` - *Brīdinājumi → Darbības → Automātiskās reģistrācijas darbības*; `configuration.internal_actions` - *Brīdinājumi → Darbības → Iekšējās darbības*. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Super admin”: `reports.system_info` - *Atskaites → Sistēmas informācija*; `reports.audit` - *Atskaites → Audita žurnāls*; `reports.action_log` - *Atskaites → Darbību žurnāls*; `configuration.event_correlation` - *Datu ievākšana → Notikumu korelācija*; `administration.media_types` - *Brīdinājumi → Mediju veidi*; `administration.scripts` - *Brīdinājumi → Skripti*; `administration.user_groups` - *Lietotāji → Lietotāju grupas*; `administration.user_roles` - *Lietotāji → Lietotāju lomas*; `administration.users` - *Lietotāji → Lietotāji*; `administration.api_tokens` - *Lietotāji → API pilnvaras*; `administration.authentication` - *Lietotāji → Autentifikācija*; `administration.general` - *Administrēšana → Vispārīgi*; `administration.audit_log` - *Administrēšana → Audita žurnāls*; `administration.housekeeping` - *Administrēšana → Tīrīšana*; `administration.proxy_groups` - *Administrēšana → Starpniekserveru grupas*; `administration.proxies` - *Administrēšana → Starpniekserveri*; `administration.macros` - *Administrēšana → Makrosi*; `administration.queue` - *Administrēšana → Rinda*. Īpašības darbība: - *obligāta*

Īpašība	Tips	Apraksts
status	integer	Vai piekļuve lietotāja saskarnes elementam ir iespējota. Iespējamās vērtības: 0 - Atspējots; 1 - *(noklusējums)* Iespējots.

Pakalpojums

Īpašība	Type	Apraksts
serviceid	ID	Pakalpojuma ID. Īpašības darbība: - *obligāts*

Servisa tags

Īpašība	Type	Apraksts
tag	string	Taga nosaukums. Ja ir norādīta tukša virkne, servisa tags netiks izmantots servisa atbilstības noteikšanai. Īpašības darbība: - *required*
value	string	Taga vērtība. Ja nav norādīta vērtība vai ir norādīta tukša virkne, servisa atbilstības noteikšanai tiks izmantots tikai taga nosaukums.

Modulis

Moduļa objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
moduleid	ID	Moduļa ID. Īpašības darbība: - *required*
status	integer	Vai piekļuve modulim ir iespējota. Iespējamās vērtības: 0 - Atspējots; 1 - *(noklusējums)* Iespējots.

Darbība

Darbības objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Darbības nosaukums. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User”, ”Admin” vai ”Super admin”: `edit_dashboards` - Izveidot un rediģēt paneļus; `edit_maps` - Izveidot un rediģēt kartes; `add_problem_comments` - Pievienot problēmu komentārus; `change_severity` - Mainīt problēmas būtiskumu; `acknowledge_problems` - Apstiprināt problēmas; `suppress_problems` - Apspiest problēmas; `close_problems` - Aizvērt problēmas; `execute_scripts` - Izpildīt skriptus; `manage_api_tokens` - Pārvaldīt API marķierus; `change_problem_ranking` - Mainīt problēmas rangu no cēloņa uz simptomu un otrādi; `edit_own_media` - Atļaut izveidot/rediģēt savu mediju. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Admin” vai ”Super admin”: `edit_maintenance` - Izveidot un rediģēt uzturēšanas periodus; `manage_scheduled_reports` - Pārvaldīt ieplānotos pārskatus, `manage_sla` - Pārvaldīt SLA. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”User” vai ”Admin”: `invoke_execute_now` - ļauj izpildīt vienums pārbaudes lietotājiem, kuriem ir tikai lasīšanas atļaujas hostā. Iespējamās vērtības, ja Role objekta `type` ir iestatīts uz ”Super admin”: `edit_user_media` - Atļaut izveidot/rediģēt lietotāju medijus. Īpašības darbība: - *obligāta*
status	integer	Vai piekļuve darbības izpildei ir iespējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.

role.create

Apraksts

`object role.create(object/array roles)`

Šī metode ļauj izveidot jaunas lomas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās lomas.

Papildus standarta lomas īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
rules	array	Lomai izveidojamie lomu noteikumi.

Atgrieztās vērtības

(`object`) Atgriež objektu, kas satur izveidoto lomu ID īpašībā `roleids`. Atgriezto ID secība atbilst nodoto lomu secībai.

Piemēri

Lomas izveide

Izveidojiet lomu ar tipu "Lietotājs" un liegtu piekļuvi diviem lietotāja saskarnes elementiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Lomas noteikumi
- Lietotāja saskarnes elements
- Modulis
- Darbība

Avots

CRole::create() failā *ui/include/classes/api/services/CRole.php*.

role.delete

Apraksts

`object role.delete(array roleids)`

Šī metode ļauj dzēst lomas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo lomu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto lomu ID zem `roleids` īpašības.

Piemēri

Vairāku lietotāju lomu dzēšana

Dzēsiet divas lietotāju lomas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "role.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CRole::delete() failā *ui/include/classes/api/services/CRole.php*.

role.get

Apraksts

`integer/array role.get(object parameters)`

Metode ļauj izgūt lomas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
roleids	ID/array	Atgriezt tikai lomas ar norādītajiem ID.
selectRules	query	Atgriezt īpašību `rules` ar lomu noteikumiem.
selectUsers	query	Atgriezt īpašību `users` ar lietotājiem, kuriem šī loma ir piešķirta. Skatiet `user.get` ierobežojumus atkarībā no lietotāja tipa.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `roleid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.

Parametrs	Tips	Apraksts
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Lomas datu izgūšana

Izgūstiet lomas ”Super admin role” datus un tās piekļuves noteikumus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "role.get",
    "params": {
        "output": "extend",
        "selectRules": "extend",
        "roleids": "3"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Lomas noteikumi
- Lietotājs

Avots

CRole::get() failā *ui/include/classes/api/services/CRole.php*.

role.update

Apraksts

`object role.update(object/array roles)`

Šī metode ļauj atjaunināt esošās lomas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Atjaunināmās lomas īpašības.

Īpašībai `roleid` jābūt definētai katrai lomai, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta lomas īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
rules	array	Piekļuves noteikumi, kas jāatjaunina lomai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto lomu ID zem `roleids` īpašības.

Piemēri

Skriptu izpildes iespējas atspējošana

Atjauniniet lomu ar ID "5", atspējojiet skriptu izpildes iespēju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "actions": [
```

```
                    {
                        "name": "execute_scripts",
                        "status": "0"
                    }
                ]
            }
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

API piekļuves ierobežošana

Atjaunināt lomu ar ID ”5”, aizliegt izsaukt jebkuras ”create”, ”update” vai ”delete” metodes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "role.update",
    "params": [
        {
            "roleid": "5",
            "rules": {
                "api.access": "1",
                "api.mode": "0",
                "api": ["*.create", "*.update", "*.delete"]
            }
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "roleids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CRole::update() failā *ui/include/classes/api/services/CRole.php*.

Mediju tips

Šī klase ir paredzēta darbam ar mediju tipiem.

Objektu atsauces:

- Mediju tips
  - Webhook parametri
  - Skripta parametri
- Ziņojuma veidne

Pieejamās metodes:

- mediatype.create - izveidot jaunus mediju tipus
- mediatype.delete - dzēst mediju tipus
- mediatype.get - izgūt mediju tipus
- mediatype.update - atjaunināt mediju tipus

Mediju tipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `mediatype` API.

Medija tips

Medija tipa objektam ir šādas īpašības.

Property	Type	Description
mediatypeid	ID	Medija tipa ID. Property behavior: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Medija tipa nosaukums. Property behavior: - *obligāts* izveides operācijām
type	integer	Pārvades veids, ko izmanto medija tips. Iespējamās vērtības: 0 - E-pasts; 1 - Skripts; 2 - SMS; 4 - Webhook.
exec_path	string	Skripta faila nosaukums (piemēram, notification.sh), kas atrodas direktorijā, kas norādīta serveris konfigurācijas parametrā AlertScriptsPath. Property behavior: - *obligāts*, ja `type` ir iestatīts uz ”Script”
gsm_modem	string	GSM modema seriālās ierīces nosaukums. Property behavior: - *obligāts*, ja `type` ir iestatīts uz ”SMS”
passwd	string	Autentifikācijas parole. Property behavior: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”Normal password”
provider	integer	E-pasta pakalpojumu sniedzējs. Iespējamās vērtības: 0 - *(noklusējums)* Generic SMTP; 1 - Gmail; 2 - Gmail relay; 3 - Office365; 4 - Office365 relay.

Property	Type	Description
smtp_email	string	E-pasta adrese, no kuras tiks sūtīti paziņojumi. Property behavior: - *obligāts*, ja `type` ir iestatīts uz ”Email”
smtp_helo	string	SMTP HELO. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
smtp_server	string	SMTP serveris. Property behavior: - *obligāts*, ja `type` ir iestatīts uz ”Email”
smtp_port	integer	SMTP servera ports, pie kura pieslēgties. Noklusējums: 25. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
smtp_security	integer	Izmantojamais SMTP savienojuma drošības līmenis. Iespējamās vērtības: 0 - *(noklusējums)* Nav; 1 - STARTTLS; 2 - SSL/TLS. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
smtp_verify_host	integer	SMTP SSL hosta verifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā. Property behavior: - *atbalstīts*, ja `smtp_security` ir iestatīts uz ”STARTTLS” vai ”SSL/TLS”
smtp_verify_peer	integer	SMTP SSL partnera verifikācija. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā. Property behavior: - *atbalstīts*, ja `smtp_security` ir iestatīts uz ”STARTTLS” vai ”SSL/TLS”
smtp_authentication	integer	Izmantojamā SMTP autentifikācijas metode. Iespējamās vērtības: 0 - *(noklusējums)* Nav; 1 - Parastā parole; 2 - OAuth marķieris. OAuth autentifikācija nav atļauta e-pasta pakalpojumu sniedzējam *Office365 relay*. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”

Property	Type	Description
redirection_url	string	Zabbix lietotāja saskarne URL, uz kuru novirzīt atpakaļ OAuth autorizāciju. Noklusējums: API iestatījumu īpašības `url` vērtība ar daļu `zabbix.php?action=oauth.authorize` Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
client_id	string	Klienta identifikators, kas reģistrēts OAuth autorizācijas serverī. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
client_secret	string	Klienta slepenā atslēga, kas reģistrēta OAuth autorizācijas serverī. Pieejama tikai lietotājam ar tipu Super Admin. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
authorization_url	string	OAuth URL ar parametriem, lai iegūtu piekļuves un atsvaidzināšanas marķierus. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
token_url	string	OAuth URL, lai apmainītu autorizācijas marķieri pret piekļuves un atsvaidzināšanas marķieriem. Šo URL serveris arī izmanto, lai atsvaidzinātu nederīgu piekļuves marķieri. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
tokens_status	integer	Marķieru statusa bitu maska. Iespējamās vērtības: 0 - (noklusējums) Abi marķieri satur nederīgu vērtību 1 - Piekļuves marķieris satur derīgu vērtību 2 - Atsvaidzināšanas marķieris satur derīgu vērtību 3 - Abi marķieri satur derīgu vērtību. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
access_token	string	OAuth piekļuves marķiera vērtība. Property behavior: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
access_token_updated	timestamp	Laika zīmogs pēdējai `access_token` izmaiņai, ko veicis serveris, atsvaidzinot ar `refresh_token`, vai API marķieru izmaiņu laikā. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
access_expires_in	integer	Laiks sekundēs, pēc kura `access_token` kļūs novecojis un būs jāveic pieprasījums uz `refresh_url`. To iestata Zabbix serveris, atsvaidzinot `access_token`, vai API marķieru izmaiņu laikā. Laika zīmogs tiek aprēķināts, pieskaitot `access_token_updated` vērtību. Property behavior: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”

Property	Type	Description
refresh_token	string	OAuth atsvaidzināšanas marķiera vērtība. Property behavior: - *obligāts*, ja `smtp_authentication` ir iestatīts uz ”OAuth token”
status	integer	Vai medija tips ir iespējots. Iespējamās vērtības: 0 - *(noklusējums)* Iespējots; 1 - Atspējots.
username	string	Lietotājvārds. Property behavior: - *atbalstīts*, ja `smtp_authentication` ir iestatīts uz ”Normal password”
maxsessions	integer	Maksimālais brīdinājumu skaits, ko var apstrādāt paralēli. Iespējamās vērtības, ja `type` ir iestatīts uz ”SMS”: 1. Iespējamās vērtības, ja `type` ir iestatīts uz ”Email”, ”Script” vai ”Webhook”: 0-100. Noklusējums: 1.
maxattempts	integer	Maksimālais mēģinājumu skaits nosūtīt brīdinājumu. Iespējamās vērtības: 1-100. Noklusējums: 3.
attempt_interval	string	Intervāls starp atkārtotiem mēģinājumiem. Pieņem sekundes un laika vienību ar sufiksu. Iespējamās vērtības: 0-1h. Noklusējums: 10s.
message_format	integer	Ziņojuma formāts. Iespējamās vērtības: 0 - Vienkāršs teksts; 1 - *(noklusējums)* HTML. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Email”
script	text	Webhook skripta saturs (JavaScript). Property behavior: - *obligāts*, ja `type` ir iestatīts uz ”Webhook”
timeout	string	Webhook skripta noildze. Pieņem sekundes un laika vienību ar sufiksu. Iespējamās vērtības: 1-60s. Noklusējums: 30s. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook”

Property	Type	Description
process_tags	integer	Apstrādāt JSON īpašību vērtības Webhook skripta atbildē kā birkas. Šīs birkas tiek pievienotas visām esošajām problēmu birkām. Iespējamās vērtības: 0 - *(noklusējums)* Ignorēt webhook skripta atbildi; 1 - Apstrādāt webhook skripta atbildi kā birkas. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook”
show_event_menu	integer	Iekļaut ierakstu notikuma izvēlnē, kas norāda uz pielāgotu URL. Tas arī pievieno īpašību `urls` izvadei no problem.get un event.get. Iespējamās vērtības: 0 - *(noklusējums)* Neiekļaut notikuma izvēlnes ierakstu vai īpašību `urls`; 1 - Iekļaut notikuma izvēlnes ierakstu un īpašību `urls`. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook”
event_menu_url	string	URL, kas izmantots notikuma izvēlnes ierakstā un īpašībā `urls`, ko atgriež problem.get un event.get. Property behavior: - *atbalstīts*, ja `show_event_menu` ir iestatīts uz ”Include event menu entry and `urls` property”
event_menu_name	string	Nosaukums, kas izmantots notikuma izvēlnes ierakstā un īpašībā `urls`, ko atgriež problem.get un event.get. Property behavior: - *atbalstīts*, ja `show_event_menu` ir iestatīts uz ”Include event menu entry and `urls` property”
parameters	array	Webhook vai skripta parametri. Property behavior: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook” vai ”Script”
description	text	Medija tipa apraksts.

Webhook parametri

Webhook parametriem ir šādas īpašības.

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāts*
value	string	Parametra vērtība, atbalsta makrosus. Atbalstītie makrosi ir aprakstīti lapā Atbalstītie makrosi.

Skripta parametri

Skripta parametriem ir šādas īpašības.

Īpašība	Tips	Apraksts
sortorder	integer	Secība, kādā parametru vērtības tiks nodotas skriptam kā komandrindas argumenti, sākot ar 0 kā pirmo. Īpašības darbība: - *obligāts*

Īpašība	Tips	Apraksts
value	string	Parametra vērtība, atbalsta makrosus. Atbalstītie makrosi ir aprakstīti lapā Atbalstītie makrosi.

Ziņojuma veidne

Ziņojuma veidnes objekts definē veidni, kas tiks izmantota kā noklusējuma ziņojums darbību operācijām, lai nosūtītu paziņojumu. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
eventsource	integer	Notikuma avots. Iespējamās vērtības: 0 - Trigeri; 1 - Atklāšana; 2 - Automātiskā reģistrācija; 3 - Iekšējs; 4 - Pakalpojumi. Īpašības darbība: - *required*
recovery	integer	Operācijas režīms. Iespējamās vērtības: 0 - Operācijas; 1 - Atkopšanas operācijas; 2 - Atjaunināšanas operācijas. Īpašības darbība: - *required*
subject	string	Ziņojuma temats.
message	string	Ziņojuma teksts.

mediatype.create

Apraksts

`object mediatype.create(object/array mediaTypes)`

Šī metode ļauj izveidot jaunus mediju tipus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet User roles papildinformācijai.

Parametri

`(object/array)` Izveidojamie mediju tipi.

Papildus standarta mediju tipu īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
message_templates	array	Ziņojumu veidnes, kas jāizveido mediju tipam.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto mediju tipu ID īpašumā `mediatypeids`. Atgriezto ID secība atbilst nodoto mediju tipu secībai.

Piemēri

E-pasta medija tipa izveide

Izveidojiet jaunu e-pasta medija tipu ar pielāgotu SMTP portu un ziņojumu veidnēm.

Pieprasījums:

```
{
    "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": "Problēma: {EVENT.NAME}",
                "message": "Problēma \"{EVENT.NAME}\" hostā \"{HOST.NAME}\" sākās {EVENT.TIME}."
            },
            {
                "eventsource": "0",
                "recovery": "1",
                "subject": "Atrisināts {EVENT.DURATION}: {EVENT.NAME}",
                "message": "Problēma \"{EVENT.NAME}\" hostā \"{HOST.NAME}\" tika atrisināta {EVENT.RECOVER
            },
            {
                "eventsource": "0",
                "recovery": "2",
                "subject": "Atjaunināta problēma {EVENT.AGE}: {EVENT.NAME}",
                "message": "{USER.FULLNAME} {EVENT.UPDATE.ACTION} problēmu \"{EVENT.NAME}\" hostā \"{HOST.
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "7"
        ]
    },
    "id": 1
}
```

Skripta mediju tipa izveide

Izveidojiet jaunu skripta mediju tipu ar pielāgotu vērtību mēģinājumu skaitam un intervālam starp tiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "8"
        ]
    },
    "id": 1
}
```

Webhook paziņojumu veida izveide

Izveidojiet jaunu webhook paziņojumu veidu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "9"
        ]
    },
    "id": 1
}
```

Avots

CMediaType::create() failā *ui/include/classes/api/services/CMediaType.php*.

mediatype.delete

Apraksts

`object mediatype.delete(array mediaTypeIds)`

Šī metode ļauj dzēst mediju tipus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo mediju tipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto mediju tipu ID zem `mediatypeids` īpašības.

Piemēri

Vairāku paziņojumu veidu dzēšana

Dzēsiet divus paziņojumu veidus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.delete",
    "params": [
        "3",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "3",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CMediaType::delete() failā *ui/include/classes/api/services/CMediaType.php*.

mediatype.get

Apraksts

`integer/array mediatype.get(object parameters)`

Šī metode ļauj izgūt mediju tipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

> **Note:**
> Pieprasot ar lietotājiem saistītu informāciju par mediju tipiem, *Admin* un *User* tipa lietotāji var iegūt datus tikai par savu lietotāju. Piemēru skatiet sadaļā Mediju tipu izgūšana kā *Admin*.

Parametrs	Tips	Apraksts
mediatypeids	ID/array	Atgriezt tikai mediju tipus ar norādītajiem ID.
mediaids	ID/array	Atgriezt tikai mediju tipus, ko izmanto norādītais medijs.
userids	ID/array	Atgriezt tikai mediju tipus, ko izmanto norādītie lietotāji.
selectActions	query	Atgriezt `actions` īpašību ar darbībām, kas izmanto mediju tipu.
selectMessageTemplates	query	Atgriezt `message_templates` īpašību ar mediju tipu ziņojumu masīvu. Parametra darbība: - *atbalstīts Super admin* tipa lietotājiem
selectUsers	query	Atgriezt `users` īpašību ar lietotājiem, kas izmanto mediju tipu. Skatiet `user.get` par ierobežojumiem atkarībā no lietotāja tipa.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `mediatypeid`.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, pēc kura veikt atbilstību. Neatbalsta `text` datu tipa īpašības.
output	query	Iespējamās Mediju tipa objekta īpašības *Admin* un *User* tipa lietotājiem: `mediatypeid`, `name`, `type`, `status`, `maxattempts`. Atgriežamās Mediju tipa objekta īpašības. *Admin* un *User* tipa lietotāji var iegūt tikai šādas īpašības: `mediatypeid`, `name`, `type`, `status`, `maxattempts`, `description`. Piemēru skatiet sadaļā Mediju tipu izgūšana kā *Admin*.
search	object	Noklusējums: `extend`. Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir virknes, ko meklēt. Ja nav norādītas papildu opcijas, tiks veikta LIKE "%…%" meklēšana. Atbalsta tikai `string` un `text` datu tipa īpašības. Iespējamās Mediju tipa objekta īpašības *Admin* un *User* tipa lietotājiem: `name`, `description`.

Parametrs	Tips	Apraksts
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Paziņojumu nesēju tipu izgūšana

Izgūst visus konfigurētos paziņojumu nesēju tipus. Tālāk redzamajā piemērā tiek atgriezti divi paziņojumu nesēju tipi:

- e-pasta paziņojumu nesēja tips;
- SMS paziņojumu nesēja tips.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "selectMessageTemplates": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Paziņojumu veidu izgūšana kā *Admin*

Kā *Admin* tipa lietotājs izgūstiet visus iespējotos paziņojumu veidus kopā ar lietotājiem, kuri izmanto šos paziņojumu veidus. Tālāk redzamajā piemērā tiek atgriezti divi paziņojumu veidi:

- e-pasta paziņojumu veids ar vienu lietotāju (tikai paša *Admin* tipa lietotāja konts);
- SMS paziņojumu veids bez lietotājiem.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": "extend",
        "filter": {
            "status": 0
        },
        "selectUsers": "extend"
```

```
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skripta un webhook paziņojumu nesēju tipu izgūšana

Šajā piemērā tiek atgriezti trīs paziņojumu nesēju tipi:

- skripta paziņojumu nesēja tips ar parametriem;
- skripta paziņojumu nesēja tips bez parametriem;
- webhook paziņojumu nesēja tips ar parametriem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.get",
    "params": {
        "output": ["mediatypeid", "name", "parameters"],
```

```
        "filter": {
            "type": [1, 4]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mediatypeid": "10",
            "name": "Skripts ar parametriem",
            "parameters": [
                {
                    "sortorder": "0",
                    "value": "{ALERT.SENDTO}"
                },
                {
                    "sortorder": "1",
                    "value": "{EVENT.NAME}"
                },
                {
                    "sortorder": "2",
                    "value": "{ALERT.MESSAGE}"
                },
                {
                    "sortorder": "3",
                    "value": "Zabbix brīdinājums"
                }
            ]
        },
        {
            "mediatypeid": "13",
            "name": "Skripts bez parametriem",
            "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
}
```

Skatiet arī

- Lietotājs

Avots

CMediaType::get() failā *ui/include/classes/api/services/CMediaType.php*.

mediatype.update

Apraksts

`object mediatype.update(object/array mediaTypes)`

Šī metode ļauj atjaunināt esošos mediju tipus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet User roles papildinformācijai.

Parametri

`(object/array)` Atjaunināmā multivides tipa rekvizīti.

Rekvizītam `mediatypeid` jābūt definētam katram multivides tipam, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta multivides tipa rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
message_templates	array	Ziņojumu veidnes, ar kurām aizstāt pašreizējās ziņojumu veidnes.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto mediju tipu ID īpašībā `mediatypeids`.

Piemēri

Mediju tipa iespējošana

Iespējojiet mediju tipu, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mediatype.update",
    "params": {
        "mediatypeid": "6",
        "status": "0"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mediatypeids": [
            "6"
        ]
    },
    "id": 1
}
```

Avots

CMediaType::update() failā *ui/include/classes/api/services/CMediaType.php*.

MFA

Šī klase ir paredzēta darbam ar MFA (daudzfaktoru autentifikācijas) metodēm.

Objektu atsauces:

- MFA

Pieejamās metodes:

- mfa.create - izveidot jaunas MFA metodes
- mfa.delete - dzēst MFA metodes
- mfa.get - izgūt MFA metodes
- mfa.update - atjaunināt MFA metodes

MFA objekts

Tālāk norādītie objekti ir tieši saistīti ar `mfa` API.

MFA

MFA (daudzfaktoru autentifikācijas) objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
mfaid	ID	MFA metodes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
type	integer	MFA metodes tips. Iespējamās vērtības: 1 - TOTP (uz laiku balstītas vienreizējās paroles); 2 - Duo Universal Prompt.
name	string	Unikāls MFA metodes nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
hash_function	integer	Jaucējfunkcijas tips TOTP kodu ģenerēšanai. Iespējamās vērtības: 1 - SHA-1; 2 - SHA-256; 3 - SHA-512. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”TOTP”

Īpašība	Tips	Apraksts
code_length	integer	Verifikācijas koda garums. Iespējamās vērtības: 6 - 6 ciparus garš; 8 - 8 ciparus garš. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”TOTP”
api_hostname	string	API resursdatora nosaukums, ko nodrošina Duo autentifikācijas pakalpojums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Duo Universal Prompt”
clientid	string	Klienta ID, ko nodrošina Duo autentifikācijas pakalpojums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Duo Universal Prompt”
client_secret	string	Klienta slepenā atslēga, ko nodrošina Duo autentifikācijas pakalpojums. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `type` ir iestatīts uz ”Duo Universal Prompt”

mfa.create

Apraksts

`object mfa.create(object/array MFA methods)`

Šī metode ļauj izveidot jaunas MFA metodes.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamās MFA metodes.

Metode pieņem MFA metodes ar standarta MFA metožu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto MFA metožu ID īpašībā `mfaids`. Atgriezto ID secība atbilst nodoto vienumu secībai.

Piemēri

MFA metožu izveide

Izveidojiet ”Zabbix TOTP” MFA metodi, izmantojot uz laiku balstītas vienreizējās paroles (TOTP), kur TOTP kodu ģenerēšanas jaucējfunkcija ir iestatīta uz SHA-1 un verifikācijas koda garums ir iestatīts uz 6 cipariem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.create",
    "params": {
        "type": 1,
        "name": "Zabbix TOTP",
        "hash_function": 1,
        "code_length": 6
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CMfa::create() failā *ui/include/classes/api/services/CMfa.php*.

mfa.delete

Apraksts

`object mfa.delete(array mfaids)`

Šī metode ļauj dzēst MFA metodes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo MFA metožu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto MFA metožu ID īpašībā `mfaids`.

Piemēri

MFA metožu dzēšana

Dzēsiet MFA metodi.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.delete",
    "params": [
        "2"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "2"
        ]
    },
    "id": 1
}
```

Avots

CMfa::delete() failā *ui/include/classes/api/services/CMfa.php*.

mfa.get

Apraksts

`integer/array mfa.get(object parameters)`

Metode ļauj izgūt MFA metodes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
mfaids	ID/masīvs	Atgriezt tikai MFA metodes ar norādītajiem ID.
selectUsrgrps	query	Atgriezt `usrgrps` īpašību ar lietotāju grupām, kas ir saistītas ar MFA metodēm. Atbalsta `count`.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Atbalstītās īpašības: `mfaid` - MFA metodes ID; `type` - MFA metodes tips.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `name`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir virknes meklēšanai. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalstītās īpašības: `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

MFA metožu atrašana pēc nosaukuma

Iegūstiet visas MFA metodes, kuru nosaukumā ir "Zabbix".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.get",
    "params": {
        "output": "extend",
        "search": {
            "name": "Zabbix"
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "mfaid": "1",
            "type": "1",
            "name": "Zabbix TOTP 1",
            "hash_function": "1",
            "code_length": "6",
            "api_hostname": "",
            "clientid": ""
        },
        {
            "mfaid": "2",
            "type": "1",
            "name": "Zabbix TOTP 2",
            "hash_function": "3",
            "code_length": "8",
            "api_hostname": "",
            "clientid": ""
        }
    ],
    "id": 1
}
```

Avots

CMfa::get() failā *ui/include/classes/api/services/CMfa.php*.

mfa.update

Apraksts

`object mfa.update(object/array MFA methods)`

Šī metode ļauj atjaunināt esošās MFA metodes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo MFA metožu īpašības.

Īpašībai `mfaid` jābūt definētai katram vienumam, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Metode pieņem MFA metodes ar standarta MFA metožu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto MFA metožu ID īpašībā `mfaids`.

Piemēri

Metodes īpašību atjaunināšana

Atjauniniet jaucējfunkciju TOTP kodu ģenerēšanai un verifikācijas koda garumu MFA metodei "Zabbix TOTP", kas izmanto uz laiku balstītas vienreizējās paroles (TOTP).

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "mfa.update",
    "params": {
        "mfaid": "1",
        "hash_function": 3,
        "code_length": 8
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "mfaids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CMfa::update() failā *ui/include/classes/api/services/CMfa.php*.

Modulis

Šī klase ir paredzēta darbam ar lietotāja saskarnes moduļiem.

Objektu atsauces:

- Module

Pieejamās metodes:

- module.create - instalēt jaunus moduļus
- module.delete - atinstalēt moduļus
- module.get - izgūt moduļus
- module.update - atjaunināt moduļus

Moduļa objekts

Tālāk norādītie objekti ir tieši saistīti ar `module` API.

Modulis

Moduļa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
moduleid	ID	Moduļa ID, kā tas ir saglabāts datubāzē. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām

Īpašība	Tips	Apraksts
id	string	Unikāls moduļa ID, ko izstrādātājs definējis moduļa failā manifest.json. Iespējamās vērtības iebūvētajiem moduļiem: skatiet īpašības "type" aprakstu sadaļā Informācijas paneļa logrīks. Īpašības darbība: - *obligāts* izveides operācijām
relative_path	string	Ceļš uz moduļa direktoriju attiecībā pret Zabbix lietotāja saskarne direktoriju. Iespējamās vērtības: `widgets/*` - iebūvēto logrīku moduļiem; `modules/*` - trešo pušu moduļiem. Īpašības darbība: - *obligāts* izveides operācijām
status	integer	Vai modulis ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* Atspējots; 1 - Iespējots.
config	object	Moduļa konfigurācija.

module.create

Apraksts

`object module.create(object/array modules)`

Šī metode ļauj instalēt jaunus lietotāja saskarnes moduļus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Attention:
Moduļu faili ir manuāli jāizpako pareizajās apakšdirektorijās, lai tās atbilstu moduļu `relative_path` īpašībai.

Parametri

`(object/array)` Moduļi, ko izveidot.

Metode pieņem moduļus ar standarta moduļu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur instalēto moduļu ID īpašumā `moduleids`. Atgriezto ID secība atbilst nodoto moduļu secībai.

Piemēri

Moduļa instalēšana

Instalējiet moduli ar statusu "Enabled".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "module.create",
    "params": {
        "id": "example_module",
        "relative_path": "modules/example_module",
        "status": 1
    },
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Modulis
- Lietotāja saskarnes moduļi

Avots

CModule::create() failā *ui/include/classes/api/services/CModule.php*.

module.delete

Apraksts

`object module.delete(array moduleids)`

Šī metode ļauj atinstalēt moduļus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Attention:
Moduļu faili ir jānoņem manuāli.

Parametri

`(array)` atinstalējamo moduļu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atinstalēto moduļu ID zem `moduleids` īpašības.

Piemēri

Vairāku moduļu atinstalēšana

Atinstalējiet moduļus ”27” un ”28”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "module.delete",
    "params": [
        "27",
        "28"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "27",
            "28"
        ]
    },
    "id": 1
}
```

Avots

CModule::delete() failā *ui/include/classes/api/services/CModule.php*.

module.get

Apraksts

`integer/array module.get(object parameters)`

Metode ļauj izgūt moduļus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
moduleids	ID/array	Atgriezt tikai moduļus ar norādītajiem ID.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `moduleid`, `relative_path`.
countOutput	boolean	Šie parametri ir aprakstīti Atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Moduļa izgūšana pēc ID

Izgūstiet visus datus par moduļiem ”1”, ”2” un ”25”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "module.get",
    "params": {
        "output": "extend",
        "moduleids": [
            "1",
            "2",
            "25"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Modulis
- Paneļa logrīks
- Lietotāja saskarnes moduļi

Avots

CModule::get() failā *ui/include/classes/api/services/CModule.php*.

module.update

Apraksts

`object module.update(object/array modules)`

Šī metode ļauj atjaunināt esošos moduļus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjauninámā moduļa īpašības.

Īpašībai `moduleid` jābūt definētai katram modulim, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai norādītās īpašības.

Metode pieņem moduļus ar standarta moduļa īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto moduļu ID zem `moduleids` īpašības.

Piemēri

Moduļa atspējošana

Atspējojiet moduli "25".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "module.update",
    "params": {
        "moduleid": "25",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "moduleids": [
            "25"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Modulis
- Lietotāja saskarnes moduļi

Avots

CModule::update() failā *ui/include/classes/api/services/CModule.php*.

Notikums

Šī klase ir paredzēta darbam ar notikumiem.

Objektu atsauces:

- Notikums
- Notikuma tags
- Mediju tipa URL

Pieejamās metodes:

- event.get - izgūt notikumus
- event.acknowledge - apstiprināt notikumus

Notikuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `event` API.

Notikums

Note:
Notikumus izveido Zabbix serveris, un tos nevar modificēt, izmantojot API.

Notikuma objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
eventid	ID	Notikuma ID.
source	integer	Notikuma tips. Iespējamās vērtības: 0 - notikums, ko izveidojis trigeris; 1 - notikums, ko izveidojis atklāšanas noteikums; 2 - notikums, ko izveidojusi aktīva aģenta automātiskā reģistrācija; 3 - iekšējs notikums; 4 - notikums, kas izveidots, atjauninot pakalpojuma statusu.
object	integer	Objekta tips, kas ir saistīts ar notikumu. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, ko izveidojis trigeris”: 0 - trigeris. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, ko izveidojis atklāšanas noteikums”: 1 - atklāts hosts; 2 - atklāts pakalpojums. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, ko izveidojusi aktīva aģenta automātiskā reģistrācija”: 3 - automātiski reģistrēts hosts. Iespējamās vērtības, ja `source` ir iestatīts uz ”iekšējs notikums”: 0 - trigeris; 4 - vienums; 5 - LLD noteikums. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, kas izveidots, atjauninot pakalpojuma statusu”: 6 - pakalpojums.
objectid	ID	Saistītā objekta ID.
acknowledged	integer	Vai notikums ir apstiprināts.
clock	timestamp	Laiks, kad notikums tika izveidots.
ns	integer	Nanosekundes, kad notikums tika izveidots.
name	string	Atrisinātais notikuma nosaukums.

Īpašība	Tips	Apraksts
value	integer	Saistītā objekta stāvoklis. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, ko izveidojis trigeris” vai ”notikums, kas izveidots, atjauninot pakalpojuma statusu”: 0 - OK; 1 - problēma. Iespējamās vērtības, ja `source` ir iestatīts uz ”notikums, ko izveidojis atklāšanas noteikums”: 0 - hosts vai pakalpojums darbojas; 1 - hosts vai pakalpojums nedarbojas; 2 - hosts vai pakalpojums atklāts; 3 - hosts vai pakalpojums zaudēts. Iespējamās vērtības, ja `source` ir iestatīts uz ”iekšējs notikums”: 0 - ”normāls” stāvoklis; 1 - ”nezināms” vai ”nav atbalstīts” stāvoklis. Īpašības darbība: - *atbalstīts*, ja `source` ir iestatīts uz ”notikums, ko izveidojis trigeris”, ”notikums, ko izveidojis atklāšanas noteikums”, ”iekšējs notikums” vai ”notikums, kas izveidots, atjauninot pakalpojuma statusu”
severity	integer	Notikuma pašreizējā nopietnība. Iespējamās vērtības: 0 - nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidēja; 4 - augsta; 5 - katastrofa.
r_eventid	ID	Atkopšanas notikuma ID.
c_eventid	ID	Notikuma ID, kas tika izmantots, lai aizstātu (aizvērtu) pašreizējo notikumu saskaņā ar globālās korelācijas noteikumu. Skatiet `correlationid`, lai noteiktu precīzu korelācijas noteikumu. Šis parametrs ir definēts tikai tad, ja notikums ir aizvērts ar globālās korelācijas noteikumu.
cause_eventid	ID	Cēloņa notikuma ID.
correlationid	ID	Korelācijas noteikuma ID, kas izraisīja problēmas aizvēršanu. Šis parametrs ir definēts tikai tad, ja notikums ir aizvērts ar globālās korelācijas noteikumu.
userid	ID	Lietotāja ID, kurš aizvēra notikumu (ja notikums tika aizvērts manuāli).
suppressed	integer	Vai notikums ir apspiests. Iespējamās vērtības: 0 - notikums ir normālā stāvoklī; 1 - notikums ir apspiests.
opdata	string	Operatīvie dati ar izvērstām makrokomandām.
urls	array	Aktīvie mediju tipa URL.

Notikuma tags

Notikuma taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Notikuma taga nosaukums.
value	string	Notikuma taga vērtība.

Mediju tipa URL

Mediju tipa URL objektam ir šādas īpašības.

Īpašība	Type	Apraksts
name	string	Mediju tipā definētais URL nosaukums.
url	string	Mediju tipā definētā URL vērtība.

Rezultātos būs iekļauti ieraksti tikai aktīviem mediju tipiem ar iespējotu notikumu izvēlnes ierakstu. Īpašībās izmantotie makrosi tiks izvērsti, bet, ja kāda no īpašībām saturēs neizvērstu makrosu, abas īpašības tiks izslēgtas no rezultātiem. Atbalstītos makrosus skatiet *Atbalstītie makrosi*.

event.acknowledge

Apraksts

`object event.acknowledge(object/array parameters)`

Šī metode ļauj atjaunināt notikumus. Var veikt šādas atjaunināšanas darbības:

- Aizvērt notikumu. Ja notikums jau ir atrisināts, šī darbība tiks izlaista.
- Apstiprināt notikumu. Ja notikums jau ir apstiprināts, šī darbība tiks izlaista.
- Atcelt notikuma apstiprinājumu. Ja notikums nav apstiprināts, šī darbība tiks izlaista.
- Pievienot ziņojumu.
- Mainīt notikuma nopietnību. Ja notikumam jau ir tāda pati nopietnība, šī darbība tiks izlaista.
- Apspiest notikumu. Ja notikums jau ir apspiests, šī darbība tiks izlaista.
- Atcelt notikuma apspiešanu. Ja notikums nav apspiests, šī darbība tiks izlaista.
- Mainīt notikuma rangu uz cēloni. Ja notikums jau ir klasificēts kā cēlonis, šī darbība tiks izlaista.
- Mainīt notikuma rangu uz simptomu. Ja notikums jau ir klasificēts kā simptoms, šī darbība tiks izlaista.

Attention:
Atjaunināt var tikai trigeru notikumus.
 Atjaunināt var tikai problēmu notikumus.
 Lai aizvērtu notikumu vai mainītu notikuma nopietnību, ir nepieciešamas trigerim piešķirtas lasīšanas/rakstīšanas tiesības.
 Lai aizvērtu notikumu, trigerī jābūt atļautai manuālai aizvēršanai.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Parametri, kas satur notikumu ID un atjaunināšanas darbības, kas jāveic.

Parametrs	Type	Apraksts
eventids	ID/array	Apstiprināmo notikumu ID. Parametra darbība: - *obligāts*

Parametrs	Type	Apraksts
action	integer	Notikumu atjaunināšanas darbība(-as). Iespējamās bitkartes vērtības: 1 - aizvērt problēmu; 2 - apstiprināt notikumu; 4 - pievienot ziņojumu; 8 - mainīt svarīguma pakāpi; 16 - noņemt notikuma apstiprinājumu; 32 - apspiest notikumu; 64 - atcelt notikuma apspiešanu; 128 - mainīt notikuma rangu uz cēloni; 256 - mainīt notikuma rangu uz simptomu. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 34 notikuma apstiprināšanai un apspiešanai). Parametra darbība: - *obligāts*
cause_eventid	ID	Cēloņa notikuma ID. Parametra darbība: - *obligāts*, ja `action` satur bitu ”mainīt notikuma rangu uz simptomu”
message	string	Ziņojuma teksts. Parametra darbība: - *obligāts*, ja `action` satur bitu ”pievienot ziņojumu”
severity	integer	Jaunā notikumu svarīguma pakāpe. Iespējamās vērtības: 0 - nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa. Parametra darbība: - *obligāts*, ja `action` satur bitu ”mainīt svarīguma pakāpi”
suppress_until	integer	Unix laikspiedols, līdz kuram notikums ir jāapslāpē. Ja iestatīts uz ”0”, apspiešana būs bez termiņa. Parametra darbība: - *obligāts*, ja `action` satur bitu ”apspiest notikumu”

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto notikumu ID īpašībā `eventids`.

Piemēri

Notikuma apstiprināšana

Apstipriniet vienu notikumu un atstājiet ziņojumu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": "20427",
        "action": 6,
```

```
        "message": "Problem resolved."
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427"
        ]
    },
    "id": 1
}
```

Notikuma nopietnības maiņa

Mainiet vairāku notikumu nopietnību un atstājiet ziņojumu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "event.acknowledge",
    "params": {
        "eventids": ["20427", "20428"],
        "action": 12,
        "message": "Maintenance required to fix it.",
        "severity": 4
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "eventids": [
            "20427",
            "20428"
        ]
    },
    "id": 1
}
```

Avots

CEvent::acknowledge() failā *ui/include/classes/api/services/CEvent.php*.

event.get

Apraksts

`integer/array event.get(object parameters)`

Metode ļauj izgūt notikumus atbilstoši norādītajiem parametriem.

Attention:
Šī metode var atgriezt dzēstas entītijas notikumus, ja šie notikumi vēl nav noņemti ar housekeeper.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
eventids	ID/array	Atgriezt tikai notikumus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai notikumus, ko izveidojuši objekti, kas pieder norādītajām hostu grupām.
hostids	ID/array	Atgriezt tikai notikumus, ko izveidojuši objekti, kas pieder norādītajiem hostiem.
objectids	ID/array	Atgriezt tikai notikumus, ko izveidojuši norādītie objekti.
source	integer	Atgriezt tikai notikumus ar norādīto tipu. Atbalstīto notikumu tipu sarakstu skatiet notikuma objekta lapā. Noklusējums: 0 - trigeru notikumi.
object	integer	Atgriezt tikai notikumus, ko izveidojuši norādītā tipa objekti. Atbalstīto objektu tipu sarakstu skatiet notikuma objekta lapā. Noklusējums: 0 - trigeris.
acknowledged	boolean	Ja iestatīts uz `true`, atgriezt tikai apstiprinātos notikumus.
action	integer	Atgriezt tikai notikumus, kuriem ir veiktas norādītās notikumu atjaunināšanas darbības. Vairāku darbību gadījumā izmantojiet jebkuru pieļaujamo bitkartes vērtību summu kā bitmasku (piemēram, 34 notikuma apstiprināšanai un apspiešanai).
action_userids	ID/array	Atgriezt tikai notikumus ar norādītajiem to lietotāju ID, kuri veica notikumu atjaunināšanas darbības.
suppressed	boolean	Ja iestatīts uz `true`, atgriezt tikai apspiestos notikumus.
symptom	boolean	Ja iestatīts uz `true`, atgriezt tikai simptomu notikumus.
severities	integer/array	Atgriezt tikai notikumus ar norādītajām notikumu nopietnības pakāpēm. Attiecas tikai tad, ja objekts ir trigeris.
trigger_severities	integer/array	Atgriezt tikai notikumus ar norādītajām trigeru nopietnības pakāpēm. Attiecas tikai tad, ja objekts ir trigeris.
evaltype	integer	Birku novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
tags	array	Atgriezt tikai notikumus ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus notikumus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
eventid_from	string	Atgriezt tikai notikumus ar ID, kas ir lielāki vai vienādi ar norādīto ID.
eventid_till	string	Atgriezt tikai notikumus ar ID, kas ir mazāki vai vienādi ar norādīto ID.
time_from	timestamp	Atgriezt tikai notikumus, kas izveidoti norādītajā laikā vai vēlāk.
time_till	timestamp	Atgriezt tikai notikumus, kas izveidoti norādītajā laikā vai agrāk.
problem_time_from	timestamp	Atgriež tikai notikumus, kas bija problēmas stāvoklī no `problem_time_from` neatkarīgi no to pašreizējā statusa. Attiecas tikai tad, ja avots ir trigera notikums un objekts ir trigeris. Šis parametrs ir obligāts, ja ir norādīts `problem_time_till`.

Parametrs	Tips	Apraksts
problem_time_till	timestamp	Atgriež tikai notikumus, kas bija problēmas stāvoklī līdz `problem_time_till` neatkarīgi no to pašreizējā statusa. Attiecas tikai tad, ja avots ir trigera notikums un objekts ir trigeris. Šis parametrs ir obligāts, ja ir norādīts `problem_time_from`.
value	integer/array	Atgriezt tikai notikumus ar norādītajām vērtībām.
selectAcknowledges	query	Atgriezt `acknowledges` īpašību ar notikumu atjauninājumiem. Notikumu atjauninājumi tiek kārtoti apgrieztā hronoloģiskā secībā. Notikuma atjauninājuma objektam ir šādas īpašības: `acknowledgeid` - `(ID)` apstiprinājuma ID; `userid` - `(ID)` tā lietotāja ID, kurš atjauninājа notikumu; `clock` - `(timestamp)` laiks, kad notikums tika atjaunināts; `message` - `(string)` ziņojuma teksts; `action` - `(integer)` veiktā atjaunināšanas darbība, skatiet `event.acknowledge`; `old_severity` - `(integer)` notikuma nopietnības pakāpe pirms šīs atjaunināšanas darbības; `new_severity` - `(integer)` notikuma nopietnības pakāpe pēc šīs atjaunināšanas darbības; `suppress_until` - `(timestamp)` laiks, līdz kuram notikums būs apspiests; `taskid` - `(ID)` uzdevuma ID, ja pašlaik notikumam tiek mainīts rangs; `username` - `(string)` tā lietotāja lietotājvārds, kurš atjauninājа notikumu; `name` - `(string)` tā lietotāja vārds, kurš atjauninājа notikumu; `surname` - `(string)` tā lietotāja uzvārds, kurš atjauninājа notikumu. Atbalsta `count`.
selectAlerts	query	Atgriezt `alerts` īpašību ar notikuma ģenerētajiem brīdinājumiem. Brīdinājumi tiek kārtoti apgrieztā hronoloģiskā secībā.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kuros atrodas objekts, kas izveidoja notikumu. Atbalstīts tikai notikumiem, ko ģenerējuši trigeri, vienumi vai LLD kārtulas.
selectRelatedObject	query	Atgriezt `relatedObject` īpašību ar objektu, kas izveidoja notikumu. Atgrieztā objekta tips ir atkarīgs no notikuma tipa.
selectSuppressionData	query	Atgriezt `suppression_data` īpašību ar aktīvo uzturēšanu un manuālo apspiešanu sarakstu: `maintenanceid` - `(ID)` uzturēšanas ID; `userid` - `(ID)` tā lietotāja ID, kurš apspieda notikumu; `suppress_until` - `(integer)` laiks, līdz kuram notikums ir apspiests.
selectTags	query	Atgriezt `tags` īpašību ar notikuma birkām.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `eventid`, `objectid`, `clock`. Iespējamās vērtības, ja lieto kopā ar `groupBy`: `objectid`. Iespējamās vērtības, ja lieto kopā ar `countOutput` un `groupBy`: `objectid`, `rowscount`.
groupBy	string/array	Grupēt rezultātus pēc norādītajām īpašībām. Norādītās īpašības tiks atgrieztas rezultātos. Iespējamās vērtības: `objectid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.

Parametrs	Tips	Apraksts
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`, bet nav izmantots parametrs `groupBy`
- objektu masīvu ar agregācijas rezultātiem, ja ir izmantots parametrs `groupBy`

Piemēri

Trigera notikumu izgūšana

Izgūstiet jaunākos notikumus no trigera "22395".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "selectAcknowledges": "extend",
        "selectSuppressionData": "extend",
        "selectTags": "extend",
        "objectids": "22395",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20",
            "source": "0",
            "object": "0",
            "objectid": "22395",
            "clock": "1728658089",
            "value": "0",
            "acknowledged": "0",
            "ns": "461809482",
            "name": "Vidējā slodze ir pārāk augsta (slodze uz CPU pārsniedz 1.5 5 minūtes)",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [],
            "opdata": "Vidējās slodzes (1m 5m 15m): (1.6328 3.0522 2.2515), CPU skaits: 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": "Vidējā slodze ir pārāk augsta (slodze uz CPU pārsniedz 1.5 5 minūtes)",
            "severity": "3",
            "r_eventid": "20",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "acknowledges": [
                {
                    "acknowledgeid": "1",
                    "userid": "1",
                    "clock": "1728657938",
                    "message": "Testēšanas vide. Lūdzu, ignorējiet šo brīdinājumu.",
                    "action": "38",
                    "old_severity": "0",
                    "new_severity": "0",
                    "suppress_until": "1728744338",
                    "taskid": "0",
                    "username": "Admin",
                    "name": "Zabbix",
                    "surname": "Administrator"
                }
            ],
            "opdata": "Vidējās slodzes (1m 5m 15m): (1.6328 3.0522 2.2515), CPU skaits: 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
}
```


Notikumu izgūšana pēc laika perioda

Izgūstiet visus notikumus, kas ir izveidoti no 2012. gada 17. līdz 18. oktobrim, apgrieztā hronoloģiskā secībā.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "event.get",
    "params": {
        "output": "extend",
        "time_from": "1350432000",
        "time_till": "1350518400",
        "sortfield": ["clock", "eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "eventid": "20617",
            "source": "0",
            "object": "0",
            "objectid": "14282",
            "clock": "1350477816",
            "value": "1",
            "acknowledged": "0",
```

```
            "ns": "0",
            "name": "Mazāk nekā 25% brīvas vietas vēstures kešatmiņā",
            "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 procesi ir noslogoti vairāk nekā par 75%",
            "severity": "0",
            "r_eventid": "0",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        },
        {
            "eventid": "20615",
            "source": "0",
            "object": "0",
            "objectid": "14281",
            "clock": "1350477541",
            "value": "1",
            "acknowledged": "0",
            "ns": "0",
            "name": "Zabbix trapper procesi ir noslogoti vairāk nekā par 75%",
            "severity": "3",
            "r_eventid": "20616",
            "c_eventid": "0",
            "correlationid": "0",
            "userid": "0",
            "cause_eventid": "0",
            "opdata": "",
            "suppressed": "0",
            "urls": []
        }
    ],
    "id": 1
}
```


Notikumu izgūšana, kurus apstiprinājis norādītais lietotājs

Notikumu izgūšana, kurus apstiprinājis lietotājs ar ID=10

Pieprasījums:


```
{
    "jsonrpc": "2.0",
```

```
    "method": "event.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Augstāko trigeru izgūšana pēc problēmu notikumu skaita

Izgūstiet 5 augstākos trigerus ar nopietnības līmeņiem ”Brīdinājums”, ”Vidējs”, ”Augsts” vai ”Katastrofa”, kopā ar problēmu notikumu skaitu norādītajā laika periodā.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Brīdinājums
- Vienums
- Hosts
- LLD noteikums
- Pakalpojums
- Trigeris

Avots

CEvent::get() failā *ui/include/classes/api/services/CEvent.php*.

Pakalpojums

Šī klase ir paredzēta darbam ar IT infrastruktūras/uzņēmējdarbības pakalpojumiem.

Objektu atsauces:

- Pakalpojums
- Statusa noteikums
- Pakalpojuma birka
- Pakalpojuma trauksme
- Problēmas birka

Pieejamās metodes:

- service.create - izveidot jaunus pakalpojumus
- service.delete - dzēst pakalpojumus
- service.get - izgūt pakalpojumus

- service.update - atjaunināt pakalpojumus

Pakalpojuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `service` API.

Pakalpojums

Pakalpojuma objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
serviceid	ID	Pakalpojuma ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
algorithm	integer	Statusa aprēķināšanas noteikums. Piemērojams tikai tad, ja pastāv bērnpakalpojumi. Iespējamās vērtības: 0 - iestatīt statusu uz OK; 1 - viskritiskākais, ja visiem bērniem ir problēmas; 2 - viskritiskākais no bērnpakalpojumiem. Īpašības darbība: - *obligāts* izveides operācijām
name	string	Pakalpojuma nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
sortorder	integer	Pakalpojuma pozīcija, ko izmanto kārtošanai. Iespējamās vērtības: 0-999. Īpašības darbība: - *obligāts* izveides operācijām
weight	integer	Pakalpojuma svars. Iespējamās vērtības: 0-1000000. Noklusējums: 0.
propagation_rule	integer	Statusa izplatīšanas noteikums. Iespējamās vērtības: 0 - *(noklusējums)* izplatīt pakalpojuma statusu tādu, kāds tas ir, bez jebkādām izmaiņām; 1 - palielināt izplatīto statusu par norādīto `propagation_value` (par 1 līdz 5 nopietnības līmeņiem); 2 - samazināt izplatīto statusu par norādīto `propagation_value` (par 1 līdz 5 nopietnības līmeņiem); 3 - ignorēt šo pakalpojumu - statuss vispār netiek izplatīts uz vecākpakalpojumu; 4 - iestatīt fiksētu pakalpojuma statusu, izmantojot norādīto `propagation_value`. Īpašības darbība: - *obligāts*, ja ir iestatīts `propagation_value`

Īpašība	Tips	Apraksts
propagation_value	integer	Statusa izplatīšanas vērtība. Iespējamās vērtības, ja `propagation_rule` ir iestatīts uz ”0” vai ”3”: 0 - Nav klasificēts. Iespējamās vērtības, ja `propagation_rule` ir iestatīts uz ”1” vai ”2”: 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Iespējamās vērtības, ja `propagation_rule` ir iestatīts uz ”4”: -1 - OK; 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Īpašības darbība: - *obligāts*, ja ir iestatīts `propagation_rule`
status	integer	Vai pakalpojums ir OK vai problēmas stāvoklī. Ja pakalpojums ir problēmas stāvoklī, `status` ir vienāds ar vienu no šīm vērtībām: - viskritiskākās problēmas nopietnības līmeni; - augstāko statusu bērnpakalpojumam, kas ir problēmas stāvoklī. Ja pakalpojums ir OK stāvoklī, `status` ir vienāds ar: -1. Īpašības darbība: - *tikai lasāms*
description	string	Pakalpojuma apraksts.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto pakalpojumu sasaistīšanai ar jau esošajiem. Tiek ģenerēts automātiski, ja nav norādīts.
created_at	integer	Unix laikspiedols, kad pakalpojums tika izveidots.
readonly	integer	Piekļuve pakalpojumam. Iespējamās vērtības: 0 - Lasīšana-rakstīšana; 1 - Tikai lasīšana. Īpašības darbība: - *tikai lasāms*

Statusa noteikums

Statusa noteikuma objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Nosacījums statusa (Jaunais statuss) iestatīšanai. Iespējamās vērtības: 0 - ja vismaz (N) bērnservisiem ir statuss (Status) vai augstāks; 1 - ja vismaz (N%) bērnservisu ir statuss (Status) vai augstāks; 2 - ja mazāk nekā (N) bērnservisiem ir statuss (Status) vai zemāks; 3 - ja mazāk nekā (N%) bērnservisu ir statuss (Status) vai zemāks; 4 - ja bērnservisu ar statusu (Status) vai augstāku svars ir vismaz (W); 5 - ja bērnservisu ar statusu (Status) vai augstāku svars ir vismaz (N%); 6 - ja bērnservisu ar statusu (Status) vai zemāku svars ir mazāks par (W); 7 - ja bērnservisu ar statusu (Status) vai zemāku svars ir mazāks par (N%). Kur: - N (W) ir `limit_value`; - (Status) ir `limit_status`; - (Jaunais statuss) ir `new_status`. Īpašības darbība: - *obligāta*
limit_value	integer	Robežvērtība. Iespējamās vērtības: - N un W: 1-100000; - N%: 1-100. Īpašības darbība: - *obligāta*
limit_status	integer	Robežstatuss. Iespējamās vērtības: -1 - Labi; 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Īpašības darbība: - *obligāta*
new_status	integer	Jaunā statusa vērtība. Iespējamās vērtības: 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Īpašības darbība: - *obligāta*

Pakalpojuma tags

Pakalpojuma taga objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Pakalpojuma taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Pakalpojuma taga vērtība.

Pakalpojuma trauksme

Note:
Pakalpojuma trauksmes nevar tieši izveidot, atjaunināt vai dzēst, izmantojot Zabbix API.

Pakalpojuma trauksmes objekti attēlo pakalpojuma stāvokļa izmaiņu. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad notikusi pakalpojuma stāvokļa izmaiņa.
value	integer	Pakalpojuma statuss. Iespējamo vērtību sarakstu skatiet pakalpojuma `status` īpašībā.

Problēmas tags

Problēmu tagi ļauj sasaistīt pakalpojumus ar problēmu notikumiem. Problēmas taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Problēmas taga nosaukums. Īpašības darbība: - *obligāts*
operator	integer	Kartēšanas nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* Vienāds ar; 2 - Satur.
value	string	Problēmas taga vērtība.

service.create

Apraksts

`object service.create(object/array services)`

Šī metode ļauj izveidot jaunus pakalpojumus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` pakalpojumi, ko izveidot.

Papildus standarta pakalpojuma īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
children	array	Pakārtotie pakalpojumi, kas jāpiesaista pakalpojumam. Pakārtotajiem pakalpojumiem jābūt definētai tikai `serviceid` īpašībai.

Parametrs	Type	Apraksts
parents	array	Vecākpakalpojumi pakalpojumi, kas jāpiesaista pakalpojumam. Vecākpakalpojumiem jābūt definētai tikai `serviceid` īpašībai.
tags	array	Pakalpojuma birkas, kas jāizveido pakalpojumam.
problem_tags	array	Problēmu birkas, kas jāizveido pakalpojumam.
status_rules	array	Statusa noteikumi, kas jāizveido pakalpojumam.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto pakalpojumu ID īpašībā `serviceids`. Atgriezto ID secība atbilst nodoto pakalpojumu secībai.

Piemēri

Pakalpojuma izveide

Izveidojiet pakalpojumu, kas tiks pārslēgts problēmas stāvoklī, ja vismaz vienam apakšpakalpojumam būs problēma.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "service.create",
    "params": {
        "name": "Server 1",
        "algorithm": 1,
        "sortorder": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CService::create() failā *ui/include/classes/api/services/CService.php*.

service.delete

Apraksts

`object service.delete(array serviceIds)`

Šī metode ļauj dzēst pakalpojumus.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo servisu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto pakalpojumu ID zem `serviceids` īpašības.

Piemēri

Vairāku pakalpojumu dzēšana

Dzēsiet divus pakalpojumus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "service.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CService::delete() failā *ui/include/classes/api/services/CService.php*.

service.get

Apraksts

`integer/array service.get(object parameters)`

Metode ļauj izgūt pakalpojumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
serviceids	ID/array	Atgriezt tikai pakalpojumus ar norādītajiem ID.
parentids	ID/array	Atgriezt tikai pakalpojumus, kas ir saistīti ar norādītajiem vecākpakalpojumiem.
deep_parentids	flag	Atgriezt visus tiešos un netiešos bērnpakalpojumus. Lieto kopā ar `parentids`.
childids	ID/array	Atgriezt tikai pakalpojumus, kas ir saistīti ar norādītajiem bērnpakalpojumiem.
evaltype	integer	Birku novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.

Parametrs	Tips	Apraksts
tags	object/array	Atgriezt tikai pakalpojumus ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus pakalpojumus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
problem_tags	object/array	Atgriezt tikai pakalpojumus ar norādītajām problēmu birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus pakalpojumus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
without_problem_tags	flag	Atgriezt tikai pakalpojumus bez problēmu birkām.
slaids	ID/array	Atgriezt tikai pakalpojumus, kas ir saistīti ar konkrēto SLA vai konkrētajiem SLA.
selectChildren	query	Atgriezt `children` īpašību ar bērnpakalpojumiem. Atbalsta `count`.
selectParents	query	Atgriezt `parents` īpašību ar vecākpakalpojumiem. Atbalsta `count`.
selectTags	query	Atgriezt `tags` īpašību ar pakalpojumu birkām. Atbalsta `count`.
selectProblemEvents	query	Atgriezt `problem_events` īpašību ar problēmu notikumu objektu masīvu. Problēmu notikuma objektam ir šādas īpašības: `eventid` - `(ID)` Notikuma ID; `severity` - `(string)` Pašreizējā notikuma nopietnība; `name` - `(string)` Atrisinātā notikuma nosaukums. Atbalsta `count`.
selectProblemTags	query	Atgriezt `problem_tags` īpašību ar problēmu birkām. Atbalsta `count`.
selectStatusRules	query	Atgriezt `status_rules` īpašību ar statusa noteikumiem. Atbalsta `count`.
selectStatusTimeline	object/array	Atgriezt `status_timeline` īpašību, kas satur pakalpojuma stāvokļa izmaiņas norādītajos periodos. Formāts `[{"period_from": "<period_from>", "period_to": "<period_to>"}, ...]` — `period_from` ir perioda sākuma datums (ieskaitot; integer timestamp), bet `period_to` ir perioda beigu datums (neieskaitot; integer timestamp) periodam, kas jūs interesē. Atgriež ierakstu masīvu, kas satur `start_value` īpašību un `alarms` masīvu ar stāvokļa izmaiņām norādītajos periodos.

Parametrs	Tips	Apraksts
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `serviceid`, `name`, `status`, `sortorder`, `created_at`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Visu pakalpojumu izgūšana

Izgūstiet visus datus par visiem pakalpojumiem un to atkarībām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "service.get",
    "params": {
        "output": "extend",
        "selectChildren": "extend",
        "selectParents": ["serviceid", "name"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CService::get() failā *ui/include/classes/api/services/CService.php*.

service.update

Apraksts

`object service.update(object/array services)`

Šī metode ļauj atjaunināt esošos pakalpojumus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` atjaunināmās pakalpojuma īpašības.

Īpašībai `serviceid` jābūt definētai katram pakalpojumam, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta pakalpojuma īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
children	array	Pakārtotie pakalpojumi, ar kuriem aizstāt pašreizējos pakārtotos pakalpojumus.
parents	array	Pakārtotajiem pakalpojumiem jābūt definētai tikai īpašībai `serviceid`. Vecākpakalpojumi pakalpojumi, ar kuriem aizstāt pašreizējos vecākpakalpojumus.
tags	array	Vecākpakalpojumiem jābūt definētai tikai īpašībai `serviceid`. Pakalpojuma birkas, ar kurām aizstāt pašreizējās pakalpojuma birkas.
problem_tags	array	Problēmu birkas, ar kurām aizstāt pašreizējās problēmu birkas.
status_rules	array	Statusa noteikumi, ar kuriem aizstāt pašreizējos statusa noteikumus.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto servisu ID zem `serviceids` īpašības.

Piemēri

Vecāka pakalpojuma iestatīšana pakalpojumam

Padariet pakalpojumu ar ID ”3” par vecāku pakalpojumu pakalpojumam ar ID ”5”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
```

```
        "serviceid": "5",
        "parents": [
            {
                "serviceid": "3"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "5"
        ]
    },
    "id": 1
}
```

Plānotas dīkstāves pievienošana

Pievienojiet dīkstāvi pakalpojumam ar ID ”4”, kas ir ieplānota katru nedēļu no pirmdienas 22:00 līdz otrdienas 10:00.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "service.update",
    "params": {
        "serviceid": "4",
        "times": [
            {
                "type": "1",
                "ts_from": "165600",
                "ts_to": "201600"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "serviceids": [
            "4"
        ]
    },
    "id": 1
}
```

Avots

CService::update() failā *ui/include/classes/api/services/CService.php*.

Panelis

Šī klase ir paredzēta darbam ar paneļiem.

Objektu atsauces:

- Panelis

- Paneļa lapa
    - Paneļa logrīks
        - Paneļa logrīka lauks
- Paneļa lietotāju grupa
- Paneļa lietotājs

Pieejamās metodes:

- dashboard.create - izveidot jaunus paneļus
- dashboard.delete - dzēst paneļus
- dashboard.get - izgūt paneļus
- dashboard.update - atjaunināt paneļus

Paneļa objekts

Tālāk norādītie objekti ir tieši saistīti ar `dashboard` API.

Panelis

Paneļa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
dashboardid	ID	Paneļa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Paneļa nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
userid	ID	Tā lietotāja ID, kurš ir paneļa īpašnieks.
private	integer	Paneļa koplietošanas tips. Iespējamās vērtības: 0 - publisks panelis; 1 - *(noklusējums)* privāts panelis.
display_period	integer	Noklusējuma lapas attēlošanas periods (sekundēs). Iespējamās vērtības: 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 30.
auto_start	integer	Automātiski sākt slaidrādi. Iespējamās vērtības: 0 - nesākt slaidrādi automātiski; 1 - *(noklusējums)* sākt slaidrādi automātiski.

Paneļa lapa

Paneļa lapas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
dashboard_pageid	ID	Paneļa lapas ID. Īpašības darbība: - *tikai lasāms*
name	string	Paneļa lapas nosaukums. Noklusējums: tukša virkne.

Īpašība	Tips	Apraksts
display_period	integer	Paneļa lapas attēlošanas periods (sekundēs). Iespējamās vērtības: 0, 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 0 (tiks izmantots noklusējuma lapas attēlošanas periods).
widgets	array	Paneļa logrīka objektu masīvs.

Informācijas paneļa logrīks

Informācijas paneļa logrīka objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
widgetid	ID	Informācijas paneļa logrīka ID. Īpašības darbība: - *tikai lasāms*
type	string	Informācijas paneļa logrīka tips. Iespējamās vērtības: actionlog - Darbību žurnāls; clock - Pulkstenis; discovery - Atklāšanas statuss; favgraphs - Iecienītākie grafiki; favmaps - Iecienītākās kartes; gauge - Mērskala; geomap - Ģeokarte; graph - Grafiks (klasisks); graphprototype - Grafika prototips; honeycomb - Šūnveida skats; hostavail - Hostu pieejamība; hostcard - Hosta karte; hostnavigator - Hosta navigators; itemcard - Vienuma karte; itemhistory - Vienuma vēsture; itemnavigator - Vienuma navigators; item - Vienuma vērtība; map - Karte; navtree - Kartes navigācijas koks; piechart - Sektoru diagramma; problemhosts - Problemātiskie hosti; problems - Problēmas; problemsbysv - Problēmas pēc svarīguma pakāpes; scatterplot - Izkliedes diagramma; slareport - SLA atskaite; svggraph - Grafiks; systeminfo - Sistēmas informācija; tophosts - Top hosti; topitems - Top vienumi; toptriggers - Top trigeri; trigover - Trigeru pārskats; url - URL; web - Tīmekļa uzraudzība. Īpašības darbība: - *obligāts*
name	string	Pielāgots logrīka nosaukums.
x	integer	Horizontālā pozīcija no informācijas paneļa kreisās malas. Iespējamās vērtības ir diapazonā no 0 līdz 71.

Īpašība	Tips	Apraksts
y	integer	Vertikālā pozīcija no informācijas paneļa augšējās malas. Iespējamās vērtības ir diapazonā no 0 līdz 63.
width	integer	Logrīka platums. Iespējamās vērtības ir diapazonā no 1 līdz 72.
height	integer	Logrīka augstums. Iespējamās vērtības ir diapazonā no 1 līdz 64.
view_mode	integer	Logrīka skata režīms. Iespējamās vērtības: 0 - *(noklusējums)* noklusējuma logrīka skats; 1 - ar paslēptu galveni;
fields	array	Masīvs ar informācijas paneļa logrīka lauka objektiem. Īpašības darbība: - skatiet atsevišķus logrīkus sadaļā Informācijas paneļa logrīku lauki

Informācijas paneļa logrīka lauks

Informācijas paneļa logrīka lauka objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Logrīka lauka tips. Iespējamās vērtības: 0 - Vesels skaitlis; 1 - Virkne; 2 - Hostu grupa; 3 - Hosts; 4 - Vienums; 5 - Vienuma prototips; 6 - Grafiks; 7 - Grafika prototips; 8 - Karte; 9 - Pakalpojums; 10 - SLA; 11 - Lietotājs; 12 - Darbība; 13 - Medija tips. Īpašības darbība: - *obligāts*
name	string	Logrīka lauka nosaukums. Iespējamās vērtības: skatiet Informācijas paneļa logrīku laukus. Īpašības darbība: - *obligāts*
value	mixed	Logrīka lauka vērtība atkarībā no tipa. Iespējamās vērtības: skatiet Informācijas paneļa logrīku laukus. Īpašības darbība: - *obligāts*

Paneļa lietotāju grupa

Paneļa atļauju saraksts, pamatojoties uz lietotāju grupām. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
usrgrpid	ID	Lietotāju grupas ID. Īpašības darbība: - *required*
permission	integer	Atļauju līmeņa tips. Iespējamās vērtības: 2 - tikai lasīšana; 3 - lasīšana-rakstīšana. Īpašības darbība: - *required*

Paneļa lietotājs

Paneļa atļauju saraksts, pamatojoties uz lietotājiem. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
userid	ID	Lietotāja ID. Īpašības darbība: - *obligāta*
permission	integer	Atļaujas līmeņa tips. Iespējamās vērtības: 2 - tikai lasīšana; 3 - lasīšana-rakstīšana. Īpašības darbība: - *obligāta*

dashboard.create

Apraksts

`object dashboard.create(object/array dashboards)`

Šī metode ļauj izveidot jaunus paneļus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Paneļi, kas jāizveido.

Papildus standarta paneļa īpašībām, metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
pages	array	Paneļa lapas, kas jāizveido panelim. Paneļa lapas tiks sakārtotas tādā pašā secībā, kā norādīts. Parametra darbība: - *obligāts*
users	array	Paneļa lietotāja koplietošanas ieraksti, kas jāizveido panelī.
userGroups	array	Paneļa lietotāju grupas koplietošanas ieraksti, kas jāizveido panelī.

Atgrieztās vērtības

(`object`) Atgriež objektu, kas satur izveidoto paneļu ID īpašībā `dashboardids`. Atgriezto ID secība atbilst nodoto paneļu secībai.

Piemēri

Informācijas paneļa izveide

Izveidojiet informācijas paneli ar nosaukumu "My dashboard" ar vienu Problēmu logrīku ar tagiem un izmantojot divus koplietošanas veidus (lietotāju grupa un lietotājs) vienā informācijas paneļa lapā.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Paneļa lapa
- Paneļa logrīks
- Paneļa logrīka lauks
- Paneļa lietotājs
- Paneļa lietotāju grupa

Avots

CDashboard::create() failā *ui/include/classes/api/services/CDashboard.php*.

dashboard.delete

Apraksts

`object dashboard.delete(array dashboardids)`

Šī metode ļauj dzēst paneļus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo paneļu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto paneļu ID īpašumā `dashboardids`.

Piemēri

Vairāku paneļu dzēšana

Dzēsiet divus paneļus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2",
            "3"
        ]
```

```
    },
    "id": 1
}
```

Avots

CDashboard::delete() failā *ui/include/classes/api/services/CDashboard.php*.

dashboard.get

Apraksts

`integer/array dashboard.get(object parameters)`

Metode ļauj izgūt paneļus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dashboardids	ID/array	Atgriezt tikai paneļus ar norādītajiem ID.
selectPages	query	Atgriezt `pages` īpašību ar paneļa lapām pareizā secībā.
selectUsers	query	Atgriezt `users` īpašību ar lietotājiem, ar kuriem panelis ir koplietots.
selectUserGroups	query	Atgriezt `userGroups` īpašību ar lietotāju grupām, ar kurām panelis ir koplietots.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `dashboardid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Informācijas izgūšana par paneli pēc ID

Izgūstiet visus datus par paneļiem "1" un "2".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.get",
```

```
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "dashboardids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Paneļa lapa
- Paneļa logrīks
- Paneļa logrīka lauks
- Paneļa lietotājs
- Paneļa lietotāju grupa

Avots

CDashboard::get() failā *ui/include/classes/api/services/CDashboard.php*.

dashboard.update

Apraksts

`object dashboard.update(object/array dashboards)`

Šī metode ļauj atjaunināt esošos paneļus.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās paneļa īpašības.

Īpašībai `dashboardid` jābūt definētai katram panelim, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta paneļa īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
pages	array	Paneļa lapas, ar kurām aizstāt esošās paneļa lapas. Paneļa lapas tiek atjauninātas pēc īpašības `dashboard_pageid`. Jauni paneļa lapu objekti tiks izveidoti objektiem bez īpašības `dashboard_pageid`, un esošās paneļa lapas tiks dzēstas, ja tās netiks izmantotas atkārtoti. Paneļa lapas tiks sakārtotas tādā pašā secībā, kā norādīts. Tiks atjauninātas tikai norādītās paneļa lapu īpašības.
users	array	Paneļa lietotāja koplietojumi, ar kuriem aizstāt esošos elementus.
userGroups	array	Paneļa lietotāju grupas koplietojumi, ar kuriem aizstāt esošos elementus.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto paneļu ID zem `dashboardids` īpašības.

Piemēri

Paneļa pārdēvēšana

Pārdēvējiet paneli par "SQL server status".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "name": "SQL server status"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Informācijas paneļa lapu atjaunināšana

Pārdēvējiet pirmo informācijas paneļa lapu, aizstājiet logrīkus otrajā informācijas paneļa lapā un pievienojiet jaunu lapu kā trešo. Dzēsiet visas pārējās informācijas paneļa lapas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "dashboard.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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Mainīt paneļa īpašnieku

Pieejams tikai administratoriem un super administratoriem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
```

```
    "method": "dashboard.update",
    "params": {
        "dashboardid": "2",
        "userid": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Paneļa lapa
- Paneļa logrīks
- Paneļa logrīka lauks
- Paneļa lietotājs
- Paneļa lietotāju grupa

Avots

CDashboard::update() failā *ui/include/classes/api/services/CDashboard.php*.

Paneļa logrīka lauki

Šajā lapā ir navigācijas saites uz paneļa logrīku parametriem un iespējamajām attiecīgo paneļa logrīka lauka objektu īpašību vērtībām.

Lai skatītu katra logrīka parametrus un īpašību vērtības, lūdzu, izmantojiet sānjoslu, lai pārietu uz atbilstošo logrīka lapu.

1 Darbību žurnāls

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Darbību žurnāla* logrīku `dashboard.create` un `dashboard.update` metodēs.

> **Attention:**
> Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Darbību žurnāla* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Darbību žurnāls*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Saņēmēji	11	userids.0	Lietotāja ID. Piezīme: Lai konfigurētu vairākus lietotājus, izveidojiet informācijas paneļa logrīka lauka objektu katram lietotājam ar palielinātu numuru īpašības nosaukumā.
Darbības	12	actionids.0	Darbības ID. Piezīme: Lai konfigurētu vairākas darbības, izveidojiet informācijas paneļa logrīka lauka objektu katrai darbībai ar palielinātu numuru īpašības nosaukumā.
Mediju tipi	13	mediatypeids.0	Mediju tipa ID. Piezīme: Lai konfigurētu vairākus mediju tipus, izveidojiet informācijas paneļa logrīka lauka objektu katram mediju tipam ar palielinātu numuru īpašības nosaukumā.
Statuss	0	statuses.0	0 - Procesā; 1 - Nosūtīts/Izpildīts; 2 - Neizdevās. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet informācijas paneļa logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Meklēšanas virkne	1	message	Jebkura virknes vērtība.
Laika periods	1	time_period._referenc	`DASHBOARD._timeperiod` - iestatīt informācijas paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	time_period.from	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`
Kārtot ierakstus pēc	0	sort_triggers	3 - Laiks (augošā secībā); 4 - *(noklusējums)* Laiks (dilstošā secībā); 5 - Tips (augošā secībā); 6 - Tips (dilstošā secībā); 7 - Statuss (augošā secībā); 8 - Statuss (dilstošā secībā); 11 - Saņēmējs (augošā secībā); 12 - Saņēmējs (dilstošā secībā).

Parametrs	type	name	value
Rādīt rindas	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 25.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Darbību žurnāls*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Darbību žurnāla logrīka konfigurēšana

Konfigurējiet *Darbību žurnāla* logrīku, kas attēlo 10 darbību operāciju detalizētus ierakstus, sakārtotus pēc laika (augošā secībā). Papildus attēlojiet detalizētu informāciju tikai par tām darbību operācijām, kurās tika mēģināts nosūtīt e-pastu lietotājam "1", bet tas neizdevās.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

2 Pulkstenis

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Pulksteņa* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *Pulksteņa* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Pulksteņa* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.

Parametrs	type	name	value
Laika tips	0	time_type	0 - *(noklusējums)* Vietējais laiks; 1 - Servera laiks; 2 - Hosta laiks.
Pulksteņa tips	0	clock_type	0 - *(noklusējums)* Analogais; 1 - Digitālais.

Tālāk norādītie parametri tiek atbalstīti, ja *Laika tips* ir iestatīts uz ”Hosta laiks”.

Parametrs	type	name	value
Vienums	4	itemid.0	Vienuma ID. Parametra darbība: - *obligāts*

Tālāk norādītie parametri tiek atbalstīti, ja *Pulksteņa tips* ir iestatīts uz ”Digitālais”.

Parametrs	type	name	value
Rādīt	0	show.0	1 - Datums; 2 - *(noklusējums)* Laiks; 3 - Laika josla. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet informācijas paneļa logrīka lauka objektu katrai vērtībai ar palielinātu numuru rekvizīta nosaukumā.

Papildu konfigurācija

Tālāk norādītie papildu konfigurācijas parametri tiek atbalstīti, ja *Pulksteņa tips* ir iestatīts uz ”Digitāls”.

Parametrs	type	name	value
Fona krāsa	1	bg_color	Heksadecimālais krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Datums

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Clock type* ir iestatīts uz ”Digital” un *Show* ir iestatīts uz ”Date”.

Parametrs	type	name	value
Treknraksts	0	date_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	date_color	Heksadecimālais krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Laiks

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Pulksteņa tips* ir iestatīts uz ”Digitāls” un *Rādīt* ir iestatīts uz ”Laiks”.

Parametrs	type	name	value
Treknraksts	0	time_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	time_color	Heksadecimālais krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Parametrs	type	name	value
Sekundes	0	time_sec	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Formāts	0	time_format	0 - *(noklusējums)* 24 stundu; 1 - 12 stundu.

Laika josla

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Pulksteņa tips* ir iestatīts uz ”Digital” un *Rādīt* ir iestatīts uz ”Laika josla”.

Parametrs	type	name	value
Treknraksts	0	tzone_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	tzone_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).
Laika josla	1	tzone_timezone	Derīga laika joslas virkne (piemēram, Europe/Riga, system, UTC utt.). Pilnu atbalstīto laika joslu sarakstu skatiet PHP dokumentācijā. Noklusējums: local. Parametra darbība: - *atbalstīts*, ja *Laika tips* ir iestatīts uz ”Local time” vai ”Server time”
Formāts	0	tzone_format	0 - *(noklusējums)* Īss; 1 - Pilns. Parametra darbība: - *atbalstīts*, ja *Laika tips* ir iestatīts uz ”Local time” vai ”Server time”

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Pulksteņa* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet dashboard.create.

Clock logrīka konfigurēšana

Konfigurējiet *Clock* logrīku, kas pielāgotā digitālajā pulkstenī attēlo vietējo datumu, laiku un laika joslu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

3 Atklāšanas statuss

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Atklāšanas statuss* metodēs `dashboard.create` un `dashboard.update`.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Atklāšanas statuss*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Atklāšanas statuss*. Papildinformāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Atklāšanas statusa logrīka konfigurēšana

Konfigurējiet *Atklāšanas statusa* logrīku, iestatot atsvaidzināšanas intervālu uz 15 minūtēm.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

4 Iecienītākie grafiki

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Favorite graphs* metodēs `dashboard.create` un `dashboard.update`.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Iecienītākie grafiki*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Iecienītākie grafiki*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Favorite graphs logrīka konfigurēšana

Konfigurējiet *Favorite graphs* logrīku, iestatot atsvaidzināšanas intervālu uz 10 minūtēm.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

5 Iecienītākās kartes

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Iecienītās kartes* `dashboard.create` un `dashboard.update` metodēs.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Iecienītās kartes*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Iecienītās kartes*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Iecienīto karšu logrīka konfigurēšana

Konfigurējiet *Iecienīto karšu* logrīku, iestatot atsvaidzināšanas intervālu uz 10 minūtēm.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

6 Mērinstruments

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Gauge* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Gauge* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Gauge* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Vienums	4	itemid.0	Item ID.
Vienums (logrīks)	1	itemid._reference	Parametra darbība: - *obligāts*, ja *Vienums (logrīks)* nav iestatīts Item ID vietā: `ABCDE._itemid` - iestatīt saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu vienumiem.
Min	1	min	Parametra darbība: - *obligāts*, ja *Vienums* nav iestatīts Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, "1d", "2w", "4K", "8G"). Noklusējums: "0".
Max	1	max	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, "1d", "2w", "4K", "8G"). Noklusējums: "100".
Vērtības loks	1	value_arc_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: "" (tukšs).
Loka fons	1	empty_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: "" (tukšs).
Fons	1	bg_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: "" (tukšs).
Rādīt	0	show.0	1 - Apraksts; 2 - Vērtība; 3 - Rādītājs; 4 - Skala; 5 - Vērtības loks. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā. Noklusējums: 1, 2, 4, 5. Vērtības "Rādītājs" un "Skala" netiek atbalstītas, ja ir spēkā abi nosacījumi: - nav iestatīts neviens dashboard logrīka lauka objekts parametram *Rādīt* ar vērtību "Vērtības loks"; - *Rādīt loku* paplašinātās konfigurācijas parametrs ir iestatīts uz "Atspējots". *Rādīt* opciju paplašinātās konfigurācijas parametri netiek atbalstīti, ja nav iestatīti dashboard logrīka lauka objekti ar attiecīgajām vērtībām.

Parametrs	type	name	value
Aizstāt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "`ABCDE`") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatīt dashboard hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Paplašinātā konfigurācija

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti *Gauge* logrīkam.

Note:
Skaitlis *Thresholds* īpašības nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu sarakstā, kas sakārtots augošā secībā. Tomēr, ja sliekšņi ir konfigurēti citā secībā, pēc logrīka konfigurācijas atjaunināšanas Zabbix lietotāja saskarnē vērtības tiks sakārtotas augošā secībā (piemēram, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametrs	type	name	value
Leņķis	0	angle	Iespējamās vērtības: 180 *(noklusējums)* vai 270.
Apraksts			
Apraksts	1	description	Jebkura virknes vērtība, ieskaitot makrosus. Atbalstītie makrosi: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makrosi. Noklusējums: {ITEM.NAME}.
Izmērs	0	desc_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 15.
Vertikālā pozīcija	0	desc_v_pos	0 - Augšā; 1 - *(noklusējums)* Apakšā.
Treknraksts	0	desc_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	desc_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `""` (tukšs).
Vērtība			
Decimālvietas	0	decimal_places	Iespējamās vērtības diapazonā no 1 līdz 10. Noklusējums: 2.
Izmērs	0	value_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 25.
Treknraksts	0	value_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	value_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `""` (tukšs).
Mērvienības			
Mērvienības (izvēles rūtiņa)	0	units_show	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Mērvienības (vērtība)	1	units	Jebkura virknes vērtība. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz "Iespējots"

Parametrs	type	name	value
Izmērs	0	units_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 25. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz ”Iespējots”
Treknraksts	0	units_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz ”Iespējots”
Pozīcija	0	units_pos	0 - Pirms vērtības; 1 - Virs vērtības; 2 - *(noklusējums)* Pēc vērtības; 3 - Zem vērtības. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz ”Iespējots” Šis parametrs tiek ignorēts, ja tas ir iestatīts uz kādu no šīm ar laiku saistītajām mērvienībām: unixtime, uptime, s.
Krāsa	1	units_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).
Vērtības loks			
Loka izmērs	0	value_arc_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 20.
Rādītājs			
Krāsa	1	needle_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs). Parametra darbība: - *atbalstīts*, ja ir iestatīts paneļa logrīka lauka objekts *Show* ar vērtību ”Value arc” vai *Show arc* ir iestatīts uz ”Iespējots”
Skala			
Rādīt mērvienības	0	scale_show_units	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz ”Iespējots” un ir iestatīts paneļa logrīka lauka objekts *Show* ar vērtību ”Value arc” vai *Show arc* ir iestatīts uz ”Iespējots”
Izmērs	0	scale_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 15. Parametra darbība: - *atbalstīts*, ja ir iestatīts paneļa logrīka lauka objekts *Show* ar vērtību ”Value arc” vai *Show arc* ir iestatīts uz ”Iespējots”
Decimālvietas	0	scale_decimal_places	Iespējamās vērtības diapazonā no 1 līdz 10. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja ir iestatīts paneļa logrīka lauka objekts *Show* ar vērtību ”Value arc” vai *Show arc* ir iestatīts uz ”Iespējots”
Sliekšņi			
Krāsa	1	thresholds.0.color	Heksadecimāls krāsas kods (piemēram, FF0000).
Slieksnis	1	thresholds.0.threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”).

Parametrs	type	name	value
Rādīt etiķetes	0	th_show_labels	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Sliekšņi* ir iestatīti un ir iestatīts paneļa logrīka lauka objekts *Show* ar vērtību ”Value arc” vai *Show arc* ir iestatīts uz ”Iespējots”
Rādīt loku	0	th_show_arc	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Sliekšņi* ir iestatīti
Loka izmērs	0	th_arc_size	Iespējamās vērtības diapazonā no 1 līdz 100. Noklusējums: 5. Parametra darbība: - *atbalstīts*, ja *Show arc* ir iestatīts uz ”Iespējots”

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Mērinstrumenta* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Gauge logrīka konfigurēšana

Konfigurējiet *Gauge* logrīku, kas attēlo vienuma vērtību vienumam ”44474” (Saskarne enp0s3: Nosūtītie biti). Papildus vizuāli pielāgojiet logrīku, izmantojot vairākas papildu opcijas, tostarp sliekšņus.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

7 Geomap

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Geomap* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Geomap* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Geomap* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā *Reference* parametru, kas iestatīts uz "`ABCDE`") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā *Reference* parametru, kas iestatīts uz "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Sākotnējais skats	1	default_view	Ar komatiem atdalīti *platums*, *garums*, *tālummaiņas līmenis* (*neobligāts*, iespējamās vērtības diapazonā no 0 līdz 30). Piemērs: `40.6892494,-74.0466891,10`.
Klasterizācija			
Klasterizācijas režīms	0	clustering_mode	0 - *(noklusējums)* Automātiski; 1 - Tālummaiņas līmenis.
Klasterizācijas tālummaiņas līmenis	0	clustering_zoom_level	Iespējamās vērtības diapazonā no 0 līdz 30. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Klasterizācijas režīms* ir iestatīts uz "Tālummaiņas līmenis"
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Geomap* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Geomap logrīka konfigurēšana

Konfigurējiet *Geomap* logrīku, kas attēlo hostus no hostu grupām "2" un "22", pamatojoties uz šādu birku konfigurāciju: birka ar nosaukumu "component" satur vērtību "node", un birka ar nosaukumu "location" ir vienāda ar vērtību "New York". Papildus iestatiet kartes sākotnējo skatu uz koordinātām "40.6892494" (platums), "-74.0466891" (garums) ar tālummaiņas līmeni "10".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

8 Graph

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu rekvizītu vērtības ļauj konfigurēt *Graph* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti netiek validēti paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *Graph* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Graph* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai informācijas panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Datu kopa

Tālāk norādītie parametri tiek atbalstīti *Datu kopas* konfigurēšanai.

Note:
Pirmais skaitlis rekvizīta nosaukumā (piemēram, ds.0.hosts.0, ds.0.items.0) apzīmē konkrēto datu kopu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto hostu vai vienumu.

Parametrs	type	name	value
Datu kopas tips	0	ds.0.dataset_type	0 - Vienumu saraksts; 1 - *(noklusējums)* Vienumu šabloni.
Vienumi	4	ds.0.itemids.0	Vienuma ID. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē. Piezīme: Lai konfigurētu vairākus vienumus, katram vienumam izveidojiet dashboard widget field objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi (logrīks)* nav iestatīts
Vienumi (logrīks)	1	ds.0.itemids.0._reference	Vienuma ID vietā: `ABCDE._itemid` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "`ABCDE`") kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, katram logrīkam izveidojiet dashboard widget field objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi* nav iestatīts
Krāsa	1	ds.0.color.0	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts"
Hostu šabloni	1	ds.0.hosts.0	Hosta nosaukums vai šablons (piemēram, "Zabbix*"). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Vienumu šabloni	1	ds.0.items.0	Vienuma nosaukums vai šablons (piemēram, "*: Apstrādāto *vērtību skaits sekundē"). Konfigurējot logrīku veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni"
Krāsa	1	ds.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" un *Krāsu palete* nav iestatīta

Parametrs	type	name	value
Krāsu palete	0	ds.0.color_palette	Krāsu paletes indekss. Iespējamās vērtības ir diapazonā no 0 līdz 11. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni” un *Krāsa* nav iestatīta
Zīmēšana	0	ds.0.type	0 - *(noklusējums)* Līnija; 1 - Punkti; 2 - Kāpņveida; 3 - Stabiņi.
Sakrauts	0	ds.0.stacked	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz ”Līnija”, ”Kāpņveida” vai ”Stabiņi”
Platums	0	ds.0.width	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz ”Līnija” vai ”Kāpņveida”
Punkta izmērs	0	ds.0.pointsize	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz ”Punkti”
Caurspīdīgums	0	ds.0.transparency	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 5.
Aizpildījums	0	ds.0.fill	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz ”Līnija” vai ”Kāpņveida”
Trūkstošie dati	0	ds.0.missingdatafunc	0 - *(noklusējums)* Nav; 1 - Savienoti; 2 - Uzskatīt par 0; 3 - Pēdējā zināmā vērtība. Parametra darbība: - *atbalstīts*, ja *Zīmēšana* ir iestatīta uz ”Līnija” vai ”Kāpņveida”
Pārrakstīt hostu	1	ds.0.override_hostid	`ABCDE` - iestatiet `_hostid` - iestatiet saderīgu logrīku (ar tā parametru ”Reference”, kas iestatīts uz ”`ABCDE`”) kā hostu datu avotu; `DASHBOARD._hostid` - iestatiet paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Y ass	0	ds.0.axisy	0 - *(noklusējums)* Kreisā; 1 - Labā.
Invertēt vērtības	0	ds.0.invert_values	0 - (noklusējums) Atspējots; 1 - Iespējots.
Laika nobīde	1	ds.0.timeshift	Derīga laika virkne (piemēram, `3600`, `1h` utt.). Varat izmantot laika sufiksus. Ir atļautas arī negatīvas vērtības. Noklusējums: `""` (tukšs).

Parametrs	type	name	value
Agregācijas funkcija	0	ds.0.aggregate_function	0 - *(noklusējums)* netiek izmantota; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Agregācijas intervāls	1	ds.0.aggregate_interval	Derīga laika virkne (piemēram, 3600, 1h utt.). Varat izmantot laika sufiksus. Noklusējums: 1h.
Agregēt	0	ds.0.aggregate_grouping	0 - *(noklusējums)* Katrs vienums; 1 - Datu kopa. Parametra darbība: - *atbalstīts*, ja *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first” vai ”last”
Aproksimācija	0	ds.0.approximation	1 - min; 2 - *(noklusējums)* avg; 4 - max; 7 - all.
Datu kopas etiķete	1	ds.0.data_set_label	Jebkura virknes vērtība. Noklusējums: "" (tukšs).
Vienumu birkas			
Novērtēšanas tips	0	ds.0.item_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	ds.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Operators	0	ds.0.item_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Birkas vērtība	1	ds.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*

Attēlošanas opcijas

Tālāk norādītie parametri tiek atbalstīti, lai konfigurētu *Attēlošanas opcijas*.

Parametrs	type	name	value
Vēsturisko datu atlase	0	source	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences.
Vienkāršie trigeri	0	simple_triggers	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Darba laiks	0	working_time	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Hostu nosaukumi etiķetēs	0	show_hostnames	0 - *(noklusējums)* Auto; 1 - Rādīt; 2 - Slēpt. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Percentiles līnija (kreisā)			
Statuss	0	percentile_left	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Left"
Vērtība	0	percentile_left_value	Iespējamās vērtības ir diapazonā no 1 līdz 100. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Left"
Percentiles līnija (labā)			
Statuss	0	percentile_right	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Right"
Vērtība	0	percentile_right_value	Iespējamās vērtības ir diapazonā no 1 līdz 100. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz "Right"

Laika periods

Tālāk norādītie parametri tiek atbalstīti *Laika perioda* konfigurēšanai.

Parametrs	type	name	value
Laika periods	1	time_period._reference	`DASHBOARD._timeperiod` - iestatīt paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod`
No	1	time_period.from	Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`

Parametrs	type	name	value
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`

Asis

Tālāk norādītie parametri tiek atbalstīti *Asis* konfigurēšanai.

Parametrs	type	name	value
Kreisā Y	0	lefty	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz ”Left”
Labā Y	0	righty	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Y ass* (konfigurācijā *Datu kopa*) ir iestatīta uz ”Right”
Mērogs	0	lefty_scale righty_scale	0 - *(noklusējums)* Lineārs; 1 - Logaritmisks.
Minimums	1	lefty_min righty_min	Jebkura skaitliska vērtība. Noklusējums: "" (tukšs).
Maksimums	1	lefty_max righty_max	Jebkura skaitliska vērtība. Noklusējums: "" (tukšs).
Vienības (tips)	0	lefty_units righty_units	0 - *(noklusējums)* Automātiski; 1 - Statiski.
Vienības (vērtība)	1	lefty_static_units righty_static_units	Jebkura virknes vērtība. Noklusējums: "" (tukšs).
X ass	0	xaxis	0 - Atspējots; 1 - *(noklusējums)* Iespējots.

Leģenda

Tālāk norādītie parametri tiek atbalstīti *Leģendas* konfigurēšanai.

Parametrs	type	name	value
Rādīt leģendu	0	legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Rādīt min/avg/max	0	legend_statistic	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz ”Iespējots”

Parametrs	type	name	value
Rādīt agregācijas funkciju	0	legend_aggregation	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz ”Iespējots”
Rindas	0	legend_lines_mode	0 - *(noklusējums)* Fiksēts; 1 - Mainīgs. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz ”Iespējots”
Rindu skaits/ Maksimālais rindu skaits	0	legend_lines	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz ”Iespējots”
Kolonnu skaits	0	legend_columns	Iespējamās vērtības ir diapazonā no 1 līdz 4. Noklusējums: 4. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz ”Iespējots” un *Rādīt min/avg/max* ir iestatīts uz ”Atspējots”

Problēmas

Tālāk norādītie parametri tiek atbalstīti *Problēmu* konfigurēšanai.

Parametrs	type	name	value
Rādīt problēmas	0	show_problems	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Tikai atlasītie vienumi	0	graph_item_problems	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Problēmu hosti	1	problemhosts.0	Hosta nosaukums. Piezīme: Skaitlis īpašības nosaukumā norāda uz konfigurēto hostu. Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu skaitli īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Svarīgums	0	severities.0	0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu skaitli īpašības nosaukumā.
Problēma	1	problem_name	Problēmas notikuma nosaukums (reģistrjutība netiek ņemta vērā, pilns nosaukums vai tā daļa).

Parametrs	type	name	value
Problēmu birkas			
Novērtēšanas veids	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*

Aizstāšanas

Tālāk norādītie parametri tiek atbalstīti *Aizstāšanu* konfigurēšanai.

Note:
Pirmais skaitlis īpašības nosaukumā (piemēram, or.0.hosts.0, or.0.items.0) apzīmē konkrēto datu kopu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto hostu vai vienumu.

Parametrs	type	name	value
Hostu šabloni	1	or.0.hosts.0	Hosta nosaukums vai šablons (piemēram, `Zabbix*`). Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Aizstāšanas*
Vienumu šabloni	1	or.0.items.0	Vienuma nosaukums vai šablons (piemēram, `*: Number of processed *values per second`). Ja logrīks tiek konfigurēts veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Aizstāšanas*
Pamatkrāsa	1	or.0.color	Heksadecimāls krāsas kods (piemēram, FF0000).
Platums	0	or.0.width	Iespējamās vērtības ir diapazonā no 1 līdz 10.
Zīmēšana	0	or.0.type	0 - Līnija; 1 - Punkti; 2 - Kāpņveida; 3 - Josla.
Caurspīdīgums	0	or.0.transparency	Iespējamās vērtības ir diapazonā no 1 līdz 10.

Parametrs	type	name	value
Aizpildījums	0	or.0.fill	Iespējamās vērtības ir diapazonā no 1 līdz 10.
Punkta izmērs	0	or.0.pointsize	Iespējamās vērtības ir diapazonā no 1 līdz 10.
Trūkstošie dati	0	or.0.missingdatafunc	0 - Nav; 1 - Savienots; 2 - Uzskatīt par 0; 3 - Pēdējā zināmā vērtība.
Y ass	0	or.0.axisy	0 - Kreisā; 1 - Labā.
Invertēt vērtības	0	or.0.invert_values	0 - (noklusējums) Atspējots; 1 - Iespējots.
Laika nobīde	1	or.0.timeshift	Derīga laika virkne (piemēram, 3600, 1h utt.). Varat izmantot laika sufiksus. Ir atļautas negatīvas vērtības.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Graph* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Graph logrīka konfigurēšana

Konfigurējiet *Graph* logrīku šādi:

- 2 datu kopas ar kopumā 9 vienumiem 1 hostā.
- Pirmā datu kopa ir tipa "Item list" un sastāv no 3 vienumiem, kas ir attēloti kā līnijas ar atšķirīgu krāsu, bet vienādu platumu, caurspīdīgumu un aizpildījumu.
- Otrā datu kopa ir tipa "Item patterns", sastāv no 6 vienumiem, tai ir konfigurēta agregācija, un tā ir attēlota kā līnija ar pielāgotu krāsu, platumu, caurspīdīgumu un aizpildījumu.
- Otrajai datu kopai ir arī pielāgota datu kopas etiķete.
- Dati grafikā tiek attēloti par pēdējo 3 stundu periodu.
- Problēmas grafikā tiek attēlotas tikai konfigurētajiem vienumiem.
- Grafikam ir divas Y asis, no kurām labā Y ass attēlo vērtības tikai otrajai datu kopai.
- Grafika leģenda attēlo konfigurētos vienumus 4 rindās, kā arī datu kopu minimālās, maksimālās un vidējās vērtības.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

9 Graph (classic)

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Graph (classic)* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Graph (classic)* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Graph (classic)*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Avots	0	source_type	0 - *(noklusējums)* Grafiks; 1 - Vienkāršs grafiks.
Grafiks	6	graphid.0	Graph ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz "Graph"
Graph (Widget)	1	graphid._reference	Graph ID vietā: `ABCDE._graphid` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu grafikiem. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz "Simple graph" un *Grafiks* nav iestatīts
Vienums	4	itemid.0	Item ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz "Simple graph" un *Item (Widget)* nav iestatīts
Item (Widget)	1	itemid._reference	Item ID vietā: `ABCDE._itemid` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu vienumiem. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz "Simple graph" un *Vienums* nav iestatīts
Laika periods	1	time_period._reference	`DASHBOARD._timeperiod` - iestatiet paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod`
No	1	time_period.from	Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Derīga laika virkne absolūtā formātā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā formātā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`
Rādīt leģendu	0	show_legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Pārrakstīt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, ABCDE vai JBPNL). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Graph (classic)*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Graph (classic) logrīka konfigurēšana

Konfigurējiet *Graph (classic)* logrīku, kas attēlo vienkāršu grafiku vienumam "42269".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

10 Graph prototype

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Graph prototype* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Graph prototype* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Graph prototype*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Avots	0	source_type	2 - *(noklusējums)* Graph prototype; 3 - Simple graph prototype.
Grafika prototips	7	graphid.0	Graph prototype ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz ”Graph prototype”
Vienuma prototips	5	itemid.0	Item prototype ID. Parametra darbība: - *obligāts*, ja *Avots* ir iestatīts uz ”Simple graph prototype”

Parametrs	type	name	value
Laika periods	1	time_period._referenceDASHBOARD._timeperiod	DASHBOARD._timeperiod - iestata paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestata saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu. Noklusējums: DASHBOARD._timeperiod Alternatīvi laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	time_period.from	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts time_period.to
Līdz	1	time_period.to	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts time_period.from
Rādīt leģendu	0	show_legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Pārrakstīt hostu	1	override_hostid._referenceABCDE._hostid	ABCDE._hostid - iestata saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostiem; DASHBOARD._hostid - iestata paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Kolonnas	0	columns	Iespējamās vērtības ir diapazonā no 1 līdz 24. Noklusējums: 2.
Rindas	0	rows	Iespējamās vērtības ir diapazonā no 1 līdz 16. Noklusējums: 1.
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, ABCDE vai JBPNL). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Graph prototype*. Plašāku informāciju par paneļa konfigurēšanu skatiet dashboard.create.

Graph prototype logrīka konfigurēšana

Konfigurējiet *Graph prototype* logrīku, kas attēlo 3 grafiku režģi (3 kolonnas, 1 rinda), kas izveidoti no vienuma prototipa (ID: "42316"), izmantojot zema līmeņa atklāšanu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

11 Honeycomb

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Honeycomb* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *Honeycomb* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Honeycomb* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Vairākiem hostiem parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hostu birkas			
Novērtēšanas tips	0	evaltype_host	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Vienumu šabloni	1	items.0	Vienuma nosaukums vai šablons. Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet dashboard logrīka lauka objektu katram vienumu šablonam ar palielinātu numuru īpašības nosaukumā. Parametra darbība: - *obligāts*
Vienumu birkas			
Novērtēšanas tips	0	evaltype_item	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*

Parametrs	type	name	value
Operators	0	item_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Birkas vērtība	1	item_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Rādīt hos-tus uz-turēšanā	0	maintenance	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt	0	show.0	1 - Primārā etiķete; 2 - Sekundārā etiķete. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā. Noklusējums: 1, 2.
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Papildu konfigurācija

Tālāk norādītie papildu konfigurācijas parametri tiek atbalstīti *Honeycomb* logrīkam.

Note:
Skaitlis *Thresholds* rekvizīta nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu sarakstā, kas sakārtots augošā secībā. Tomēr, ja sliekšņi ir konfigurēti citā secībā, pēc logrīka konfigurācijas atjaunināšanas Zabbix lietotāja saskarnē vērtības tiks sakārtotas augošā secībā (piemēram, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametrs	type	name	value
Primārā etiķete			
Tips	0	primary_label_type	0 - *(noklusējums)* Teksts; 1 - Vērtība.

Parametrs	type	name	value
Teksts	1	primary_label	Jebkura virknes vērtība, ieskaitot makrosus. Atbalstītie makrosi: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makrosi. Noklusējums: {HOST.NAME} Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Teksts"
Decimāldaļu skaits	0	primary_label_decimals	Iespējamās vērtības ir diapazonā no 0 līdz 6. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība"
Izmērs (tips)	0	primary_label_size_type	0 - *(noklusējums)* Automātiski; 1 - Pielāgots.
Izmērs	0	primary_label_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 20. Parametra darbība: - *atbalstīts*, ja *Izmērs* (tips) ir iestatīts uz "Pielāgots"
Treknraksts	0	primary_label_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	primary_label_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: atkarīgs no Settings object un User object `theme`: `1F2C33` priekš "blue-theme", "blue-classic-theme" vai "hc-light"; `EEEEEE` priekš "dark-theme", "dark-classic-theme" vai "hc-dark".
Mērvienības (izvēles rūtiņa)	0	primary_label_units_show	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība"
Mērvienības (vērtība)	1	primary_label_units	Jebkura virknes vērtība. "" (tukšs) Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība" un *Mērvienības* (izvēles rūtiņa) ir iestatītas uz "Iespējots"
Pozīcija	0	primary_label_units_pos	0 - Pirms vērtības; 1 - *(noklusējums)* Pēc vērtības. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Vērtība" un *Mērvienības* (izvēles rūtiņa) ir iestatītas uz "Iespējots" Šis parametrs tiek ignorēts, ja ir iestatīta kāda no šīm ar laiku saistītajām mērvienībām: unixtime, uptime, s.
Sekundārā etiķete			
Tips	0	secondary_label_type	0 - Teksts; 1 - *(noklusējums)* Vērtība.
Teksts	1	secondary_label	Jebkura virknes vērtība, ieskaitot makrosus. Atbalstītie makrosi: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makrosi. Noklusējums: {{ITEM.LASTVALUE}.fmtnum(2)} Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz "Teksts"

Parametrs	type	name	value
Decimāldaļu skaits	0	secondary_label_decimal_places	Iespējamās vērtības ir diapazonā no 0 līdz 6. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz ”Vērtība”
Izmērs (tips)	0	secondary_label_size_type	0 - *(noklusējums)* Automātiski; 1 - Pielāgots.
Izmērs	0	secondary_label_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 30. Parametra darbība: - *atbalstīts*, ja *Izmērs* (tips) ir iestatīts uz ”Pielāgots”
Treknraksts	0	secondary_label_bold	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Krāsa	1	secondary_label_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: atkarīgs no Settings object un User object `theme`: `1F2C33` priekš ”blue-theme”, ”blue-classic-theme” vai ”hc-light”; `EEEEEE` priekš ”dark-theme”, ”dark-classic-theme” vai ”hc-dark”.
Mērvienības (izvēles rūtiņa)	0	secondary_label_units_show	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz ”Vērtība”
Mērvienības (vērtība)	1	secondary_label_units	Jebkura virknes vērtība. "" (tukšs) Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz ”Vērtība” un *Mērvienības* (izvēles rūtiņa) ir iestatītas uz ”Iespējots”
Pozīcija	0	secondary_label_position	0 - Pirms vērtības; 1 - *(noklusējums)* Pēc vērtības. Parametra darbība: - *atbalstīts*, ja *Tips* ir iestatīts uz ”Vērtība” un *Mērvienības* (izvēles rūtiņa) ir iestatītas uz ”Iespējots” Šis parametrs tiek ignorēts, ja ir iestatīta kāda no šīm ar laiku saistītajām mērvienībām: unixtime, uptime, s.
Fona krāsa			
Fona krāsa	1	bg_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: atkarīgs no Settings object un User object `theme`: `D9E7ED` priekš ”blue-theme” un ”blue-classic-theme”; `3D5059` priekš ”dark-theme” un ”dark-classic-theme”; `AAD7E9` priekš ”hc-light”; `335463` priekš ”hc-dark”.
Sliekšņi			
Krāsu interpolācija	0	interpolation	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Krāsa	1	thresholds.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`).
Slieksnis	1	thresholds.0.threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”).

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Honeycomb* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Honeycomb logrīka konfigurēšana

Konfigurējiet *Honeycomb* logrīku, kas attēlo Zabbix servera procesu noslodzi. Papildus mainiet honeycomb šūnu primāro etiķeti un vizuāli precizējiet logrīku ar sliekšņiem.

Pieprasījums:

```
{
    "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
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

12 Hosta pieejamība

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Host availability* logrīku `dashboard.create` un `dashboard.update` metodēs.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Host availability*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Saskarnes tips	0	interface_type.0	0 - Nav; 1 - Zabbix aģents (pasīvās pārbaudes); 2 - SNMP; 3 - IPMI; 4 - JMX; 5 - Zabbix aģents (aktīvās pārbaudes). Noklusējums: 1, 2, 3, 4, 5 (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls.
Rādīt hostus uzturēšanā	0	maintenance	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt tikai kopsummas	0	only_totals	0 - *(noklusējums)* Atspējots; 1 - Iespējots.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Host availability*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Host availability logrīka konfigurēšana

Konfigurējiet *Host availability* logrīku, kas attēlo pieejamības informāciju (vertikālā izkārtojumā) hostiem hostu grupā "4", kuriem ir konfigurētas "Zabbix aģents" un "SNMP" saskarnes.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

13 Hosta karte

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Host card* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Host card* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Host card*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Nav atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hosts	3	hostid.0	Host ID. Parametra darbība: - *obligāts*, ja *Hosts (Widget/Dashboard)* nav iestatīts Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosts (Widget/Dashboard)	1	hostid._reference	Host ID vietā: `DASHBOARD.hostid` - iestata paneļa hostu atlasītāju kā datu avotu hostam; `ABCDE._hostid` - iestata saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostam. Parametra darbība: - *obligāts*, ja *Hosts* nav iestatīts Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Rādīt apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt	0	sections.0	0 - Hostu grupas; 1 - Apraksts; 2 - Uzraudzība; 3 - Pieejamība; 4 - Uzrauga; 5 - Veidnes; 6 - Inventārs; 7 - Birkas. Piezīme: skaitlis rekvizīta nosaukumā norāda sadaļas secību sadaļu sarakstā. Lai konfigurētu vairākas sadaļas, izveidojiet paneļa logrīka lauka objektu katrai sadaļai, rekvizīta nosaukumā palielinot skaitli.

Tālāk norādītie parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Inventārs”.

Parametrs	type	name	value
Inventāra lauki	0	inventory.0	Inventory ID. Piezīme: lai konfigurētu vairākus inventāra laukus, izveidojiet paneļa logrīka lauka objektu katram inventāra laukam, rekvizīta nosaukumā palielinot skaitli.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Host card* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Host card logrīka konfigurēšana

Konfigurējiet *Host card* logrīku, kas attēlo šādas sadaļas: ”Uzraudzība”, ”Pieejamība”, ”Uzrauga”, ”Inventārs” un ”Birkas”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

14 Hostu navigators

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu rekvizītu vērtības ļauj konfigurēt *Host navigator* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti netiek validēti paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Host navigator* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Host navigator*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hostu šabloni	1	hosts.0	Hosta nosaukums vai šablons. Piezīme: Lai konfigurētu vairākus hostu šablonus, izveidojiet dashboard logrīka lauka objektu katram hostu šablonam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hostu šabloni, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder hosti, kas atbilst konfigurētajiem hostu šabloniem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosta sta-tuss	0	status	-1 - *(noklusējums)* Jebkurš; 0 - Iespējots; 1 - Atspējots. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu birkas			
Novērtēšanas tips	0	host_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Svarīguma pakāpe	0	severities.0	0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Rādīt hostus uzturēšanā	0	maintenance	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt problēmas	0	show_problems	0 - Visas; 1 - *(noklusējums)* Nenomāktās; 2 - Nav.
Grupēt pēc			
Atribūts	0	group_by.0.attribute	0 - Hostu grupa; 1 - Birkas vērtība; 2 - Svarīguma pakāpe. Piezīme: Numurs īpašības nosaukumā norāda atribūta secību grupēšanas atribūtu sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc*
Vērtība	1	group_by.0.tag_name	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda grupēšanas atribūtu, kas iestatīts parametrā *Atribūts*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc* un *Atribūts* ir iestatīts uz ”Birkas vērtība”
Hostu limits	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 9999. Noklusējums: 100. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Host navigator* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Host navigator logrīka konfigurēšana

Konfigurējiet *Host navigator* logrīku, kas attēlo hostus, grupētus vispirms pēc to hostu grupas un pēc tam pēc birkas ”city” vērtības.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

15 Vienuma kartīte

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Vienuma kartītes* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Vienuma kartītes* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Vienuma kartīte*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Neatsvaidzināt; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Vienums	4	itemid.0	Vienuma ID. Parametra darbība: - *obligāts*, ja *Vienums (logrīks)* nav iestatīts
Vienums (logrīks)	1	itemid._reference	Vienuma ID vietā: `ABCDE._itemid` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu vienumam. Parametra darbība: - *obligāts*, ja *Vienums* nav iestatīts

Parametrs	type	name	value
Rādīt	0	sections.0	0 - Apraksts; 1 - Kļūdas teksts; 2 - Metrikas; 3 - Jaunākie dati; 4 - Informācijas tips; 5 - Trigeri; 6 - Hosta saskarne; 7 - Tips; 8 - Hosta inventārs; 9 - Birkas. Piezīme: skaitlis rekvizīta nosaukumā norāda sadaļas secību sadaļu sarakstā. Lai konfigurētu vairākas sadaļas, izveidojiet dashboard widget field objektu katrai sadaļai ar palielinātu skaitli rekvizīta nosaukumā.
Pārrakstīt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatīt dashboard hosta atlasītāju kā datu avotu hostiem. Noklusējums: `""` (tukšs) Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Dzirksteļdiagramma

Tālāk norādītie parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz "Jaunākie dati".

Parametrs	type	name	value
Platums	0	sparkline.width	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 1.
Aizpildījums	0	sparkline.fill	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 3.
Krāsa	1	sparkline.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `42A5F5`.
Laika periods	1	sparkline.time_period._reference	`DASHBOARD._timeperiod` - iestata paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestata saderīgu logrīku (ar tā parametru `reference`, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `""` (tukšs) Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	sparkline.time_period.from	Laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Noklusējums: now-1h. Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `sparkline.time_period.to`

Parametrs	type	name	value
Līdz	1	sparkline.time_period.to	Laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Noklusējums: now. Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Vēstures dati	0	sparkline.history	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Vienuma kartītes* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Vienuma kartītes logrīka konfigurēšana

Konfigurējiet *Vienuma kartītes* logrīku, kas attēlo šīs sadaļas: "Apraksts", "Jaunākie dati", "Trigeri" un "Birkas".

Pieprasījums:

```
{
    "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
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```


Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

16 Vienumu vēsture

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt logrīku *Vienuma vēsture* metodēs `dashboard.create` un `dashboard.update`.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Vienuma vēsture* izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Vienumu vēsture*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls.
Kolonnas (skatīt zemāk)			
Rādīt rindas	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 25.
Pārrakstīt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatīt paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Paplašinātā konfigurācija (skatīt zemāk)			
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *required*

Kolonnas

Kolonnām ir kopīgi parametri un papildu parametri atkarībā no parametra *Vienums* konfigurācijas.

Note:
Visiem ar kolonnām saistītajiem parametriem skaitlis īpašības nosaukumā (piemēram, columns.0.name) norāda kolonnu, kurai parametrs ir konfigurēts.

Visām kolonnām tiek atbalstīti šādi parametri.

Parametrs	type	name	value
Nosaukums	1	columns.0.name	Jebkura virknes vērtība. Parametra darbība: - *obligāts*
Vienums	4	columns.0.itemid	Vienuma ID. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē. Parametra darbība: - *obligāts*
Pamatkrāsa	1	columns.0.base_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `""` (tukšs).

Tālāk norādītie kolonnas parametri tiek atbalstīti, ja konfigurētais *vienums* ir skaitliska tipa vienums.

Parametrs	type	name	value
Attēlojums	0	columns.0.display	1 - *(noklusējums)* Kā ir; 2 - Josla; 3 - Indikatori.
Min	1	columns.0.min	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Max	1	columns.0.max	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Sliekšņi			
Krāsa	1	columns.0.thresholds.color	Heksadecimāls krāsas kods (piemēram, FF0000).
Slieksnis	1	columns.0.thresholds.threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”).
Vēstures dati	0	columns.0.history	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences.

Tālāk norādītie kolonnas parametri tiek atbalstīti, ja konfigurētais *vienums* ir rakstzīmju, teksta vai žurnāla tipa vienums.

Parametrs	type	name	value
Izcēlumi			
Izcēlums	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, FF0000).
Slieksnis	1	columns.0.highlights.pattern	Jebkura regulārā izteiksme.
Attēlošana	0	columns.0.display	1 - *(noklusējums)* Kā ir; 4 - HTML; 5 - Vienā rindā.
Viena rinda	0	columns.0.max_length	Iespējamās vērtības ir diapazonā no 1 līdz 500. Noklusējums: 100. Parametra darbība: - *atbalstīts*, ja *Attēlošana* ir iestatīta uz ”Vienā rindā”
Izmantot vienplatuma fontu	0	columns.0.monospace_font	0 - *(noklusējums)* Izmantot noklusējuma fontu; 1 - Izmantot vienplatuma fontu.
Rādīt vietējo laiku	0	columns.0.local_time	0 - *(noklusējums)* Rādīt laikspiedolu; 1 - Rādīt vietējo laiku. Parametra darbība: - *atbalstīts*, ja *vienums* ir iestatīts kā žurnāla tipa vienums un *Rādīt laikspiedolu* ir iestatīts uz ”Iespējots”

Tālāk norādītie kolonnas parametri tiek atbalstīti, ja konfigurētais *vienums* ir binārā tipa vienums.

Parametrs	type	nosaukums	vērtība
Rādīt sīktēlu	1	columns.0.show_thumbnail	0 - *(noklusējums)* Atspējots; 1 - Iespējots.

Paplašinātā konfigurācija

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti logrīkam *Vienuma vēsture*.

Parametrs	type	name	value
Jaunās vērtības	0	sortorder	0 - *(noklusējums)* Augšā; 1 - Apakšā.
Rādīt laik-spiedolu	0	show_timestamp	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Rādīt kolon-nas gal-veni	0	show_column_header	0 - Izslēgts; 1 - Horizontāli; 2 - *(noklusējums)* Vertikāli.
Laika pe-ri-ods	1	time_period._reference	`DASHBOARD._timeperiod` - iestatīt paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod`
No	1	time_period.from	Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Vienuma vēsture*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Vienuma vēstures logrīka konfigurēšana

Konfigurējiet *Vienuma vēstures* logrīku, kas attēlo jaunākos datus diviem skaitliskiem vienumiem "42269" un "42270". Papildus konfigurējiet vienumu kolonnas attēlošanai vertikāli, kolonnu nosaukumus attēlojot horizontāli; ierobežojiet attēlojumu līdz 15 datu rindām un iekļaujiet atsevišķu laikspiedola kolonnu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

17 Vienumu navigators

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Vienumu navigatora* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Vienumu navigatora* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Vienumu navigators*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostid` - iestatiet paneļa hosta atlasītāju kā hostu datu avotu; `ABCDE._hostid` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu birkas			
Novērtēšanas veids	0	host_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Vienumu šabloni	1	items.0	Vienuma nosaukums vai šablons. Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet dashboard logrīka lauka objektu katram vienumu šablonam ar palielinātu numuru īpašības nosaukumā.
Vienumu birkas			
Novērtēšanas veids	0	item_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Operators	0	item_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Birkas vērtība	1	item_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Stāvoklis	0	state	-1 - *(noklusējums)* Visi; 0 - Normāls; 1 - Netiek atbalstīts.
Rādīt prob-lē-mas	0	show_problems	0 - Visas; 1 - *(noklusējums)* Nenoklusētās; 2 - Nerādīt.
Grupēt pēc			
Atribūts	0	group_by.0.attribute	0 - Hostu grupa; 1 - Hosta nosaukums; 2 - Hosta birkas vērtība; 3 - Vienuma birkas vērtība. Piezīme: Numurs īpašības nosaukumā norāda atribūta secību grupēšanas atribūtu sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc*
Vērtība	1	group_by.0.tag_name	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda grupēšanas atribūtu, kas iestatīts parametrā *Atribūts*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Grupēt pēc* un *Atribūts* ir iestatīts uz ”Hosta birkas vērtība” vai ”Vienuma birkas vērtība”

Parametrs	type	name	value
Vienumu lim-its	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 9999. Noklusējums: 100.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Vienumu navigatora* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Vienumu navigatora logrīka konfigurēšana

Konfigurējiet *Vienumu navigatora* logrīku, kas attēlo līdz 1000 vienumiem, grupētiem pēc to hosta un pēc tam pēc vienuma birkas "component" vērtības.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

18 Vienuma vērtība

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauka objektiem ļauj konfigurēt *Vienuma vērtība* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Vienuma vērtība* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Vienuma vērtība*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Vienums	4	itemid.0	Vienuma ID. Parametra darbība: - *obligāts*, ja *Vienums (logrīks)* nav iestatīts
Vienums (logrīks)	1	itemid._reference	Vienuma ID vietā: `ABCDE._itemid` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu vienumiem. Parametra darbība: - *obligāts*, ja *Vienums* nav iestatīts
Rādīt	0	show.0	1 - Apraksts; 2 - Vērtība; 3 - Laiks; 4 - Izmaiņu indikators; 5 - Dzirksteles diagramma. Noklusējums: 1, 2, 3, 4. Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet informācijas paneļa logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Pārrakstīt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatīt informācijas paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes informācijas panelī.

Paplašinātā konfigurācija

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti logrīkam *Vienuma vērtība*.

Note:
Skaitlis īpašības *Thresholds* nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu sarakstā, kas sakārtots augošā secībā. Tomēr, ja sliekšņi ir konfigurēti citā secībā, pēc logrīka konfigurācijas atjaunināšanas Zabbix lietotāja saskarnē vērtības tiks sakārtotas augošā secībā (piemēram, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametrs	type	name	value
Fona krāsa	1	bg_color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `""` (tukšs).
Sliekšņi			
Krāsa	1	thresholds.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`).
Slieksnis	1	thresholds.0.threshold	Jebkura virknes vērtība.

Parametrs	type	name	value
Agregācijas funkcija	0	aggregate_function	0 - *(noklusējums)* netiek izmantota; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Laika periods	1	time_period._reference	`DASHBOARD._timeperiod` - iestatīt paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā parametru `reference`, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”
No	1	time_period.from	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts un *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts un *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `time_period.from`
Vēstures dati	0	history	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences.

Apraksts

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Apraksts”.

Parametrs	type	name	value
Apraksts	1	description	Jebkura virknes vērtība, ieskaitot makrosus. Atbalstītie makrosi: {HOST.*}, {ITEM.*}, {INVENTORY.*}, lietotāja makrosi. Noklusējums: {ITEM.NAME}.
Horizontālā pozīcija	0	desc_h_pos	0 - Pa kreisi; 1 - *(noklusējums)* Centrā; 2 - Pa labi.
Vertikālā pozīcija	0	desc_v_pos	Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*. 0 - Augšā; 1 - Vidū; 2 - *(noklusējums)* Apakšā. Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*.

Parametrs	type	name	value
Izmērs	0	desc_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 15.
Treknraksts	0	desc_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	desc_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Vērtība

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Vērtība”.

Parametrs	type	name	value
Decimālvietas			
Decimālvietas	0	decimal_places	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 2.
Izmērs	0	decimal_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 35.
Pozīcija			
Horizontālā pozīcija	0	value_h_pos	0 - Pa kreisi; 1 - *(noklusējums)* Centrā; 2 - Pa labi. Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*.
Vertikālā pozīcija	0	value_v_pos	0 - Augšā; 1 - *(noklusējums)* Vidū; 2 - Apakšā. Diviem vai vairākiem elementiem (Apraksts, Vērtība, Laiks) nevar būt vienāda *Horizontālā pozīcija* un *Vertikālā pozīcija*.
Izmērs	0	value_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 45.
Treknraksts	0	value_bold	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Krāsa	1	value_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).
Mērvienības			
Mērvienības (izvēles rūtiņa)	0	units_show	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Mērvienības (vērtība)	1	units	Jebkura virknes vērtība.
Pozīcija	0	units_pos	0 - Pirms vērtības; 1 - Virs vērtības; 2 - *(noklusējums)* Pēc vērtības; 3 - Zem vērtības.
Izmērs	0	units_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 35.
Treknraksts	0	units_bold	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Krāsa	1	units_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Laiks

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Laiks”.

Parametrs	type	name	value
Horizontālā pozīcija	0	time_h_pos	0 - Pa kreisi; 1 - *(noklusējums)* Centrā; 2 - Pa labi. Divi vai vairāki elementi (Apraksts, Vērtība, Laiks) nevar koplietot vienu un to pašu *Horizontālo pozīciju* un *Vertikālo pozīciju*.
Vertikālā pozīcija	0	time_v_pos	0 - *(noklusējums)* Augšā; 1 - Vidū; 2 - Apakšā. Divi vai vairāki elementi (Apraksts, Vērtība, Laiks) nevar koplietot vienu un to pašu *Horizontālo pozīciju* un *Vertikālo pozīciju*.
Izmērs	0	time_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 15.
Treknraksts	0	time_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Krāsa	1	time_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Izmaiņu indikators

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Izmaiņu indikators”.

Parametrs	type	name	value
Izmaiņu indikatora ↑ krāsa	1	up_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).
Izmaiņu indikatora ↓ krāsa	1	down_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).
Izmaiņu indikatora ↕ krāsa	1	updown_color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: "" (tukšs).

Dzirksteļdiagramma

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti, ja *Rādīt* ir iestatīts uz ”Dzirksteļdiagramma”.

Parametrs	type	name	value
Platums	0	sparkline.width	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 1.
Aizpildījums	0	sparkline.fill	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 3.
Krāsa	1	sparkline.color	Heksadecimāls krāsas kods (piemēram, FF0000). Noklusējums: 42A5F5.

Parametrs	type	name	value
Laika periods	1	sparkline.time_period.from	DASHBOARD._timeperiod — iestatīt paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod — iestatīt saderīgu logrīku (ar tā `reference` parametru, kas vienāds ar ABCDE) kā datu avotu. Noklusējums: DASHBOARD._timeperiod Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	sparkline.time_period.from	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w utt.). Parametra darbība: - *obligāts*, ja ir iestatīts sparkline.time_period.to
Līdz	1	sparkline.time_period.to	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w utt.). Parametra darbība: - *obligāts*, ja ir iestatīts sparkline.time_period.from
Vēstures dati	0	sparkline.history	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Vienuma vērtība*. Plašāku informāciju par paneļa konfigurēšanu skatiet dashboard.create.

Vienuma vērtības logrīka konfigurēšana

Konfigurējiet *Vienuma vērtības* logrīku, kas attēlo vienuma vērtību vienumam "42266" (Zabbix aģenta pieejamība). Papildus tam vizuāli pielāgojiet logrīku, izmantojot vairākas papildu opcijas, tostarp dinamisku fona krāsu, kas mainās atkarībā no Zabbix aģenta pieejamības statusa.

Pieprasījums:

```
{
    "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
}
```


Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

19 Karte

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Map* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Map* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Kartes* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.
Karte	8	sysmapid.0	Kartes ID. Parametra darbība: - *obligāts*, ja *Karte (logrīks)* nav iestatīts
Karte (logrīks)	1	sysmapid._reference	`ABCDE._mapid` - iestatīt *Kartes navigācijas koka* logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā karšu datu avotu. Parametra darbība: - *obligāts*, ja *Karte* nav iestatīta
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai informācijas panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir aprakstīt tikai paneļa logrīka lauku objektu konfigurāciju *Kartes* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Map logrīka konfigurēšana

Konfigurējiet *Map* logrīku, kas attēlo karti "1".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Saistīta *Map* logrīka konfigurēšana

Konfigurējiet *Map* logrīku, kas ir saistīts ar *Map navigation tree* logrīku.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`
- Kartes navigācijas koks

20 Kartes navigācijas koks

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Kartes navigācijas koks* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Kartes navigācijas koks* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Kartes navigācijas koks*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.
Rādīt nepieejamās kartes	1	show_unavailable	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai informācijas panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Tālāk norādītie parametri tiek atbalstīti kartes navigācijas koka elementu konfigurēšanai.

Parametrs	type	name	value
Nosaukums	1	navtree.1.name	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā nosaka elementa numuru.
Saistītā karte	8	navtree.1.sysmapid	Kartes ID. Piezīme: Skaitlis īpašības nosaukumā norāda elementu, ar kuru karte ir saistīta.
Parametri elementu hierarhijas izveidei	0	navtree.1.parent	Vecākelementa numurs. Piezīme: Skaitlis īpašības nosaukumā norāda bērnelementu. Īpašības vērtība norāda vecākelementu.
	0	navtree.1.order	Elementa pozīcija kartes navigācijas kokā. Piezīme: Skaitlis īpašības nosaukumā norāda elementa numuru. Īpašības vērtība norāda elementa pozīciju kartes navigācijas kokā. Vecākelementa pozīcija tiek noteikta visā kartes navigācijas kokā. Bērnelementa pozīcija tiek noteikta vecākelementa ietvaros.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Kartes navigācijas koks*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Kartes navigācijas koka logrīka konfigurēšana

Konfigurējiet *Kartes navigācijas koka* logrīku, kas attēlo šādu kartes navigācijas koku:

- Elements A
  - Elements A1
  - Elements A2
- Elements B
  - Elements B1 (satur saistīto karti "1", ko var attēlot saistītā *Kartes logrīkā*)

        - Elements B2
    - Elements C

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`
- Karte

21 Sektoru diagramma

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Sektoru diagrammas* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Sektoru diagrammas* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Sektoru diagramma*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.

Datu kopa

Tālāk norādītie parametri tiek atbalstīti *Datu kopas* konfigurēšanai.

Note:
Pirmais skaitlis īpašības nosaukumā (piem., ds.0.hosts.0, ds.0.items.0) apzīmē konkrēto datu kopu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto hostu vai vienumu.

Parametrs	type	name	value
Datu kopas tips	0	ds.0.dataset_type	0 - Vienumu saraksts; 1 - *(noklusējums)* Vienumu šabloni.
Vienumi	4	ds.0.itemids.0	Item ID. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē. Piezīme: Lai konfigurētu vairākus vienumus, katram vienumam izveidojiet paneļa logrīka lauka objektu ar palielinātu numuru īpašības nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi (logrīks)* nav iestatīts
Vienumi (logrīks)	1	ds.0.itemids.0._reference	Item ID vietā: `ABCDE._itemid` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, katram logrīkam izveidojiet paneļa logrīka lauka objektu ar palielinātu numuru īpašības nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts" un *Vienumi* nav iestatīts
Krāsa	1	ds.0.color.0	Heksadecimāls krāsas kods (piem., FF0000). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts"
Vienuma tips	0	ds.0.type.0	0 - *(noklusējums)* Normāls; 1 - Kopā. Vērtību "Kopā" var iestatīt tikai vienam vienumam visā diagrammā. Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu saraksts"
Hostu šabloni	1	ds.0.hosts.0	Host nosaukums vai šablons (piem., "Zabbix*"). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Vienumu šabloni	1	ds.0.items.0	Item nosaukums vai šablons (piem., "*: Number of processed *values per second"). Konfigurējot logrīku veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni"
Krāsa	1	ds.0.color	Heksadecimāls krāsas kods (piem., FF0000). Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" un *Krāsu palete* nav iestatīta
Krāsu palete	0	ds.0.color_palette	Krāsu paletes indekss. Iespējamās vērtības ir diapazonā no 0 līdz 11. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz "Vienumu šabloni" un *Krāsa* nav iestatīta
Agregācijas funkcija	0	ds.0.aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - *(noklusējums)* last.
Datu kopas agregācija	0	ds.0.dataset_aggregation	0 - *(noklusējums)* nav; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum. Parametra darbība: - *atbalstīts*, ja *Vienuma tips* ir iestatīts uz "Kopā"
Datu kopas etiķete	1	ds.0.data_set_label	Jebkura virknes vērtība. Noklusējums: "" (tukšs).
Vienumu birkas			
Novērtēšanas tips	0	ds.0.item_tags_evaltype	0 - *(noklusējums)* And/Or; 2 - Or.
Birkas nosaukums	1	ds.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*

Parametrs	type	name	value
Operators	0	ds.0.item_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Birkas vērtība	1	ds.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*

Attēlošanas opcijas

Tālāk norādītie parametri tiek atbalstīti *Attēlošanas opciju* konfigurēšanai.

Parametrs	type	name	value
Vēsturisko datu atlase	0	source	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences.
Zīmējums	0	draw_type	0 - *(noklusējums)* Sektoru diagramma; 1 - Gredzenveida diagramma.
Platums	0	width	20 - 20% no rādiusa; 30 - 30% no rādiusa; 40 - 40% no rādiusa; 50 - *(noklusējums)* 50% no rādiusa. Parametra darbība: - *atbalstīts*, ja *Zīmējums* ir iestatīts uz ”Gredzenveida diagramma”
Kontūras platums	0	stroke	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Zīmējums* ir iestatīts uz ”Gredzenveida diagramma”
Rādīt kopējo vērtību	0	total_show	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Zīmējums* ir iestatīts uz ”Gredzenveida diagramma”
Izmērs	0	value_size_type	0 - *(noklusējums)* Auto; 1 - Pielāgots. Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”
Izmērs (pielāgotā izmēra vērtība)	0	value_size	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 20. Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”

Parametrs	type	name	value
Decimāldaļu skaits	0	decimal_places	Iespējamās vērtības ir diapazonā no 0 līdz 6. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”
Mērvienības (izvēles rūtiņa)	0	units_show	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”
Mērvienības (vērtība)	1	units	Jebkura virknes vērtība. Parametra darbība: - *atbalstīts*, ja *Mērvienības* (izvēles rūtiņa) ir iestatīta uz ”Iespējots”
Treknraksts	0	value_bold	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”
Krāsa	1	value_color	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *atbalstīts*, ja *Rādīt kopējo vērtību* ir iestatīts uz ”Iespējots”
Atstarpe starp sek-to-riem	0	space	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 1.
Apvienot sek-torus, kas ir mazāki par N% (izvēles rūtiņa)	0	merge	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Apvienot sek-torus, kas ir mazāki par N% (vērtība)	0	merge_percent	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Apvienot sektorus, kas ir mazāki par N%* (izvēles rūtiņa) ir iestatīta uz ”Iespējots”
Apvienot sek-torus, kas ir mazāki par N% (krāsa)	1	merge_color	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *atbalstīts*, ja *Apvienot sektorus, kas ir mazāki par N%* (izvēles rūtiņa) ir iestatīta uz ”Iespējots”

Laika periods

Tālāk norādītie parametri tiek atbalstīti *Laika perioda* konfigurēšanai.

Parametrs	type	name	value
Laika periods	1	time_period._reference	`DASHBOARD._timeperiod` - iestata paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestata saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	time_period.from	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`

Leģenda

Tālāk norādītie parametri tiek atbalstīti *Leģendas* konfigurēšanai.

Parametrs	type	name	value
Rādīt leģendu	0	legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Rādīt vērtību	0	legend_value	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Rādīt agregācijas funkciju	0	legend_aggregation	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Rindas	0	legend_lines_mode	0 - *(noklusējums)* Fiksēts; 1 - Mainīgs. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Rindu skaits/ Maksimālais rindu skaits	0	legend_lines	Iespējamās vērtības ir diapazonā no 1 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Kolonnu skaits	0	legend_columns	Iespējamās vērtības ir diapazonā no 1 līdz 4. Noklusējums: 4. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots" un *Rādīt vērtību* ir iestatīts uz "Atspējots"

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Pie chart* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Sektoru diagrammas logrīka konfigurēšana

Konfigurējiet *Sektoru diagrammas* logrīku šādi:

- 2 datu kopas ar kopumā 9 vienumiem 1 hostā.
- Pirmā datu kopa ir tipa ”Vienumu saraksts” un sastāv no 3 vienumiem, kuri visi ir ar tipu ”Parasts” un attēloti dažādās krāsās.
- Otrā datu kopa ir tipa ”Vienumu šabloni”, sastāv no 6 vienumiem, tai ir konfigurēta agregācija katram vienumam, un tā ir attēlota ar pielāgotu krāsu.
- Otrajai datu kopai ir arī pielāgota datu kopas etiķete.
- Dati sektoru diagrammā tiek attēloti kā gredzenveida diagramma ar pielāgotu platumu un kopējo vērtību ar mērvienībām centrā.
- Dati sektoru diagrammā tiek attēloti un agregēti pielāgotam pēdējo 3 stundu laika periodam.
- Sektoru diagrammas leģenda attēlo konfigurētos vienumus 4 rindās.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- dashboard.create
- dashboard.update

22 Problemātiskie hosti

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Problem hosts* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *Problem hosts* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Problem hosts*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "`ABCDE`") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Izslēgt hostu grupas	2	exclude_groupids.0	Hostu grupas ID. Piezīme: Lai izslēgtu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrnejutīgs, pilns nosaukums vai tā daļa).
Svarīgums	0	severities.0	0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Problēmu birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Rādīt apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.

Parametrs	type	name	value
Paslēpt grupas bez problēmām	0	hide_empty_groups	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēmu atšķirtspēja	0	ext_ack	0 - *(noklusējums)* Visas; 1 - Tikai neapstiprinātās; 2 - Atdalītas.
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Problēmu hosti*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Problem hosts logrīka konfigurēšana

Konfigurējiet *Problem hosts* logrīku, kas attēlo hostus no hostu grupām "2" un "4", kuriem ir problēmas ar nosaukumu, kas ietver virkni "CPU", un kuriem ir šādas nopietnības pakāpes: "Brīdinājums", "Vidēja", "Augsta", "Katastrofa".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

23 Problēmas

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt logrīku *Problems* metodēs

`dashboard.create` un `dashboard.update`.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu logrīka *Problems* izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Problems*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Rādīt	0	show	1 - *(noklusējums)* Nesenās problēmas; 2 - Vēsture; 3 - Problēmas.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Izslēgt hostu grupas	2	exclude_groupids.0	Hostu grupas ID. Piezīme: Lai izslēgtu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā hostu datu avotu; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrjutība netiek ņemta vērā, pilns nosaukums vai tā daļa).
Svarīgums	0	severities.0	0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Problēmu birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Rādīt birkas	0	show_tags	0 - *(noklusējums)* Nav; 1 - 1; 2 - 2; 3 - 3.
Birkas nosaukums (formāts)	0	tag_name_format	0 - *(noklusējums)* Pilns; 1 - Saīsināts; 2 - Nav. Parametra darbība: - *atbalstīts*, ja *Rādīt birkas* ir iestatīts uz "1", "2" vai "3"

Parametrs	type	name	value
Birku atlašošanas prioritāte	1	tag_priority	Ar komatiem atdalīts birku saraksts. Parametra darbība: - *atbalstīts*, ja *Rādīt birkas* ir iestatīts uz ”1”, ”2” vai ”3”
Rādīt operatīvos datus	0	show_opdata	0 - *(noklusējums)* Nav; 1 - Atsevišķi; 2 - Kopā ar problēmas nosaukumu.
Rādīt apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Apstiprinājuma statuss	0	acknowledgement_status	0 - *(noklusējums)* visi; 1 - Neapstiprināts; 2 - Apstiprināts.
Manis apstiprināts	0	acknowledged_by_me	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Kārtot ierakstus pēc	0	sort_triggers	1 - Svarīgums (dilstoši); 2 - Hosts (augoši); 3 - Laiks (augoši); 4 - *(noklusējums)* Laiks (dilstoši); 13 - Svarīgums (augoši); 14 - Hosts (dilstoši); 15 - Problēma (augoši); 16 - Problēma (dilstoši). Visām vērtībām, izņemot ”Laiks (dilstoši)” un ”Laiks (augoši)”, parametram *Rādīt laika skalu* jābūt iestatītam uz ”Atspējots”. Vērtības ”Hosts (augoši)” un ”Hosts (dilstoši)” netiek atbalstītas, ja logrīks tiek konfigurēts veidnes panelī.
Rādīt laika skalu	0	show_timeline	0 - Atspējots; 1 - *(noklusējums)* Iespējots. Parametra darbība: - *atbalstīts*, ja *Kārtot ierakstus pēc* ir iestatīts uz ”Laiks (dilstoši)” vai ”Laiks (augoši)”
Izcelt visu rindu	0	highlight_row	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Šis parametrs netiek atbalstīts augsta kontrasta tēmās.
Rādīt rindas	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 25.
Reference	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Problēmas*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Problēmas logrīka konfigurēšana

Konfigurējiet *Problēmas* logrīku, kas attēlo problēmas hostu grupai "4" un atbilst šādiem nosacījumiem:

- Problēmas, kurām ir tags ar nosaukumu "scope", kas satur vērtības "performance" vai "availability", vai "capacity".
- Problēmas ar šādiem svarīguma līmeņiem: "Warning", "Average", "High", "Disaster".

Papildus konfigurējiet logrīku tā, lai tas rādītu tagus un darbības datus.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

24 Problēmas pēc svarīguma pakāpes

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt logrīku *Problēmas pēc nopietnības* `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu logrīka *Problēmas pēc nopietnības* izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Problēmas pēc svarīguma*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Izslēgt hostu grupas	2	exclude_groupids.0	Hostu grupas ID. Piezīme: Lai izslēgtu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "`ABCDE`") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrnejutīgs, pilns nosaukums vai tā daļa).
Svarīgums	0	severities.0	0 - Neklasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Problēmu birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*

Parametrs	type	name	value
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Rādīt	0	show_type	0 - *(noklusējums)* Hostu grupas; 1 - Kopsummas. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī, un pēc noklusējuma ir iestatīts uz ”Kopsummas”.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls. Parametra darbība: - *atbalstīts*, ja *Rādīt* ir iestatīts uz ”Kopsummas”
Rādīt datus	0	show_opdata	0 - *(noklusējums)* Nekas; 1 - Atsevišķi; 2 - Kopā ar problēmas nosaukumu.
Rādīt apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Paslēpt grupas bez problēmām	0	hide_empty_groups	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt* ir iestatīts uz ”Hostu grupas” Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēmu attēlojums	0	ext_ack	0 - *(noklusējums)* Visas; 1 - Tikai neapstiprinātās; 2 - Atdalītas.
Rādīt laika skalu	0	show_timeline	0 - Atspējots; 1 - *(noklusējums)* Iespējots.

Parametrs	type	name	value
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Problēmas pēc smaguma pakāpes*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Problēmas pēc smaguma pakāpes logrīka konfigurēšana

Konfigurējiet *Problēmas pēc smaguma pakāpes* logrīku, kas parāda problēmu kopsummas visām hostu grupām.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

25 Izkliedes diagramma

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu rekvizītu vērtības ļauj konfigurēt *Scatter plot* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` rekvizīti paneļa izveides vai atjaunināšanas laikā netiek validēti. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Scatter plot* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Izkliedes diagrammas* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai informācijas panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Datu kopa

Tālāk norādītie parametri tiek atbalstīti *Datu kopas* konfigurēšanai.

Note:
Pirmais skaitlis rekvizīta nosaukumā (piemēram, ds.0.hosts.0, ds.0.items.0) apzīmē konkrēto datu kopu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto hostu vai vienumu.

Parametrs	type	name	value
Datu kopas tips	0	ds.0.dataset_type	0 - Vienumu saraksts; 1 - *(noklusējums)* Vienumu šabloni.

Parametrs	type	name	value
Datu kopa: Vienumu saraksts			
Krāsa	1	ds.0.color.0	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu saraksts”
X ass	4	ds.0.x_axis_itemids.0	Vienuma ID. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē. Piezīme: Lai konfigurētu vairākus vienumus, katram vienumam izveidojiet paneļa logrīka lauka objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu saraksts” un *X ass (logrīks)* nav iestatīta
X ass (logrīks)	1	ds.0.x_axis_itemids.0	Vienuma atsauce vietā: ABCDE._itemid - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz ”ABCDE”) kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, katram logrīkam izveidojiet paneļa logrīka lauka objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu saraksts” un *X ass* nav iestatīta
Y ass	4	ds.0.y_axis_itemids.0	Vienuma ID. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē. Piezīme: Lai konfigurētu vairākus vienumus, katram vienumam izveidojiet paneļa logrīka lauka objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu saraksts” un *Y ass (logrīks)* nav iestatīta
Y ass (logrīks)	1	ds.0.y_axis_itemids.0	Vienuma atsauce vietā: ABCDE._itemid - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz ”ABCDE”) kā vienumu datu avotu. Piezīme: Lai konfigurētu vairākus logrīkus, katram logrīkam izveidojiet paneļa logrīka lauka objektu ar palielinātu skaitli rekvizīta nosaukumā. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu saraksts” un *Y ass* nav iestatīta
Aizstāt hostu	1	ds.0.override_hostid.0	ABCDE._hostid - iestatiet saderīgu logrīku (ar tā parametru ”Atsauce”, kas iestatīts uz ”ABCDE”) kā hostu datu avotu; DASHBOARD._hostid - iestatiet paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Datu kopa: Vienumu šabloni			
Krāsa	1	ds.0.color	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni” un *Krāsu palete* nav iestatīta
Krāsu palete	0	ds.0.color_palette	Krāsu paletes indekss. Iespējamo vērtību diapazons: 0-11. Noklusējums: 0. Parametra darbība: - *atbalstīts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni” un *Krāsa* nav iestatīta
Hostu šabloni	1	ds.0.hosts.0	Hosta nosaukums vai šablons (piemēram, ”Zabbix*”). Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni” Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
X ass vienumu šabloni	1	ds.0.x_axis_items.0	Vienuma nosaukums vai šablons (piemēram, ”*: Apstrādāto *vērtību skaits sekundē”). Konfigurējot logrīku veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni”
Y ass vienumu šabloni	1	ds.0.y_axis_items.0	Vienuma nosaukums vai šablons (piemēram, ”*: Apstrādāto *vērtību skaits sekundē”). Konfigurējot logrīku veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja *Datu kopas tips* ir iestatīts uz ”Vienumu šabloni”
Hostu gru-pas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, katrai hostu grupai izveidojiet paneļa logrīka lauka objektu ar palielinātu skaitli rekvizīta nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu gru-pas (lo-grīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz ”ABCDE”) kā hostu grupu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Aizstāt hostu	1	ds.0.override_hostid.enable	`ABCDE._hostid`- iestatiet saderīgu logrīku (ar tā parametru ”Atsauce”, kas iestatīts uz ”ABCDE”) kā hostu datu avotu; `DASHBOARD._hostid` - iestatiet paneļa hostu atlasītāju kā hostu datu avotu. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu birkas			
Novērtēšanas tips	0	evaltype_host	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis rekvizīta nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Marķieris	0	ds.0.marker	0 - *(noklusējums)* Elipse; 1 - Kvadrāts; 2 - Trijstūris; 3 - Rombs; 4 - Zvaigzne; 5 - Krusts.
Marķiera izmērs	0	ds.0.marker_size	0 - *(noklusējums)* Mazs (6px); 1 - Vidējs (9px); 2 - Liels (15px).
Laika nobīde	1	ds.0.timeshift	Derīga laika virkne (piemēram, 3600, 1h utt.). Varat izmantot laika sufiksus. Ir atļautas arī negatīvas vērtības. Noklusējums: "" (tukšs).

Parametrs	type	name	value
Agregācijas intervāls	1	ds.0.aggregate_interval	Derīga laika virkne (piemēram, 3600, 1h utt.). Varat izmantot laika sufiksus. Noklusējums: 15m.
Agregācijas funkcija	0	ds.0.aggregate_function	0 - netiek izmantota; 1 - min; 2 - max; 3 - *(noklusējums)* avg; 4 - count; 5 - sum; 6 - first; 7 - last.

Attēlošanas opcijas

Tālāk norādītie parametri tiek atbalstīti, lai konfigurētu *Attēlošanas opcijas*.

Parametrs	type	name	value
Vēsturisko datu atlase	0	source	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences.
Hostu nosaukumi etiķetēs	0	show_hostnames	0 - *(noklusējums)* Automātiski; 1 - Rādīt; 2 - Slēpt. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Laika periods

Tālāk norādītie parametri tiek atbalstīti *Laika perioda* konfigurēšanai.

Parametrs	type	name	value
Laika periods	1	time_period._reference	DASHBOARD._timeperiod - iestatīt paneļa laika perioda atlasītāju kā datu avotu; ABCDE._timeperiod - iestatīt saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu. Noklusējums: DASHBOARD._timeperiod Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*.
No	1	time_period.from	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w u.c.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts time_period.to
Līdz	1	time_period.to	Derīga laika virkne absolūtā (YYYY-MM-DD hh:mm:ss) vai relatīvā laika sintaksē (now, now/d, now/w-1w u.c.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts time_period.from

Asis

Tālāk norādītie parametri tiek atbalstīti *asu* konfigurēšanai.

Parametrs	type	name	value
X ass	0	x_axis	0 - Atspējota; 1 - *(noklusējums)* Iespējota.
Y ass	0	y_axis	0 - *(noklusējums)* Atspējota; 1 - Iespējota.
Min	1	x_axis_min	Jebkura skaitliska vērtība. Noklusējums: "" (tukšs).
		y_axis_min	
Max	1	x_axis_max	Jebkura skaitliska vērtība. Noklusējums: "" (tukšs).
		y_axis_max	
Vienības (tips)	0	x_axis_units	0 - *(noklusējums)* Automātiski; 1 - Statiski.
		y_axis_units	
Vienības (vērtība)	1	x_axis_static_units	Jebkura virknes vērtība. Noklusējums: "" (tukšs).
		y_axis_static_units	

Leģenda

Tālāk norādītie parametri tiek atbalstīti *Leģendas* konfigurēšanai.

Parametrs	type	name	value
Rādīt leģendu	0	legend	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Rādīt apkopošanas funkciju	0	legend_aggregation	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Rindas	0	legend_lines_mode	0 - *(noklusējums)* Fiksēts; 1 - Mainīgs. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Rindu skaits/ Maksimālais rindu skaits	0	legend_lines	Iespējamais vērtību diapazons: 1-10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"
Kolonnu skaits	0	legend_columns	Iespējamais vērtību diapazons: 1-4. Noklusējums: 4. Parametra darbība: - *atbalstīts*, ja *Rādīt leģendu* ir iestatīts uz "Iespējots"

Sliekšņi

Tālāk norādītie parametri tiek atbalstīti *Sliekšņu* konfigurēšanai.

Note:
Skaitlis *Sliekšņu* īpašības nosaukumā (piemēram, thresholds.0.color) norāda sliekšņa vietu sarakstā, kas sakārtots augošā secībā. Tomēr, ja sliekšņi ir konfigurēti citā secībā, pēc logrīka konfigurācijas atjaunināšanas Zabbix lietotāja saskarnē vērtības tiks sakārtotas augošā secībā (piemēram, `"thresholds.0.threshold":"5"` → `"thresholds.0.threshold":"1"`; `"thresholds.1.threshold":"1"` → `"thresholds.1.threshold": "5"`).

Parametrs	type	name	value
Krāsu interpolācija	0	interpolation	0 - *(noklusējums)* Atspējota; 1 - Iespējota.
Krāsa	1	thresholds.0.color	Heksadecimāls krāsas kods (piemēram, FF0000).
X ass	1	thresholds.0.x_axis_threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”).
Y ass	1	thresholds.0.y_axis_threshold	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”).

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Izkliedes diagramma*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Scatter plot logrīka konfigurēšana

Konfigurējiet *Scatter plot* logrīku šādi:

- 5 datu kopas ar tipu ”Vienumu šabloni”, katra ar noklusējuma agregāciju (avg, 15m) un atšķirīgiem marķieriem.
- Leģenda attēlo konfigurētos vienumus vienā rindā un divās kolonnās.
- Sliekšņi ir konfigurēti tā, lai marķieris tiktu iekrāsots sarkans, kad tā vērtība sasniedz 80 uz jebkuras ass.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

26 SLA atskaite

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *SLA report* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *SLA report* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *SLA atskaites* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - *(noklusējums)* Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
SLA	10	slaid.0	SLA ID. Parametra darbība: - *obligāts*
Pakalpojums	9	serviceid.0	Pakalpojums ID.
Rādīt periodus	0	show_periods	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 20.
No	1	date_from	Derīga datuma virkne formātā `YYYY-MM-DD`. Tiek atbalstīti relatīvie datumi ar modifikatoriem `d`, `w`, `M`, `y` (piemēram, `now`, `now/d`, `now/w-1w` utt.).

Parametrs	type	name	value
Līdz	1	date_to	Derīga datuma virkne formātā `YYYY-MM-DD`. Tiek atbalstīti relatīvie datumi ar modifikatoriem `d`, `w`, `M`, `y` (piemēram, `now`, `now/d`, `now/w-1w` utt.).

Piemēri

Tālāk sniegto piemēru mērķis ir aprakstīt tikai paneļa logrīka lauku objektu konfigurāciju *SLA report* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

SLA atskaites logrīka konfigurēšana

Konfigurējiet *SLA atskaites* logrīku, kas attēlo SLA atskaiti SLA "4" pakalpojumam "2" par pēdējām 30 dienām.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

27 Sistēmas informācija

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Sistēmas informācija* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, bet vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Sistēmas informācija* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Sistēmas informācija*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - *(noklusējums)* 15 minūtes.
Rādīt	0	info_type	0 - *(noklusējums)* Sistēmas statistika; 1 - Augstas pieejamības mezgli.

Parametrs	type	name	value
Rādīt programmatūras atjauninājumu pārbaudes detalizēto informāciju	0	show_software_update_check_details	0 - (noklusējums) Atspējots; 1 - Iespējots. Parametra darbība: - *supported* ja `AllowSoftwareUpdateCheck` Zabbix servera konfigurācijā ir iespējots un *Rādīt* ir iestatīts uz ”Sistēmas statistika”

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Sistēmas informācija*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Sistēmas informācijas logrīka konfigurēšana

Konfigurējiet *Sistēmas informācijas* logrīku, kas attēlo sistēmas statistiku ar atsvaidzināšanas intervālu 10 minūtes un iespējotu programmatūras atjauninājumu pārbaudi.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

28 Populārākie hosti

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Top hosts* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Top hosts* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Top hosts*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "`ABCDE`") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "`ABCDE`") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Rādīt hostus uzturēšanā	0	maintenance	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Kolonnas (skatiet tālāk)			
Kārtot pēc	0	column	Kolonnas skaitliskā vērtība no konfigurētajām kolonnām.
Secība	0	order	2 - *(noklusējums)* Top N; 3 - Bottom N.
Hostu limits	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 1000. Noklusējums: 10. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Kolonnas

Kolonnām ir kopīgi parametri un papildu parametri atkarībā no parametra *Dati* konfigurācijas.

Note:
Visiem ar kolonnām saistītajiem parametriem skaitlis īpašības nosaukumā (piemēram, columns.0.name) norāda kolonnu, kurai parametrs ir konfigurēts.

Visām kolonnām tiek atbalstīti šādi parametri.

Parametrs	type	name	value
Nosaukums	1	columns.0.name	Jebkura virknes vērtība. Parametra darbība: - *obligāts*
Dati	0	columns.0.data	1 - vienuma vērtība; 2 - hosta nosaukums; 3 - Teksts. Parametra darbība: - *obligāts*
Pamatkrāsa	1	columns.0.base_color	Heksadecimāls krāsas kods (piemēram, FF0000). Parametra darbība: - *obligāts*

Vienuma vērtība

Tālāk norādītie parametri tiek atbalstīti, ja *Data* ir iestatīts uz "Item value".

Note:
Pirmais skaitlis *Thresholds* īpašības nosaukumā (piemēram, columnsthresholds.0.color.0) norāda kolonnu, kurai ir konfigurēti sliekšņi, savukārt otrais skaitlis norāda sliekšņa vietu sarakstā, kas sakārtots augošā secībā. Tomēr, ja sliekšņi ir konfigurēti citā secībā, pēc logrīka konfigurācijas atjaunināšanas Zabbix lietotāja saskarnē vērtības tiks sakārtotas augošā secībā (piemēram, `"threshold.0.threshold":"5"` → `"threshold.0.threshold":"1"`; `"threshold.1.threshold":"1"` → `"threshold.1.threshold": "5"`).

Parametrs	type	name	value
Vienums	1	columns.0.item	Derīgs vienuma nosaukums. Konfigurējot logrīku veidnes panelī, jānorāda tikai tie vienumi, kas konfigurēti veidnē.
Rādīt vienuma vērtību kā	0	columns.0.display_value_as	0 - *(noklusējums)* Skaitlisks; 1 - Teksts; 2 - Binārs.
Attēlojums	0	columns.0.display	1 - *(noklusējums)* Kā ir; 2 - Josla; 3 - Indikatori; 6 - Dzirksteles diagramma. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric”
Min	1	columns.0.min	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric” un *Attēlojums* ir iestatīts uz ”Bar” vai ”Indicators”
Max	1	columns.0.max	Jebkura skaitliska vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric” un *Attēlojums* ir iestatīts uz ”Bar” vai ”Indicators”
Decimālvietas	0	columns.0.decimal_places	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 2. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric”
Dzirksteles diagramma			
Platums	0	columns.0.sparkline.width	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric” un *Attēlojums* ir iestatīts uz ”Sparkline”
Aizpildījums	0	columns.0.sparkline.fill	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric” un *Attēlojums* ir iestatīts uz ”Sparkline”

Parametrs	type	name	value
Krāsa	1	columns.0.sparkline.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `42A5F5`. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric" un *Attēlojums* ir iestatīts uz "Sparkline"
Laika periods	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - iestata paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestata saderīgu logrīku (ar tā `reference` parametru, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *From* un *To*. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric" un *Attēlojums* ir iestatīts uz "Sparkline"
No	1	columns.0.sparkline.time_period.from	Datu laika diapazona virkne absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric" un *Attēlojums* ir iestatīts uz "Sparkline" - *obligāts*, ja ir iestatīts `columns.0.sparkline.time_period.to`
Līdz	1	columns.0.sparkline.time_period.to	Datu laika diapazona virkne absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric" un *Attēlojums* ir iestatīts uz "Sparkline" - *obligāts*, ja ir iestatīts `columns.0.sparkline.time_period.from`
Vēstures dati	0	columns.0.sparkline.history	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric" un *Attēlojums* ir iestatīts uz "Sparkline"
Sliekšņi			
Krāsa	1	columnsthresholds.0.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric"
Slieksnis	1	columnsthresholds.0.threshold	Jebkura vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Numeric"
Izcēlumi			
Krāsa	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Text"
Raksts	1	columns.0.highlights.pattern	Jebkura virknes vērtība. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz "Text"

Parametrs	type	name	value
Rādīt sīktēlu	0	columns.0.show_thumbnail	0 - *(noklusējums)* Atspējots; 1 - Iespējots. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Binary”
Agregācijas funkcija	0	columns.0.aggregate_function	0 - *(noklusējums)* netiek izmantota; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last.
Laika periods	1	columns.0.time_period._reference	`DASHBOARD._timeperiod` - iestata paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestata saderīgu logrīku (ar tā `reference` parametru, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *From* un *To*. Parametra darbība: - *atbalstīts*, ja *Aggregation function* ir iestatīts uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”
No	1	columns.0.time_period.from	Datuma laika virkne absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Time period* nav iestatīts un *Aggregation function* ir iestatīts uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `columns.0.time_period.to`
Līdz	1	columns.0.time_period.to	Datuma laika virkne absolūtajā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Time period* nav iestatīts un *Aggregation function* ir iestatīts uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `columns.0.time_period.from`
Vēstures dati	0	columns.0.history	0 - *(noklusējums)* Auto; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Rādīt vienuma vērtību kā* ir iestatīts uz ”Numeric”
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Teksts

Tālāk norādītie parametri tiek atbalstīti, ja *Data* ir iestatīts uz ”Text”.

Parametrs	type	name	value
Text	1	columns.0.text	Jebkura virknes vērtība, ieskaitot makrosus. Atbalstītie makrosi: {HOST.*}, {INVENTORY.*}. Parametra darbība: - *obligāts*, ja *Data* ir iestatīts uz "Text"

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Top hosts*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Top hosts logrīka konfigurēšana

Konfigurējiet *Top hosts* logrīku, kas attēlo augstākās noslodzes hostus pēc CPU izmantojuma hostu grupā "4". Papildus konfigurējiet šādas pielāgotās kolonnas: "Hosta nosaukums", "CPU izmantojums %", "1m vid.", "5m vid.", "15m vid.", "Procesi".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

29 Populārākie vienumi

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Top items* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku logrīkus izveidot vai atjaunināt nepareizi. Lai nodrošinātu veiksmīgu *Top items* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Top items*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu birkas			

Parametrs	type	name	value
Novērtēšanas tips	0	host_tags_evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas nosaukums	1	host_tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Operators	0	host_tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Birkas vērtība	1	host_tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Hostu birkas* Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls.
Rādīt problēmas	0	show_problems	0 - Visas; 1 - *(noklusējums)* Nenoklusinātās; 2 - Nav.
Vienumi (skatīt zemāk)			

Paplašinātā konfigurācija

Tālāk norādītie paplašinātās konfigurācijas parametri tiek atbalstīti logrīkam *Top items*.

Parametrs	type	name	value
Hostu kārtošana			
Kārtot pēc	0	host_ordering_order_by	0 - *(noklusējums)* Hosta nosaukums; 3 - Vienuma vērtība.

Parametrs	type	name	value
Vienumu šabloni	1	host_ordering_item.0	Vienuma nosaukums vai šablons (piemēram, ”*: Apstrādāto *vērtību skaits sekundē”). Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet dashboard widget field object katram vienuma šablonam, īpašības nosaukumā palielinot numuru. Konfigurējot logrīku veidnes panelī, jānorāda tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*, ja *Kārtot pēc* ir iestatīts uz ”Vienuma vērtība”
Secība	0	host_ordering_order	2 - *(noklusējums)* Augšējie N; 3 - Apakšējie N.
Limits	0	host_ordering_limit	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 10.
Vienumu kār-tošana			
Kārtot pēc	0	item_ordering_order_by	1 - Hosts; 2 - Vienuma nosaukums; 3 - *(noklusējums)* Vienuma vērtība.
Hostu šabloni	1	item_ordering_host.0	Hosta nosaukums vai šablons. Piezīme: Lai konfigurētu vairākus hostu šablonus, izveidojiet dashboard widget field object katram hosta šablonam, īpašības nosaukumā palielinot numuru. Parametra darbība: - *obligāts*, ja *Kārtot pēc* ir iestatīts uz ”Hosts”
Secība	0	item_ordering_order	2 - *(noklusējums)* Augšējie N; 3 - Apakšējie N.
Limits	0	item_ordering_limit	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 10.

Kolonnas

Note:
Pirmais skaitlis īpašības nosaukumā (piemēram, columns.0.items.0, columns.0.item_tags_evaltype) apzīmē konkrēto kolonnu, savukārt otrais skaitlis, ja tāds ir, apzīmē konfigurēto entītiju (piemēram, vienuma šablonu, birku).

Parametrs	type	name	value
Vienumu šabloni	1	columns.0.items.0	Vienuma nosaukums vai šablons (piemēram, ”*: Apstrādāto *vērtību skaits sekundē”). Piezīme: Lai konfigurētu vairākus vienumu šablonus, izveidojiet dashboard logrīka lauka objektu katram vienuma šablonam ar palielinātu otro skaitli īpašības nosaukumā. Konfigurējot logrīku veidnes dashboard, jāiestata tikai to vienumu šabloni, kas konfigurēti veidnē. Parametra darbība: - *obligāts*
Vienumu birkas			

Parametrs	type	name	value
Novērtēšanas tips	0	columns.0.item_tags.0.evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	columns.0.item_tags.0.tag	Jebkura virknes vērtība. Piezīme: Otrais skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Operators	0	columns.0.item_tags.0.operator	0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Otrais skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Birkas vērtība	1	columns.0.item_tags.0.value	Jebkura virknes vērtība. Piezīme: Otrais skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Vienumu birkas*
Pamatkrāsa	1	columns.0.base_color	Heksadecimāls krāsas kods (piemēram, FF0000).
Rādīt vērtību kā	0	columns.0.display_value_as	1 - *(noklusējums)* Skaitliski; 2 - Teksts.
Attēlojums	0	columns.0.display	1 - *(noklusējums)* Kā ir; 2 - Josla; 3 - Indikatori; 6 - Sparkline.
Min	1	columns.0.min	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski” un *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Max	1	columns.0.max	Jebkura skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski” un *Attēlojums* ir iestatīts uz ”Josla” vai ”Indikatori”
Sparkline			
Platums	0	columns.0.sparkline.width	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 1. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
Aizpildījums	0	columns.0.sparkline.fill	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 3. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”

Parametrs	type	name	value
Krāsa	1	columns.0.sparkline.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Noklusējums: `42A5F5`. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
Laika periods	1	columns.0.sparkline.time_period	`DASHBOARD._timeperiod` - iestatīt dashboard laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā `reference` parametru, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
No	1	columns.0.sparkline.time_period.from	Datu laika formāts absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline” - *obligāts*, ja ir iestatīts `columns.0.sparkline.time_period.to`
Līdz	1	columns.0.sparkline.time_period.to	Datu laika formāts absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline” - *obligāts*, ja ir iestatīts `columns.0.sparkline.time_period.from`
Vēstures dati	0	columns.0.sparkline.history	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Attēlojums* ir iestatīts uz ”Sparkline”
Sliekšņi			
Krāsa	1	columns.0.thresholds.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski”
Slieksnis	1	columns.0.thresholds.value	Skaitliska vērtība. Tiek atbalstīti sufiksi (piemēram, ”1d”, ”2w”, ”4K”, ”8G”). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski”
Izcēlumi			
Izcēlums	1	columns.0.highlights.color	Heksadecimāls krāsas kods (piemēram, `FF0000`). Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Teksts”
Slieksnis	1	columns.0.highlights.pattern	Jebkura regulārā izteiksme. Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Teksts”
Decimālzīmju skaits	0	columns.0.decimal_places	Iespējamās vērtības ir diapazonā no 0 līdz 10. Noklusējums: 2.

Parametrs	type	name	value
Paplašinātā konfigurācija			
Agregācijas funkcija	0	columns.0.aggregate_function	Ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski”: 0 - *(noklusējums)* netiek izmantota; 1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum; 6 - first; 7 - last. Ja *Rādīt vērtību kā* ir iestatīts uz ”Teksts”: 0 - *(noklusējums)* netiek izmantota; 4 - count; 6 - first; 7 - last.
Laika periods	1	columns.0.time_period.name	`DASHBOARD._timeperiod` - iestatīt dashboard laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā `reference` parametru, kas vienāds ar `ABCDE`) kā datu avotu. Noklusējums: `DASHBOARD._timeperiod` Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Parametra darbība: - *atbalstīts*, ja *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last”
No	1	columns.0.time_period.from	Definēja laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts un *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `columns.0.time_period.to`
Līdz	1	columns.0.time_period.to	Definēja laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts un *Agregācijas funkcija* ir iestatīta uz ”min”, ”max”, ”avg”, ”count”, ”sum”, ”first”, ”last” - *obligāts*, ja ir iestatīts `columns.0.time_period.from`
Vēstures dati	0	columns.0.history	0 - *(noklusējums)* Automātiski; 1 - Vēsture; 2 - Tendences. Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski”
Agregēt	0	columns.0.aggregate_column	0 - *(noklusējums)* Katrs vienums atsevišķi 1 - Apvienoti. Parametra darbība: - *atbalstīts*, ja *Rādīt vērtību kā* ir iestatīts uz ”Skaitliski” un *Attēlojums* ir iestatīts uz ”Kā ir”, ”Josla” vai ”Indikatori”.

Parametrs	type	name	value
Apvienotā agregā-ci-jas funkcija	0	columns.0.column_aggregate_function	1 - min; 2 - max; 3 - avg; 4 - count; 5 - sum *(noklusējums)*. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Agregēt*.
Apvienotās kolon-nas nosaukums	1	columns.0.combined_column_name	Jebkura virkne vērtība. Parametra darbība: - *obligāts*, ja tiek konfigurēts *Agregēt*.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *Top items* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Top items logrīka konfigurēšana

Konfigurējiet *Top items* logrīku, kas attēlo datus hostam "10084" un tikai tiem vienumiem, kuriem tags ar nosaukumu "component" satur vērtību "cpu". Papildus attēlojiet datus ar hostiem augšpusē un šūnu attēlošanai izmantojiet krāsainu mērinstrumenta joslu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

30 populārākie trigeri

Apraksts

Šie parametri un iespējamās īpašību vērtības attiecīgajiem paneļa logrīka lauku objektiem ļauj konfigurēt *Top trigeri* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Top trigeri* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Top trigeri*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - *(noklusējums)* Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Vairākiem hostiem parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēma	1	problem	Problēmas notikuma nosaukums (reģistrjutība netiek ņemta vērā, pilns nosaukums vai tā daļa).
Svarīgums	0	severities.0	Trigeru svarīguma līmeņi. 0 - Nav klasificēts; 1 - Informācija; 2 - Brīdinājums; 3 - Vidējs; 4 - Augsts; 5 - Katastrofa. Noklusējums: tukšs (visi iespējoti). Piezīme: Lai konfigurētu vairākas vērtības, izveidojiet dashboard logrīka lauka objektu katrai vērtībai ar palielinātu numuru īpašības nosaukumā.
Problēmu birkas			
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.

Parametrs	type	name	value
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā.
Operators	0	tags.0.operator	Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas* 0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā.
Birkas vērtība	1	tags.0.value	Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas* Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā.
Laika periods	1	time_period._reference	Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas* `DASHBOARD._timeperiod` - iestatīt paneļa laika perioda atlasītāju kā datu avotu; `ABCDE._timeperiod` - iestatīt saderīgu logrīku (ar tā parametru *Reference*, kas iestatīts uz "ABCDE") kā datu avotu. Noklusējums: `DASHBOARD._timeperiod`
No	1	time_period.from	Alternatīvi, laika periodu var iestatīt tikai parametros *No* un *Līdz*. Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.to`
Līdz	1	time_period.to	Derīga laika virkne absolūtā (`YYYY-MM-DD hh:mm:ss`) vai relatīvā laika sintaksē (`now`, `now/d`, `now/w-1w` utt.). Parametra darbība: - *atbalstīts*, ja *Laika periods* nav iestatīts - *obligāts*, ja ir iestatīts `time_period.from`
Trigeru limits	0	show_lines	Iespējamās vērtības ir diapazonā no 1 līdz 100. Noklusējums: 10.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Top trigeri*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Top trigeri logrīka konfigurēšana

Konfigurējiet *Top trigeri* logrīku, kas attēlo 5 populārākos trigerus hostu grupai "4" ar visu problēmu skaitu katram trigerim. Logrīks attēlo tikai tos trigerus, kuriem ir nopietnības līmeņi "Warning", "Average", "High" vai "Disaster", un problēmas, kurām ir birka ar nosaukumu "scope", kas satur vērtības "performance" vai "availability", vai "capacity".

Pieprasījums:

```
{
    "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
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
```

```
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

31 Trigeru pārskats

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Trigeru pārskata* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, taču vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Trigeru pārskata* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *Trigeru pārskata* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Rādīt	0	show	1 - *(noklusējums)* Nesenās problēmas; 2 - Jebkuras; 3 - Problēmas.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Vairākiem hostiem parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Parametrs	type	name	value
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Problēmu birkas			
Novērtēšanas veids	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Numurs īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Problēmu birkas*
Rādīt apspiestās problēmas	0	show_suppressed	0 - *(noklusējums)* Atspējots; 1 - Iespējots.
Izkārtojums	0	layout	0 - *(noklusējums)* Horizontāls; 1 - Vertikāls.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Trigeru pārskats*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Trigera pārskata logrīka konfigurēšana

Konfigurējiet *Trigera pārskata* logrīku, kas attēlo trigeru stāvokļus visām hostu grupām, kurām ir trigeri ar birku, kuras nosaukums ir "scope" un kuras vērtība satur "availability".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

32 URL

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *URL* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, bet vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *URL* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet tālāk esošajās tabulās aprakstīto parametru darbību.

Parametri

Tālāk norādītie parametri tiek atbalstīti *URL* logrīkam.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - *(noklusējums)* Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
URL	1	url	Derīga URL virkne. Parametra darbība: - *obligāts*
Pārrakstīt hostu	1	override_hostid._reference	`ABCDE._hostid` - iestatīt saderīgu logrīku (kuram parametrs *Reference* ir iestatīts uz "ABCDE") kā datu avotu hostiem; `DASHBOARD._hostid` - iestatīt paneļa hostu atlasītāju kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju *URL* logrīkam. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

URL logrīka konfigurēšana

Konfigurējiet *URL* logrīku, kas attēlo Zabbix rokasgrāmatas sākumlapu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

33 Tīmekļa uzraudzība

Apraksts

Šie parametri un iespējamās attiecīgo paneļa logrīka lauku objektu īpašību vērtības ļauj konfigurēt *Tīmekļa uzraudzības* logrīku `dashboard.create` un `dashboard.update` metodēs.

Attention:
Logrīka `fields` īpašības netiek validētas paneļa izveides vai atjaunināšanas laikā. Tas ļauj lietotājiem modificēt iebūvētos logrīkus un izveidot pielāgotus logrīkus, bet vienlaikus rada arī risku izveidot vai atjaunināt logrīkus nepareizi. Lai nodrošinātu veiksmīgu *Tīmekļa uzraudzības* logrīka izveidi vai atjaunināšanu, lūdzu, skatiet parametru darbību, kas aprakstīta tālāk esošajās tabulās.

Parametri

Tālāk norādītie parametri tiek atbalstīti logrīkam *Tīmekļa uzraudzība*.

Parametrs	type	name	value
Atsvaidzināšanas intervāls	0	rf_rate	0 - Bez atsvaidzināšanas; 10 - 10 sekundes; 30 - 30 sekundes; 60 - *(noklusējums)* 1 minūte; 120 - 2 minūtes; 600 - 10 minūtes; 900 - 15 minūtes.
Hostu grupas	2	groupids.0	Hostu grupas ID. Piezīme: Lai konfigurētu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hostu grupas (logrīks)	1	groupids._reference	Hostu grupas ID vietā: `ABCDE._hostgroupids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostu grupām. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Izslēgt hostu grupas	2	exclude_groupids.0	Hostu grupas ID. Piezīme: Lai izslēgtu vairākas hostu grupas, izveidojiet dashboard logrīka lauka objektu katrai hostu grupai ar palielinātu numuru īpašības nosaukumā. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti	3	hostids.0	Hosta ID. Piezīme: Lai konfigurētu vairākus hostus, izveidojiet dashboard logrīka lauka objektu katram hostam ar palielinātu numuru īpašības nosaukumā. Ja ir vairāki hosti, parametrs *Hostu grupas* vai nu vispār nedrīkst būt konfigurēts, vai arī tam jābūt konfigurētam ar vismaz vienu hostu grupu, kurai pieder konfigurētie hosti. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Hosti (logrīks/panelis)	1	hostids._reference	Hosta ID vietā: `DASHBOARD.hostids` - iestatiet paneļa hostu atlasītāju kā datu avotu hostiem; `ABCDE._hostids` - iestatiet saderīgu logrīku (ar tā parametru *Atsauce*, kas iestatīts uz "ABCDE") kā datu avotu hostiem. Šis parametrs netiek atbalstīts, ja logrīks tiek konfigurēts veidnes panelī.
Scenāriju birkas			

Parametrs	type	name	value
Novērtēšanas tips	0	evaltype	0 - *(noklusējums)* Un/Vai; 2 - Vai.
Birkas nosaukums	1	tags.0.tag	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Scenāriju birkas*
Operators	0	tags.0.operator	0 - Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Scenāriju birkas*
Birkas vērtība	1	tags.0.value	Jebkura virknes vērtība. Piezīme: Skaitlis īpašības nosaukumā norāda birkas secību birku novērtēšanas sarakstā. Parametra darbība: - *obligāts*, ja tiek konfigurētas *Scenāriju birkas*
Rādīt hostus uzturēšanā	0	maintenance	0 - Atspējots; 1 - *(noklusējums)* Iespējots.
Atsauce	1	reference	Jebkura virknes vērtība, kas sastāv no 5 rakstzīmēm (piemēram, `ABCDE` vai `JBPNL`). Šai vērtībai jābūt unikālai panelī, kuram pieder logrīks. Parametra darbība: - *obligāts*

Piemēri

Tālāk sniegto piemēru mērķis ir tikai aprakstīt paneļa logrīka lauku objektu konfigurāciju logrīkam *Tīmekļa uzraudzība*. Plašāku informāciju par paneļa konfigurēšanu skatiet `dashboard.create`.

Web monitoring logrīka konfigurēšana

Konfigurējiet *Web monitoring* logrīku, kas attēlo aktīvo tīmekļa uzraudzības scenāriju statusa kopsavilkumu hostu grupai "4".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Informācijas paneļa logrīka lauks
- `dashboard.create`
- `dashboard.update`

Problēma

Šī klase ir paredzēta darbam ar problēmām.

Objektu atsauces:

- Problem
  - Multivides tipa URL
- Problēmas tags

Pieejamās metodes:

- problem.get - izgūt problēmas

Problēmas objekts

Tālāk norādītie objekti ir tieši saistīti ar `problem` API.

Problēma

> **Note:**
> Problēmas izveido Zabbix serveris, un tās nevar mainīt, izmantojot API.

Problēmas objektam ir šādas īpašības.

Property	Type	Description
eventid	ID	Problēmas notikuma ID.
source	integer	Problēmas notikuma tips. Iespējamās vērtības: 0 - notikums, ko izveidojis trigeris; 3 - iekšējs notikums; 4 - notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā.
object	integer	Ar problēmas notikumu saistītā objekta tips. Iespējamās vērtības, ja `source` ir iestatīts uz "notikums, ko izveidojis trigeris": 0 - trigeris. Iespējamās vērtības, ja `source` ir iestatīts uz "iekšējs notikums": 0 - trigeris; 4 - vienums; 5 - LLD kārtula. Iespējamās vērtības, ja `source` ir iestatīts uz "notikums, kas izveidots pakalpojuma statusa atjaunināšanas laikā": 6 - pakalpojums.
objectid	ID	Saistītā objekta ID.
clock	timestamp	Laiks, kad tika izveidots problēmas notikums.
ns	integer	Nanosekundes, kad tika izveidots problēmas notikums.
r_eventid	ID	Atkopšanas notikuma ID.
r_clock	timestamp	Laiks, kad tika izveidots atkopšanas notikums.
r_ns	integer	Nanosekundes, kad tika izveidots atkopšanas notikums.
cause_eventid	ID	Cēloņa notikuma ID.
correlationid	ID	Korelācijas kārtulas ID, ja šis notikums tika atkopots ar globālu korelācijas kārtulu.
userid	ID	Tā lietotāja ID, kurš aizvēra problēmu (ja problēma tika aizvērta manuāli).
name	string	Atrisinātās problēmas nosaukums.
acknowledged	integer	Problēmas apstiprināšanas stāvoklis. Iespējamās vērtības: 0 - nav apstiprināta; 1 - apstiprināta.
severity	integer	Pašreizējā problēmas nopietnība. Iespējamās vērtības: 0 - nav klasificēta; 1 - informācija; 2 - brīdinājums; 3 - vidēja; 4 - augsta; 5 - katastrofa.

Property	Type	Description
suppressed	integer	Vai problēma ir apspiesta. Iespējamās vērtības: 0 - problēma ir normālā stāvoklī; 1 - problēma ir apspiesta.
opdata	string	Operatīvie dati ar izvērstām makrosiem.
urls	array	Aktīvie mediju tipa URL.

Mediju tipa URL

Mediju tipa URL objektam ir šādas īpašības.

Īpašība	Type	Apraksts
name	string	Mediju tipā definētā URL nosaukums.
url	string	Mediju tipā definētā URL vērtība.

Rezultātos būs iekļauti ieraksti tikai aktīviem mediju tipiem ar iespējotu notikumu izvēlnes ierakstu. Īpašībās izmantotie makrosi tiks izvērsti, bet, ja kāda no īpašībām satur neizvērstu makrosu, abas īpašības tiks izslēgtas no rezultātiem. Atbalstītos makrosus skatiet *Atbalstītie makrosi*.

Problēmas tags

Problēmas taga objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Problēmas taga nosaukums.
value	string	Problēmas taga vērtība.

problem.get

Apraksts

`integer/array problem.get(object parameters)`

Metode ļauj izgūt problēmas atbilstoši norādītajiem parametriem.

Šī metode ir paredzēta neatrisinātu problēmu izgūšanai. Ir arī iespējams, ja tas ir norādīts, papildus izgūt nesen atrisinātas problēmas. Periods, kas nosaka, cik sens ir "nesen", ir definēts sadaļā *Administrēšana* → *Vispārīgi*. Problēmas, kas tika atrisinātas pirms šī perioda, netiek glabātas problēmu tabulā. Lai izgūtu problēmas, kas tika atrisinātas vēl senākā pagātnē, izmantojiet metodi `event.get`.

Attention:
Šī metode var atgriezt dzēstas entītijas problēmas, ja housekeeper tās vēl nav noņēmis.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
eventids	ID/array	Atgriezt tikai problēmas ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai problēmas, ko izveidojuši objekti, kas pieder norādītajām hostu grupām.

Parametrs	Tips	Apraksts
hostids	ID/array	Atgriezt tikai problēmas, ko izveidojuši objekti, kas pieder norādītajiem hostiem.
objectids	ID/array	Atgriezt tikai problēmas, ko izveidojuši norādītie objekti.
source	integer	Atgriezt tikai problēmas ar norādīto tipu. Atbalstīto notikumu tipu sarakstu skatiet problēmas notikuma objekta lapā. Noklusējums: 0 - problēmu izveidojis trigeris.
object	integer	Atgriezt tikai problēmas, ko izveidojuši norādītā tipa objekti. Atbalstīto objektu tipu sarakstu skatiet problēmas notikuma objekta lapā. Noklusējums: 0 - trigeris.
acknowledged	boolean	Ja iestatīts uz `true`, atgriezt tikai apstiprinātās problēmas.
action	integer	Atgriezt tikai problēmas, kurām ir veiktas norādītās notikumu atjaunināšanas darbības. Vairākām darbībām izmantojiet jebkuru pieļaujamo bitkartes vērtību summu kā bitmasku (piemēram, 34 notikuma apstiprināšanai un apspiešanai).
action_userids	ID/array	Atgriezt tikai problēmas ar norādītajiem to lietotāju ID, kuri veica problēmas notikuma atjaunināšanas darbības.
suppressed	boolean	Ja iestatīts uz `true`, atgriezt tikai apspiestās problēmas.
symptom	boolean	Ja iestatīts uz `true`, atgriezt tikai simptomu problēmu notikumus.
severities	integer/array	Atgriezt tikai problēmas ar norādītajām notikumu nopietnības pakāpēm. Attiecas tikai tad, ja objekts ir trigeris.
evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
tags	array	Atgriezt tikai problēmas ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visas problēmas. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
recent	boolean	Ja iestatīts uz `true`, atgriezt tikai nesen atrisinātās problēmas (atkarīgs no `ok_period`).
eventid_from	string	Atgriezt tikai problēmas ar ID, kas ir lielāki vai vienādi ar norādīto ID.
eventid_till	string	Atgriezt tikai problēmas ar ID, kas ir mazāki vai vienādi ar norādīto ID.
time_from	timestamp	Atgriezt tikai problēmas, kas izveidotas pēc norādītā laika vai tajā pašā laikā.
time_till	timestamp	Atgriezt tikai problēmas, kas izveidotas pirms norādītā laika vai tajā pašā laikā.

Parametrs	Tips	Apraksts
selectAcknowledges	query	Atgriezt `acknowledges` īpašību ar problēmu atjauninājumiem. Problēmu atjauninājumi tiek kārtoti apgrieztā hronoloģiskā secībā. Problēmas atjauninājuma objektam ir šādas īpašības: `acknowledgeid` - `(ID)` atjauninājuma ID; `userid` - `(ID)` lietotāja ID, kurš atjaunināja notikumu; `eventid` - `(ID)` atjauninātā notikuma ID; `clock` - `(timestamp)` laiks, kad notikums tika atjaunināts; `message` - `(string)` ziņojuma teksts; `action` - `(integer)` atjaunināšanas darbības tips (skatiet `event.acknowledge`); `old_severity` - `(integer)` notikuma nopietnības pakāpe pirms šīs atjaunināšanas darbības; `new_severity` - `(integer)` notikuma nopietnības pakāpe pēc šīs atjaunināšanas darbības; `suppress_until` - `(timestamp)` laiks, līdz kuram notikums būs apspiests; `taskid` - `(ID)` uzdevuma ID, ja pašlaik notiek pašreizējā notikuma ranga maiņa. Atbalsta `count`.
selectTags	query	Atgriezt `tags` īpašību ar problēmas birkām. Izvades formāts: `[{"tag": "<tag>", "value": "<value>"}, ...]`.
selectSuppressionData	query	Atgriezt `suppression_data` īpašību ar aktīvo uzturēšanas periodu un manuālo apspiešanu sarakstu: `maintenanceid` - `(ID)` uzturēšanas perioda ID; `userid` - `(ID)` lietotāja ID, kurš apspieda problēmu; `suppress_until` - `(integer)` laiks, līdz kuram problēma ir apspiesta.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs salīdzināšanai. Neatbalsta `text` datu tipa īpašības.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `eventid`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Trigera problēmu notikumu izgūšana

Izgūst nesenos notikumus no trigera "15112."

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "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 aģents uz localhost nav sasniedzams 5 minūtes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "acknowledgeid": "14443",
                    "userid": "1",
                    "eventid": "1245463",
                    "clock": "1472457281",
                    "message": "problēma atrisināta",
                    "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
}
```

Norādītā lietotāja apstiprināto problēmu izgūšana

Lietotāja ar ID=10 apstiprināto problēmu izgūšana

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "problem.get",
    "params": {
        "output": "extend",
        "action": 2,
        "action_userids": [10],
        "selectAcknowledges": ["userid", "action"],
        "sortfield": ["eventid"],
        "sortorder": "DESC"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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 aģents uz localhost nav sasniedzams 5 minūtes",
            "acknowledged": "1",
            "severity": "3",
            "cause_eventid": "0",
            "opdata": "",
            "acknowledges": [
                {
                    "userid": "10",
                    "action": "2"
                }
            ],
            "suppressed": "0"
        }
    ],
    "id": 1
}
```

Skatiet arī

- Brīdinājums
- Vienums

- Hosts
- LLD noteikums
- Trigeris

Avots

CEvent::get() failā *ui/include/classes/api/services/CProblem.php*.

Pārskats

Šī klase ir paredzēta darbam ar ieplānotiem pārskatiem.

Objektu atsauces:

- Report
- Users
- User groups

Pieejamās metodes:

- report.create - izveidot jaunus ieplānotus pārskatus
- report.delete - dzēst ieplānotus pārskatus
- report.get - izgūt ieplānotus pārskatus
- report.update - atjaunināt ieplānotus pārskatus

Pārskata objekts

Tālāk norādītie objekti ir tieši saistīti ar `report` API.

Atskaite

Atskaites objektam ir šādas īpašības:

Property	Type	Description
reportid	ID	Atskaites ID. Property behavior: - *read-only* - *required* atjaunināšanas operācijām
userid	ID	Lietotāja ID, kurš izveidoja atskaiti. Property behavior: - *required* izveides operācijām
name	string	Unikāls atskaites nosaukums. Property behavior: - *required* izveides operācijām
dashboardid	ID	Paneļa ID, uz kura pamata atskaite ir izveidota. Property behavior: - *required* izveides operācijām
period	integer	Periods, par kuru tiks sagatavota atskaite. Iespējamās vērtības: 0 - *(noklusējums)* iepriekšējā diena; 1 - iepriekšējā nedēļa; 2 - iepriekšējais mēnesis; 3 - iepriekšējais gads.
cycle	integer	Perioda atkārtošanās grafiks. Iespējamās vērtības: 0 - *(noklusējums)* katru dienu; 1 - katru nedēļu; 2 - katru mēnesi; 3 - katru gadu.

Property	Type	Description
start_time	integer	Diennakts laiks sekundēs, kad atskaite tiks sagatavota nosūtīšanai. Noklusējums: 0.
weekdays	integer	Nedēļas dienas atskaites nosūtīšanai. Iespējamās bitkartes vērtības: 1 - pirmdiena; 2 - otrdiena; 4 - trešdiena; 8 - ceturtdiena; 16 - piektdiena; 32 - sestdiena; 64 - svētdiena. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 21 pirmdienai, trešdienai un piektdienai). Noklusējums: 0. Property behavior: - *required* ja `cycle` ir iestatīts uz "weekly".
active_since	string	Datums, no kura sākt. Iespējamās vērtības: tukša virkne - *(noklusējums)* nav norādīts (glabājas kā 0); konkrēts datums YYYY-MM-DD formātā (glabājas kā dienas sākuma laikspiedols (00:00:00)).
active_till	string	Datums, līdz kuram beigt. Iespējamās vērtības: tukša virkne - *(noklusējums)* nav norādīts (glabājas kā 0); konkrēts datums YYYY-MM-DD formātā (glabājas kā dienas beigu laikspiedols (23:59:59)).
subject	string	Atskaites ziņojuma tēma.
message	string	Atskaites ziņojuma teksts.
status	integer	Vai atskaite ir iespējota vai atspējota. Iespējamās vērtības: 0 - Atspējota; 1 - *(noklusējums)* Iespējota.
description	text	Atskaites apraksts.
state	integer	Atskaites stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* atskaite vēl nav apstrādāta; 1 - atskaite tika ģenerēta un veiksmīgi nosūtīta visiem saņēmējiem; 2 - atskaites ģenerēšana neizdevās; "info" satur kļūdas informāciju; 3 - atskaite tika ģenerēta, bet nosūtīšana dažiem (vai visiem) saņēmējiem neizdevās; "info" satur kļūdas informāciju. Property behavior: - *read-only*
lastsent	timestamp	Pēdējās veiksmīgi nosūtītās atskaites Unix laikspiedols. Property behavior: - *read-only*
info	string	Kļūdas apraksts vai papildu informācija. Property behavior: - *read-only*

Lietotāji

Lietotāju objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
userid	ID	Lietotāja ID, kuram nosūtīt atskaiti. Īpašības darbība: - *obligāts*
access_userid	ID	Lietotāja ID, kura vārdā tiks ģenerēta atskaite. 0 - *(noklusējums)* Ģenerēt atskaiti saņēmēja vārdā.
exclude	integer	Vai izslēgt lietotāju no adresātu saraksta. Iespējamās vērtības: 0 - *(noklusējums)* Iekļaut; 1 - Izslēgt.

Lietotāju grupas

Lietotāju grupu objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
usrgrpid	ID	Lietotāju grupas ID, kurai nosūtīt atskaiti. Īpašības darbība: - *obligāts*
access_userid	ID	Lietotāja ID, kura vārdā tiks ģenerēta atskaite. 0 - *(noklusējums)* Ģenerēt atskaiti saņēmēja vārdā.

report.create

Apraksts

`object report.create(object/array reports)`

Šī metode ļauj izveidot jaunus ieplānotus pārskatus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Ieplānotie pārskati, ko izveidot.

Papildus standarta ieplānotā pārskata īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
users	object/array	Lietotāji, kuriem nosūtīt pārskatu. Parametra darbība: - *obligāts*, ja `user_groups` nav iestatīts
user_groups	object/array	Lietotāju grupas, kurām nosūtīt pārskatu. Parametra darbība: - *obligāts*, ja `users` nav iestatīts

Atgrieztās vērtības

(object) Atgriež objektu, kas satur izveidoto ieplānoto atskaišu ID īpašībā `reportids`. Atgriezto ID secība atbilst nodoto ieplānoto atskaišu secībai.

Piemēri

Ieplānota pārskata izveide

Izveidojiet nedēļas pārskatu, kas tiks sagatavots par iepriekšējo nedēļu katru pirmdienu–piektdienu plkst. 12:00 no 2021-04-01 līdz 2021-08-31.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Lietotāji
- Lietotāju grupas

Avots

CReport::create() failā *ui/include/classes/api/services/CReport.php*.

report.delete

Apraksts

`object report.delete(array reportids)`

Šī metode ļauj dzēst ieplānotos pārskatus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Dzēšamo ieplānoto atskaišu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto ieplānoto atskaišu ID zem `reportids` īpašības.

Piemēri

Vairāku ieplānoto atskaišu dzēšana

Dzēsiet divas ieplānotās atskaites.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "report.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Avots

CReport::delete() failā *ui/include/classes/api/services/CReport.php*.

report.get

Apraksts

`integer/array report.get(object parameters)`

Metode ļauj izgūt ieplānotos pārskatus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
reportids	ID/array	Atgriezt tikai ieplānotos pārskatus ar norādītajiem pārskatu ID.
expired	boolean	Ja iestatīts uz `true`, atgriezt tikai ieplānotos pārskatus, kuriem ir beidzies derīguma termiņš.
selectUsers	query	Atgriezt `users` īpašību ar lietotājiem, kuriem ir konfigurēta pārskata nosūtīšana.
selectUserGroups	query	Atgriezt `user_groups` īpašību ar lietotāju grupām, kurām ir konfigurēta pārskata nosūtīšana.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `reportid`, `name`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Pārskata datu izgūšana

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "report.get",
    "params": [
        "output": "extend",
        "selectUsers": "extend",
        "selectUserGroups": "extend",
        "reportids": ["1", "2"]
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "reportid": "1",
            "userid": "1",
```

```
            "name": "Nedēļas pārskats",
            "dashboardid": "1",
            "period": "1",
            "cycle": "1",
            "start_time": "43200",
            "weekdays": "31",
            "active_since": "2021-04-01",
            "active_till": "2021-08-31",
            "subject": "Nedēļas pārskats",
            "message": "Pārskatam pievienotais teksts",
            "status": "1",
            "description": "Pārskata apraksts",
            "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": "Mēneša pārskats",
            "dashboardid": "2",
            "period": "2",
            "cycle": "2",
            "start_time": "0",
            "weekdays": "0",
            "active_since": "2021-05-01",
            "active_till": "",
            "subject": "Mēneša pārskats",
            "message": "Pārskatam pievienotais teksts",
            "status": "1",
            "description": "",
            "state": "0",
            "lastsent": "0",
            "info": "",
            "users": [
                {
                    "userid": "1",
                    "access_userid": "1",
                    "exclude": "0"
                }
            ],
            "user_groups": []
        }
    ],
```

```
    "id": 1
}
```

Skatiet arī

- Lietotāji
- Lietotāju grupas

Avots

CReport::get() failā *ui/include/classes/api/services/CReport.php*.

report.update

Apraksts

`object report.update(object/array reports)`

Šī metode ļauj atjaunināt esošos ieplānotos pārskatus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjauninámo plānoto atskaišu rekvizīti.

Rekvizītam `reportid` jābūt definētam katrai plānotajai atskaitei, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta plānotās atskaites rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
users	object/array	Lietotāji, ar kuriem aizstāt pašreizējos plānotajai atskaitei piešķirtos lietotājus. Parametra darbība: - *obligāts*, ja `user_groups` nav iestatīts
user_groups	object/array	Lietotāju grupas, ar kurām aizstāt pašreizējās plānotajai atskaitei piešķirtās lietotāju grupas. Parametra darbība: - *obligāts*, ja `users` nav iestatīts

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto ieplānoto atskaišu ID īpašībā `reportids`.

Piemēri

Ieplānotā pārskata atspējošana

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "report.update",
    "params": {
        "reportid": "1",
        "status": "0"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "reportids": [
            "1"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Lietotāji
- Lietotāju grupas

Avots

CReport::update() failā *ui/include/classes/api/services/CReport.php*.

Regulārā izteiksme

Šī klase ir paredzēta darbam ar globālajām regulārajām izteiksmēm.

Objektu atsauces:

- Regulārā izteiksme
- Izteiksmes

Pieejamās metodes:

- regexp.create - izveidot jaunas regulārās izteiksmes
- regexp.delete - dzēst regulārās izteiksmes
- regexp.get - izgūt regulārās izteiksmes
- regexp.update - atjaunināt regulārās izteiksmes

Regulārās izteiksmes objekts

Tālāk norādītie objekti ir tieši saistīti ar `regexp` API.

Regulārā izteiksme

Globālajam regulārās izteiksmes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
regexpid	ID	Regulārās izteiksmes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Regulārās izteiksmes nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
test_string	string	Testa virkne.

Izteiksmes

Izteiksmju objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
expression	string	Regulārā izteiksme. Īpašības darbība: - *obligāta*

Īpašība	Tips	Apraksts
expression_type	integer	Regulārās izteiksmes tips. Iespējamās vērtības: 0 - Iekļauta rakstzīmju virkne; 1 - Iekļauta jebkura rakstzīmju virkne; 2 - Rakstzīmju virkne nav iekļauta; 3 - Rezultāts ir TRUE; 4 - Rezultāts ir FALSE. Īpašības darbība: - *obligāta*
exp_delimiter	string	Izteiksmes atdalītājs. Noklusējuma vērtība: `","`. Iespējamās vērtības: `","` vai `"."`, vai `"/"`. Īpašības darbība: - *atbalstīta*, ja `expression_type` ir iestatīts uz ”Iekļauta jebkura rakstzīmju virkne”
case_sensitive	integer	Reģistra jutība. Noklusējuma vērtība: 0. Iespējamās vērtības: 0 - Reģistrnejutīga; 1 - Reģistrjutīga.

regexp.create

Apraksts

`object regexp.create(object/array regularExpressions)`

Šī metode ļauj izveidot jaunas globālās regulārās izteiksmes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Regulārās izteiksmes, ko izveidot.

Papildus standarta īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
expressions	array	Izteiksmju opcijas. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto regulāro izteiksmju ID īpašībā `regexpids`.

Piemēri

Jaunas globālās regulārās izteiksmes izveide.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16"
        ]
    },
    "id": 1
}
```

Avots

CRegexp::create() failā *ui/include/classes/api/services/CRegexp.php*.

regexp.delete

Apraksts

`object regexp.delete(array regexpids)`

Šī metode ļauj dzēst globālās regulārās izteiksmes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` dzēšamo regulāro izteiksmju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto regulāro izteiksmju ID īpašībā `regexpids`.

Piemēri

Vairāku globālo regulāro izteiksmju dzēšana.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.delete",
    "params": [
        "16",
        "17"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "16",
            "17"
        ]
    },
    "id": 1
}
```

Avots

CRegexp::delete() failā *ui/include/classes/api/services/CRegexp.php*.

regexp.get

Apraksts

`integer/array regexp.get(object parameters)`

Metode ļauj izgūt globālās regulārās izteiksmes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super Admin*. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
regexpids	ID/array	Atgriezt tikai regulārās izteiksmes ar norādītajiem ID.
selectExpressions	query	Atgriezt `expressions` īpašību.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `regexpid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Globālo regulāro izteiksmju izgūšana.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "regexp.get",
    "params": {
        "output": ["regexpid", "name"],
        "selectExpressions": ["expression", "expression_type"],
        "regexpids": [1, 2],
        "preservekeys": true
    },
    "id": 1
}
```

Atbilde:

```
{
  "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
}
```

Avots

CRegexp::get() failā *ui/include/classes/api/services/CRegexp.php*.

regexp.update

Apraksts

`object regexp.update(object/array regularExpressions)`

Šī metode ļauj atjaunināt esošās globālās regulārās izteiksmes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Atjaunināmāmo regulārās izteiksmes rekvizīti.

Katram objektam jābūt definētam rekvizītam `regexpid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
expressions	array	Izteiksmju opcijas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto regulāro izteiksmju ID īpašībā `regexpids`.

Piemēri

Globālās regulārās izteiksmes atjaunināšana failu sistēmu atklāšanai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "regexpids": [
            "1"
        ]
```

```
    },
    "id": 1
}
```

Avots

CRegexp::update() failā *ui/include/classes/api/services/CRegexp.php*.

Savienotājs

Šī klase ir paredzēta darbam ar savienotājiem.

Objektu atsauces:

- Savienotājs
- Birkas filtrs

Pieejamās metodes:

- connector.create - izveidot jaunus savienotājus
- connector.delete - dzēst savienotājus
- connector.get - izgūt savienotājus
- connector.update - atjaunināt savienotājus

Savienotāja objekts

Tālāk norādītie objekti ir tieši saistīti ar `connector` API.

Savienotājs

Savienotāja objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
connectorid	ID	Savienotāja ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Savienotāja nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
url	string	Galapunkta URL, tas ir, saņēmēja URL. Tiek atbalstīti lietotāja makrosi. Īpašības darbība: - *obligāts* izveides operācijām
protocol	integer	Saziņas protokols. Iespējamās vērtības: 0 - *(noklusējums)* Zabbix straumēšanas protokols v1.0.
data_type	integer	Datu tips. Iespējamās vērtības: 0 - *(noklusējums)* Vienumu vērtības; 1 - Notikumi.

Īpašība	Tips	Apraksts
item_value_type	integer	Nosūtāmo vienumu vērtību tipu summa. Iespējamās vērtības: 1 - Skaitlisks (peldošā komata); 2 - Rakstzīme; 4 - Žurnāls; 8 - Skaitlisks (bez zīmes); 16 - Teksts; 32 - Binārs; 64 - JSON. Noklusējums: 31 - Visi vienumu tipi (izņemot bināro un JSON). Īpašības darbība: - *atbalstīts*, ja `data_type` ir iestatīts uz "Vienumu vērtības".
max_records	integer	Maksimālais notikumu vai vienumu skaits, ko var nosūtīt vienā ziņojumā. Iespējamās vērtības: 0-2147483647 (32 bitu vesela skaitļa ar zīmi maksimālā vērtība). Noklusējums: 0 - Neierobežots.
max_senders	integer	Šim savienotājam palaižamo sūtītāja procesu skaits. Iespējamās vērtības: 1-100. Noklusējums: 1.
max_attempts	integer	Mēģinājumu skaits. Iespējamās vērtības: 1-5. Noklusējums: 1.
attempt_interval	string	Intervāls starp atkārtotiem mēģinājumiem. Pieņem sekundes. Iespējamās vērtības: 0s-10s. Noklusējums: 5s. Īpašības darbība: - *atbalstīts*, ja `max_attempts` ir lielāks par 1.
timeout	string	Noildze. Tiek atbalstīti laika sufiksi (piemēram, 30s, 1m). Tiek atbalstīti lietotāja makrosi. Iespējamās vērtības: 1s-60s. Noklusējums: 5s.
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne, norādīta kā *[protocol]://[username[:password]@]proxy.example.com[:port]*. Tiek atbalstīti lietotāja makrosi.
authtype	integer	HTTP autentifikācijas metode. Iespējamās vērtības: 0 - *(noklusējums)* Nav; 1 - Basic; 2 - NTLM; 3 - Kerberos; 4 - Digest; 5 - Bearer.

Īpašība	Tips	Apraksts
username	string	Lietotājvārds. Tiek atbalstīti lietotāja makrosi. Īpašības darbība: - *atbalstīts*, ja `authtype` ir iestatīts uz ”Basic”, ”NTLM”, ”Kerberos” vai ”Digest”
password	string	Parole. Tiek atbalstīti lietotāja makrosi. Īpašības darbība: - *atbalstīts*, ja `authtype` ir iestatīts uz ”Basic”, ”NTLM”, ”Kerberos” vai ”Digest”
token	string	Bearer marķieris. Tiek atbalstīti lietotāja makrosi. Īpašības darbība: - *obligāts*, ja `authtype` ir iestatīts uz ”Bearer”
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - Nepārbaudīt; 1 - *(noklusējums)* Pārbaudīt.
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Iespējamās vērtības: 0 - Nepārbaudīt; 1 - *(noklusējums)* Pārbaudīt.
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Tiek atbalstīti lietotāja makrosi.
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Tiek atbalstīti lietotāja makrosi.
ssl_key_password	string	SSL atslēgas faila parole. Tiek atbalstīti lietotāja makrosi.
description	text	Savienotāja apraksts.
status	integer	Vai savienotājs ir iespējots. Iespējamās vērtības: 0 - Atspējots; 1 - *(noklusējums)* Iespējots.
tags_evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.

Birku filtrs

Birku filtrs ļauj eksportēt tikai atbilstošas vienumu vērtības vai notikumus. Ja tas nav iestatīts, tiks eksportēts viss. Birku filtra objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Birkas nosaukums. Īpašības darbība: - *obligāta*

Īpašība	Type	Apraksts
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - *(noklusējums)* Vienāds ar; 1 - Nav vienāds ar; 2 - Satur; 3 - Nesatur; 12 - Eksistē; 13 - Neeksistē.
value	string	Birkas vērtība.

connector.create

Apraksts

`object connector.create(object/array connectors)`

Šī metode ļauj izveidot jaunus savienotāja objektus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie savienotāja objekti.

Papildus standarta savienotāja īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
tags	array	Savienotāja birkas filtrs.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto savienotāju ID īpašībā `connectorids`. Atgriezto ID secība atbilst nodoto savienotāju secībai.

Piemēri

Savienotāja izveide

Izveidojiet savienotāju, lai eksportētu trigeru notikumus ar birku filtru. HTTP autentifikācija tiks veikta, izmantojot Bearer token.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorid": [
            "3"
        ]
    },
    "id": 1
}
```

Avots

CConnector::create() failā *ui/include/classes/api/services/CConnector.php*.

connector.delete

Apraksts

`object connector.delete(array connectorids)`

Šī metode ļauj dzēst savienotāju ierakstus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo savienotāju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto savienotāju ID zem `connectorids` īpašības.

Piemēri

Vairāku savienotāju dzēšana

Dzēsiet divus savienotāju ierakstus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "connector.delete",
    "params": [
        3,
        5
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3",
            "5"
        ]
    },
```

```
    "id": 1
}
```

Avots

CConnector::delete() failā *ui/include/classes/api/services/CConnector.php*.

connector.get

Apraksts

`integer/array connector.get(object parameters)`

Metode ļauj izgūt savienotāja objektus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
connectorids	ID/masīvs	Atgriezt tikai savienotājus ar norādītajiem ID.
selectTags	query	Atgriezt `tags` īpašību ar savienotāja birku filtru. Atbalsta `count`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `connectorid`, `name`, `data_type`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Visu savienotāju izgūšana

Izgūstiet visus datus par visiem savienotājiem un to īpašībām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "connector.get",
    "params": {
        "output": "extend",
        "selectTags": ["tag", "operator", "value"],
```

```
        "preservekeys": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "connectorid": "1",
            "name": "Vienumu vērtību eksportēšana",
            "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": "Notikumu eksportēšana",
            "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
}
```

Avots

CConnector:get() failā *ui/include/classes/api/services/CConnector.php*.

connector.update

Apraksts

`object connector.update(object/array connectors)`

Šī metode ļauj atjaunināt esošos savienotājus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmā savienotāja rekvizīti.

Katram savienotājam jābūt definētam rekvizītam `connectorid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta savienotāja rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
tags	array	Savienotāja birkas filtrs, kas aizstās pašreizējo birku filtru.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto savienotāju ID zem `connectorids` īpašības.

Piemēri

HTTP autentifikācijas tipa maiņa

Mainiet HTTP autentifikācijas tipu uz Bearer savienotājam ar ID ”3”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "connector.update",
    "params": {
        "connectorid": 3,
        "authtype": 5,
        "token": "{$DATA_EXPORT_BEARER_TOKEN}"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "3"
        ]
    },
    "id": 1
}
```

Atjauninot birku filtru

Mainiet birku filtru savienotājam ar ID ”5”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "connectorids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CConnector::update() failā *ui/include/classes/api/services/CConnector.php*.

Skripts

Šī klase ir paredzēta darbam ar skriptiem.

Objektu atsauces:

- Skripts
  - Webhook parametri
- Atkļūdošana

- Žurnāla ieraksts

Pieejamās metodes:

- script.create - izveidot jaunus skriptus
- script.delete - dzēst skriptus
- script.execute - palaist skriptus
- script.get - izgūt skriptus
- script.getscriptsbyevents - izgūt skriptus notikumiem
- script.getscriptsbyhosts - izgūt skriptus hostiem
- script.update - atjaunināt skriptus

Skripta objekts

Tālāk norādītie objekti ir tieši saistīti ar `script` API.

Skripts

Skripta objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
scriptid	ID	Skripta ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Skripta nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
type	integer	Skripta tips. Iespējamās vērtības, ja `scope` ir iestatīts uz ”action operation”: 0 - Skripts; 1 - IPMI; 2 - SSH; 3 - TELNET; 5 - Webhook. Iespējamās vērtības, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”: 6 - URL. Īpašības darbība: - *obligāts* izveides operācijām
command	string	Komanda, ko izpildīt. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Script”, ”IPMI”, ”SSH”, ”TELNET” vai ”Webhook”
scope	integer	Skripta tvērums. Iespējamās vērtības: 1 - darbības operācija; 2 - manuāla hosta darbība; 4 - manuāla notikuma darbība. Īpašības darbība: - *obligāts* izveides operācijām

Īpašība	Tips	Apraksts
execute_on	integer	Kur izpildīt skriptu. Iespējamās vērtības: 0 - izpildīt uz Zabbix aģenta; 1 - izpildīt uz Zabbix servera. Tas tiek *atbalstīts* tikai tad, ja Zabbix serverī ir iespējota globālo skriptu izpilde; 2 - *(noklusējums)* izpildīt uz Zabbix servera vai starpniekservera. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Script"
menu_path	string	Mapes, kas atdalītas ar slīpsvītru un veido izvēlnei līdzīgu navigāciju lietotāja saskarnē, noklikšķinot uz hosta vai notikuma. Īpašības darbība: - *atbalstīts*, ja `scope` ir iestatīts uz "manual host action" vai "manual event action"
authtype	integer	Autentifikācijas metode, kas izmantota SSH skripta tipam. Iespējamās vērtības: 0 - parole; 1 - publiskā atslēga. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "SSH"
username	string	Lietotājvārds, kas izmantots autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH" vai "TELNET"
password	string	Parole, kas izmantota SSH skriptiem ar paroles autentifikāciju un TELNET skriptiem. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "SSH" un `authtype` ir iestatīts uz "password", vai `type` ir iestatīts uz "TELNET"
publickey	string	Publiskās atslēgas faila nosaukums, kas izmantots SSH skriptiem ar publiskās atslēgas autentifikāciju. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH" un `authtype` ir iestatīts uz "public key"
privatekey	string	Privātās atslēgas faila nosaukums, kas izmantots SSH skriptiem ar publiskās atslēgas autentifikāciju. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH" un `authtype` ir iestatīts uz "public key"
port	string	Porta numurs, kas izmantots SSH un TELNET skriptiem. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "SSH" vai "TELNET"
groupid	ID	Hostu grupas ID, uz kuras var izpildīt skriptu. Ja iestatīts uz "0", skripts būs pieejams visās hostu grupās. Noklusējums: 0.

Īpašība	Tips	Apraksts
usrgrpid	ID	Lietotāju grupas ID, kurai būs atļauts izpildīt skriptu. Ja iestatīts uz ”0”, skripts būs pieejams visām lietotāju grupām. Noklusējums: 0. Īpašības darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”
host_access	integer	Hosta atļaujas, kas nepieciešamas skripta izpildei. Iespējamās vērtības: 2 - *(noklusējums)* lasīšana; 3 - rakstīšana. Īpašības darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”
confirmation	string	Apstiprinājuma uznirstošā loga teksts. Mēģinot izpildīt skriptu no Zabbix lietotāja saskarnes, tiks parādīts uznirstošais logs. Īpašības darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”
timeout	string	Webhook skripta izpildes taimauts sekundēs. Tiek atbalstīti laika sufiksi (piemēram, 30s, 1m). Iespējamās vērtības: 1-60s. Noklusējums: 30s. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Webhook”
parameters	array	webhook ievades parametru masīvs. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Webhook”
description	string	Skripta apraksts.
url	string	Lietotāja definēts URL. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”URL”
new_window	integer	Atvērt URL jaunā logā. Iespējamās vērtības: 0 - Nē; 1 - *(noklusējums)* Jā. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”URL”
manualinput	integer	Norāda, vai skripts pieņem lietotāja ievadītu ievadi. Iespējamās vērtības: 0 - *(noklusējums)* Atspējots; 1 - Iespējots; Īpašības darbība: - *atbalstīts*, ja `scope` ir iestatīts uz ”manual host action” vai ”manual event action”

Īpašība	Tips	Apraksts
manualinput_prompt	string	Manuālās ievades uzvednes teksts. Īpašības darbība: - *obligāts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_validator	string	Rakstzīmju virknes lauks, kas izmantots lietotāja ievadītās ievades validēšanai. Virkne sastāv vai nu no regulārās izteiksmes, vai no vērtību kopas, kas atdalītas ar komatiem. Īpašības darbība: - *obligāts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_validator_type	integer	Nosaka sagaidāmās lietotāja ievades tipu. Iespējamās vērtības: 0 - *(noklusējums)* Virkne. Norāda, ka manualinput_validator ir jāapstrādā kā regulārā izteiksme; 1 - Saraksts. Norāda, ka manualinput_validator ir jāapstrādā kā ar komatiem atdalīts iespējamo ievades vērtību saraksts. Īpašības darbība: - *atbalstīts*, ja `manualinput` ir iestatīts uz ”Enabled”
manualinput_default_value	string	Noklusējuma vērtība lietotāja ievades automātiskai aizpildīšanai. Īpašības darbība: - *atbalstīts*, ja `manualinput_validator_type` ir iestatīts uz ”String”

Webhook parametri

Parametriem, kas tiek nodoti webhook skriptam tā izsaukšanas laikā, ir šādas īpašības.

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāts*
value	string	Parametra vērtība. Atbalsta makro.

Atkļūdošana

Izpildītā webhook skripta atkļūdošanas informācija. Atkļūdošanas objektam ir šādas īpašības.

Īpašība	Type	Apraksts
logs	array	Žurnāla ierakstu masīvs.
ms	string	Skripta izpildes ilgums milisekundēs.

Žurnāla ieraksts

Žurnāla ieraksta objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
level	integer	Žurnāla līmenis.
ms	string	Milisekundēs izteiktais laiks, kas pagājis kopš skripta palaišanas, pirms tika pievienots žurnāla ieraksts.
message	string	Žurnāla ziņojums.

script.create

Apraksts

`object script.create(object/array scripts)`

Šī metode ļauj izveidot jaunus skriptus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Skripti, ko izveidot.

Metode pieņem skriptus ar standarta skripta īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto skriptu ID īpašībā `scriptids`. Atgriezto ID secība atbilst nodoto skriptu secībai.

Piemēri

Izveidot webhook skriptu

Izveidojiet webhook skriptu, kas nosūta HTTP pieprasījumu uz ārēju servisu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3"
        ]
    },
    "id": 1
}
```

Izveidot SSH skriptu

Izveidojiet SSH skriptu ar publiskās atslēgas autentifikāciju, ko var izpildīt uz hosta un kam ir konteksta izvēlne.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "5"
        ]
    },
    "id": 1
}
```

Izveidot pielāgotu skriptu

Izveidojiet pielāgotu skriptu, kas pārstartēs serveri. Skriptam būs nepieciešama rakstīšanas piekļuve hostam, un tas aicinās lietotāju ievadīt manuālu ievadi. Pēc veiksmīgas ievades iesniegšanas skripts parādīs apstiprinājuma ziņojumu lietotāja saskarnē.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "4"
        ]
```

```
    },
    "id": 1
}
```

Izveidot URL tipa skriptu

Izveidojiet URL tipa skriptu hosta tvērumam, kas paliek tajā pašā logā un kam ir apstiprinājuma teksts.

Pieprasījums:

```
{
    "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": "Rediģēt hostu {HOST.NAME}?",
        "new_window": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "56"
        ]
    },
    "id": 1
}
```

Izveidojiet URL tipa skriptu ar manuālu ievadi

Izveidojiet URL tipa skriptu notikuma tvērumam, kas tiek atvērts jaunā logā un kam ir manuāla ievade.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "57"
        ]
    },
```

```
    "id": 1
}
```

Avots

CScript::create() failā *ui/include/classes/api/services/CScript.php*.

script.delete

Apraksts

`object script.delete(array scriptIds)`

Šī metode ļauj dzēst skriptus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` dzēšamo skriptu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto skriptu ID zem `scriptids` īpašības.

Piemēri

Vairāku skriptu dzēšana

Dzēsiet divus skriptus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.delete",
    "params": [
        "3",
        "4"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "3",
            "4"
        ]
    },
    "id": 1
}
```

Avots

CScript::delete() failā *ui/include/classes/api/services/CScript.php*.

script.execute

Apraksts

`object script.execute(object parameters)`

Šī metode ļauj izpildīt skriptu hostā vai notikumā. Izņemot URL tipa skriptus. Tie nav izpildāmi.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur izpildāmā skripta ID, vai nu hosta ID, vai notikuma ID un manualinput vērtību.

Parametrs	Type	Apraksts
scriptid	ID	Izpildāmā skripta ID. Parametra darbība: - *obligāts*
hostid	ID	Hosta ID, uz kura izpildīt skriptu. Parametra darbība: - *obligāts*, ja `eventid` nav iestatīts
eventid	ID	Notikuma ID, uz kura izpildīt skriptu. Parametra darbība: - *obligāts*, ja `hostid` nav iestatīts
manualinput	string	Lietotāja norādīta vērtība skripta izpildei, aizstājot makrosu {MANUALINPUT}.

Atgrieztās vērtības

`(object)` Atgriež skripta izpildes rezultātu.

Īpašība	Type	Apraksts
response	string	Vai skripts tika izpildīts veiksmīgi. Iespējamā vērtība - `success`.
value	string	Skripta izvade.
debug	object	Satur debug objektu, ja tiek izpildīts webhook skripts. Citiem skriptu tipiem tas satur tukšu objektu.

Piemēri

Palaist webhook skriptu

Palaidiet webhook skriptu, kas nosūta HTTP pieprasījumu uz ārēju servisu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "4",
        "hostid": "30079"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Palaist pielāgotu skriptu

Palaidiet ”ping” skriptu hostā.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "1",
        "hostid": "30079"
    },
    "id": 1
}
```


Atbilde:


```
{
    "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
}
```


Palaidiet pielāgotu skriptu ar manuālu ievadi

Palaidiet ”ping” skriptu ar komandu ”ping -c {MANUALINPUT} {HOST.CONN}; case $? in [01]) true;; *) false;; esac” hostā.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "script.execute",
    "params": {
        "scriptid": "7",
        "hostid": "30079",
        "manualinput": "2"
    },
    "id": 1
}
```


Atbilde:


```
{
    "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
}
```

Avots

CScript::execute() failā *ui/include/classes/api/services/CScript.php*.

script.get

Apraksts

`integer/array script.get(object parameters)`

Šī metode ļauj izgūt skriptus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai skriptus, kurus var palaist norādītajās hostu grupās.
hostids	ID/array	Atgriezt tikai skriptus, kurus var palaist norādītajos hostos.
scriptids	ID/array	Atgriezt tikai skriptus ar norādītajiem ID.
usrgrpids	ID/array	Atgriezt tikai skriptus, kurus var palaist lietotāji no norādītajām lietotāju grupām.
selectHostGroups	query	Atgriezt `hostgroups` īpašību ar hostu grupām, kurās skriptu var palaist.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kuros skriptu var palaist.
selectActions	query	Atgriezt `actions` īpašību ar darbībām, ar kurām skripts ir saistīts.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `scriptid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt visus skriptus

Iegūt visus konfigurētos skriptus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.get",
    "params": {
```

```
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Hosts
- Hostu grupa

Avots

CScript::get() failā *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyevents

Apraksts

`object script.getscriptsbyevents(object parameters)`

Šī metode ļauj izgūt visus pieejamos skriptus norādītajam notikumam vai konkrētu skriptu, ja ir norādīts skripta ID. Ja ir norādīts manualinput, tas aizstāj {MANUALINPUT} makrosu ar norādīto vērtību.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Metode pieņem objektu vai objektu masīvu ar šādiem parametriem.

Parametrs	Tips	Apraksts
eventid	ID	Atgriežamo skriptu notikuma ID. Jābūt unikālam. Parametra darbība: - *obligāts*
scriptid	ID	Atgriežamā skripta ID.
manualinput	string	Lietotāja norādītās {MANUALINPUT} makro vērtība.

Atgrieztās vērtības

`(object)` Atgriež objektu, kurā notikumu ID ir kā īpašības un pieejamo skriptu masīvi ir kā vērtības. Ja ir norādīts skripta ID, saistītā vērtība ir masīvs, kas satur konkrēto skriptu.

Note:
Metode automātiski izvērsīs makrosus `confirmation` tekstā, `manualinput prompt` tekstā un `url`.
Ja ir norādīts parametrs manualinput, makross {MANUALINPUT} tiks atrisināts uz norādīto vērtību.

Piemēri

Skriptu izgūšana pēc notikumu ID

Izgūst visus skriptus, kas pieejami notikumiem ”632” un ”614”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
         "eventid":  "632"
      },
      {
         "eventid":  "614"
      }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "632": [
            {
                "scriptid": "3",
                "name": "Noteikt operētājsistēmu",
                "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": "http://localhost/ui/zabbix.php?action=*UNKNOWN*",
                "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": "Atvērt Zabbix lapu",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Vai tiešām vēlaties atvērt lapu *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": "Atveramā Zabbix lapa:",
            "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": "Noteikt operētājsistēmu",
                "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": "Atvērt Zabbix lapu",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Vai tiešām vēlaties atvērt lapu *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": "Atveramā Zabbix lapa:",
            "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
}
```

Konkrēta skripta izgūšana ar manualinput vērtību.

Izgūstiet skriptu ar ID "4" notikumam "632" ar manualinput vērtību "dashboard.view".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyevents",
    "params": [
      {
        "eventid":  "632",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CScript::getScriptsByEvents() failā *ui/include/classes/api/services/CScript.php*.

script.getscriptsbyhosts

Apraksts

`object script.getscriptsbyhosts(object parameters)`

Šī metode ļauj iegūt visus pieejamos skriptus norādītajā hostā vai konkrētu skriptu, ja ir norādīts skripta ID. Ja ir norādīts manualinput, tas aizstāj makrosu {MANUALINPUT} ar norādīto vērtību.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Metode pieņem objektu vai objektu masīvu ar šādiem parametriem.

Parametrs	Type	Apraksts
hostid	ID	Hosta ID, kuram atgriezt skriptus. Jābūt unikālam. Parametra darbība: - *obligāts*
scriptid	ID	Atgriežamā skripta ID.
manualinput	string	Lietotāja norādītā {MANUALINPUT} makro vērtība.

Atgrieztās vērtības

`(object)` Atgriež objektu, kurā hostu ID ir īpašības un pieejamo skriptu masīvi ir vērtības. Ja ir norādīts skripta ID, saistītā vērtība ir masīvs, kas satur konkrēto skriptu.

Note:
Metode automātiski izvērsīs makrosus `confirmation` tekstā, `manualinput prompt` tekstā un `url`.
Ja ir norādīts parametrs manualinput, makross {MANUALINPUT} tiks atrisināts uz norādīto vērtību.

Piemēri

Skriptu izgūšana pēc hostu ID

Izgūst visus skriptus, kas pieejami hostos ”30079” un ”30073”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
       {
          "hostid":  "30079"
       },
       {
          "hostid":  "30073"
       }
    ],
    "id": 1
}
```

Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": {
        "30079": [
            {
                "scriptid": "3",
                "name": "Noteikt operētājsistēmu",
                "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": "Atvērt Zabbix lapu",
            "command": "",
            "host_access": "2",
            "usrgrpid": "0",
            "groupid": "0",
            "description": "",
            "confirmation": "Vai tiešām vēlaties atvērt lapu *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": "Atveramā Zabbix lapa:",
            "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": "Noteikt operētājsistēmu",
            "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": "Atvērt Zabbix lapu",
            "command": "",
            "host_access": "2",
```

```
                "usrgrpid": "0",
                "groupid": "0",
                "description": "",
                "confirmation": "Vai tiešām vēlaties atvērt lapu *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": "Atveramā Zabbix lapa:",
                "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
}
```


Konkrēta skripta izgūšana ar manualinput vērtību.

Izgūstiet skriptu ar ID "4" hostam "30079" ar manualinput vērtību "dashboard.view".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.getscriptsbyhosts",
    "params": [
      {
        "hostid":  "30079",
        "scriptid": "4",
        "manualinput": "dashboard.view"
      }
  ],
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CScript::getScriptsByHosts() failā *ui/include/classes/api/services/CScript.php*.

script.update

Apraksts

`object script.update(object/array scripts)`

Šī metode ļauj atjaunināt esošos skriptus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Skripta īpašības, kas jāatjaunina.

Īpašībai `scriptid` jābūt definētai katram skriptam, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas. Izņēmums ir īpašības `type` maiņa no 5 (Webhook) uz citu vērtību: īpašība `parameters` tiks notīrīta.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto skriptu ID īpašībā `scriptids`.

Piemēri

Mainīt skripta komandu

Mainīt skripta komandu uz "/bin/ping -c 10 {HOST.CONN} 2>&1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
        "scriptid": "1",
        "command": "/bin/ping -c 10 {HOST.CONN} 2>&1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Mainīt skripta komandu un pievienot manuālu ievadi

Mainiet skripta komandu uz "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "script.update",
    "params": {
      "scriptid": "1",
      "command": "/bin/ping -c {MANUALINPUT} {HOST.CONN} 2>&1",
      "manualinput": "1",
      "manualinput_prompt": "Norādiet ar ping komandu nosūtāmo ICMP pakešu skaitu",
      "manualinput_validator": "^(?:[1-9]|10)$",
      "manualinput_validator_type": "0",
      "manualinput_default_value": "10"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "scriptids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CScript::update() failā *ui/include/classes/api/services/CScript.php*.

SLA

Šī klase ir paredzēta darbam ar SLA (Service Level Agreement) objektiem, ko izmanto IT infrastruktūras un biznesa pakalpojumu veiktspējas novērtēšanai.

Objektu atsauces:

- SLA
- SLA grafiks
- SLA izslēgtā dīkstāve
- SLA pakalpojuma tags

Pieejamās metodes:

- sla.create - izveidot jaunus SLA
- sla.delete - dzēst SLA
- sla.get - izgūt SLA
- sla.getsli - izgūt Service Level Indicator (SLI) datus SLA
- sla.update - atjaunināt SLA

SLA objekts

Tālāk norādītie objekti ir tieši saistīti ar `sla` (Service Level Agreement) API.

SLA

SLA objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
slaid	ID	SLA ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	SLA nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
period	integer	SLA pārskata periods. Iespējamās vērtības: 0 - katru dienu; 1 - katru nedēļu; 2 - katru mēnesi; 3 - katru ceturksni; 4 - katru gadu. Īpašības darbība: - *obligāts* izveides operācijām

Īpašība	Tips	Apraksts
slo	float	Minimāli pieļaujamais pakalpojuma līmeņa mērķis, izteikts procentos. Ja pakalpojuma līmeņa indikators (SLI) nokrītas zemāk, SLA tiek uzskatīts par problemātisku/neizpildītu. Iespējamās vērtības: 0-100 (līdz 4 cipariem aiz komata). Īpašības darbība: - *obligāts* izveides operācijām
effective_date	integer	SLA spēkā stāšanās datums. Iespējamās vērtības: datuma laikspiedols UTC.
timezone	string	Pārskata laika josla, piemēram: `Europe/London`, `UTC`. Pilnu atbalstīto laika joslu sarakstu skatiet PHP dokumentācijā. Īpašības darbība: - *obligāts* izveides operācijām
status	integer	SLA statuss. Iespējamās vērtības: 0 - *(noklusējums)* SLA ir atspējots; 1 - SLA ir iespējots.
description	string	SLA apraksts.

SLA grafiks

SLA grafika objekts definē periodus, kuros saistītajam(-iem) pakalpojumam(-iem) ir jābūt darba kārtībā. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
period_from	integer	Atkārtota nedēļas laika perioda sākuma laiks (ieskaitot). Iespējamās vērtības: sekunžu skaits (skaitot no svētdienas). Īpašības darbība: - *required*
period_to	integer	Atkārtota nedēļas laika perioda beigu laiks (neieskaitot). Iespējamās vērtības: sekunžu skaits (skaitot no svētdienas). Īpašības darbība: - *required*

SLA izslēgtais dīkstāves laiks

Izslēgtā dīkstāves laika objekts definē periodus, kuros saistītajam pakalpojumam(-iem) ir plānota nedarbošanās, neietekmējot SLI, piemēram, plānotas uzturēšanas laikā. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
name	string	Izslēgtā dīkstāves perioda nosaukums. Īpašības darbība: - *obligāts*
period_from	integer	Izslēgtā dīkstāves perioda sākuma laiks (ieskaitot). Iespējamās vērtības: timestamp. Īpašības darbība: - *obligāts*

Īpašība	Type	Apraksts
period_to	integer	Izslēgtā dīkstāves perioda beigu laiks (neieskaitot). Iespējamās vērtības: timestamp. Īpašības darbība: - *obligāts*

SLA pakalpojuma birka

SLA pakalpojuma birkas objekts sasaista pakalpojumus, kas jāiekļauj SLA aprēķinos. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	SLA pakalpojuma birkas nosaukums. Īpašības darbība: - *obligāts*
operator	integer	SLA pakalpojuma birkas operators. Iespējamās vērtības: 0 - *(noklusējums)* Vienāds; 2 - Satur.
value	string	SLA pakalpojuma birkas vērtība.

sla.create

Apraksts

`object sla.create(object/array SLAs)`

Šī metode ļauj izveidot jaunus SLA objektus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` SLA objekti, ko izveidot.

Papildus standarta SLA īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
service_tags	array	SLA pakalpojumu birkas, kas jāizveido SLA. Parametra darbība: - *obligāts*
schedule	array	SLA grafiks, kas jāizveido SLA. Norādot tukšu parametru, tas tiks interpretēts kā 24x7 grafiks. Noklusējums: 24x7 grafiks.
excluded_downtimes	array	SLA izslēgtās dīkstāves, kas jāizveido SLA.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto SLA ID zem `slaids` īpašības. Atgriezto ID secība atbilst nodoto SLA secībai.

Piemēri

SLA izveide

Norādiet izveidot SLA ierakstu šādam mērķim: * darbspējas laika uzskaite ar SQL dzini saistītiem pakalpojumiem; * pielāgots grafiks visām darba dienām, izņemot pēdējo stundu sestdienā; * spēkā stāšanās datums — 2022. gada pēdējā diena; * ar 1 stundu

un 15 minūtes ilgu plānoto dīkstāvi, kas sākas 4. jūlijā pusnaktī; * SLA nedēļas pārskata aprēķināšana būs ieslēgta; * minimālais pieļaujamais SLO būs 99.9995%.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "sla.create",
    "params": [
        {
            "name": "Datubāzes darbspējas laiks",
            "slo": "99.9995",
            "period": "1",
            "timezone": "America/Toronto",
            "description": "Nodrošināt izcilu galveno datubāzes dzinēju darbspējas laiku.",
            "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": "Programmatūras versijas jauninašanas izvēršana",
                    "period_from": "1648760400",
                    "period_to": "1648764900"
                }
            ]
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CSla::create() failā *ui/include/classes/api/services/CSla.php*.

sla.delete

Apraksts

`object sla.delete(array slaids)`

Šī metode ļauj dzēst SLA ierakstus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo SLA ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto SLA ID zem `slaids` īpašības.

Piemēri

Vairāku SLA dzēšana

Dzēsiet divus SLA ierakstus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "sla.delete",
    "params": [
        "4",
        "5"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "4",
            "5"
        ]
    },
    "id": 1
}
```

Avots

CSla::delete() failā *ui/include/classes/api/services/CSla.php*.

sla.get

Apraksts

`integer/array sla.get(object parameters)`

Metode ļauj izgūt SLA objektus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
slaids	ID/array	Atgriezt tikai SLA ar norādītajiem ID.
serviceids	ID/array	Atgriezt tikai SLA, kas atbilst konkrētajiem pakalpojumiem.
selectSchedule	query	Atgriezt `schedule` īpašību ar SLA grafikiem. Atbalsta `count`.
selectExcludedDowntimes	query	Atgriezt `excluded_downtimes` īpašību ar SLA izslēgtajām dīkstāvēm. Atbalsta `count`.
selectServiceTags	query	Atgriezt `service_tags` īpašību ar SLA pakalpojumu birkām. Atbalsta `count`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `slaid`, `name`, `period`, `slo`, `effective_date`, `timezone`, `status`, `description`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Visu SLA izgūšana

Izgūstiet visus datus par visiem SLA un to īpašībām.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "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
}
```


Avots

CSla:get() failā *ui/include/classes/api/services/CSla.php*.

sla.getsli

Apraksts

`object sla.getsli(object parameters)`

Šī metode ļauj aprēķināt Service Level Indicator (SLI) datus Service Level Agreement (SLA).

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur SLA ID, pārskata periodus un pēc izvēles pakalpojumu ID — lai aprēķinātu SLI.

Parametrs	Tips	Apraksts
slaid	ID	SLA ID, kuram jāatgriež pieejamības informācija. Parametra darbība: - *obligāts*
period_from	timestamp	Sākuma laikspiedols (ieskaitot), par kuru jāveido SLI pārskats. Iespējamās vērtības: Unix laikspiedols.

Parametrs	Tips	Apraksts
period_to	timestamp	Beigu laikspiedols (ieskaitot), par kuru jāveido SLI pārskats. Iespējamās vērtības: Unix laikspiedols.
periods	integer	Pārskatāmo periodu skaits. Iespējamās vērtības: 1-100
serviceids	ID/array	Pakalpojumu ID, kuriem jāatgriež SLI.

Periodu sadalījums

Tālāk redzamajā tabulā ir parādīts atgriezto periodu nogriežņu izkārtojums atkarībā no parametru kombinācijām.

Note:
Atgrieztie periodi nesākas pirms pirmā pieejamā perioda (pamatojoties uz SLA spēkā stāšanās datumu) un nepārsniedz pašreizējo periodu.

Parametri			Atgrieztie periodi
period_from	**period_to**	**periods**	
-	-	-	Pēdējie 20 periodi, ieskaitot pašreizējo.
-	-	N	Pēdējie N periodi.
-	norādīts	-	Pēdējie 20 periodi pirms `period_to`.
-	norādīts	N	Pēdējie N periodi pirms `period_to`.
norādīts	-	-	Pirmie 20 periodi, sākot no `period_from`.
norādīts	-	N	Pirmie N periodi, sākot no `period_from`.
norādīts	norādīts	-	Līdz 100 periodiem norādītajā diapazonā.
norādīts	norādīts	N	N periodi norādītajā diapazonā.

Atgrieztās vērtības

`(object)` Atgriež aprēķina rezultātus.

Īpašība	Type	Apraksts
periods	array	Paziņoto periodu saraksts. Katrs paziņotais periods ir attēlots kā objekts, kas sastāv no: - `period_from` - `(timestamp)` Paziņotā perioda sākuma laikspiedols (ieskaitot). - `period_to` - `(timestamp)` Paziņotā perioda beigu laikspiedols (neieskaitot). Periodi ir sakārtoti pēc `period_from`, vispirms attēlojot agrīnākos periodus.
serviceids	array	Paziņotajos periodos esošo pakalpojumu ID saraksts. Saraksta kārtošanas secība nav noteikta. Pat ja `serviceids` parametrs tika nodots metodei `sla.getsli`.
sli	array	SLI dati (**divdimensiju masīva** veidā) katram paziņotajam periodam un pakalpojumam. Īpašības `periods` indekss tiek izmantots kā īpašības `sli` **pirmā** dimensija. Īpašības `serviceids` indekss tiek izmantots kā īpašības `sli` **otrā** dimensija.

SLI dati

SLI dati, kas tiek atgriezti par katru pārskata periodu un pakalpojumu, sastāv no:

Īpašība	Type	Apraksts
uptime	integer	Laika apjoms, kurā pakalpojums plānotajā darbības laikā atradās *OK* stāvoklī, atskaitot izslēgtās dīkstāves.
downtime	integer	Laika apjoms, kurā pakalpojums plānotajā darbības laikā atradās *not OK* stāvoklī, atskaitot izslēgtās dīkstāves.
sli	float	SLI (procenti no kopējā darbības laika), pamatojoties uz uptime un downtime.
error_budget	integer	Kļūdu budžets (sekundēs), pamatojoties uz SLI un SLO.
excluded_downtimes	array	Izslēgto dīkstāves periodu masīvs šajā pārskata periodā. Katrs objekts saturēs šādus parametrus: - `name` - (`string`) Izslēgtās dīkstāves nosaukums. - `period_from` - (`timestamp`) Izslēgtās dīkstāves sākuma laikspiedols (ieskaitot). - `period_to` - (`timestamp`) Izslēgtās dīkstāves beigu laikspiedols (neieskaitot). Izslēgtās dīkstāves ir sakārtotas pēc `period_from`, vispirms attēlojot agrīnākos periodus.

Piemēri

SLI aprēķināšana dienas SLA

Iegūstiet SLI datus pakalpojumiem ar ID "1" un "4", kas ir saistīti ar SLA ar ID "1". Iegūstiet datus vienam periodam līdz "1761861599" (30. okt. 2025 23:59:59 GMT+0200). Tā kā SLA atskaites periods ir diena, SLI dati tiek iegūti no "1761775200" (30. okt. 2025 00:00:00 GMT+0200) līdz "1761861600" (31. okt. 2025 00:00:00 GMT+0200).

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "1",
        "serviceids": [
            1,
            4
        ],
        "periods": 1,
        "period_to": 1761861599
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


SLI aprēķināšana mēneša SLA

Iegūstiet SLI datus par pakalpojumiem ar ID ”50”, ”60” un ”70”, kas ir saistīti ar SLA ar ID ”5”. Iegūstiet datus par trim periodiem, sākot no ”1635724800” (2021. gada 1. novembris 00:00:00 UTC). Tā kā SLA pārskata periods ir mēnesis, SLI dati tiek iegūti par šādiem trim mēnešiem:

- No ”1635724800” (2021. gada 1. novembris 00:00:00 UTC) līdz ”1638316800” (2021. gada 1. decembris 00:00:00 UTC)
- No ”1638316800” (2021. gada 1. decembris 00:00:00 UTC) līdz ”1640995200” (2022. gada 1. janvāris 00:00:00 UTC)
- No ”1640995200” (2022. gada 1. janvāris 00:00:00 UTC) līdz ”1643673600” (2022. gada 1. februāris 00:00:00 UTC)

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "sla.getsli",
    "params": {
        "slaid": "5",
        "serviceids": [
            50,
            60,
            70
        ],
        "periods": 3,
        "period_from": 1635724800
    },
    "id": 1
}
```


Atbilde:


```
{
    "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": []
                },
                {
                    "uptime": 1674000,
                    "downtime": 0,
                    "sli": 100,
                    "error_budget": 0,
                    "excluded_downtimes": []
                }
            ]
        ]
    },
    "id": 1
}
```


Avots

CSla::getSli() failā *ui/include/classes/api/services/CSla.php*

sla.update

Apraksts

`object sla.update(object/array slaids)`

Šī metode ļauj atjaunināt esošos SLA ierakstus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` atjaunināmie SLA rekvizīti.

Rekvizītam `slaid` jābūt definētam katram SLA, visi pārējie rekvizīti ir neobligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta SLA rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
service_tags	array	SLA pakalpojumu birkas, ar kurām aizstāt pašreizējās SLA pakalpojumu birkas.
schedule	array	SLA grafiks, ar ko aizstāt pašreizējo. Ja parametrs tiek norādīts kā tukšs, tas tiks interpretēts kā 24x7 grafiks.
excluded_downtimes	array	SLA izslēgtās dīkstāves, ar kurām aizstāt pašreizējās.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto SLA ID īpašumā `slaids`.

Piemēri

Pakalpojumu birku atjaunināšana

Padariet SLA ar ID ”5” aprēķināmu ikmēneša intervālos ar NoSQL saistītiem pakalpojumiem, nemainot tā grafiku vai izslēgtās dīkstāves; iestatiet SLO uz 95%.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA grafika maiņa

Pārslēdziet SLA ar ID ”5” uz 24x7 grafiku.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "sla.update",
    "params": {
        "slaid": "5",
        "schedule": []
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

SLA izslēgto dīkstāvju maiņa

Pievienojiet plānotu 4 stundas ilgu RAM jauninašanas dīkstāvi 2022. gada 6. aprīlī, vienlaikus saglabājot (to nepieciešams definēt no jauna) iepriekš esošu programmatūras jauninašanas plānu 4. jūlijā SLA ar ID ”5”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "slaids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CSla::update() failā *ui/include/classes/api/services/CSla.php*.

Starpniekserveris

Šī klase ir paredzēta darbam ar starpniekserveriem.

Objektu atsauces:

- Proxy

Pieejamās metodes:

- proxy.create - izveidot jaunus starpniekserverus
- proxy.delete - dzēst starpniekserverus
- proxy.get - izgūt starpniekserverus
- proxy.update - atjaunināt starpniekserverus

Starpniekservera objekts

Tālāk norādītie objekti ir tieši saistīti ar `proxy` API.

Starpniekserveris

Starpniekservera objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
proxyid	ID	Starpniekservera ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Starpniekservera nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
proxy_groupid	ID	Starpniekserveru grupas ID. 0 - starpniekserveris nav piešķirts nevienai starpniekserveru grupai.
local_address	string	Adrese aktīvajiem aģentiem. IP adrese vai DNS nosaukums, pie kura pieslēgties. Īpašības darbība: - *obligāts*, ja `proxy_groupid` nav 0

Īpašība	Tips	Apraksts
local_port	string	Lokālā starpniekservera porta numurs, pie kura pieslēgties. Tiek atbalstīti lietotāja makrosi. Noklusējums: 10051. Īpašības darbība: - *atbalstīts*, ja `proxy_groupid` nav 0
operating_mode	integer	Starpniekservera tips. Iespējamās vērtības: 0 - aktīvs starpniekserveris; 1 - pasīvs starpniekserveris. Īpašības darbība: - *obligāts* izveides operācijām
description	text	Starpniekservera apraksts.
lastaccess	timestamp	Laiks, kad starpniekserveris pēdējo reizi pieslēdzās serverim. Īpašības darbība: - *tikai lasāms*
address	string	IP adrese vai DNS nosaukums, pie kura pieslēgties. Tiek atbalstīti lietotāja makrosi. Īpašības darbība: - *obligāts*, ja Zabbix starpniekservera darbības režīms ir pasīvs
port	string	Porta numurs, pie kura pieslēgties. Tiek atbalstīti lietotāja makrosi. Noklusējums: 10051. Īpašības darbība: - *atbalstīts*, ja Zabbix starpniekservera darbības režīms ir pasīvs
allowed_addresses	string	Ar komatiem atdalītas aktīvā Zabbix starpniekservera IP adreses vai DNS nosaukumi.
tls_connect	integer	Savienojumi ar hostu. Iespējamās vērtības: 1 - *(noklusējums)* Bez šifrēšanas; 2 - PSK; 4 - sertifikāts.
tls_accept	integer	Savienojumi no hosta. Iespējamās bitkartes vērtības: 1 - *(noklusējums)* Bez šifrēšanas; 2 - PSK; 4 - sertifikāts. Šis ir bitmaskas lauks; ir pieņemama jebkura iespējamo bitkartes vērtību summa (piemēram, 6 PSK un sertifikātam).
tls_issuer	string	Sertifikāta izdevējs.
tls_subject	string	Sertifikāta subjekts.

Īpašība	Tips	Apraksts
tls_psk_identity	string	PSK identitāte; tai jābūt sapārotai tikai ar vienu PSK (starp autoregistration, hosts un proxies). Neiekļaujiet sensitīvu informāciju PSK identitātē, jo tā tiek nosūtīta nešifrēta pa tīklu, lai informētu saņēmēju, kuru PSK izmantot. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur ”PSK” bitu
tls_psk	string	Iepriekš koplietotā atslēga (PSK); tai jābūt vismaz 32 heksadecimālām zīmēm garai. Īpašības darbība: - *tikai rakstāms* - *obligāts*, ja `tls_connect` ir iestatīts uz ”PSK” vai `tls_accept` satur ”PSK” bitu
custom_timeouts	integer	Vai ignorēt globālos vienumu noildzes iestatījumus starpniekservera līmenī. Iespējamās vērtības: 0 - *(noklusējums)* izmantot globālos iestatījumus; 1 - ignorēt noildzes.
timeout_zabbix_agent	string	Apstrādei netērēt vairāk par `timeout_*` sekundēm. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: ””. Īpašības darbība: - *obligāts*, ja `custom_timeouts` ir iestatīts uz 1.
timeout_simple_check timeout_snmp_agent timeout_external_check timeout_db_monitor timeout_http_agent timeout_ssh_agent timeout_telnet_agent timeout_script timeout_browser		
version	integer	Starpniekservera versija. Trīsdaļīgs Zabbix versijas numurs, kur katrai daļai tiek izmantoti divi decimālcipari, piemēram, 60401 versijai 6.4.1, 70002 versijai 7.0.2 utt. 0 - Nezināma starpniekservera versija. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
compatibility	integer	Starpniekservera versija salīdzinājumā ar Zabbix servera versiju. Iespējamās vērtības: 0 - Nedefinēta; 1 - Pašreizējā versija (starpniekserverim un serverim ir viena un tā pati galvenā versija); 2 - Novecojusi versija (starpniekservera versija ir vecāka par servera versiju, bet tiek daļēji atbalstīta); 3 - Neatbalstīta versija (starpniekservera versija ir vecāka par servera iepriekšējā LTS laidiena versiju vai servera galvenā versija ir vecāka par starpniekservera galveno versiju). Īpašības darbība: - *tikai lasāms*
state	integer	Starpniekservera stāvoklis. Iespējamās vērtības: 0 - Nezināms; 1 - Bezsaistē; 2 - Tiešsaistē. Īpašības darbība: - *tikai lasāms*

proxy.create

Apraksts

`object proxy.create(object/array proxies)`

Šī metode ļauj izveidot jaunus starpniekserverus.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Starpniekserveri, ko izveidot.

Papildus standarta starpniekservera īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hosts	array	Hosti, kurus uzraudzīs starpniekserveris. Ja hosts jau tiek uzraudzīts ar citu starpniekserveri, tas tiks pārdalīts pašreizējam starpniekserverim. Hostiem jābūt definētai tikai `hostid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto starpniekserveru ID īpašībā `proxyids`. Atgriezto ID secība atbilst nodoto starpniekserveru secībai.

Piemēri

Izveidot aktīvu starpniekserveri

Izveidojiet aktīvu starpniekserveri "Active proxy" un piešķiriet tam hostu uzraudzībai.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "operating_mode": "0",
        "hosts": [
            {
                "hostid": "10279"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10280"
        ]
    },
    "id": 1
}
```

Izveidot pasīvu starpniekserveri

Izveidojiet pasīvu starpniekserveri ”Passive proxy” un piešķiriet tam divus hostus uzraudzībai.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10284"
        ]
    },
    "id": 1
}
```

Starpniekservera izveide un tā pievienošana starpniekserveru grupai

Izveidojiet aktīvu starpniekserveri "Active proxy" un pievienojiet to starpniekserveru grupai ar ID "1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "name": "Active proxy",
        "proxy_groupid": "1",
        "operating_mode": "0"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Host
- Starpniekserveru grupa

Avots

CProxy::create() failā *ui/include/classes/api/services/CProxy.php*.

proxy.delete

Apraksts

`object proxy.delete(array proxies)`

Šī metode ļauj dzēst starpniekserverus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo starpniekserveru ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto starpniekserveru ID īpašībā `proxyids`.

Piemēri

Dzēst vairākus starpniekserverus

Dzēsiet divus starpniekserverus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.delete",
    "params": [
        "10286",
        "10285"
    ],
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10286",
            "10285"
        ]
    },
    "id": 1
}
```

Avots

CProxy::delete() failā *ui/include/classes/api/services/CProxy.php*.

proxy.get

Apraksts

`integer/array proxy.get(object parameters)`

Metode ļauj izgūt starpniekserverus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
proxyids	ID/array	Atgriezt tikai starpniekserverus ar norādītajiem ID.
proxy_groupids	ID/array	Atgriezt tikai starpniekserverus, kas pieder norādītajām starpniekserveru grupām.
selectAssignedHosts	query	Atgriezt `assignedHosts` īpašību ar hostiem, kas piešķirti starpniekserverim. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kurus uzrauga starpniekserveris. Atbalsta `count`.
selectProxyGroup	query	Atgriezt `proxyGroup` īpašību ar starpniekserveru grupas objektu.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `proxyid`, `name`, `operating_mode`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	

Parametrs	Tips	Apraksts
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt visus starpniekserverus

Iegūst visus konfigurētos starpniekserverus un to saskarnes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Hosts
- Starpniekserveru grupa

Avots

CProxy::get() failā *ui/include/classes/api/services/CProxy.php*.

proxy.update

Apraksts

`object proxy.update(object/array proxies)`

Šī metode ļauj atjaunināt esošos starpniekserverus.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmie starpniekservera rekvizīti.

Katram starpniekserverim ir jābūt norādītam rekvizītam `proxyid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta starpniekservera rekvizītiem metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
hosts	array	Hosti, kurus uzrauga starpniekserveris. Ja hosts jau tiek uzraudzīts ar citu starpniekserveri, tas tiks pārpiešķirts pašreizējam starpniekserverim. Hostiem jābūt definētam tikai rekvizītam `hostid`.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto starpniekserveru ID īpašībā `proxyids`.

Piemēri

Mainīt starpniekservera uzraudzītos hostus

Atjauniniet starpniekserveri, lai tas uzraudzītu divus norādītos hostus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "hosts": [
            {
                "hostid": "10294"
            },
            {
                "hostid": "10295"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Mainīt starpniekservera statusu

Mainiet starpniekserveri uz aktīvu starpniekserveri un pārdēvējiet to par ”Active proxy”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.update",
    "params": {
        "proxyid": "10293",
        "name": "Active proxy",
        "operating_mode": "0"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "10293"
        ]
    },
    "id": 1
}
```

Pievienot starpniekserveri starpniekserveru grupai

Atjauniniet starpniekserveri ar ID ”5” un pievienojiet to starpniekserveru grupai ar ID ”1”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxy.create",
    "params": {
        "proxyid": "5",
        "proxy_groupid": "1",
        "local_address": "127.0.0.1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxyids": [
            "5"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Hosts
- Starpniekserveru grupa

Avots

CProxy::update() failā *ui/include/classes/api/services/CProxy.php*.

Starpniekserveru grupa

Šī klase ir paredzēta darbam ar starpniekserveru grupām.

Objektu atsauces:

- Starpniekserveru grupa

Pieejamās metodes:

- proxygroup.create - izveidot jaunas starpniekserveru grupas
- proxygroup.delete - dzēst starpniekserveru grupas
- proxygroup.get - izgūt starpniekserveru grupas
- proxygroup.update - atjaunināt starpniekserveru grupas

Starpniekserveru grupas objekts

Tālāk norādītie objekti ir tieši saistīti ar `proxygroup` API.

Starpniekserveru grupa

Starpniekserveru grupas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
proxy_groupid	ID	Starpniekserveru grupas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
name	string	Starpniekserveru grupas nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
description	text	Starpniekserveru grupas apraksts.
failover_delay	string	Periods, kura laikā starpniekserverim starpniekserveru grupā ir jāsazinās ar Zabbix serveris, lai tas tiktu uzskatīts par tiešsaistē esošu. Laika sufiksi tiek atbalstīti, piemēram, 30s, 1m. Lietotāja makrosi tiek atbalstīti. Iespējamās vērtības: 10s-15m. Noklusējums: 1m.
min_online	string	Minimālais tiešsaistē esošo starpniekserveru skaits, kas nepieciešams, lai starpniekserveru grupa paliktu tiešsaistē. Lietotāja makrosi tiek atbalstīti. Iespējamo vērtību diapazons: 1-1000. Noklusējums: 1.
state	integer	Starpniekserveru grupas stāvoklis. Iespējamās vērtības: 0 - Nezināms; 1 - Bezsaistē; 2 - Atkopjas; 3 - Tiešsaistē; 4 - Pasliktinās. Īpašības darbība: - *tikai lasāms*

proxygroup.create

Apraksts

`object proxygroup.create(object/array proxyGroups)`

Šī metode ļauj izveidot jaunas starpniekserveru grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Starpniekserveru grupas, ko izveidot.

Metode pieņem starpniekserveru grupas ar standarta starpniekserveru grupu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto starpniekserveru grupu ID īpašībā `proxy_groupids`. Atgriezto ID secība atbilst nodoto starpniekserveru grupu secībai.

Piemēri

Izveidot starpniekserveru grupu

Izveidojiet starpniekserveru grupu ar pielāgotiem iestatījumiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.create",
    "params": {
        "name": "Proxy group",
        "failover_delay": "5m",
        "min_online": "10"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CProxyGroup::create() failā *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.delete

Apraksts

`object proxygroup.delete(array proxyGroupIds)`

Šī metode ļauj dzēst starpniekserveru grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo starpniekserveru grupu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto starpniekserveru grupu ID īpašībā `proxy_groupids`.

Piemēri

Dzēst vairākas starpniekserveru grupas

Dzēst divas starpniekserveru grupas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.delete",
    "params": [
        "5",
        "10"
    ],
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5",
            "10"
        ]
    },
    "id": 1
}
```

Avots

CProxyGroup::delete() failā *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.get

Apraksts

`integer/array proxygroup.get(object parameters)`

Šī metode ļauj izgūt starpniekserveru grupas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā User roles.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
proxy_groupids	ID/array	Atgriezt tikai starpniekserveru grupas ar norādītajiem ID.
proxyids	ID/array	Atgriezt tikai tās starpniekserveru grupas, kas satur norādītos starpniekserverus.
selectProxies	query	Atgriezt `proxies` īpašību ar starpniekserveriem, kas pieder starpniekserveru grupai. Atbalsta `count`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `proxy_groupid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt visas starpniekserveru grupas

Iegūstiet visas konfigurētās starpniekserveru grupas ar starpniekserveriem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.get",
    "params": {
        "output": "extend",
        "selectProxies": ["proxyid", "name"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Starpniekserveris

Avots

CProxyGroup::get() failā *ui/include/classes/api/services/CProxyGroup.php*.

proxygroup.update

Apraksts

`object proxygroup.update(object/array proxyGroups)`

Šī metode ļauj atjaunināt esošās starpniekserveru grupas.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās starpniekserveru grupas īpašības.

Īpašībai `proxy_groupid` jābūt definētai katrai starpniekserveru grupai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Metode pieņem starpniekserveru grupas ar standarta starpniekserveru grupu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto starpniekserveru grupu ID īpašumā `proxy_groupids`.

Piemēri

Mainīt minimālo tiešsaistē esošo starpniekserveru skaitu

Mainīt minimālo tiešsaistē esošo starpniekserveru skaitu, kas nepieciešams, lai grupa būtu tiešsaistē.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "proxygroup.update",
    "params": {
        "proxy_groupid": "5",
        "min_online": "3"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "proxy_groupids": [
            "5"
        ]
    },
    "id": 1
}
```

Avots

CProxyGroup::update() failā *ui/include/classes/api/services/CProxyGroup.php*.

Token

Šī klase ir paredzēta darbam ar tokeniem.

Objektu atsauces:

- Token

Pieejamās metodes:

- token.create - izveidot jaunus tokenus
- token.delete - dzēst tokenus
- token.get - izgūt tokenus
- token.update - atjaunināt tokenus
- token.generate - ģenerēt tokenus

Tokena objekts

Tālāk norādītie objekti ir tieši saistīti ar `token` API.

Tokens

Tokena objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tokenid	ID	Tokena ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Tokena nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
description	text	Tokena apraksts.
userid	ID	Tā lietotāja ID, kuram tokens ir piešķirts. Noklusējums: *pašreizējais lietotājs*. Īpašības darbība: - *konstants*
lastaccess	timestamp	Pēdējais datums un laiks, kad tokens tika autentificēts. ”0”, ja tokens nekad nav ticis autentificēts. Īpašības darbība: - *tikai lasāms*
status	integer	Tokena statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots tokens; 1 - atspējots tokens.
expires_at	timestamp	Tokena derīguma termiņa beigu datums un laiks. ”0” tokeniem bez derīguma termiņa.
created_at	timestamp	Tokena izveides datums un laiks. Īpašības darbība: - *tikai lasāms*
creator_userid	ID	Tā lietotāja ID, kurš izveidoja tokenu. Īpašības darbība: - *tikai lasāms*

token.create

Apraksts

`object token.create(object/array tokens)`

Šī metode ļauj izveidot jaunus tokenus.

> **Note:**
> Lai lietotāja loma varētu pārvaldīt tokenus citiem lietotājiem, ir nepieciešama *Manage API tokens* atļauja.

> **Attention:**
> Ar šo metodi izveidots tokens pirms lietošanas ir arī jāģenerē.

Parametri

`(object/array)` Izveidojamie tokeni.

Metode pieņem tokenus ar standarta tokenu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto tokenu ID īpašībā `tokenids`. Atgriezto ID secība atbilst nodoto tokenu secībai.

Piemēri

Izveidot tokenu

Izveidojiet iespējotu tokenu, kam nekad nebeidzas derīguma termiņš un kas autentificē lietotāju ar ID 2.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "userid": "2"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188"
        ]
    },
    "id": 1
}
```

Izveidojiet atspējotu tokenu, kura derīguma termiņš beidzas 2021. gada 21. janvārī. Šis tokens autentificēs pašreizējo lietotāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "token.create",
    "params": {
        "name": "Your token",
        "status": "1",
        "expires_at": "1611238072"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "189"
        ]
    },
    "id": 1
}
```

Avots

CToken::create() failā *ui/include/classes/api/services/CToken.php*.

token.delete

Apraksts

`object token.delete(array tokenids)`

Šī metode ļauj dzēst tokenus.

> **Note:**
> Lai lietotāja loma varētu pārvaldīt citu lietotāju tokenus, ir nepieciešama *Manage API tokens* atļauja.

Parametri

`(array)` dzēšamo tokenu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto tokenu ID zem `tokenids` īpašības.

Piemēri

Dzēst vairākus tokenus

Dzēst divus tokenus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "token.delete",
    "params": [
        "188",
        "192"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "tokenids": [
            "188",
            "192"
        ]
    },
    "id": 1
}
```

Avots

CToken::delete() failā *ui/include/classes/api/services/CToken.php*.

token.generate

Apraksts

`object token.generate(array tokenids)`

Šī metode ļauj ģenerēt tokenus.

> **Note:**
> Lietotāja lomai ir nepieciešama *Manage API tokens* atļauja, lai pārvaldītu tokenus citiem lietotājiem.

> **Attention:**
> Tokenu ar šo metodi var ģenerēt tikai tad, ja tas ir izveidots.

Parametri

`(array)` ģenerējamo tokenu ID.

Atgrieztās vērtības

`(array)` Atgriež objektu masīvu, kas satur ģenerētā marķiera ID īpašībā `tokenid` un ģenerēto autorizācijas virkni īpašībā `token`.

Īpašība	Type	Apraksts
tokenid	ID	Marķiera ID.
token	string	Šim marķierim ģenerētā autorizācijas virkne.

Piemēri

Ģenerēt vairākus tokenus

Ģenerējiet divus tokenus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "token.generate",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "tokenid": "1",
            "token": "bbcfce79a2d95037502f7e9a534906d3466c9a1484beb6ea0f4e7be28e8b8ce2"
        },
        {
            "tokenid": "2",
            "token": "fa1258a83d518eabd87698a96bd7f07e5a6ae8aeb8463cae33d50b91dd21bd6d"
        }
    ],
    "id": 1
}
```

Avots

CToken::generate() failā *ui/include/classes/api/services/CToken.php*.

token.get

Apraksts

`integer/array token.get(object parameters)`

Metode ļauj izgūt tokenus atbilstoši norādītajiem parametriem.

> **Note:**
> Tikai *Super admin* lietotāja tipam ir atļauts skatīt citu lietotāju tokenus.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
tokenids	ID/array	Atgriezt tikai tokenus ar norādītajiem ID.
userids	ID/array	Atgriezt tikai tokenus, kas izveidoti norādītajiem lietotājiem.
token	string	Atgriezt tikai tokenus, kas izveidoti norādītajam *Auth token*.
valid_at	timestamp	Atgriezt tikai tokenus, kas ir derīgi (nav beidzies derīguma termiņš) norādītajā datumā un laikā.
expired_at	timestamp	Atgriezt tikai tokenus, kuriem ir beidzies derīguma termiņš (nav derīgi) norādītajā datumā un laikā.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `tokenid`, `name`, `lastaccess`, `status`, `expires_at`, `created_at`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt tokenu

Iegūt visus datus par tokenu ar ID "2".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "token.get",
    "params": {
        "output": "extend",
        "tokenids": "2"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Avots

CToken::get() failā *ui/include/classes/api/services/CToken.php*.

token.update

Apraksts

`object token.update(object/array tokens)`

Šī metode ļauj atjaunināt esošos tokenus.

> **Note:**
> Lai lietotāja loma varētu pārvaldīt tokenus citiem lietotājiem, ir nepieciešama *Manage API tokens* atļauja.

Parametri

`(object/array)` Atjaunināmo tokenu īpašības.

Īpašībai `tokenid` jābūt definētai katram tokenam, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Metode pieņem tokenus ar standarta tokenu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto tokenu ID zem `tokenids` īpašības.

Piemēri

Noņemt pilnvaras termiņa beigas

Noņemt pilnvarai derīguma termiņa datumu.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "token.update",
    "params": {
        "tokenid": "2",
        "expires_at": "0"
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "tokenids": [
            "2"
        ]
    },
    "id": 1
}
```

Avots

CToken::update() failā *ui/include/classes/api/services/CToken.php*.

Trends

Šī klase ir paredzēta darbam ar tendenču datiem.

Objektu atsauces:

- Peldošā komata tendence
- Veselā skaitļa tendence

Pieejamās metodes:

- trend.get - izgūt tendences

Tendenču objekts

Tālāk norādītie objekti ir tieši saistīti ar `trend` API.

Note:
Tendenču objekti atšķiras atkarībā no vienuma informācijas tipa. Tos izveido Zabbix serveris, un tos nevar modificēt, izmantojot API.

Peldošā komata trends

Peldošā komata trenda objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Tās stundas laikspiedols, kurai tika aprēķināta vērtība. Piemēram, laikspiedols ”04:00:00” nozīmē vērtības, kas aprēķinātas periodam ”04:00:00-04:59:59”.
itemid	ID	Saistītā vienuma ID.
num	integer	Stundai pieejamo vērtību skaits.
value_min	float	Stundas minimālā vērtība.
value_avg	float	Stundas vidējā vērtība.
value_max	float	Stundas maksimālā vērtība.

Veselu skaitļu trends

Veselu skaitļu trenda objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Tās stundas laikspiedols, kurai tika aprēķināta vērtība. Piemēram, laikspiedols ”04:00:00” nozīmē vērtības, kas aprēķinātas periodam ”04:00:00-04:59:59”.
itemid	ID	Saistītā vienuma ID.
num	integer	Stundai pieejamo vērtību skaits.
value_min	integer	Stundas minimālā vērtība.
value_avg	integer	Stundas vidējā vērtība.
value_max	integer	Stundas maksimālā vērtība.

trend.get

Apraksts

`integer/array trend.get(object parameters)`

Šī metode ļauj izgūt tendenču datus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
itemids	ID/masīvs	Atgriezt tikai trendus ar norādītajiem vienumu ID.
time_from	timestamp	Atgriezt tikai vērtības, kas ir savāktas pēc norādītā laika vai tajā.
time_till	timestamp	Atgriezt tikai vērtības, kas ir savāktas pirms norādītā laika vai tajā.
countOutput	boolean	Skaitīt iegūto objektu skaitu.
limit	integer	Ierobežot iegūto objektu skaitu.
output	query	Iestatīt atgriežamās Trend object īpašības.

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Vienuma tendenču datu izgūšana

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trend.get",
    "params": {
        "output": [
            "itemid",
            "clock",
            "num",
            "value_min",
            "value_avg",
            "value_max"
        ],
        "itemids": [
            "23715"
        ],
        "limit": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23715",
            "clock": "1446199200",
```

```
            "num": "60",
            "value_min": "0.165",
            "value_avg": "0.2168",
            "value_max": "0.35"
        }
    ],
    "id": 1
}
```

Avots

CTrend::get() failā *ui/include/classes/api/services/CTrend.php*.

Trigera prototips

Šī klase ir paredzēta darbam ar trigera prototipiem.

Objektu atsauces:

- Trigera prototips
- Trigera prototipa tags

Pieejamās metodes:

- triggerprototype.create - izveidot jaunus trigera prototipus
- triggerprototype.delete - dzēst trigera prototipus
- triggerprototype.get - izgūt trigera prototipus
- triggerprototype.update - atjaunināt trigera prototipus

Trigera prototipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `triggerprototype` API.

Trigera prototips

Trigera prototipa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
triggerid	ID	Trigera prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
description	string	Trigera prototipa nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
expression	string	Trigera izteiksme. Jāsatur vismaz viens vienuma prototips. Vienumu prototipus ar `value_type`, kas iestatīts uz 5 (binārs) vai 6 (JSON), nevar izmantot trigera izteiksmēs. Īpašības darbība: - *obligāts* izveides operācijām
event_name	string	Trigera ģenerētā notikuma nosaukums.
opdata	string	Operatīvie dati.
comments	string	Papildu komentāri trigera prototipam.

Īpašība	Tips	Apraksts
priority	integer	Trigera prototipa nozīmīgums. Iespējamās vērtības: 0 - *(noklusējums)* nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa.
status	integer	Vai trigera prototips ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
flags	integer	Trigera prototipa izcelsme. Iespējamās vērtības: 2 - trigera prototips; 6 - atklāts trigera prototips. Īpašības darbība: - *tikai lasāms*
templateid	ID	Vecākās veidnes trigera prototipa ID. Īpašības darbība: - *tikai lasāms*
type	integer	Vai trigera prototips var ģenerēt vairākus problēmu notikumus. Iespējamās vērtības: 0 - *(noklusējums)* neģenerēt vairākus notikumus; 1 - ģenerēt vairākus notikumus.
url	string	Ar trigera prototipu saistītais URL.
url_name	string	Etiķete URL, kas saistīts ar trigera prototipu.
recovery_mode	integer	OK notikuma ģenerēšanas režīms. Iespējamās vērtības: 0 - *(noklusējums)* izteiksme; 1 - atkopšanas izteiksme; 2 - nav.
recovery_expression	string	Trigera atkopšanas izteiksme. Jāsatur vismaz viens vienuma prototips. Vienumu prototipus ar `value_type`, kas iestatīts uz 5 (binārs) vai 6 (JSON), nevar izmantot trigera atkopšanas izteiksmēs.
correlation_mode	integer	OK notikums aizver. Iespējamās vērtības: 0 - *(noklusējums)* visas problēmas; 1 - visas problēmas, ja atbilst birku vērtības.
correlation_tag	string	Birka atbilstības noteikšanai.
manual_close	integer	Atļaut manuālu aizvēršanu. Iespējamās vērtības: 0 - *(noklusējums)* nē; 1 - jā.
discover	integer	Trigera prototipa atklāšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* tiks atklāti jauni trigeri; 1 - jauni trigeri netiks atklāti, un esošie trigeri tiks atzīmēti kā zaudēti.

Īpašība	Tips	Apraksts
uuid	string	Universāls unikāls identifikators, ko izmanto importēto trigera prototipu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja trigera prototips pieder veidnei

Trigera prototipa tags

Trigera prototipa taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Trigera prototipa taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Trigera prototipa taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne; 1 - hosts; 3 - vienums; 4 - vienuma prototips. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

triggerprototype.create

Apraksts

`object triggerprototype.create(object/array triggerPrototypes)`

Šī metode ļauj izveidot jaunus trigeru prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie trigeru prototipi.

Papildus standarta trigeru prototipu īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
dependencies	array	Trigeri un trigeru prototipi, no kuriem ir atkarīgs trigeru prototips. Trigeriem jābūt definētai tikai `triggerid` īpašībai.
tags	array	Trigeru prototipu birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto trigeru prototipu ID īpašumā `triggerids`. Atgriezto ID secība atbilst nodoto trigeru prototipu secībai.

Piemēri

Trigera prototipa izveide

Izveidojiet trigera prototipu, lai noteiktu, kad failu sistēmā ir mazāk nekā 20% brīvas diska vietas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.create",
    "params": {
        "description": "Brīvā diska vieta ir mazāka par 20% sējumā {#FSNAME}",
        "expression": "last(/Zabbix server/vfs.fs.size[{#FSNAME},pfree])<20",
        "tags": [
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17372"
        ]
    },
    "id": 1
}
```

Avots

CTriggerPrototype::create() failā *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.delete

Apraksts

`object triggerprototype.delete(array triggerPrototypeIds)`

Šī metode ļauj dzēst trigeru prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Dzēšamo trigeru prototipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto trigeru prototipu ID zem `triggerids` īpašības.

Piemēri

Vairāku trigeru prototipu dzēšana

Dzēsiet divus trigeru prototipus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Avots

CTriggerPrototype::delete() failā *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.get

Apraksts

`integer/array triggerprototype.get(object parameters)`

Metode ļauj izgūt trigeru prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
active	flag	Atgriezt tikai iespējotus trigeru prototipus, kas pieder uzraudzītiem hostiem.
discoveryids	ID/array	Atgriezt tikai trigeru prototipus, kas pieder norādītajiem LLD noteikumiem.
functions	string/array	Atgriezt tikai trigerus, kas izmanto norādītās funkcijas. Atbalstīto funkciju sarakstu skatiet lapā Atbalstītās funkcijas.
group	string	Atgriezt tikai trigeru prototipus, kas pieder hostiem vai veidnēm no hostu grupām vai veidņu grupām ar norādīto nosaukumu.
groupids	ID/array	Atgriezt tikai trigeru prototipus, kas pieder hostiem vai veidnēm no norādītajām hostu grupām vai veidņu grupām.
host	string	Atgriezt tikai trigeru prototipus, kas pieder hostiem ar norādīto nosaukumu.
hostids	ID/array	Atgriezt tikai trigeru prototipus, kas pieder norādītajiem hostiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotos trigeru prototipus.

Parametrs	Tips	Apraksts
maintenance	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotus trigeru prototipus, kas pieder hostiem uzturēšanas režīmā.
min_severity	integer	Atgriezt tikai trigeru prototipus, kuru nopietnības līmenis ir lielāks vai vienāds ar norādīto nopietnības līmeni.
monitored	flag	Atgriezt tikai iespējotus trigeru prototipus, kas pieder uzraudzītiem hostiem un satur tikai iespējotus vienumus.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai trigeru prototipus, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai trigeru prototipus, kas pieder norādītajām veidnēm.
triggerids	ID/array	Atgriezt tikai trigeru prototipus ar norādītajiem ID.
expandExpression	flag	Izvērst funkcijas un makrosus trigera izteiksmē.
selectDependencies	query	Atgriezt trigeru prototipus un trigerus, no kuriem trigeru prototips ir atkarīgs, īpašībā `dependencies`.
selectDiscoveryData	query	Atgriezt īpašību `discoveryData` ar trigera prototipa atklāšanas objekta datiem. Trigera prototipa atklāšanas objekts sasaista atklātu trigera prototipu ar trigera prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_triggerid` - `(ID)` tā trigera prototipa ID, no kura trigera prototips ir izveidots; `status` - `(int)` trigera prototipa atklāšanas statuss: 0 - *(noklusējums)* trigera prototips ir atklāts, 1 - trigera prototips vairs netiek atklāts; `ts_delete` - `(timestamp)` laiks, kad trigera prototips, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad trigera prototips, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` norāde, vai trigera prototips tika atspējots ar LLD noteikumu vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD noteikumu.
selectDiscoveryRule	query	Atgriezt LLD noteikumu, kuram pieder trigera prototips, īpašībā `discoveryRule`.
selectDiscoveryRulePrototype	query	Atgriezt īpašību `discoveryRulePrototype` ar vecāka LLD noteikuma prototipu, kuram pieder trigera prototips.
selectFunctions	query	Atgriezt trigera prototipā izmantotās funkcijas īpašībā `functions`. Funkciju objekti attēlo trigera izteiksmē izmantotās funkcijas, un tiem ir šādas īpašības: `functionid` - `(ID)` funkcijas ID; `itemid` - `(ID)` funkcijā izmantotā vienuma ID; `function` - `(string)` funkcijas nosaukums; `parameter` - `(string)` funkcijai nodotais parametrs. Vaicājuma parametrs atgrieztajā virknē tiek aizstāts ar simbolu $.
selectHostGroups	query	Atgriezt hostu grupas, kurām pieder trigera prototips, īpašībā `hostgroups`.
selectHosts	query	Atgriezt hostus, kuriem pieder trigera prototips, īpašībā `hosts`.
selectInheritedTags	query	Atgriezt īpašību `inheritedTags` ar birkām, kas mantotas no veidnēm, piesaistītajām veidnēm un hostiem, kā arī no vienumu prototipiem, uz kuriem ir atsauce trigera vai atkopšanas izteiksmēs.
selectItems	query	Atgriezt vienumus un vienumu prototipus, ko izmanto trigera prototips, īpašībā `items`.
selectTags	query	Atgriezt trigera prototipa birkas īpašībā `tags`.
selectTemplateGroups	query	Atgriezt veidņu grupas, kurām pieder trigera prototips, īpašībā `templategroups`.

Parametrs	Tips	Apraksts
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - tā hosta tehniskais nosaukums, kuram pieder trigera prototips; `hostid` - tā hosta ID, kuram pieder trigera prototips.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectHosts` - rezultāti tiks kārtoti pēc `host`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `triggerid`, `description`, `status`, `priority`, `discovered`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Iegūt trigeru prototipus no LLD kārtulas

Iegūstiet visus trigeru prototipus un to funkcijas no LLD kārtulas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": "extend",
        "selectFunctions": "extend",
        "discoveryids": "22450"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13272",
            "expression": "{12598}<20",
            "description": "Brīvo inode skaits ir mazāks par 20% sējumā {#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": "$",
                    "function": "last"
                }
            ]
        },
        {
            "triggerid": "13266",
            "expression": "{13500}<20",
            "description": "Brīvā diska vieta ir mazāka par 20% sējumā {#FSNAME}",
            "url": "",
            "status": "0",
            "value": "0",
            "priority": "2",
            "lastchange": "0",
            "comments": "",
            "error": "",
            "templateid": "0",
            "type": "0",
            "state": "0",
            "flags": "2",
            "recovery_mode": "0",
            "recovery_expression": "",
            "correlation_mode": "0",
            "correlation_tag": "",
            "manual_close": "0",
            "opdata": "",
            "discover": "0",
            "event_name": "",
            "uuid": "74a1fc62bfe24b7eabe4e244c70dc384",
            "url_name": "",
            "functions": [
                {
                    "functionid": "13500",
                    "itemid": "22686",
                    "triggerid": "13266",
                    "parameter": "$",
```

```
                    "function": "last"
                }
            ]
        }
    ],
    "id": 1
}
```

Konkrēta trigera prototipa izgūšana ar tagiem

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.get",
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "17373",
            "description": "Brīvā diska vieta ir mazāka par 20% sējumā {#FSNAME}",
            "tags": [
                {
                    "tag": "volume",
                    "value": "{#FSNAME}"
                },
                {
                    "tag": "type",
                    "value": "{#FSTYPE}"
                }
            ]
        }
    ],
    "id": 1
}
```

Skatīt arī

- Atklāšanas kārtula
- Vienums
- Hosts
- Hostu grupa
- Veidņu grupa

Avots

CTriggerPrototype::get() failā *ui/include/classes/api/services/CTriggerPrototype.php*.

triggerprototype.update

Apraksts

`object triggerprototype.update(object/array triggerPrototypes)`

Šī metode ļauj atjaunināt esošos trigeru prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Papildinformāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo trigeru prototipu rekvizīti.

Katram trigeru prototipam jābūt definētam rekvizītam `triggerid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta trigeru prototipu rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
dependencies	array	Trigeri un trigeru prototipi, no kuriem trigeru prototips ir atkarīgs. Trigeriem jābūt definētam tikai rekvizītam `triggerid`.
tags	array	Trigeru prototipu birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto trigeru prototipu ID īpašumā `triggerids`.

Piemēri

Trigera prototipa iespējošana

Iespējojiet trigera prototipu, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Trigera prototipa tagu aizstāšana

Aizstāj tagus vienam trigera prototipam.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "triggerprototype.update",
    "params": {
        "triggerid": "17373",
        "tags": [
```

```
            {
                "tag": "volume",
                "value": "{#FSNAME}"
            },
            {
                "tag": "type",
                "value": "{#FSTYPE}"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17373"
        ]
    },
    "id": 1
}
```

Avots

CTriggerPrototype::update() failā *ui/include/classes/api/services/CTriggerPrototype.php*.

Trigeris

Šī klase ir paredzēta darbam ar trigeriem.

Objektu atsauces:

- Trigeris
- Trigera tags

Pieejamās metodes:

- trigger.create - izveidot jaunus trigerus
- trigger.delete - dzēst trigerus
- trigger.get - izgūt trigerus
- trigger.update - atjaunināt trigerus

Trigera objekts

Tālāk norādītie objekti ir tieši saistīti ar `trigger` API.

Trigeris

Trigera objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
triggerid	ID	Trigera ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
description	string	Trigera nosaukums. Īpašības darbība: - *obligāts* izveides darbībām

Īpašība	Tips	Apraksts
expression	string	Trigera izteiksme. Vienumus, kuriem `value_type` ir iestatīts uz 5 (binārs) vai 6 (JSON), nevar izmantot trigeru izteiksmēs. Īpašības darbība: - *obligāts* izveides darbībām
event_name	string	Trigera ģenerētā notikuma nosaukums.
opdata	string	Operatīvie dati.
comments	string	Trigera papildu apraksts.
error	string	Kļūdas teksts, ja, atjauninot trigera stāvokli, ir radušās kādas problēmas. Īpašības darbība: - *tikai lasāms*
flags	integer	Trigera izcelsme. Iespējamās vērtības: 0 - *(noklusējums)* parasts trigeris; 4 - no prototipa pārveidots trigeris. Īpašības darbība: - *tikai lasāms*
lastchange	timestamp	Laiks, kad trigeris pēdējo reizi mainīja savu stāvokli. Īpašības darbība: - *tikai lasāms*
priority	integer	Trigera būtiskums. Iespējamās vērtības: 0 - *(noklusējums)* nav klasificēts; 1 - informācija; 2 - brīdinājums; 3 - vidējs; 4 - augsts; 5 - katastrofa.
state	integer	Trigera izteiksmes stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* trigera stāvoklis ir aktuāls; 1 - pašreizējais trigera stāvoklis nav zināms. Īpašības darbība: - *tikai lasāms*
status	integer	Vai trigeris ir iespējots vai atspējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.
templateid	ID	Vecākveidnes trigera ID. Īpašības darbība: - *tikai lasāms*
type	integer	Vai trigeris var ģenerēt vairākus problēmu notikumus. Iespējamās vērtības: 0 - *(noklusējums)* neģenerēt vairākus notikumus; 1 - ģenerēt vairākus notikumus.
url	string	Ar trigeri saistītais URL.
url_name	string	Ar trigeri saistītā URL etiķete.

Īpašība	Tips	Apraksts
value	integer	Vai trigeris ir OK vai problēmas stāvoklī. Iespējamās vērtības: 0 - *(noklusējums)* OK; 1 - problēma. Īpašības darbība: - *tikai lasāms*
recovery_mode	integer	OK notikuma ģenerēšanas režīms. Iespējamās vērtības: 0 - *(noklusējums)* Izteiksme; 1 - Atkopšanas izteiksme; 2 - Nav.
recovery_expression	string	Trigera atkopšanas izteiksme. Vienumus, kuriem `value_type` ir iestatīts uz 5 (binārs) vai 6 (JSON), nevar izmantot trigera atkopšanas izteiksmēs.
correlation_mode	integer	OK notikums aizver. Iespējamās vērtības: 0 - *(noklusējums)* visas problēmas; 1 - visas problēmas, ja atbilst birku vērtības.
correlation_tag	string	Birka atbilstības noteikšanai.
manual_close	integer	Atļaut manuālu aizvēršanu. Iespējamās vērtības: 0 - *(noklusējums)* Nē; 1 - Jā.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto trigeru sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja trigeris pieder veidnei

Trigera tags

Trigera taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Trigera taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Trigera taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne; 1 - hosts; 3 - vienums. Īpašības darbība: - *tikai lasāms*.

Īpašība	Tips	Apraksts
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.
automatic	integer	Trigera taga tips: Iespējamās vērtības: 0 - *(noklusējums)* tags, ko izveidojis lietotājs; 1 - tags, ko izveidojis LLD;Īpašības darbība: - *tikai lasāms*.

trigeris.update

Apraksts

`object trigger.update(object/array triggers)`

Šī metode ļauj atjaunināt esošos trigerus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmie trigeru rekvizīti.

Katram trigerim ir jābūt norādītam rekvizītam `triggerid`, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta trigera rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
dependencies	array	Trigeri, no kuriem trigeris ir atkarīgs. Trigeriem jābūt definētam tikai rekvizītam `triggerid`.
tags	array	Trigera birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto trigeru ID zem `triggerids` īpašības.

Piemēri

Trigera iespējošana

Iespējojiet trigeri, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "13938",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Trigera tagu aizstāšana

Aizstāt tagus trigerim.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "13938"
        ]
    },
    "id": 1
}
```

Atkarību aizstāšana

Aizstājiet trigera atkarības.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.update",
    "params": {
        "triggerid": "22713",
        "dependencies": [
            {
                "triggerid": "22712"
            },
            {
                "triggerid": "22772"
            }
        ]
    },
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "22713"
        ]
    },
    "id": 1
}
```

Avots

CTrigger::update() failā *ui/include/classes/api/services/CTrigger.php*.

trigger.create

Apraksts

`object trigger.create(object/array triggers)`

Šī metode ļauj izveidot jaunus trigerus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Trigeri, ko izveidot.

Papildus standarta trigeru īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
dependencies	array	Trigeri, no kuriem trigeris ir atkarīgs.
tags	array	Trigeriem jābūt definētai tikai `triggerid` īpašībai. Trigeru birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto trigeru ID īpašībā `triggerids`. Atgriezto ID secība atbilst nodoto trigeru secībai.

Piemēri

Trigera izveide

Izveidojiet divus trigerus, no kuriem katrs ir atkarīgs no viena cita trigera.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.create",
    "params": [
        {
            "description": "Procesora slodze ir pārāk augsta uz {HOST.NAME}",
            "expression": "last(/Linux server/system.cpu.load[percpu,avg1])>5",
            "dependencies": [
                {
                    "triggerid": "17367"
                }
            ]
```

```
        },
        {
            "description": "Pakalpojums ir apstājies",
            "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "17369",
            "17370"
        ]
    },
    "id": 1
}
```

Avots

CTrigger::create() failā *ui/include/classes/api/services/CTrigger.php*.

trigger.delete

Apraksts

`object trigger.delete(array triggerIds)`

Šī metode ļauj dzēst trigerus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo trigeru ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto trigeru ID zem `triggerids` īpašības.

Piemēri

Dzēst vairākus trigerus

Dzēst divus trigerus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.delete",
    "params": [
        "12002",
        "12003"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "triggerids": [
            "12002",
            "12003"
        ]
    },
    "id": 1
}
```

Avots

CTrigger::delete() failā *ui/include/classes/api/services/CTrigger.php*.

trigger.get

Apraksts

`integer/array trigger.get(object parameters)`

Metode ļauj izgūt trigerus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
triggerids	ID/array	Atgriezt tikai trigerus ar norādītajiem ID.
groupids	ID/array	Atgriezt tikai trigerus, kas pieder hostiem vai veidnēm no norādītajām hostu grupām vai veidņu grupām.
templateids	ID/array	Atgriezt tikai trigerus, kas pieder norādītajām veidnēm.
hostids	ID/array	Atgriezt tikai trigerus, kas pieder norādītajiem hostiem.
itemids	ID/array	Atgriezt tikai trigerus, kas satur norādītos vienumus.
functions	string/array	Atgriezt tikai trigerus, kas izmanto norādītās funkcijas. Atbalstīto funkciju sarakstu skatiet lapā supported function.
group	string	Atgriezt tikai trigerus, kas pieder hostiem vai veidnēm no hostu grupas vai veidņu grupas ar norādīto nosaukumu.
host	string	Atgriezt tikai trigerus, kas pieder hostam ar norādīto tehnisko nosaukumu.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotos trigerus.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai trigerus, kas pieder veidnēm.
dependent	boolean	Ja iestatīts uz `true`, atgriezt tikai trigerus, kuriem ir atkarības. Ja iestatīts uz `false`, atgriezt tikai trigerus, kuriem nav atkarību.
monitored	flag	Atgriezt tikai iespējotus trigerus, kas pieder uzraudzītiem hostiem un satur tikai iespējotus vienumus.

Parametrs	Tips	Apraksts
active	flag	Atgriezt tikai iespējotus trigerus, kas pieder uzraudzītiem hostiem.
maintenance	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotus trigerus, kas pieder hostiem apkopes režīmā.
withUnacknowledgedEvents	flag	Atgriezt tikai trigerus, kuriem ir neapstiprināti notikumi.
withAcknowledgedEvents	flag	Atgriezt tikai trigerus, kuriem visi notikumi ir apstiprināti.
withLastEventUnacknowledged	flag	Atgriezt tikai trigerus, kuriem pēdējais notikums nav apstiprināts.
skipDependent	flag	Izlaist trigerus problēmas stāvoklī, kas ir atkarīgi no citiem trigeriem. Ņemiet vērā, ka citi trigeri tiek ignorēti, ja tie ir atspējoti, tiem ir atspējoti vienumi vai atspējotu vienumu hosti.
lastChangeSince	timestamp	Atgriezt tikai trigerus, kas mainījuši savu stāvokli pēc norādītā laika.
lastChangeTill	timestamp	Atgriezt tikai trigerus, kas mainījuši savu stāvokli pirms norādītā laika.
only_true	flag	Atgriezt tikai trigerus, kas nesen bijuši problēmas stāvoklī.
min_severity	integer	Atgriezt tikai trigerus, kuru nopietnības līmenis ir lielāks vai vienāds ar norādīto.
evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* And/Or; 2 - Or.
tags	array	Atgriezt tikai trigerus ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus trigerus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
expandComment	flag	Izvērst makrosus trigera aprakstā.
expandDescription	flag	Izvērst makrosus trigera nosaukumā.
expandExpression	flag	Izvērst funkcijas un makrosus trigera izteiksmē.
selectHostGroups	query	Atgriezt hostu grupas, kurām trigeris pieder, īpašībā `hostgroups`.
selectHosts	query	Atgriezt hostus, kuriem trigeris pieder, īpašībā `hosts`.
selectItems	query	Atgriezt trigerī iekļautos vienumus īpašībā `items`.
selectFunctions	query	Atgriezt trigerī izmantotās funkcijas īpašībā `functions`. Funkciju objekti attēlo trigera izteiksmē izmantotās funkcijas, un tiem ir šādas īpašības: `functionid` - `(ID)` funkcijas ID; `itemid` - `(ID)` funkcijā izmantotā vienuma ID; `function` - `(string)` funkcijas nosaukums; `parameter` - `(string)` funkcijai nodotais parametrs. Vaicājuma parametrs atgrieztajā virknē tiek aizstāts ar simbolu $.
selectDependencies	query	Atgriezt trigerus, no kuriem trigeris ir atkarīgs, īpašībā `dependencies`.

Parametrs	Tips	Apraksts
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar trigera atklāšanas objekta datiem. Trigera atklāšanas objekts sasaista atklātu trigeri ar trigera prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_triggerid` - `(ID)` trigera prototipa ID, no kura trigeris ir izveidots; `status` - `(int)` trigera atklāšanas statuss: 0 - *(noklusējums)* trigeris ir atklāts, 1 - trigeris vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad trigeris, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad trigeris, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` indikators, vai trigeri atspējoja LLD kārtula vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD kārtulu.
selectDiscoveryRule	query	Atgriezt zema līmeņa atklāšanas kārtulu, kas izveidoja trigeri, īpašībā `discoveryRule`.
selectLastEvent	query	Atgriezt pēdējo nozīmīgo trigera notikumu īpašībā `lastEvent`.
selectTags	query	Atgriezt trigera birkas īpašībā `tags`.
inheritedTags	boolean	Atgriezt trigerus, kuriem norādītās `tags` ir arī veidnē/hostā/piesaistītajās veidnēs. Iespējamās vērtības: `true` - veidnei/hostam/piesaistītajām veidnēm arī jābūt norādītajām birkām; `false` - *(noklusējums)* birkas no veidnes/hosta/piesaistītajām veidnēm tiek ignorētas.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas mantotas no veidnēm, piesaistītajām veidnēm un hostiem, kā arī no vienumiem, uz kuriem atsaucas trigera vai atkopšanas izteiksmes.
selectTemplateGroups	query	Atgriezt veidņu grupas, kurām trigeris pieder, īpašībā `templategroups`.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta, kuram trigeris pieder, tehniskais nosaukums; `hostid` - hosta, kuram trigeris pieder, ID.
limitSelects	integer	Ierobežo apakšvaicājumu atgriezto ierakstu skaitu. Attiecas uz šādiem apakšvaicājumiem: `selectHosts` - rezultāti tiks kārtoti pēc `host`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `triggerid`, `description`, `status`, `priority`, `lastchange`, `hostname`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentāros.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	

Parametrs	Tips	Apraksts
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectTriggerDiscovery	query	Atgriezt trigera atklāšanas objektu īpašībā `triggerDiscovery`. Trigera atklāšanas objekti sasaista trigeri ar trigera prototipu, no kura tas tika izveidots. Šis vaicājums ir **novecojis**, lūdzu, tā vietā izmantojiet `selectDiscoveryData`.

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Datu izgūšana pēc trigera ID

Izgūstiet visus datus un funkcijas, kas tiek izmantotas trigerī "14062".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "triggerids": "14062",
        "output": "extend",
        "selectFunctions": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Trigeru izgūšana problēmas stāvoklī

Izgūstiet visu trigeru ID, nosaukumu un svarīguma līmeni, kas ir problēmas stāvoklī, un kārtojiet tos pēc svarīguma līmeņa dilstošā secībā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
    "params": {
        "output": [
            "triggerid",
            "description",
            "priority"
        ],
        "filter": {
            "value": 1
        },
        "sortfield": "priority",
        "sortorder": "DESC"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "triggerid": "13907",
            "description": "Zabbix pašuzraudzības procesi ir aizņemti mazāk nekā 100%",
            "priority": "4"
        },
        {
            "triggerid": "13824",
            "description": "Zabbix atklāšanas procesi ir aizņemti vairāk nekā 75%",
            "priority": "3"
        }
    ],
    "id": 1
}
```

Konkrēta trigera ar tagiem izgūšana

Izgūstiet konkrētu trigeri ar tagiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "trigger.get",
```

```
    "params": {
        "output": [
            "triggerid",
            "description"
        ],
        "selectTags": "extend",
        "triggerids": [
            "17578"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatiet arī

- Atklāšanas kārtula
- Vienums
- Hosts
- Hostu grupa
- Veidņu grupa

Avots

CTrigger::get() failā *ui/include/classes/api/services/CTrigger.php*.

Tīmekļa scenārijs

Šī klase ir paredzēta darbam ar tīmekļa scenārijiem.

Objektu atsauces:

- Tīmekļa scenārijs
- Scenārija solis
  - HTTP lauks
- Tīmekļa scenārija tags

Pieejamās metodes:

- httptest.create - izveidot jaunus tīmekļa scenārijus
- httptest.delete - dzēst tīmekļa scenārijus
- httptest.get - izgūt tīmekļa scenārijus
- httptest.update - atjaunināt tīmekļa scenārijus

Tīmekļa scenārija objekts

Tālāk norādītie objekti ir tieši saistīti ar `webcheck` API.

Tīmekļa scenārijs

Tīmekļa scenārija objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
httptestid	ID	Tīmekļa scenārija ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām
hostid	ID	Hosta ID, kuram pieder tīmekļa scenārijs. Īpašības darbība: - *nemainīgs* - *obligāts* izveides darbībām
name	string	Tīmekļa scenārija nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
agent	string	Lietotāja aģenta virkne, ko izmantos tīmekļa scenārijs. Noklusējums: Zabbix
authentication	integer	Autentifikācijas metode, ko izmantos tīmekļa scenārijs. Iespējamās vērtības: 0 - *(noklusējums)* nav; 1 - pamata HTTP autentifikācija; 2 - NTLM autentifikācija; 3 - Kerberos autentifikācija; 4 - Digest autentifikācija.
delay	string	Tīmekļa scenārija izpildes intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d), vai lietotāja makrosu. Noklusējums: 1m.
headers	array	HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu.
http_password	string	Parole, ko izmanto pamata HTTP, NTLM, Kerberos vai Digest autentifikācijai.
http_proxy	string	Starpniekserveris, ko izmantos tīmekļa scenārijs, norādīts kā *http://[username[:password]@]proxy.example.com[:port]*.
http_user	string	Lietotājvārds, ko izmanto pamata HTTP, NTLM, Kerberos vai Digest autentifikācijai.
retries	integer	Reižu skaits, cik tīmekļa scenārijs mēģinās izpildīt katru soli pirms neveiksmes. Noklusējums: 1.
ssl_cert_file	string	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai (jābūt PEM formātā).
ssl_key_file	string	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai (jābūt PEM formātā).
ssl_key_password	string	SSL privātās atslēgas parole.
status	integer	Vai tīmekļa scenārijs ir iespējots. Iespējamās vērtības: 0 - *(noklusējums)* iespējots; 1 - atspējots.

Īpašība	Tips	Apraksts
templateid	ID	Vecākveidnes tīmekļa scenārija ID. Īpašības darbība: - *tikai lasāms*
variables	array	Tīmekļa scenārija mainīgie.
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst nosaukumam hosta sertifikātā. Iespējamās vērtības: 0 - *(noklusējums)* izlaist hosta pārbaudi; 1 - pārbaudīt hostu.
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* izlaist otras puses pārbaudi; 1 - pārbaudīt otru pusi.
uuid	string	Globāli unikāls identifikators, ko izmanto importēto tīmekļa scenāriju sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja tīmekļa scenārijs pieder veidnei

Scenārija solis

Scenārija soļa objekts definē konkrētu tīmekļa scenārija pārbaudi. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
name	string	Scenārija soļa nosaukums. Īpašības darbība: - *obligāta*
no	integer	Soļa kārtas numurs tīmekļa scenārijā. Īpašības darbība: - *obligāta*
url	string	Pārbaudāmais URL. Īpašības darbība: - *obligāta*
follow_redirects	integer	Vai sekot HTTP pāradresācijām. Iespējamās vērtības: 0 - nesekot pāradresācijām; 1 - *(noklusējums)* sekot pāradresācijām.
headers	array	HTTP galvenes, kas tiks nosūtītas, izpildot pieprasījumu. Scenārija soļa galvenes pārrakstīs tīmekļa scenārijam norādītās galvenes.
posts	string/array	HTTP POST mainīgie kā virkne (neapstrādāti POST dati) vai kā HTTP lauku masīvs (veidlapas lauku dati).
required	string	Teksts, kam jābūt ietvertam atbildē.
retrieve_mode	integer	HTTP atbildes daļa, kas scenārija solim jāiegūst. Iespējamās vērtības: 0 - *(noklusējums)* tikai pamatteksts; 1 - tikai galvenes; 2 - galvenes un pamatteksts.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem.

Īpašība	Tips	Apraksts
timeout	string	Pieprasījuma noildze sekundēs. Pieņem sekundes, laika vienību ar sufiksu vai lietotāja makro. Noklusējums: 15s. Maksimums: 1h. Minimums: 1s.
variables	array	Scenārija soļa mainīgie.
query_fields	array	Vaicājuma lauki — HTTP lauku masīvs, kas tiks pievienots URL, izpildot pieprasījumu.

HTTP lauks

HTTP lauka objekts definē nosaukumu un vērtību, ko izmanto, lai norādītu tīmekļa scenārija mainīgos, HTTP galvenes un POST laukus vai vaicājuma laukus. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
name	string	Galvenes/mainīgā/POST vai GET lauka nosaukums. Property behavior: - *required*
value	string	Galvenes/mainīgā/POST vai GET lauka vērtība. Property behavior: - *required*

Tīmekļa scenārija tags

Tīmekļa scenārija taga objektam ir šādas īpašības.

Īpašība	Type	Apraksts
tag	string	Tīmekļa scenārija taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Tīmekļa scenārija taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne; 1 - hosts. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

httptest.create

Apraksts

`object httptest.create(object/array webScenarios)`

Šī metode ļauj izveidot jaunus tīmekļa scenārijus.

> **Note:**
> Izveidojot tīmekļa scenāriju, tiks automātiski izveidota kopa ar tīmekļa uzraudzības vienumiem.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Papildinformāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Tīmekļa scenāriji, ko izveidot.

Papildus standarta tīmekļa scenārija īpašībām, metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
steps	array	Scenārija soļi. Parametra darbība: - *obligāts*
tags	array	Tīmekļa scenārija birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto web scenāriju ID īpašībā `httptestids`. Atgriezto ID secība atbilst nodoto web scenāriju secībai.

Piemēri

Tīmekļa scenārija izveide

Izveidojiet tīmekļa scenāriju, lai uzraudzītu uzņēmuma mājaslapu. Scenārijam būs divi soļi, lai pārbaudītu sākumlapu un lapu "Par mums", kā arī pārliecinātos, ka tās atgriež HTTP statusa kodu 200.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Scenārija solis

Avots

CHttpTest::create() failā *ui/include/classes/api/services/CHttpTest.php*.

httptest.delete

Apraksts

`object httptest.delete(array webScenarioIds)`

Šī metode ļauj dzēst tīmekļa scenārijus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo tīmekļa scenāriju ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto tīmekļa scenāriju ID īpašībā `httptestids`.

Piemēri

Vairāku tīmekļa scenāriju dzēšana

Dzēsiet divus tīmekļa scenārijus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.delete",
    "params": [
        "2",
        "3"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "2",
            "3"
        ]
    },
    "id": 1
}
```

Avots

CHttpTest::delete() failā *ui/include/classes/api/services/CHttpTest.php*.

httptest.get

Apraksts

`integer/array httptest.get(object parameters)`

Metode ļauj izgūt tīmekļa scenārijus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai tīmekļa scenārijus, kas pieder norādītajām hostu grupām.
hostids	ID/array	Atgriezt tikai tīmekļa scenārijus, kas pieder norādītajiem hostiem.
httptestids	ID/array	Atgriezt tikai tīmekļa scenārijus ar norādītajiem ID.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai tīmekļa scenārijus, kas mantoti no veidnes.
inheritedTags	boolean	Atgriezt tīmekļa scenārijus, kuriem norādītās `tags` ir arī veidnē/hostā/piesaistītajās veidnēs. Iespējamās vērtības: `true` - veidnei/hostam/piesaistītajām veidnēm arī jābūt norādītajām birkām; `false` - *(noklusējums)* birkas no veidnes/hosta/piesaistītajām veidnēm tiek ignorētas.
monitored	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotus tīmekļa scenārijus, kas pieder uzraudzītiem hostiem.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas atrodas veidnē/hostā/piesaistītajās veidnēs.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai tīmekļa scenārijus, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai tīmekļa scenārijus, kas pieder norādītajām veidnēm.
expandName	flag	Izvērst makrosus tīmekļa scenārija nosaukumā.
expandStepName	flag	Izvērst makrosus scenārija soļu nosaukumos.
evaltype	integer	Birku novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
tags	array	Atgriezt tikai tīmekļa scenārijus ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus tīmekļa scenārijus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
selectHosts	query	Atgriezt hostus, kuriem pieder tīmekļa scenārijs, kā masīvu `hosts` īpašībā.
selectSteps	query	Atgriezt tīmekļa scenārija soļus `steps` īpašībā. Atbalsta `count`.
selectTags	query	Atgriezt tīmekļa scenārija birkas `tags` īpašībā.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `httptestid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	

Parametrs	Tips	Apraksts
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Tīmekļa scenārija izgūšana

Izgūstiet visus datus par tīmekļa scenāriju ”9”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.get",
    "params": {
        "output": "extend",
        "selectSteps": "extend",
        "httptestids": "9"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Hosts
- Scenārija solis

Avots

CHttpTest::get() failā *ui/include/classes/api/services/CHttpTest.php*.

httptest.update

Apraksts

`object httptest.update(object/array webScenarios)`

Šī metode ļauj atjaunināt esošos tīmekļa scenārijus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo tīmekļa scenāriju rekvizīti.

Rekvizītam `httptestid` jābūt definētam katram tīmekļa scenārijam, visi pārējie rekvizīti nav obligāti. Tiks atjaunināti tikai nodotie rekvizīti, visi pārējie paliks nemainīti.

Papildus standarta tīmekļa scenārija rekvizītiem metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
steps	array	Scenārija soļi, ar kuriem aizstāt esošos soļus.
tags	array	Tīmekļa scenārija birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto tīmekļa scenāriju ID īpašībā `httptestid`.

Piemēri

Tīmekļa scenārija iespējošana

Iespējojiet tīmekļa scenāriju, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "httptest.update",
    "params": {
        "httptestid": "5",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "httptestids": [
            "5"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Scenārija solis

Avots

CHttpTest::update() failā *ui/include/classes/api/services/CHttpTest.php*.

Uzdevums

Šī klase ir paredzēta darbam ar uzdevumiem (piemēram, vienumu vai zema līmeņa atklāšanas kārtulu pārbaudei bez konfigurācijas pārlādes).

Objektu atsauces:

- Task
  - ’Izpildīt tagad’ pieprasījuma objekts
  - ’Atsvaidzināt starpniekservera konfigurāciju’ pieprasījuma objekts
  - ’Diagnostikas informācijas’ pieprasījuma objekts
    * Statistikas pieprasījuma objekts
    * Statistikas rezultāta objekts

Pieejamās metodes:

- task.create - izveidot jaunus uzdevumus

- task.get - izgūt uzdevumus

Uzdevuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `task` API.

Uzdevums

Uzdevuma objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
taskid	ID	Uzdevuma ID. Īpašības darbība: - *tikai lasāms*
type	integer	Uzdevuma tips. Iespējamās vērtības: 1 - Diagnostikas informācija; 2 - starpniekservera konfigurācijas atsvaidzināšana; 6 - Izpildīt tagad. *Admin* un *User* tipa lietotāji var izveidot tikai uzdevumus ”Izpildīt tagad”. Ņemiet vērā, ka `task.get` vienmēr atgriež ”7” (uzdevuma izpildes kopsavilkums). Īpašības darbība: - *obligāts*
status	integer	Uzdevuma statuss. Iespējamās vērtības: 1 - jauns uzdevums; 2 - uzdevums tiek izpildīts; 3 - uzdevums ir pabeigts; 4 - uzdevuma derīguma termiņš ir beidzies. Īpašības darbība: - *tikai lasāms*
clock	timestamp	Laiks, kad uzdevums tika izveidots. Īpašības darbība: - *tikai lasāms*
ttl	integer	Laiks sekundēs, pēc kura uzdevuma derīguma termiņš beidzas. Īpašības darbība: - *tikai lasāms*
proxyid	ID	Tā starpniekservera ID, par kuru tiek apkopota diagnostikas informācijas statistika. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Diagnostikas informācija” vai ”starpniekservera konfigurācijas atsvaidzināšana”
request	object	Uzdevuma pieprasījuma objekts atbilstoši uzdevuma tipam: Objekts uzdevumam ”Izpildīt tagad” ir detalizēti aprakstīts tālāk; Objekts uzdevumam ”starpniekservera konfigurācijas atsvaidzināšana” ir detalizēti aprakstīts tālāk; Objekts uzdevumam ”Diagnostikas informācija” ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāts*

Īpašība	Tips	Apraksts
result	object	Diagnostikas informācijas uzdevuma rezultāta objekts. Var saturēt NULL, ja rezultāts vēl nav gatavs. Rezultāta objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *tikai lasāms*

Objekts 'Izpildīt tagad' pieprasījumam

Uzdevuma pieprasījuma objektam 'Izpildīt tagad' ir šādas īpašības.

Īpašība	Tips	Apraksts
itemid	ID	Vienuma un zema līmeņa atklāšanas kārtulu ID. *Admin* un *User* tipa lietotāji var izmantot 'Izpildīt tagad' vienumiem hostos, kuriem viņiem ir *read-write* atļauja, vai *read* atļauja un viņu lomai iespējota `invoke_execute_now` darbība. Tas pats attiecas uz *Admin* tipa lietotājiem zema līmeņa atklāšanas (LLD) kārtulām.

'Atsvaidzināt starpniekservera konfigurāciju' pieprasījuma objekts

Uzdevuma 'Atsvaidzināt starpniekservera konfigurāciju' pieprasījuma objektam ir šādas īpašības.

Īpašība	Type	Apraksts
proxyids	array	Starpniekserveru ID.

'Diagnostikas informācijas' pieprasījuma objekts

Diagnostikas informācijas uzdevuma pieprasījuma objektam ir šādas īpašības. Statistikas pieprasījuma objekts visiem īpašību tipiem ir detalizēti aprakstīts zemāk.

Īpašība	Tips	Apraksts
historycache	object	Vēstures kešatmiņas statistikas pieprasījums. Pieejams serverī un starpniekserverī.
valuecache	object	Vienumu kešatmiņas statistikas pieprasījums. Pieejams serverī.
preprocessing	object	Pirmsapstrādes pārvaldnieka statistikas pieprasījums. Pieejams serverī un starpniekserverī.
alerting	object	Brīdinājumu pārvaldnieka statistikas pieprasījums. Pieejams serverī.
lld	object	LLD pārvaldnieka statistikas pieprasījums. Pieejams serverī.

Statistikas pieprasījuma objekts

Statistikas pieprasījuma objekts tiek izmantots, lai noteiktu, kāda veida informācija būtu jāapkopo par servera/starpniekservera iekšējiem procesiem. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
stats	query	Atgriežamās statistikas objekta īpašības. Pieejamo lauku saraksts katram diagnostikas informācijas statistikas tipam ir detalizēti aprakstīts zemāk. Noklusējums: `extend` atgriezīs visus pieejamos statistikas laukus.
top	object	Objekts atgriezto statistikas vērtību kārtošanai un ierobežošanai. Pieejamo lauku saraksts katram diagnostikas informācijas statistikas tipam ir detalizēti aprakstīts zemāk. Piemērs: { "source.alerts": 10 }

Katram diagnostikas informācijas pieprasījuma tipam pieejamo statistikas lauku saraksts

Tālāk norādītos statistikas laukus var pieprasīt katram diagnostikas informācijas pieprasījuma rekvizīta tipam.

Diagnostikas tips	Pieejamie lauki	Apraksts
historycache	items	Kešatmiņā saglabāto vienumu skaits.
	values	Kešatmiņā saglabāto vērtību skaits.
	memory	Koplietojamās atmiņas statistika (brīvā vieta, izmantoto bloku skaits, brīvo bloku skaits, brīvā bloka maksimālais izmērs).
	memory.data	Vēstures datu kešatmiņas koplietojamās atmiņas statistika.
	memory.index	Vēstures indeksa kešatmiņas koplietojamās atmiņas statistika.
valuecache	items	Kešatmiņā saglabāto vienumu skaits.
	values	Kešatmiņā saglabāto vērtību skaits.
	memory	Koplietojamās atmiņas statistika (brīvā vieta, izmantoto bloku skaits, brīvo bloku skaits, brīvā bloka maksimālais izmērs).
	mode	Vērtību kešatmiņas režīms.
preprocessing	values	Rindā esošo vērtību skaits.
	preproc.values	Rindā esošo vērtību skaits ar priekšapstrādes soļiem.
alerting	alerts	Rindā esošo brīdinājumu skaits.
lld	rules	Rindā esošo noteikumu skaits.
	values	Rindā esošo vērtību skaits.

Katram diagnostikas informācijas pieprasījuma tipam pieejamo kārtošanas lauku saraksts

Tālāk norādītos statistikas laukus var izmantot, lai kārtotu un ierobežotu pieprasīto informāciju.

Diagnostikas tips	Pieejamie lauki	Tips
historycache	values	integer
valuecache	values	integer
	request.values	integer
preprocessing	values	integer
alerting	media.alerts	integer
	source.alerts	integer
lld	values	integer

Statistikas rezultāta objekts

Statistikas rezultāta objekts tiek iegūts uzdevuma objekta laukā `result`.

Īpašība	Tips	Apraksts
status	integer	Uzdevuma rezultāta statuss. Iespējamās vērtības: -1 - uzdevuma izpildes laikā radās kļūda; 0 - uzdevuma rezultāts ir izveidots. Īpašības darbība: - *tikai lasāms*
data	string/object	Rezultāti atbilstoši konkrētā diagnostikas informācijas uzdevuma statistikas pieprasījuma objektam. Satur kļūdas ziņojuma virkni, ja uzdevuma izpildes laikā radās kļūda.

task.create

Apraksts

`object task.create(object/array tasks)`

Šī metode ļauj izveidot uzdevumus.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Izveidojamais uzdevums.

Metode pieņem uzdevumus ar standarta uzdevuma īpašībām.

Ņemiet vērā, ka uzdevumus "Izpildīt tagad" var izveidot tikai šādu tipu vienumiem/atklāšanas kārtulām:

- Zabbix aģents (pasīvs)
- Vienkāršā pārbaude
- SNMP aģents (v1/v2/v3)
- Zabbix iekšējais
- Ārējā pārbaude
- Datubāzes monitors
- HTTP aģents
- IPMI aģents
- SSH aģents
- TELNET aģents
- JMX aģents
- Aprēķināts
- Atkarīgais vienums
- Skripts
- Pārlūks

Ja vienuma/atklāšanas kārtulas tips ir "Atkarīgais vienums", tad arī tā galvenajam vienumam jābūt vienam no iepriekš minētajiem tipiem.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto uzdevumu ID īpašībā `taskids`. Katram vienumam un zemā līmeņa atklāšanas kārtulai tiek izveidots viens uzdevums. Atgriezto ID secība atbilst nodoto `itemids` secībai.

Piemēri

Uzdevuma izveide

Izveidojiet uzdevumu 'Execute now' vienumam un zema līmeņa atklāšanas kārtulai.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 6,
            "request": {
                "itemid": "10092"
            }
        },
        {
            "type": 6,
            "request": {
                "itemid": "10093"
            }
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "taskids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Izveidojiet uzdevumu ’Refresh proxy configuration’ diviem starpniekserveriem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "task.create",
    "params": [
        {
            "type": 2,
            "request": {
                "proxyids": ["10459", "10460"]
            }
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "1"
        ]
    },
    "id": 1
}
```

Izveidojiet uzdevumu ’Diagnostic information’.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "taskids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Uzdevums
- 'Izpildīt tagad' pieprasījuma objekts
- 'Atsvaidzināt starpniekservera konfigurāciju' pieprasījuma objekts
- 'Diagnostikas informācijas' pieprasījuma objekts

Avots

CTask::create() failā *ui/include/classes/api/services/CTask.php*.

task.get

Apraksts

`integer/array task.get(object parameters)`

Metode ļauj izgūt uzdevumus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
taskids	ID/array	Atgriezt tikai uzdevumus ar norādītajiem ID.
output	query	Šie parametri ir aprakstīti atsauces komentārā.
preservekeys	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež objektu masīvu.

Piemēri

Iegūt uzdevumu pēc ID

Iegūt visus datus par uzdevumu ar ID "1".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "task.get",
    "params": {
        "output": "extend",
        "taskids": "1"
```

```
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatīt arī

- Uzdevums
- ’Izpildīt tagad’ pieprasījuma objekts
- ’Atsvaidzināt starpniekservera konfigurāciju’ pieprasījuma objekts
- ’Diagnostikas informācijas’ pieprasījuma objekts

Avots

CTask::get() failā *ui/include/classes/api/services/CTask.php*.

Uzturēšana

Šī klase ir paredzēta darbam ar uzturēšanām.

Objektu atsauces:

- Uzturēšana
- Laika periods
- Problēmas tags

Pieejamās metodes:

- maintenance.create - izveidot jaunas uzturēšanas
- maintenance.delete - dzēst uzturēšanas
- maintenance.get - izgūt uzturēšanas
- maintenance.update - atjaunināt uzturēšanas

Uzturēšanas objekts

Tālāk norādītie objekti ir tieši saistīti ar `maintenance` API.

Uzturēšana

Uzturēšanas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
maintenanceid	ID	Uzturēšanas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Uzturēšanas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
active_since	timestamp	Laiks, kad uzturēšana kļūst aktīva (ieskaitot). Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Īpašības darbība: - *obligāts* izveides operācijām
active_till	timestamp	Laiks, kad uzturēšana pārstāj būt aktīva (neieskaitot). Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Īpašības darbība: - *obligāts* izveides operācijām
description	string	Uzturēšanas apraksts.
maintenance_type	integer	Uzturēšanas tips. Iespējamās vērtības: 0 - *(noklusējums)* ar datu vākšanu; 1 - bez datu vākšanas.
tags_evaltype	integer	Problēmu birku novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.

Laika periods

Laika perioda objekts tiek izmantots, lai definētu periodus, kad uzturēšanai jāstājas spēkā. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
period	integer	Uzturēšanas perioda ilgums sekundēs. Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Iespējamo vērtību diapazons: 300 - 86399940. Noklusējums: 3600.
timeperiod_type	integer	Laika perioda tips. Iespējamās vērtības: 0 - *(noklusējums)* tikai vienu reizi; 2 - katru dienu; 3 - katru nedēļu; 4 - katru mēnesi.
start_date	timestamp	Datums, kad uzturēšanas periodam jāstājas spēkā. Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Noklusējums: pašreizējais datums. Īpašības darbība: - *atbalstīts*, ja `timeperiod_type` ir iestatīts uz ”tikai vienu reizi”
start_time	integer	Diennakts laiks sekundēs, kad sākas uzturēšana. Norādītā vērtība tiks noapaļota uz leju līdz minūtēm. Noklusējums: 0. Īpašības darbība: - *atbalstīts*, ja `timeperiod_type` ir iestatīts uz ”katru dienu”, ”katru nedēļu” vai ”katru mēnesi”
every	integer	Dienas un nedēļas periodiem `every` nosaka dienu vai nedēļu intervālus, kuros uzturēšanai jāstājas spēkā. Noklusējuma vērtība, ja `timeperiod_type` ir iestatīts uz ”katru dienu” vai ”katru nedēļu”: 1. Mēneša periodiem, kad ir iestatīts `day`, īpašība `every` nosaka mēneša dienu, kad uzturēšanai jāstājas spēkā. Noklusējuma vērtība, ja `timeperiod_type` ir iestatīts uz ”katru mēnesi” un `day` ir iestatīts: 1. Mēneša periodiem, kad ir iestatīts `dayofweek`, īpašība `every` nosaka mēneša nedēļu, kad uzturēšanai jāstājas spēkā. Iespējamās vērtības, ja `timeperiod_type` ir iestatīts uz ”katru mēnesi” un `dayofweek` ir iestatīts: 1 - *(noklusējums)* pirmā nedēļa; 2 - otrā nedēļa; 3 - trešā nedēļa; 4 - ceturtā nedēļa; 5 - pēdējā nedēļa. Īpašības darbība: - *atbalstīts*, ja `timeperiod_type` ir iestatīts uz ”katru dienu”, ”katru nedēļu” vai ”katru mēnesi”

Īpašība	Tips	Apraksts
dayofweek	integer	Nedēļas dienas, kad uzturēšanai jāstājas spēkā. Iespējamās bitkartes vērtības: 1 - pirmdiena; 2 - otrdiena; 4 - trešdiena; 8 - ceturtdiena; 16 - piektdiena; 32 - sestdiena; 64 - svētdiena. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 21 pirmdienai, trešdienai un piektdienai). Īpašības darbība: - *obligāts*, ja `timeperiod_type` ir iestatīts uz ”katru nedēļu” vai ja `timeperiod_type` ir iestatīts uz ”katru mēnesi” un `day` nav iestatīts
day	integer	Mēneša diena, kad uzturēšanai jāstājas spēkā. Īpašības darbība: - *obligāts*, ja `timeperiod_type` ir iestatīts uz ”katru mēnesi” un `dayofweek` nav iestatīts
month	integer	Mēneši, kad uzturēšanai jāstājas spēkā. Iespējamās bitkartes vērtības: 1 - janvāris; 2 - februāris; 4 - marts; 8 - aprīlis; 16 - maijs; 32 - jūnijs; 64 - jūlijs; 128 - augusts; 256 - septembris; 512 - oktobris; 1024 - novembris; 2048 - decembris. Šis ir bitmaskas lauks; ir pieļaujama jebkura iespējamo bitkartes vērtību summa (piemēram, 585 janvārim, aprīlim, jūlijam un oktobrim). Īpašības darbība: - *obligāts*, ja `timeperiod_type` ir iestatīts uz ”katru mēnesi”

Problēmas tags

Problēmas taga objekts tiek izmantots, lai noteiktu, kuras problēmas ir jāapslāpē, kad stājas spēkā uzturēšana. Tagus var norādīt tikai tad, ja Uzturēšanas objekta `maintenance_type` ir iestatīts uz ”ar datu vākšanu”. Tam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Problēmas taga nosaukums. Īpašības darbība: - *obligāts*
operator	integer	Nosacījuma operators. Iespējamās vērtības: 0 - Vienāds; 2 - *(noklusējums)* Satur.
value	string	Problēmas taga vērtība.

maintenance.create

Apraksts

`object maintenance.create(object/array maintenances)`

Šī metode ļauj izveidot jaunas apkopes.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Uzturēšanas ieraksti, ko izveidot.

Papildus standarta uzturēšanas īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Hostu grupas, kurām tiks veikta uzturēšana. Hostu grupām jābūt definētai tikai `groupid` īpašībai. Parametra darbība: - *obligāts*, ja `hosts` nav iestatīts
hosts	object/array	Hosti, kuriem tiks veikta uzturēšana. Hostiem jābūt definētai tikai `hostid` īpašībai. Parametra darbība: - *obligāts*, ja `groups` nav iestatīts
timeperiods	object/array	Uzturēšanas laika periodi. Parametra darbība: - *obligāts*
tags	object/array	Problēmu birkas. Definē, kuras problēmas ir jānomāc. Ja birkas nav norādītas, tiks nomāktas visas aktīvās uzturēšanas hostu problēmas. Parametra darbība: - *atbalstīts*, ja Uzturēšanas objekta `maintenance_type` ir iestatīts uz "with data collection"

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto uzturēšanu ID īpašībā `maintenanceids`. Atgriezto ID secība atbilst nodoto uzturēšanu secībai.

Piemēri

Uzturēšanas izveide

Izveidojiet uzturēšanu ar datu vākšanu hostu grupai ar ID "2" un ar problēmu tagiem **service:mysqld** un **error**. Tai jābūt aktīvai no 17.03.2026 līdz 17.03.2027, jāstājas spēkā katru svētdienu plkst. 18:00 un jāturpinās vienu stundu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Laika periods

Avots

CMaintenance::create() failā *ui/include/classes/api/services/CMaintenance.php*.

maintenance.delete

Apraksts

`object maintenance.delete(array maintenanceIds)`

Šī metode ļauj dzēst uzturēšanas periodus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo uzturēšanas periodu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto uzturēšanas periodu ID īpašībā `maintenanceids`.

Piemēri

Vairāku uzturēšanas periodu dzēšana

Dzēsiet divus uzturēšanas periodus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.delete",
    "params": [
        "3",
        "1"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3",
            "1"
        ]
    },
    "id": 1
}
```

Avots

CMaintenance::delete() failā *ui/include/classes/api/services/CMaintenance.php*.

maintenance.get

Apraksts

`integer/array maintenance.get(object parameters)`

Metode ļauj izgūt uzturēšanas atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai tās uzturēšanas, kas ir piešķirtas norādītajām hostu grupām.
hostids	ID/array	Atgriezt tikai tās uzturēšanas, kas ir piešķirtas norādītajiem hostiem.
maintenanceids	ID/array	Atgriezt tikai uzturēšanas ar norādītajiem ID.
selectHostGroups	query	Atgriezt `hostgroups` īpašību ar hostu grupām, kas piešķirtas uzturēšanai.
selectHosts	query	Atgriezt `hosts` īpašību ar hostiem, kas piešķirti uzturēšanai.
selectTags	query	Atgriezt `tags` īpašību ar uzturēšanas problēmu birkām.
selectTimeperiods	query	Atgriezt `timeperiods` īpašību ar uzturēšanas laika periodiem.

Parametrs	Tips	Apraksts
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `maintenanceid`, `name`, `maintenance_type`, `active_since`, `active_till`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vai nu:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Uzturēšanas periodu izgūšana

Izgūstiet visus konfigurētos uzturēšanas periodus un datus par piešķirtajām hostu grupām, definētajiem laika periodiem un problēmu tagiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.get",
    "params": {
        "output": "extend",
        "selectHostGroups": "extend",
        "selectTimeperiods": "extend",
        "selectTags": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Hosts
- Hostu grupa
- Laika periods

Avots

CMaintenance::get() failā *ui/include/classes/api/services/CMaintenance.php*.

maintenance.update

Apraksts

`object maintenance.update(object/array maintenances)`

Šī metode ļauj atjaunināt esošās uzturēšanas.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Uzturēšanas īpašības, kas jāatjaunina.

Īpašībai `maintenanceid` jābūt definētai katrai uzturēšanai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta uzturēšanas īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Hostu grupas, kas aizstās pašreizējās grupas. Hostu grupām jābūt definētai tikai īpašībai `groupid`. Parametra darbība: - *obligāts*, ja `hosts` nav iestatīts
hosts	object/array	Hosti, kas aizstās pašreizējos hostus. Hostiem jābūt definētai tikai īpašībai `hostid`. Parametra darbība: - *obligāts*, ja `groups` nav iestatīts
timeperiods	object/array	Uzturēšanas laika periodi, kas aizstās pašreizējos periodus.
tags	object/array	Problēmu birkas, kas aizstās pašreizējās birkas. Parametra darbība: - *atbalstīts*, ja Uzturēšanas objekta `maintenance_type` ir iestatīts uz "with data collection"

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto uzturēšanas ierakstu ID īpašībā `maintenanceids`.

Piemēri

Dažādu hostu piešķiršana

Aizstājiet hostus, kas pašlaik piešķirti uzturēšanai, ar diviem citiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "maintenance.update",
    "params": {
        "maintenanceid": "3",
        "hosts": [
            {"hostid": "10085"},
            {"hostid": "10084"}
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "maintenanceids": [
            "3"
        ]
    },
    "id": 1
}
```

Skatīt arī

- Laika periods

Avots

CMaintenance::update() failā *ui/include/classes/api/services/CMaintenance.php*.

Veidne

Šī klase ir paredzēta darbam ar veidnēm.

Objektu atsauces:

- Veidne
- Veidnes tags

Pieejamās metodes:

- template.create - izveidot jaunas veidnes
- template.delete - dzēst veidnes
- template.get - izgūt veidnes
- template.massadd - pievienot saistītos objektus veidnēm
- template.massremove - noņemt saistītos objektus no veidnēm
- template.update - atjaunināt veidnes

Veidnes objekts

Tālāk norādītie objekti ir tieši saistīti ar `template` API.

Veidne

Veidnes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
templateid	ID	Veidnes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
description	text	Veidnes apraksts.
host	string	Veidnes tehniskais nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
name	string	Veidnes redzamais nosaukums. Noklusējums: `host` īpašības vērtība.
readme	text	Veidnei specifiskas konfigurācijas instrukcijas, kas tiek parādītas Host Wizard. Atbalsta Markdown formatējumu.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto veidņu sasaistīšanai ar jau esošajām. Tiek ģenerēts automātiski, ja nav norādīts.
vendor_name	string	Veidnes piegādātāja nosaukums. Izveides operācijām gan `vendor_name`, gan `vendor_version` ir jābūt vai nu iestatītiem, vai atstātiem tukšiem. Atjaunināšanas operācijām `vendor_version` var atstāt tukšu, ja tam datubāzē jau ir vērtība.
vendor_version	string	Veidnes piegādātāja versija. Izveides operācijām gan `vendor_name`, gan `vendor_version` ir jābūt vai nu iestatītiem, vai atstātiem tukšiem. Atjaunināšanas operācijām `vendor_name` var atstāt tukšu, ja tam datubāzē jau ir vērtība.
wizard_ready	integer	Vai veidne ir pieejama atlasei Host Wizard. Iespējamās vērtības: 0 - *(noklusējums)* Nav pieejama. 1 - Pieejama.

Veidnes tags

Veidnes taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Veidnes taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Veidnes taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

veidne.create

Apraksts

`object template.create(object/array templates)`

Šī metode ļauj izveidot jaunas veidnes.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Veidnes, ko izveidot.

Papildus standarta veidnes īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Veidņu grupas, kurām pievienot veidni. Veidņu grupām jābūt definētai tikai īpašībai `groupid`. Parametra darbība: - *obligāts*
tags	object/array	Veidņu birkas.
templates	object/array	Veidnes, kas jāpiesaista veidnei. Veidnēm jābūt definētai tikai īpašībai `templateid`.
macros	object/array	Lietotāja makrosi, kas jāizveido veidnei.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto veidņu ID īpašībā `templateids`. Atgriezto ID secība atbilst nodoto veidņu secībai.

Piemēri

Veidnes izveide

Izveidojiet veidni ar tagiem un piesaistiet šai veidnei divas veidnes.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "11117"
        ]
    },
    "id": 1
}
```

Avots

CTemplate::create() failā *ui/include/classes/api/services/CTemplate.php*.

veidne.delete

Apraksts

`object template.delete(array templateIds)`

Šī metode ļauj dzēst veidnes.

Dzēšot veidni, no visiem hostiem vai veidnēm, ar kuriem tā ir saistīta, tiek noņemtas arī visas tās entītijas (vienumi, trigeri, grafiki u.c.). Lai šīs entītijas saglabātu, vispirms atsaistiet veidni no attiecīgajiem hostiem vai veidnēm, izmantojot metodes `host.update`, `host.massremove`, `template.update` vai `template.massremove`.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo veidņu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto veidņu ID zem `templateids` īpašības.

Piemēri

Vairāku veidņu dzēšana

Dzēsiet divas veidnes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.delete",
    "params": [
        "13",
        "32"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "13",
            "32"
        ]
    },
    "id": 1
}
```

Avots

CTemplate::delete() failā *ui/include/classes/api/services/CTemplate.php*.

veidne.get

Apraksts

`integer/array template.get(object parameters)`

Metode ļauj izgūt veidnes atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
templateids	ID/array	Atgriezt tikai veidnes ar norādītajiem veidņu ID.
groupids	ID/array	Atgriezt tikai veidnes, kas pieder norādītajām veidņu grupām.
parentTemplateids	ID/array	Atgriezt tikai veidnes, ar kurām norādītā veidne ir sasaistīta.
hostids	ID/array	Atgriezt tikai veidnes, kas ir sasaistītas ar norādītajiem hostiem/veidnēm.
graphids	ID/array	Atgriezt tikai veidnes, kas satur norādītos grafikus.
itemids	ID/array	Atgriezt tikai veidnes, kas satur norādītos vienumus.
triggerids	ID/array	Atgriezt tikai veidnes, kas satur norādītos trigerus.
with_items	flag	Atgriezt tikai veidnes, kurām ir vienumi.
with_triggers	flag	Atgriezt tikai veidnes, kurām ir trigeri.
with_graphs	flag	Atgriezt tikai veidnes, kurām ir grafiki.
with_httptests	flag	Atgriezt tikai veidnes, kurām ir tīmekļa scenāriji.

Parametrs	Tips	Apraksts
evaltype	integer	Birkas novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
tags	object/array	Atgriezt tikai veidnes ar norādītajām birkām. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visas veidnes. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
inheritedTags	boolean	Atgriezt veidnes, kurām norādītās `tags` ir arī sasaistītajās veidnēs. Iespējamās vērtības: `true` - sasaistītajām veidnēm arī jābūt norādītajām birkām; `false` - *(noklusējums)* birkas no sasaistītajām veidnēm tiek ignorētas.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas atrodas sasaistītajās veidnēs.
selectTags	query	Atgriezt veidnes birkas `tags` īpašībā.
selectDiscoveryRules	query	Atgriezt `discoveryRules` īpašību ar veidnes LLD noteikumiem. Atbalsta `count`.
selectHosts	query	Atgriezt hostus, kas ir sasaistīti ar veidni, `hosts` īpašībā. Atbalsta `count`.
selectTemplateGroups	query	Atgriezt veidņu grupas, kurām veidne pieder, `templategroups` īpašībā.
selectTemplates	query	Atgriezt veidnes, ar kurām norādītā veidne ir sasaistīta, `templates` īpašībā. Atbalsta `count`.
selectParentTemplates	query	Atgriezt veidnes, kas ir sasaistītas ar norādīto veidni, `parentTemplates` īpašībā. Atbalsta `count`.
selectHttpTests	query	Atgriezt tīmekļa scenārijus no veidnes `httpTests` īpašībā. Atbalsta `count`.
selectItems	query	Atgriezt vienumus no veidnes `items` īpašībā. Atbalsta `count`.
selectTriggers	query	Atgriezt trigerus no veidnes `triggers` īpašībā. Atbalsta `count`.
selectGraphs	query	Atgriezt grafikus no veidnes `graphs` īpašībā. Atbalsta `count`.
selectMacros	query	Atgriezt makrosus no veidnes `macros` īpašībā.
selectDashboards	query	Atgriezt paneļus no veidnes `dashboards` īpašībā. Atbalsta `count`.
selectValueMaps	query	Atgriezt `valuemaps` īpašību ar veidnes vērtību kartēm.

Parametrs	Tips	Apraksts
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectTemplates` - rezultāti tiks kārtoti pēc `name`; `selectHosts` - kārtoti pēc `host`; `selectParentTemplates` - kārtoti pēc `host`; `selectItems` - kārtoti pēc `name`; `selectDiscoveryRules` - kārtoti pēc `name`; `selectTriggers` - kārtoti pēc `description`; `selectGraphs` - kārtoti pēc `name`; `selectDashboards` - kārtoti pēc `name`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `hostid`, `host`, `name`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
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	Atgriezt zema līmeņa atklāšanas no veidnes `discoveries` īpašībā. Atbalsta `count`. Šis vaicājums ir **novecojis**, lūdzu, tā vietā izmantojiet `selectDiscoveryRules`.

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Veidņu izgūšana pēc nosaukuma

Izgūstiet visus datus par divām veidnēm ar nosaukumiem ”Linux by Zabbix agent” un ”Windows by Zabbix agent”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent",
                "Windows by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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": "Šī ir oficiāla Linux veidne. Tai ir nepieciešams Zabbix aģents 8.0 vai jaunāks
            "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": "## Pārskats\r\n\r\nŠī ir oficiāla Linux veidne. Tai ir nepieciešams Zabbix aģents 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": "Šī ir oficiāla Windows veidne. Tai ir nepieciešams Zabbix aģents 8.0 vai jaunā
            "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": "## Pārskats\r\n\r\nŠī ir oficiāla Windows veidne. Tai ir nepieciešams Zabbix aģents
        }
    ],
    "id": 1
}
```


Veidņu grupu izgūšana

Izgūstiet veidņu grupas, kuru dalībnieks ir veidne ”Linux by Zabbix aģents”.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTemplateGroups": "extend",
        "filter": {
            "host": [
                "Linux by Zabbix agent"
            ]
        }
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10001",
            "templategroups": [
                {
                    "groupid": "10",
                    "name": "Templates/Operating systems",
                    "uuid": "846977d1dfed4968bc5f8bdb363285bc"
                }
            ]
        }
    ],
    "id": 1
}
```


Hostu izgūšana pēc veidnes

Izgūstiet hostus, kuriem ir piesaistīta veidne ”10001” (*Linux by Zabbix aģents*).

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": "templateid",
        "templateids": "10001",
        "selectHosts": ["hostid", "name"]
    },
    "id": 1
}
```


Atbilde:


```
{
    "jsonrpc": "2.0",
```

```
    "result": [
        {
            "templateid": "10001",
            "hosts": [
                {
                    "hostid": "10084",
                    "name": "Zabbix server"
                },
                {
                    "hostid": "10603",
                    "name": "Host 1"
                },
                {
                    "hostid": "10604",
                    "name": "Host 2"
                }
            ]
        }
    ],
    "id": 1
}
```

Meklēšana pēc veidņu tagiem

Iegūstiet veidnes, kurām tags ”host-name” ir vienāds ar ”{HOST.NAME}”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.get",
    "params": {
        "output": ["hostid"],
        "selectTags": "extend",
        "evaltype": 0,
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}",
                "operator": 1
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "templateid": "10402",
            "tags": [
                {
                    "tag": "host-name",
                    "value": "{HOST.NAME}"
                }
            ]
        }
    ],
    "id": 1
}
```

Skatiet arī

- Veidņu grupa
- Veidne
- Lietotāja makross
- Hosta saskarne

Avots

CTemplate::get() failā *ui/include/classes/api/services/CTemplate.php*.

veidne.massadd

Apraksts

`object template.massadd(object parameters)`

Šī metode ļauj vienlaikus pievienot vairākus saistītus objektus norādītajām veidnēm.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo veidņu ID un objektus, kas jāpievieno veidnēm.

Metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
templates	object/array	Veidnes, kas jāatjaunina. Veidnēm jābūt definētai tikai `templateid` īpašībai. Parametra darbība: - *obligāts*
groups	object/array	Veidņu grupas, kurām pievienot norādītās veidnes. Veidņu grupām jābūt definētai tikai `groupid` īpašībai.
macros	object/array	Lietotāja makrosi, kas jāizveido norādītajām veidnēm.
templates_link	object/array	Veidnes, kas jāpiesaista norādītajām veidnēm. Veidnēm jābūt definētai tikai `templateid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu ID īpašībā `templateids`.

Piemēri

Piesaistīt grupu veidnēm

Pievienot veidņu grupu ”2” divām veidnēm.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10085"
            },
            {
                "templateid": "10086"
            }
        ],
        "groups": [
```

```
            {
                "groupid": "2"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
    "id": 1
}
```

Saistīt divas veidnes ar veidni

Saistiet veidnes ”10106” un ”10104” ar veidni ”10073”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.massadd",
    "params": {
        "templates": [
            {
                "templateid": "10073"
            }
        ],
        "templates_link": [
            {
                "templateid": "10106"
            },
            {
                "templateid": "10104"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10073"
        ]
    },
    "id": 1
}
```

Skatiet arī

- template.update
- Hosts
- Veidņu grupa
- Lietotāja makross

Avots

CTemplate::massAdd() failā *ui/include/classes/api/services/CTemplate.php*.

veidne.massremove

Apraksts

`object template.massremove(object parameters)`

Šī metode ļauj noņemt saistītos objektus no vairākām veidnēm.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo veidņu ID un objektus, kas jānoņem.

Parametrs	Type	Apraksts
templateids	ID/array	Atjauninámo veidņu ID. Parametra darbība: - *required*
groupids	ID/array	To veidņu grupu ID, no kurām noņemt norādītās veidnes.
macros	string/array	To lietotāja makrosu ID, kas jādzēš no norādītajām veidnēm.
templateids_clear	ID/array	To veidņu ID, kuras jāatsaista un jānotīra no norādītajām veidnēm (augšupējās).
templateids_link	ID/array	To veidņu ID, kuras jāatsaista no norādītajām veidnēm (augšupējās).

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu ID zem `templateids` īpašības.

Piemēri

Veidņu noņemšana no grupas

Noņemiet divas veidnes no grupas "2".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": [
            "10085",
            "10086"
        ],
        "groupids": "2"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085",
            "10086"
        ]
    },
```

```
    "id": 1
}
```

Veidņu atsaistīšana no hosta

Atsaistīt veidnes ”10106” un ”10104” no veidnes ”10085”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.massremove",
    "params": {
        "templateids": "10085",
        "templateids_link": [
            "10106",
            "10104"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10085"
        ]
    },
    "id": 1
}
```

Skatiet arī

- template.update
- Lietotāja makross

Avots

CTemplate::massRemove() failā *ui/include/classes/api/services/CTemplate.php*.

veidne.update

Apraksts

`object template.update(object/array templates)`

Šī metode ļauj atjaunināt esošās veidnes.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās veidnes īpašības.

Katram veidnei ir jābūt definētai `templateid` īpašībai, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai norādītās īpašības, visas pārējās paliks nemainītas.

Papildus standarta veidnes īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Veidņu grupas, ar kurām aizstāt pašreizējās veidņu grupas, kurām veidnes pieder. Veidņu grupām jābūt definētai tikai `groupid` īpašībai.
tags	object/array	Veidņu birkas, ar kurām aizstāt pašreizējās veidņu birkas.
macros	object/array	Lietotāja makrosi, ar kuriem aizstāt pašreizējos lietotāja makrosus norādītajās veidnēs.
templates	object/array	Veidnes, ar kurām aizstāt pašlaik piesaistītās veidnes. Veidnes, kas netiek nodotas, tiks tikai atsaistītas. Veidnēm jābūt definētai tikai `templateid` īpašībai.
templates_clear	object/array	Veidnes, ko atsaistīt un notīrīt no norādītajām veidnēm. Veidnēm jābūt definētai tikai `templateid` īpašībai.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu ID īpašībā `templateids`.

Piemēri

Veidnes standarta īpašību mainīšana

Mainiet veidnes tehnisko nosaukumu uz ”Linux by Zabbix agent Custom”, redzamo nosaukumu uz ”My template” un atjauniniet veidnes aprakstu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Veidņu grupu atjaunināšana

Aizstājiet visas dotās veidnes grupas ar citu grupu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "groups": [
            {
                "groupid": "24"
```

```
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Vairāku veidņu grupu atjaunināšana

Aizstājiet visas veidņu grupas vairākām veidnēm ar citām.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "templateids": [
            "10086",
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Veidnes birku atjaunināšana

Aizstājiet visas veidnes birkas ar citu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "tags": [
            {
                "tag": "host-name",
                "value": "{HOST.NAME}"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Veidnes makrosu atjaunināšana

Aizstājiet visus veidnes makrosus ar citu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "macros": [
            {
                "macro": "{$MY_MACRO}",
                "value": "new_value"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
```

```
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Vairāku veidņu makrosu atjaunināšana

Aizstājiet visus lietotāja makrosus ar norādīto lietotāja makrosu vairākās veidnēs.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
            {
                "templateid": "10087",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu. Darbojas
                    }
                ]
            },
            {
                "templateid": "10088",
                "macros": [
                    {
                        "macro": "{$AGENT.TIMEOUT}",
                        "value": "5m",
                        "description": "Noildze, pēc kuras aģents tiek uzskatīts par nepieejamu. Darbojas
                    }
                ]
            }
        ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Veidnei piesaistīto veidņu atjaunināšana

Atvienojiet (neveicot notīrīšanu) visas veidnes no dotās veidnes un piesaistiet tai citu veidni.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": {
        "templateid": "10086",
        "templates": [
```

```
            {
                "templateid": "10001"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10086"
        ]
    },
    "id": 1
}
```

Vairāku veidnei piesaistīto veidņu notīrīšana

Atsaistiet un notīriet konkrētu piesaistīto veidni no dotajām veidnēm.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "template.update",
    "params": [
        {
            "templateid": "10087",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        },
        {
            "templateid": "10088",
            "templates_clear": [
                {
                    "templateid": "10001"
                }
            ]
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "templateids": [
            "10087",
            "10088"
        ]
    },
    "id": 1
}
```

Avots

CTemplate::update() failā *ui/include/classes/api/services/CTemplate.php*.

Veidņu grupa

Šī klase ir paredzēta darbam ar veidņu grupām.

Objektu atsauces:

- Veidņu grupa

Pieejamās metodes:

- templategroup.create - izveidot jaunas veidņu grupas
- templategroup.delete - dzēst veidņu grupas
- templategroup.get - izgūt veidņu grupas
- templategroup.massadd - pievienot saistītos objektus veidņu grupām
- templategroup.massremove - noņemt saistītos objektus no veidņu grupām
- templategroup.propagate - izplatīt atļaujas uz veidņu grupu apakšgrupām
- templategroup.update - atjaunināt veidņu grupas

Veidņu grupas objekts

Tālāk norādītie objekti ir tieši saistīti ar `templategroup` API.

Veidņu grupa

Veidņu grupas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
groupid	ID	Veidņu grupas ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
name	string	Veidņu grupas nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
uuid	string	Universāli unikāls identifikators, ko izmanto importēto veidņu grupu sasaistīšanai ar jau esošajām. Tiek ģenerēts automātiski, ja nav norādīts.

templategroup.create

Apraksts

`object templategroup.create(object/array templateGroups)`

Šī metode ļauj izveidot jaunas veidņu grupas.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet User roles papildinformācijai.

Parametri

`(object/array)` Veidņu grupas, ko izveidot. Metode pieņem veidņu grupas ar standarta veidņu grupu īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto veidņu grupu ID īpašībā `groupids`. Atgriezto ID secība atbilst nodoto veidņu grupu secībai.

Piemēri

Veidņu grupas izveide

Izveidojiet veidņu grupu ar nosaukumu "Templates/Databases".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.create",
    "params": {
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107820"
        ]
    },
    "id": 1
}
```

Avots

CTemplateGroup::create() failā *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.delete

Apraksts

`object templategroup.delete(array templateGroupIds)`

Šī metode ļauj dzēst veidņu grupas.

Veidņu grupu nevar dzēst, ja tajā ir veidnes, kas pieder tikai šai grupai.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(array)` dzēšamo veidņu grupu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto veidņu grupu ID zem `groupids` īpašības.

Piemēri

Vairāku veidņu grupu dzēšana

Dzēsiet divas veidņu grupas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.delete",
    "params": [
        "107814",
        "107815"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "107814",
            "107815"
        ]
    },
    "id": 1
}
```

Avots

CTemplateGroup::delete() failā *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.get

Apraksts

`integer/array templategroup.get(object parameters)`

Metode ļauj izgūt veidņu grupas atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
graphids	ID/array	Atgriezt tikai veidņu grupas, kurās ir veidnes ar norādītajiem grafikiem.
groupids	ID/array	Atgriezt tikai veidņu grupas ar norādītajiem veidņu grupu ID.
templateids	ID/array	Atgriezt tikai veidņu grupas, kurās ir norādītās veidnes.
triggerids	ID/array	Atgriezt tikai veidņu grupas, kurās ir veidnes ar norādītajiem trigeriem.
with_graphs	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar grafikiem.
with_graph_prototypes	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar grafiku prototipiem.
with_httptests	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar tīmekļa pārbaudēm.
with_items	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar vienumiem. Aizstāj parametrus `with_simple_graph_items`.
with_item_prototypes	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar vienumu prototipiem. Aizstāj parametru `with_simple_graph_item_prototypes`.
with_simple_graph_item_prototypes	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar vienumu prototipiem, kuri ir iespējoti izveidei un kuriem ir skaitlisks informācijas tips.
with_simple_graph_items	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar skaitliskiem vienumiem.
with_templates	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes.
with_triggers	boolean	Atgriezt tikai veidņu grupas, kurās ir veidnes ar trigeriem.
selectTemplates	query	Atgriezt `templates` īpašību ar veidnēm, kas pieder veidņu grupai. Atbalsta `count`.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectTemplates` - rezultāti tiks kārtoti pēc `template`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `groupid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.

Parametrs	Tips	Apraksts
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Datu izgūšana pēc nosaukuma

Izgūstiet visus datus par divām veidņu grupām ar nosaukumiem ”Templates/Databases” un ”Templates/Modules”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.get",
    "params": {
        "output": "extend",
        "filter": {
            "name": [
                "Templates/Databases",
                "Templates/Modules"
            ]
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "groupid": "13",
            "name": "Templates/Databases",
            "uuid": "748ad4d098d447d492bb935c907f652f"
        },
        {
            "groupid": "8",
            "name": "Templates/Modules",
            "uuid": "57b7ae836ca64446ba2c296389c009b7"
        }
    ],
    "id": 1
}
```

Skatiet arī

- Veidne

Avots

CTemplateGroup::get() failā *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massadd

Apraksts

`object templategroup.massadd(object parameters)`

Šī metode ļauj vienlaikus pievienot vairākus saistītos objektus visām norādītajām veidņu grupām.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo veidņu grupu ID un objektus, kas jāpievieno visām veidņu grupām.

Metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Veidņu grupas, kas jāatjaunina. Veidņu grupām jābūt definētai tikai `groupid` īpašībai. Parametra darbība: - *obligāts*
templates	object/array	Veidnes, kas jāpievieno visām veidņu grupām. Veidnēm jābūt definētai tikai `templateid` īpašībai. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu grupu ID īpašībā `groupids`.

Piemēri

Veidņu pievienošana veidņu grupām

Pievienojiet divas veidnes veidņu grupām ar ID 12 un 13.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massadd",
    "params": {
        "groups": [
            {
                "groupid": "12"
            },
            {
                "groupid": "13"
            }
        ],
        "templates": [
            {
                "templateid": "10486"
            },
            {
                "templateid": "10487"
            }
        ]
```

```
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "12",
            "13"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Veidne

Avots

CTemplateGroup::massAdd() failā *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.massremove

Apraksts

`object templategroup.massremove(object parameters)`

Šī metode ļauj noņemt saistītos objektus no vairākām veidņu grupām.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas satur atjauninámo veidņu grupu ID un objektus, kas jānoņem.

Parametrs	Type	Apraksts
groupids	ID/array	Atjauninámo veidņu grupu ID. Parametra darbība: - *required*
templateids	ID/array	Veidņu ID, kas jānoņem no visām veidņu grupām. Parametra darbība: - *required*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu grupu ID īpašībā `groupids`.

Piemēri

Veidņu noņemšana no veidņu grupām

Noņemiet divas veidnes no norādītajām veidņu grupām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.massremove",
    "params": {
```

```
        "groupids": [
            "5",
            "6"
        ],
        "templateids": [
            "30050",
            "30001"
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "5",
            "6"
        ]
    },
    "id": 1
}
```

Avots

CTemplateGroup::massRemove() failā *ui/include/classes/api/services/CTemplateGroup.php*.

templategroup.propagate

Apraksts

`object templategroup.propagate(object parameters)`

Šī metode ļauj piemērot atļaujas visām veidņu grupu apakšgrupām.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāju tipiem.
> Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos.
> Plašāku informāciju skatiet [Lietotāju
> lomas](/manual/web_interface/frontend_sections/users/user_roles).

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
groups	object/array	Veidņu grupas, ko izplatīt. Veidņu grupām jābūt definētai tikai `groupid` īpašībai. Parametra darbība: - *obligāts*
permissions	boolean	Iestatiet `true`, ja nepieciešams izplatīt atļaujas. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur propagēto veidņu grupu ID zem `groupids` īpašības.

Piemēri

Veidņu grupas atļauju propagēšana uz tās apakšgrupām.

Propagēt veidņu grupas atļaujas uz tās apakšgrupām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.propagate",
    "params": {
        "groups": [
            {
                "groupid": "15"
            }
        ],
        "permissions": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "15",
        ]
    },
    "id": 1
}
```

Skatiet arī

- templategroup.update
- templategroup.massadd
- Veidne

Avots

CTemplateGroup::propagate() failā

ui/include/classes/api/services/CTemplateGroup.php.

templategroup.update

Apraksts

`object templategroup.update(object/array templateGroups)`

Šī metode ļauj atjaunināt esošās veidņu grupas.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Veidņu grupas īpašības, kas jāatjaunina.

Īpašībai `groupid` jābūt definētai katrai veidņu grupai, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai norādītās īpašības, visas pārējās paliks nemainītas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidņu grupu ID īpašībā `groupids`.

Piemēri

Veidņu grupas pārdēvēšana

Pārdēvējiet veidņu grupu par ”Templates/Databases”

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templategroup.update",
    "params": {
        "groupid": "7",
        "name": "Templates/Databases"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "groupids": [
            "7"
        ]
    },
    "id": 1
}
```

Avots

CTemplateGroup::update() failā *ui/include/classes/api/services/CTemplateGroup.php*.

Veidņu paneļi

Šī klase ir paredzēta darbam ar veidņu paneļiem.

Objektu atsauces:

- Veidnes panelis
- Veidnes paneļa lapa
  - Veidnes paneļa logrīks
    - Veidnes paneļa logrīka lauks

Pieejamās metodes:

- templatedashboard.create - izveidot jaunus veidņu paneļus
- templatedashboard.delete - dzēst veidņu paneļus
- templatedashboard.get - izgūt veidņu paneļus
- templatedashboard.update - atjaunināt veidņu paneļus

Veidnes paneļa objekts

Tālāk norādītie objekti ir tieši saistīti ar `templatedashboard` API.

Veidnes panelis

Veidnes paneļa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
dashboardid	ID	Veidnes paneļa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām

Īpašība	Tips	Apraksts
name	string	Veidnes paneļa nosaukums. Īpašības darbība: - *obligāts* izveides operācijām
templateid	ID	Veidnes ID, kurai pieder panelis. Īpašības darbība: - *nemainīgs* - *obligāts* izveides operācijām
display_period	integer	Noklusējuma lapas attēlošanas periods (sekundēs). Iespējamās vērtības: 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 30.
auto_start	integer	Automātiski sākt slaidrādi. Iespējamās vērtības: 0 - nesākt slaidrādi automātiski; 1 - *(noklusējums)* sākt slaidrādi automātiski.
uuid	string	Universāls unikāls identifikators, ko izmanto importēto veidņu paneļu sasaistīšanai ar jau esošajiem. Tiek ģenerēts automātiski, ja nav norādīts.

Veidnes paneļa lapa

Veidnes paneļa lapas objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
dashboard_pageid	ID	Paneļa lapas ID. Īpašības darbība: - *tikai lasāms*
name	string	Paneļa lapas nosaukums. Noklusējums: tukša virkne.
display_period	integer	Paneļa lapas attēlošanas periods (sekundēs). Iespējamās vērtības: 0, 10, 30, 60, 120, 600, 1800, 3600. Noklusējums: 0 (tiks izmantots noklusējuma lapas attēlošanas periods).
widgets	array	Veidnes paneļa logrīka objektu masīvs.

Veidnes paneļa logrīks

Veidnes paneļa logrīka objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
widgetid	ID	Paneļa logrīka ID. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
type	string	Paneļa logrīka tips. Iespējamās vērtības: actionlog - Darbību žurnāls; clock - Pulkstenis; discovery - Atklāšanas statuss; favgraphs - Iecienītākie grafiki; favmaps - Iecienītākās kartes; gauge - Mērskala; graph - Grafiks (klasisks); graphprototype - Grafika prototips; honeycomb - Šūnveida skats; hostavail - Hosta pieejamība; hostcard - Hosta kartīte; hostnavigator - Hosta navigators; itemcard - Vienuma kartīte; itemnavigator - Vienuma navigators; item - Vienuma vērtība; map - Karte; navtree - Kartes navigācijas koks; piechart - Sektoru diagramma; plaintext - Vienkāršs teksts; problemhosts - Problemātiskie hosti; problems - Problēmas; problemsbysv - Problēmas pēc svarīguma pakāpes; scatterplot - Izkliedes diagramma; slareport - SLA atskaite; svggraph - Grafiks; systeminfo - Sistēmas informācija; tophosts - Top hosti; topitems - Top vienumi; toptriggers - Top trigeri; trigover - Trigeru pārskats; url - URL; web - Tīmekļa uzraudzība. Īpašības darbība: - *obligāts*
name	string	Pielāgots logrīka nosaukums.
x	integer	Horizontālā pozīcija no paneļa kreisās malas. Iespējamās vērtības ir diapazonā no 0 līdz 71.
y	integer	Vertikālā pozīcija no paneļa augšējās malas. Iespējamās vērtības ir diapazonā no 0 līdz 63.
width	integer	Logrīka platums. Iespējamās vērtības ir diapazonā no 1 līdz 72.
height	integer	Logrīka augstums. Iespējamās vērtības ir diapazonā no 1 līdz 64.
view_mode	integer	Logrīka skata režīms. Iespējamās vērtības: 0 - *(noklusējums)* noklusējuma logrīka skats; 1 - ar paslēptu galveni;
fields	array	Veidnes paneļa logrīka lauka objektu masīvs.

Veidnes paneļa logrīka lauks

Veidnes paneļa logrīka lauka objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Logrīka lauka tips. Iespējamās vērtības: 0 - Vesels skaitlis; 1 - Virkne; 4 - Vienums; 5 - Vienuma prototips; 6 - Grafiks; 7 - Grafika prototips; 8 - Karte; 9 - Pakalpojums; 10 - SLA; 11 - Lietotājs; 12 - Darbība; 13 - Medija tips. Īpašības darbība: - *obligāta*
name	string	Logrīka lauka nosaukums. Iespējamās vērtības: skatiet Paneļa logrīku laukus. Ņemiet vērā, ka daži ar hostu saistītie parametri (piemēram, *Hostu grupas*, *Izslēgt hostu grupas* un *Hosti* logrīkā *Problēmas*, *Hostu grupas* logrīkā *Hostu pieejamība* utt.) nav pieejami, konfigurējot logrīku veidnes panelī. Tas ir tāpēc, ka veidnes paneļi attēlo datus tikai no hosta, ar kuru veidne ir saistīta. Īpašības darbība: - *obligāta*
value	mixed	Logrīka lauka vērtība atkarībā no tipa. Iespējamās vērtības: skatiet Paneļa logrīku laukus. Ņemiet vērā, ka daži ar hostu saistītie parametri (piemēram, *Hostu grupas*, *Izslēgt hostu grupas* un *Hosti* logrīkā *Problēmas*, *Hostu grupas* logrīkā *Hostu pieejamība* utt.) nav pieejami, konfigurējot logrīku veidnes panelī. Tas ir tāpēc, ka veidnes paneļi attēlo datus tikai no hosta, ar kuru veidne ir saistīta. Īpašības darbība: - *obligāta*

templatedashboard.create

Apraksts

`object templatedashboard.create(object/array templateDashboards)`

Šī metode ļauj izveidot jaunus veidņu paneļus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamie veidņu paneļi.

Papildus standarta veidnes paneļa īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
pages	array	Izveidojamās veidnes paneļa lapas. Paneļa lapas tiks sakārtotas tādā pašā secībā, kādā tās ir norādītas. Parametra darbība: - *obligāts*

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto veidņu paneļu ID īpašībā `dashboardids`. Atgriezto ID secība atbilst nodoto veidņu paneļu secībai.

Piemēri

Veidnes paneļa izveide

Izveidojiet veidnes paneli ar nosaukumu “Graphs”, kurā ir viens Graph logrīks vienā paneļa lapā.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "32"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Veidnes paneļa lapa
- Veidnes paneļa logrīks
- Veidnes paneļa logrīka lauks

Avots

CTemplateDashboard::create() failā *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.delete

Apraksts

`object templatedashboard.delete(array templateDashboardIds)`

Šī metode ļauj dzēst veidņu paneļus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` Dzēšamo veidņu paneļu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto veidnes paneļu ID īpašībā `dashboardids`.

Piemēri

Vairāku veidņu paneļu dzēšana

Dzēsiet divus veidņu paneļus.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.delete",
    "params": [
        "45",
        "46"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "45",
            "46"
        ]
    },
    "id": 1
}
```

Avots

CTemplateDashboard::delete() failā *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.get

Apraksts

`integer/array templatedashboard.get(object parameters)`

Metode ļauj izgūt veidņu paneļus atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
dashboardids	ID/array	Atgriezt tikai veidņu paneļus ar norādītajiem ID.
templateids	ID/array	Atgriezt tikai tos veidņu paneļus, kas pieder norādītajām veidnēm.
selectPages	query	Atgriezt īpašību `pages` ar veidnes paneļa lapām pareizā secībā.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `dashboardid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Veidņu paneļu izgūšana

Izgūst visus norādītās veidnes paneļus ar logrīkiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.get",
    "params": {
        "output": "extend",
        "selectPages": "extend",
        "templateids": "10001"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Skatiet arī

- Veidnes paneļa lapa
- Veidnes paneļa logrīks
- Veidnes paneļa logrīka lauks

Avots

CTemplateDashboard::get() failā *ui/include/classes/api/services/CTemplateDashboard.php*.

templatedashboard.update

Apraksts

`object templatedashboard.update(object/array templateDashboards)`

Šī metode ļauj atjaunināt esošos veidņu paneļus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmās veidnes paneļa īpašības.

Īpašība `dashboardid` ir jānorāda katram panelim, visas pārējās īpašības ir neobligātas. Tiks atjauninātas tikai norādītās īpašības.

Papildus standarta veidnes paneļa īpašībām metode pieņem šādus parametrus.

Parameter	Type	Description
pages	array	Veidnes paneļa lapas, ar kurām aizstāt esošās paneļa lapas. Paneļa lapas tiek atjauninātas pēc īpašības `dashboard_pageid`. Jaunās paneļa lapas tiks izveidotas objektiem bez īpašības `dashboard_pageid`, un esošās paneļa lapas tiks dzēstas, ja tās netiks izmantotas atkārtoti. Paneļa lapas tiks sakārtotas tādā pašā secībā, kādā tās ir norādītas. Tiks atjauninātas tikai norādītās paneļa lapu īpašības. Īpašībai `pages` ir nepieciešams vismaz viens paneļa lapas objekts.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto veidnes paneļu ID īpašībā `dashboardids`.

Piemēri

Veidnes paneļa pārdēvēšana

Pārdēvējiet veidnes paneli par "Performance graphs".

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "23",
        "name": "Performance graphs"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "23"
        ]
    },
    "id": 1
}
```

Veidnes paneļa lapu atjaunināšana

Pārdēvējiet pirmo paneļa lapu, aizstājiet logrīkus otrajā paneļa lapā un pievienojiet jaunu lapu kā trešo. Dzēsiet visas pārējās paneļa lapas.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "templatedashboard.update",
    "params": {
        "dashboardid": "2",
        "pages": [
            {
                "dashboard_pageid": 1,
                "name": "Pārdēvēta lapa"
            },
            {
                "dashboard_pageid": 2,
```

```
                "widgets": [
                    {
                        "type": "clock",
                        "x": 0,
                        "y": 0,
                        "width": 12,
                        "height": 3
                    }
                ]
            },
            {
                "display_period": 60
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "dashboardids": [
            "2"
        ]
    },
    "id": 1
}
```

Skatiet arī

- Veidnes paneļa logrīks
- Veidnes paneļa logrīka lauks

Avots

CTemplateDashboard::update() failā *ui/include/classes/api/services/CTemplateDashboard.php*.

Vienuma prototips

Šī klase ir paredzēta darbam ar vienumu prototipiem.

Objektu atsauces:

- Vienuma prototips
  - HTTP galvene
  - HTTP vaicājuma lauks
- Vienuma prototipa tags
- Vienuma prototipa priekšapstrāde

Pieejamās metodes:

- itemprototype.create - izveidot jaunus vienumu prototipus
- itemprototype.delete - dzēst vienumu prototipus
- itemprototype.get - izgūt vienumu prototipus
- itemprototype.update - atjaunināt vienumu prototipus

Vienuma prototipa objekts

Tālāk norādītie objekti ir tieši saistīti ar `itemprototype` API.

Vienuma prototips

Vienuma prototipa objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
itemid	ID	Vienuma prototipa ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
delay	string	Vienuma prototipa atjaunināšanas intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus un LLD makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”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), vai ja `type` ir iestatīts uz ”Zabbix agent (active)” (7) un `key_` nesatur ”mqtt.get”
hostid	ID	Hosta ID, kuram pieder vienuma prototips. Īpašības darbība: - *konstants* - *obligāts* izveides operācijām
interfaceid	ID	Vienuma prototipa hosta saskarnes ID. Īpašības darbība: - *obligāts*, ja vienuma prototips pieder hostam un `type` ir iestatīts uz ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” vai ”SNMP agent” - *atbalstīts*, ja vienuma prototips pieder hostam un `type` ir iestatīts uz ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent”
key_	string	Vienuma prototipa atslēga. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
name	string	Vienuma prototipa nosaukums. Atbalsta lietotāja makrosus. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
type	integer	Vienuma prototipa tips. Iespējamās vērtības: 0 - Zabbix aģents; 2 - Zabbix trapper; 3 - vienkāršā pārbaude; 5 - Zabbix iekšējais; 7 - Zabbix aģents (aktīvs); 10 - ārējā pārbaude; 11 - datubāzes monitors; 12 - IPMI aģents; 13 - SSH aģents; 14 - TELNET aģents; 15 - aprēķināts; 16 - JMX aģents; 17 - SNMP trap; 18 - atkarīgais vienums; 19 - HTTP aģents; 20 - SNMP aģents; 21 - skripts; 22 - pārlūks. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
url	string	URL virkne. Atbalsta LLD makrosus, lietotāja makrosus, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
value_type	integer	Vienuma prototipa informācijas tips. Iespējamās vērtības: 0 - skaitlis ar peldošo komatu; 1 - rakstzīme; 2 - žurnāls; 3 - bezzīmes skaitlis; 4 - teksts; 5 - binārs; 6 - JSON. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem
allow_traps	integer	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. 0 - *(noklusējums)* Neatļaut pieņemt ienākošos datus; 1 - Atļaut pieņemt ienākošos datus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent”

Īpašība	Tips	Apraksts
authtype	integer	Autentifikācijas metode. Iespējamās vērtības, ja `type` ir iestatīts uz ”SSH agent”: 0 - *(noklusējums)* parole; 1 - publiskā atslēga. Iespējamās vērtības, ja `type` ir iestatīts uz ”HTTP agent”: 0 - *(noklusējums)* nav; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
description	string	Vienuma prototipa apraksts.
flags	integer	Vienuma prototipa izcelsme. Iespējamās vērtības: 2 - vienuma prototips; 6 - atklāts vienuma prototips Īpašības darbība: - *tikai lasāms*
follow_redirects	integer	Sekot atbildes pāradresācijām datu aptaujas laikā. Iespējamās vērtības: 0 - Nesekot pāradresācijām; 1 - *(noklusējums)* Sekot pāradresācijām. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
history	string	Laika vienība, kas nosaka, cik ilgi jāsaglabā vēstures dati. Pieņem arī lietotāja makrosu un LLD makrosu. Noklusējums: 31d.
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ipmi_sensor	string	IPMI sensors. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”IPMI agent” un `key_` nav iestatīts uz ”ipmi.get” - *atbalstīts*, ja `type` ir iestatīts uz ”IPMI agent” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
jmx_endpoint	string	JMX aģenta pielāgota savienojuma virkne. Noklusējums: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”JMX agent”
logtimefmt	string	Laika formāts žurnāla ierakstos. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”log” - *tikai lasāms* mantotajiem objektiem
master_itemid	ID	Galvenā vienuma ID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotajiem objektiem
output_format	integer	Vai atbilde jākonvertē uz JSON. Iespējamās vērtības: 0 - *(noklusējums)* Saglabāt neapstrādātu; 1 - Konvertēt uz JSON. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
params	string	Papildu parametri atkarībā no vienuma prototipa tipa: - izpildāmais skripts SSH aģenta un TELNET aģenta vienuma prototipiem; - SQL vaicājums datubāzes monitora vienuma prototipiem; - formula aprēķināto vienuma prototipiem; - skripts skripta un pārlūka vienuma prototipiem. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”Script” vai ”Browser”)
parameters	object/array	Papildu parametri, ja `type` ir iestatīts uz ”Script” vai ”Browser”. Objektu masīvs ar `name` un `value` īpašībām, kur `name` jābūt unikālam. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem
password	string	Parole autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”JMX agent” un `username` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
post_type	integer	`posts` īpašībā saglabātā POST datu pamatteksta tips. Iespējamās vērtības: 0 - *(noklusējums)* Neapstrādāti dati. 2 - JSON dati. 3 - XML dati. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
posts	string	HTTP(S) pieprasījuma pamatteksta dati. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”JSON data” vai ”XML data” - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” un `post_type` ir iestatīts uz ”Raw data” - *tikai lasāms* mantotajiem objektiem
privatekey	string	Privātās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
publickey	string	Publiskās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” un `authtype` ir iestatīts uz ”public key”
query_fields	array	Vaicājuma lauku masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
retrieve_mode	integer	Kura atbildes daļa jāsaglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”GET”, ”POST” vai ”PUT”: 0 - *(noklusējums)* Pamatteksts; 1 - Galvenes; 2 - Tiks saglabāts gan pamatteksts, gan galvenes. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”HEAD”: 1 - Galvenes. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
snmp_oid	string	SNMP OID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotajiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem

Īpašība	Tips	Apraksts
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
status	integer	Vienuma prototipa statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots vienuma prototips; 1 - atspējots vienuma prototips; 3 - neatbalstīts vienuma prototips.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalsta arī lietotāja makrosus vai LLD makrosus kā daļu no ar komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
templateid	ID	Vecākveidnes vienuma prototipa ID. Īpašības darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus un LLD makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: ”” - izmantot starpniekservera/globālos iestatījumus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Zabbix agent” (0), ”Simple check” (3) un `key_` nesākas ar ”`vmware.`” un ”`icmpping`”, ”Zabbix agent (active)” (7), ”External check” (10), ”Database monitor” (11), ”SSH agent” (13), ”TELNET agent” (14), ”HTTP agent” (19), ”SNMP agent” (20) un `snmp_oid` sākas ar ”`walk[`” vai ”`get[`”, ”Script” (21), ”Browser” (22) - *tikai lasāms* mantotajiem objektiem
trapper_hosts	string	Atļautie hosti. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Zabbix trapper”, vai ja `type` ir iestatīts uz ”HTTP agent” un `allow_traps` ir iestatīts uz ”Allow to accept incoming data”
trends	string	Laika vienība, kas nosaka, cik ilgi jāsaglabā tendenču dati. Pieņem arī lietotāja makrosu un LLD makrosu. Noklusējums: 365d. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned”

Īpašība	Tips	Apraksts
units	string	Vērtības mērvienības. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float” vai ”numeric unsigned” - *tikai lasāms* mantotajiem objektiem
username	string	Lietotājvārds autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SSH agent” vai ”TELNET agent”, vai ja `type` ir iestatīts uz ”JMX agent” un `password` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”Database monitor” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”)
uuid	string	Universāls unikāls identifikators, ko izmanto importēto vienumu prototipu sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja vienuma prototips pieder veidnei
valuemapid	ID	Saistītās vērtību kartes ID. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float”, ”character” vai ”numeric unsigned” - *tikai lasāms* mantotajiem objektiem
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam nosaukumam. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem
discover	integer	Vienuma prototipa atklāšanas statuss. Iespējamās vērtības: 0 - *(noklusējums)* tiks atklāti jauni vienumi; 1 - jauni vienumi netiks atklāti, un esošie vienumi tiks atzīmēti kā zaudēti.

HTTP galvene

Galvenes objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	HTTP galvenes nosaukums. Īpašības darbība: - *obligāta*
value	string	Galvenes vērtība. Īpašības darbība: - *obligāta*

HTTP vaicājuma lauks

Vaicājuma lauka objekts definē nosaukumu un vērtību, ko izmanto URL parametra norādīšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāts*
value	string	Parametra vērtība. Īpašības darbība: - *obligāts*

Vienuma prototipa tags

Vienuma prototipa taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Vienuma prototipa taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Vienuma prototipa taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne; 1 - hosts. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

Vienuma prototipa priekšapstrāde

Vienuma prototipa priekšapstrādes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 1 - Pielāgots reizinātājs; 2 - Apgriezt no labās puses; 3 - Apgriezt no kreisās puses; 4 - Apgriezt; 5 - Regulārā izteiksme; 6 - Būla vērtība uz decimāldaļu; 7 - Oktāls uz decimāldaļu; 8 - Heksadecimāls uz decimāldaļu; 9 - Vienkāršas izmaiņas; 10 - Izmaiņas sekundē; 11 - XML XPath; 12 - JSONPath; 13 - Diapazonā; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - Pārbaudīt kļūdu JSON; 17 - Pārbaudīt kļūdu XML; 18 - Pārbaudīt kļūdu, izmantojot regulāro izteiksmi; 19 - Atmest, ja nav izmaiņu; 20 - Atmest, ja nav izmaiņu, ar sirdspukstu; 21 - JavaScript; 22 - Prometheus modelis; 23 - Prometheus uz JSON; 24 - CSV uz JSON; 25 - Aizstāt; 26 - Pārbaudīt neatbalstītu; 27 - XML uz JSON; 28 - SNMP walk vērtība; 29 - SNMP walk uz JSON; 30 - SNMP get vērtība. Īpašības darbība: - *obligāta*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri ir atdalīti ar jaunās rindas (\n) rakstzīmi. Ja `type` ir iestatīts uz ”Pārbaudīt neatbalstītu”, parametri izmanto sintaksi `<scope>[\n<pattern>]`, kur *pattern* ir regulārā izteiksme, un *scope* ir viens no šiem: -1 - atbilst jebkurai kļūdai; 0 - pārbaudīt, vai kļūdas ziņojums atbilst *pattern*; 1 - pārbaudīt, vai kļūdas ziņojums neatbilst *pattern*. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Apgriezt no labās puses” (2), ”Apgriezt no kreisās puses” (3), ”Apgriezt” (4), ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Pārbaudīt kļūdu, izmantojot regulāro izteiksmi” (18), ”Atmest, ja nav izmaiņu, ar sirdspukstu” (20), ”JavaScript” (21), ”Prometheus modelis” (22), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Aizstāt” (25), ”Pārbaudīt neatbalstītu” (26), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)

Īpašība	Tips	Apraksts
error_handler	integer	Darbības tips, ko izmanto priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Iespējamās vērtības, ja `type` ir iestatīts uz ”Pārbaudīt neatbalstītu”: 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Regulārā izteiksme” (5), ”Būla vērtība uz decimāldaļu” (6), ”Oktāls uz decimāldaļu” (7), ”Heksadecimāls uz decimāldaļu” (8), ”Vienkāršas izmaiņas” (9), ”Izmaiņas sekundē” (10), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Pārbaudīt kļūdu, izmantojot regulāro izteiksmi” (18), ”Prometheus modelis” (22), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Pārbaudīt neatbalstītu” (26), ”XML uz JSON” (27), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādātāja parametri. Īpašības darbība: - *obligāta*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Katram priekšapstrādes tipam tiek atbalstīti šādi parametri un kļūdu apstrādātāji.

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
1	Pielāgots reizinātājs	skaitlis^1, 6^			0, 1, 2, 3
2	Apgriezt no labās puses	rakstzīmju saraksts[2]			
3	Apgriezt no kreisās puses	rakstzīmju saraksts[2]			
4	Apgriezt	rakstzīmju saraksts[2]			
5	Regulārā izteiksme	modelis[3]	izvade[2]		0, 1, 2, 3
6	Būla vērtība uz decimāldaļu				0, 1, 2, 3
7	Oktāls uz decimāldaļu				0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
8	Heksadecimāls uz decimāldaļu				0, 1, 2, 3
9	Vienkāršas izmaiņas				0, 1, 2, 3
10	Izmaiņas sekundē				0, 1, 2, 3
11	XML XPath	ceļš[4]			0, 1, 2, 3
12	JSONPath	ceļš[4]			0, 1, 2, 3
13	Diapazona kārtas	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	modelis[3]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	modelis[3]			0, 1, 2, 3
16	Pārbaudīt kļūdu JSON	ceļš[4]			0, 1, 2, 3
17	Pārbaudīt kļūdu XML	ceļš[4]			0, 1, 2, 3
18	Pārbaudīt kļūdu, izmantojot regulāro izteiksmi	modelis[3]	izvade[2]		0, 1, 2, 3
19	Atmest, ja nav izmaiņu				
20	Atmest, ja nav izmaiņu, ar sirdspukstu	sekundes^5, 6^			
21	JavaScript	skripts[2]			
22	Prometheus modelis	modelis^6, 7^	`value`, `label`, `function`	izvade^8, 9^	0, 1, 2, 3
23	Prometheus uz JSON	modelis^6, 7^			0, 1, 2, 3
24	CSV uz JSON	rakstzīme[2]	rakstzīme[2]	0,1	0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
25	Aizstāt	meklējamā virkne[2]	aizstāšana[2]		
26	Pārbaudīt neatbalstītu	vērtības tips[1]	modelis^3, 6^		1, 2, 3
27	XML uz JSON				0, 1, 2, 3
28	SNMP walk vērtība	OID[2]	Formāts: 0 - Bez izmaiņām 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[10]	Lauka nosaukums[2]	OID prefikss[2]	Formāts: 0 - Bez izmaiņām 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS			0, 1, 2, 3

[1] vesels skaitlis vai peldošā komata skaitlis
 [2] virkne
 [3] regulārā izteiksme
 [4] JSONPath vai XML XPath
 [5] pozitīvs vesels skaitlis (ar laika sufiksu atbalstu, piemēram, 30s, 1m, 2h, 1d)
 [6] lietotāja makross, LLD makross
 [7] Prometheus modelis, kas atbilst sintaksei: `<metric name>{<label name>="<label value>", ...} == <value>`. Katrs Prometheus modeļa komponents (metrika, etiķetes nosaukums, etiķetes vērtība un metrikas vērtība) var būt lietotāja makross vai LLD makross.
 [8] Prometheus izvade, kas atbilst sintaksei: `<label name>` (var būt lietotāja makross vai LLD makross), ja kā otrais parametrs ir atlasīts `label`.
 [9] Viena no agregācijas funkcijām: `sum`, `min`, `max`, `avg`, `count`, ja kā otrais parametrs ir atlasīts `function`.
 [10] Atbalsta vairākus "Lauka nosaukums,OID prefikss,Formāts ierakstus", kas atdalīti ar jaunās rindas rakstzīmi.

itemprototype.create

Apraksts

`object itemprototype.create(object/array itemPrototypes)`

Šī metode ļauj izveidot jaunus vienumu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Izveidojamais vienuma prototips.

Papildus standarta vienuma prototipa īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
ruleid	ID	LLD kārtulas ID, kurai pieder vienuma prototips. Parametra darbība: - *obligāts*
preprocessing	array	Vienuma prototipa priekšapstrādes opcijas.
tags	array	Vienuma prototipa birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto vienumu prototipu ID īpašībā `itemids`. Atgriezto ID secība atbilst nodoto vienumu prototipu secībai.

Piemēri

Vienuma prototipa izveide

Izveidojiet vienuma prototipu, lai uzraudzītu brīvo diska vietu atklātajā failu sistēmā. Atklātajiem vienumiem jābūt skaitliskiem Zabbix aģenta vienumiem, kas tiek atjaunināti ik pēc 30 sekundēm.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Brīvā diska vieta uz {#FSNAME}",
        "key_": "vfs.fs.size[{#FSNAME},free]",
        "hostid": "10197",
        "ruleid": "27665",
        "type": 0,
        "value_type": 3,
        "interfaceid": "112",
        "delay": "30s"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27666"
        ]
    },
    "id": 1
}
```

Vienuma prototipa izveide ar priekšapstrādi

Izveidojiet vienumu, izmantojot izmaiņas sekundē un pielāgotu reizinātāju kā otro soli.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.create",
    "params": {
        "name": "Ienākošā tīkla datplūsma uz {#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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Atkarīgā vienuma prototipa izveide

Izveidojiet atkarīgā vienuma prototipu galvenā vienuma prototipam ar ID 44211. Ir atļautas tikai atkarības uz tā paša hosta (veidnes/atklāšanas kārtulas), tādēļ galvenajam un atkarīgajam vienumam jābūt vienādam hostid un ruleid.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44212"
        ]
    },
    "id": 1
}
```

Izveidot HTTP aģenta vienuma prototipu

Izveidojiet vienuma prototipu ar URL, izmantojot lietotāja makrosu, vaicājuma laukus un pielāgotas galvenes.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Izveidot skripta vienuma prototipu

Izveidojiet vienkāršu datu vākšanu, izmantojot skripta vienuma prototipu.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Avots

CItemPrototype::create() failā *ui/include/classes/api/services/CItemPrototype.php*.

itemprototype.update

Apraksts

`object itemprototype.update(object/array itemPrototypes)`

Šī metode ļauj atjaunināt esošos vienumu prototipus.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Atjauninámo vienumu prototipu īpašības.

Īpašībai `itemid` jābūt definētai katram vienuma prototipam, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta vienumu prototipu īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
preprocessing	array	Vienuma prototipa priekšapstrādes opcijas, ar kurām aizstāt pašreizējās priekšapstrādes opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem
tags	array	Vienuma prototipa birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto vienumu prototipu ID īpašumā `itemids`.

Piemēri

Vienuma prototipa saskarnes maiņa

Mainiet hosta saskarni, kas tiks izmantota atklātajiem vienumiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "27428",
        "interfaceid": "132"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "27428"
        ]
    },
    "id": 1
}
```

Atjaunināt atkarīgā vienuma prototipu

Atjauniniet atkarīgā vienuma prototipu ar jaunu galvenā vienuma prototipa ID. Ir atļautas tikai atkarības uz tā paša hosta (veidnes/atklāšanas kārtulas), tādēļ galvenajam un atkarīgajam vienumam jābūt ar vienādu hostid un ruleid.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "master_itemid": "25570",
        "itemid": "189030"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "189030"
        ]
    },
    "id": 1
}
```

Atjaunināt HTTP aģenta vienuma prototipu

Mainīt vaicājuma laukus un noņemt visas pielāgotās galvenes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid":"28305",
        "query_fields": [
            {
                "name": "random",
                "value": "qwertyuiopasdfghjklzxcvbnm"
            }
```

```
        ],
        "headers": []
    }
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "28305"
        ]
    },
    "id": 1
}
```

Vienuma prototipa priekšapstrādes opciju atjaunināšana

Atjauniniet vienuma prototipu ar vienuma priekšapstrādes noteikumu “Pielāgots reizinātājs”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "44211",
        "preprocessing": [
            {
                "type": 1,
                "params": "4",
                "error_handler": 2,
                "error_handler_params": "5"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Skripta vienuma prototipa atjaunināšana

Atjauniniet skripta vienuma prototipu ar citu skriptu un noņemiet nevajadzīgos parametrus, kas tika izmantoti iepriekšējā skriptā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "script": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
```

```
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Avots

CItemPrototype::update() failā *ui/include/classes/api/services/CItemPrototype.php*.

vienumaprototips.delete

Apraksts

`object itemprototype.delete(array itemPrototypeIds)`

Šī metode ļauj dzēst vienumu prototipus.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo vienumu prototipu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto vienumu prototipu ID zem `prototypeids` īpašības.

Piemēri

Vairāku vienumu prototipu dzēšana

Dzēsiet divus vienumu prototipus.
 Atkarīgie vienumu prototipi tiek noņemti automātiski, ja tiek dzēsts galvenais vienums vai vienuma prototips.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.delete",
    "params": [
        "27352",
        "27356"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "prototypeids": [
            "27352",
            "27356"
        ]
    },
```

```
    "id": 1
}
```

Avots

CItemPrototype::delete() failā *ui/include/classes/api/services/CItemPrototype.php*.

vienumaprototips.get

Apraksts

`integer/array itemprototype.get(object parameters)`

Metode ļauj izgūt vienumu prototipus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
discoveryids	ID/array	Atgriezt tikai tos vienumu prototipus, kas pieder norādītajiem LLD noteikumiem.
graphids	ID/array	Atgriezt tikai tos vienumu prototipus, kas tiek izmantoti norādītajos grafiku prototipos.
hostids	ID/array	Atgriezt tikai tos vienumu prototipus, kas pieder norādītajiem hostiem.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai no veidnes mantotos vienumu prototipus.
itemids	ID/array	Atgriezt tikai tos vienumu prototipus, kuriem ir norādītie ID.
monitored	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotos vienumu prototipus, kas pieder uzraudzītiem hostiem.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai tos vienumu prototipus, kas pieder veidnēm.
templateids	ID/array	Atgriezt tikai tos vienumu prototipus, kas pieder norādītajām veidnēm.
triggerids	ID/array	Atgriezt tikai tos vienumu prototipus, kas tiek izmantoti norādītajos trigeru prototipos.
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar vienumu prototipa atklāšanas objekta datiem. Vienumu prototipa atklāšanas objekts sasaista atklātu vienumu prototipu ar vienumu prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_itemid` - `(string)` tā vienumu prototipa ID, no kura vienumu prototips tika izveidots; `key_` - `(string)` vienumu prototipa atslēga; `status` - `(int)` vienumu prototipa atklāšanas statuss: 0 - *(noklusējums)* vienumu prototips ir atklāts, 1 - vienumu prototips vairs nav atklāts; `ts_delete` - `(timestamp)` laiks, kad vienumu prototips, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad vienumu prototips, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` norāde, vai vienumu prototips tika atspējots ar LLD noteikumu vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD noteikumu.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar zema līmeņa atklāšanas noteikumu, kuram pieder vienumu prototips.
selectDiscoveryRulePrototype	query	Atgriezt `discoveryRulePrototype` īpašību ar vecāka LLD noteikuma prototipu, kuram pieder vienumu prototips.

Parametrs	Tips	Apraksts
selectGraphs	query	Atgriezt `graphs` īpašību ar grafiku prototipiem, kuros tiek izmantots vienumu prototips. Atbalsta `count`.
selectHosts	query	Atgriezt `hosts` īpašību ar hostu masīvu, kuriem pieder vienumu prototips.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar birkām, kas atrodas uz veidnes/hosta/piesaistītajām veidnēm.
selectTags	query	Atgriezt vienumu prototipa birkas īpašībā `tags`.
selectTriggers	query	Atgriezt `triggers` īpašību ar trigeru prototipiem, kuros tiek izmantots vienumu prototips. Atbalsta `count`.
selectPreprocessing	query	Atgriezt `preprocessing` īpašību ar vienumu prototipa priekšapstrādes opcijām.
selectValueMap	query	Atgriezt `valuemap` īpašību ar vienumu prototipa vērtību kartējumu.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kur atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram pieder vienumu prototips.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādiem apakšatlases vaicājumiem: `selectGraphs` - rezultāti tiks kārtoti pēc `name`; `selectTriggers` - rezultāti tiks kārtoti pēc `description`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `name`, `key_`, `delay`, `type`, `status`, `history`, `trends`, `discovered`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Vienumu prototipu izgūšana no LLD kārtulas

Izgūst visus vienumu prototipus norādītajam LLD kārtulas ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
```

```
    "params": {
        "output": "extend",
        "discoveryids": "27426"
    },
    "id": 1
}
```

Atbilde:


```
{
    "jsonrpc": "2.0",
    "result": [
        {
            "itemid": "23077",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10079",
            "name": "Incoming network traffic on en0",
            "key_": "net.if.in[en0]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "3",
            "trapper_hosts": "",
            "units": "bps",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "",
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
```

```
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
            "enabled_lifetime": "0",
            "parameters": []
        },
        {
            "itemid": "10010",
            "type": "0",
            "snmp_oid": "",
            "hostid": "10001",
            "name": "Processor load (1 min average per core)",
            "key_": "system.cpu.load[percpu,avg1]",
            "delay": "1m",
            "history": "1w",
            "trends": "365d",
            "status": "0",
            "value_type": "0",
            "trapper_hosts": "",
            "units": "",
            "formula": "",
            "logtimefmt": "",
            "templateid": "0",
            "valuemapid": "0",
            "params": "",
            "ipmi_sensor": "",
            "authtype": "0",
            "username": "",
            "password": "",
            "publickey": "",
            "privatekey": "",
            "flags": "2",
            "interfaceid": "0",
            "description": "The processor load is calculated as system CPU load divided by number of CPU c
            "inventory_link": "0",
            "lifetime": "7d",
            "evaltype": "0",
            "jmx_endpoint": "",
            "master_itemid": "0",
            "timeout": "",
            "url": "",
            "query_fields": [],
            "posts": "",
            "status_codes": "200",
            "follow_redirects": "1",
            "post_type": "0",
            "http_proxy": "",
            "headers": [],
            "retrieve_mode": "0",
            "request_method": "0",
            "output_format": "0",
            "ssl_cert_file": "",
            "ssl_key_file": "",
            "ssl_key_password": "",
            "verify_peer": "0",
            "verify_host": "0",
            "allow_traps": "0",
            "discover": "0",
            "uuid": "",
            "lifetime_type": "0",
            "enabled_lifetime_type": "2",
```

```
            "enabled_lifetime": "0",
            "parameters": []
        }
    ],
    "id": 1
}
```

Atkarīgā vienuma atrašana

Atrodiet vienu atkarīgo vienumu konkrētam vienuma ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "output": "extend",
        "filter": {
            "type": 18,
            "master_itemid": "25545"
        },
        "limit": "1"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Atrast HTTP aģenta vienuma prototipu

Atrodiet HTTP aģenta vienuma prototipu ar pieprasījuma metodi HEAD konkrētam hosta ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "itemprototype.get",
    "params": {
        "hostids": "10254",
        "filter": {
            "type": 19,
            "request_method": 3
        }
    },
    "id": 1
}
```

Atbilde:


```
{
    "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
}
```


Skatiet arī

- Hosts
- Grafika prototips
- Trigera prototips

Avots

CItemPrototype::get() failā *ui/include/classes/api/services/CItemPrototype.php*.

Vienums

Šī klase ir paredzēta darbam ar vienumiem.

Objektu atsauces:

- Vienums
  - HTTP galvene
  - HTTP vaicājuma lauks
- Vienuma tags
- Vienuma priekšapstrāde

Pieejamās metodes:

- item.create - izveidot jaunus vienumus
- item.delete - dzēst vienumus
- item.get - izgūt vienumus
- item.update - atjaunināt vienumus

Vienuma objekts

Tālāk norādītie objekti ir tieši saistīti ar `vienums` API.

Vienums

Note:
Tīmekļa vienumus nevar tieši izveidot, atjaunināt vai dzēst, izmantojot Zabbix API.

Vienuma objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
itemid	ID	Vienuma ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas operācijām
delay	string	Vienuma atjaunināšanas intervāls. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m, 2h, 1d) un pēc izvēles vienu vai vairākus pielāgotus intervālus, kas visi atdalīti ar semikoliem. Pielāgotie intervāli var būt elastīgo un plānošanas intervālu kombinācija. Pieņem lietotāja makrosus. Ja tie tiek izmantoti, vērtībai jābūt vienam makrosam. Vairāki makrosi vai makrosi kopā ar tekstu netiek atbalstīti. Elastīgos intervālus var rakstīt kā divus makrosus, kas atdalīti ar slīpsvītru (piemēram, `{$FLEX_INTERVAL}/{$FLEX_PERIOD}`). Piemērs: `1h;wd1-5h9-18;{$Macro1}/1-7,00:00-24:00;0/6-7,12:00-24:00;{$Macr` Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "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), vai ja `type` ir iestatīts uz "Zabbix agent (active)" (7) un `key_` nesatur "mqtt.get"
hostid	ID	Hosta vai veidnes ID, kam pieder vienums. Īpašības darbība: - *nemainīgs* - *obligāts* izveides operācijām

Īpašība	Tips	Apraksts
interfaceid	ID	Vienuma hosta saskarnes ID. Īpašības darbība: - *obligāts*, ja vienums pieder hostam un `type` ir iestatīts uz ”Zabbix agent”, ”IPMI agent”, ”JMX agent”, ”SNMP trap” vai ”SNMP agent” - *atbalstīts*, ja vienums pieder hostam un `type` ir iestatīts uz ”Simple check”, ”External check”, ”SSH agent”, ”TELNET agent” vai ”HTTP agent” - *tikai lasāms* atklātajiem objektiem
key_	string	Vienuma atslēga. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
name	string	Vienuma nosaukums. Atbalsta lietotāja makrosus. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
name_resolved	string	Vienuma nosaukums ar atrisinātiem lietotāja makrosiem. Īpašības darbība: - *tikai lasāms*
type	integer	Vienuma tips. Iespējamās vērtības: 0 - Zabbix aģents; 2 - Zabbix trapper; 3 - vienkāršā pārbaude; 5 - Zabbix iekšējais; 7 - Zabbix aģents (aktīvs); 9 - tīmekļa vienums; 10 - ārējā pārbaude; 11 - datubāzes monitors; 12 - IPMI aģents; 13 - SSH aģents; 14 - TELNET aģents; 15 - aprēķinātais; 16 - JMX aģents; 17 - SNMP slazds; 18 - atkarīgais vienums; 19 - HTTP aģents; 20 - SNMP aģents; 21 - skripts; 22 - pārlūks. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
url	string	URL virkne. Atbalsta lietotāja makrosus, {HOST.IP}, {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.NAME}, {HOST.PORT}, {ITEM.ID}, {ITEM.KEY}. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem

Īpašība	Tips	Apraksts
value_type	integer	Vienuma informācijas tips. Iespējamās vērtības: 0 - skaitlis ar peldošo komatu; 1 - rakstzīme; 2 - žurnāls; 3 - bezparaksta skaitlis; 4 - teksts; 5 - binārs; 6 - JSON. Īpašības darbība: - *obligāts* izveides operācijām - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
allow_traps	integer	Atļaut aizpildīt vērtību līdzīgi kā trapper vienumam. 0 - *(noklusējums)* Neatļaut pieņemt ienākošos datus; 1 - Atļaut pieņemt ienākošos datus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* atklātajiem objektiem
authtype	integer	Autentifikācijas metode. Iespējamās vērtības, ja `type` ir iestatīts uz ”SSH agent”: 0 - *(noklusējums)* parole; 1 - publiskā atslēga. Iespējamās vērtības, ja `type` ir iestatīts uz ”HTTP agent”: 0 - *(noklusējums)* nav; 1 - basic; 2 - NTLM; 3 - Kerberos; 4 - Digest. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”SSH agent” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”) vai atklātajiem objektiem
description	string	Vienuma apraksts. Īpašības darbība: - *tikai lasāms* atklātajiem objektiem
error	string	Kļūdas teksts, ja rodas problēmas ar vienuma vērtības atjaunināšanu. Īpašības darbība: - *tikai lasāms*
flags	integer	Vienuma izcelsme. Iespējamās vērtības: 0 - parasts vienums; 4 - vienums, kas pārveidots no prototipa. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
follow_redirects	integer	Sekot atbildes pāradresācijām datu aptaujas laikā. Iespējamās vērtības: 0 - Nesekot pāradresācijām; 1 - *(noklusējums)* Sekot pāradresācijām. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
headers	array	Galveņu masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
history	string	Laika vienība, cik ilgi jāglabā vēstures dati. Pieņem arī lietotāja makrosu. Noklusējums: `31d`. Īpašības darbība: - *tikai lasāms* atklātajiem objektiem
http_proxy	string	HTTP(S) starpniekservera savienojuma virkne. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
inventory_link	integer	Hosta inventāra lauka ID, ko aizpilda vienums. Skatiet hosta inventāra lapu, lai iegūtu atbalstīto hosta inventāra lauku un to ID sarakstu. Noklusējums: `0`. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz "numeric float", "character", "numeric unsigned" vai "text" - *tikai lasāms* atklātajiem objektiem
ipmi_sensor	string	IPMI sensors. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "IPMI agent" un `key_` nav iestatīts uz "ipmi.get" - *atbalstīts*, ja `type` ir iestatīts uz "IPMI agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
jmx_endpoint	string	JMX aģenta pielāgota savienojuma virkne. Noklusējuma vērtība: service:jmx:rmi:///jndi/rmi://{HOST.CONN}:{HOST.PORT}/jmxrmi Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "JMX agent" - *tikai lasāms* atklātajiem objektiem
lastclock	timestamp	Laiks, kad vienuma vērtība pēdējo reizi tika atjaunināta. Pēc noklusējuma tiek parādītas tikai tās vērtības, kas ietilpst pēdējās 24 stundās. Jūs varat pagarināt šo laika periodu, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration → General*. Īpašības darbība: - *tikai lasāms*

Īpašība	Tips	Apraksts
lastns	integer	Nanosekundes, kad vienuma vērtība pēdējo reizi tika atjaunināta. Pēc noklusējuma tiek parādītas tikai tās vērtības, kas ietilpst pēdējās 24 stundās. Jūs varat pagarināt šo laika periodu, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration → General*. Īpašības darbība: - *tikai lasāms*
lastvalue	string	Vienuma pēdējā vērtība. Pēc noklusējuma tiek parādītas tikai tās vērtības, kas ietilpst pēdējās 24 stundās. Jūs varat pagarināt šo laika periodu, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration → General*. Īpašības darbība: - *tikai lasāms*
logtimefmt	string	Laika formāts žurnāla ierakstos. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”log” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
master_itemid	ID	Galvenā vienuma ID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Dependent item” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
output_format	integer	Vai atbilde jāpārveido uz JSON. 0 - *(noklusējums)* Glabāt neapstrādātu; 1 - Pārveidot uz JSON. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
params	string	Papildu parametri atkarībā no vienuma tipa: - izpildāmais skripts SSH aģenta un TELNET aģenta vienumiem; - SQL vaicājums datubāzes monitora vienumiem; - formula aprēķinātajiem vienumiem; - skripts skripta un pārlūka vienumiem. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”Database monitor”, ”SSH agent”, ”TELNET agent”, ”Calculated”, ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”Script” vai ”Browser”) vai atklātajiem objektiem
parameters	object/array	Papildu parametri, ja `type` ir iestatīts uz ”Script” vai ”Browser”. Objektu masīvs ar `name` un `value` īpašībām, kur `name` jābūt unikālam. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”Script” vai ”Browser” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
password	string	Parole autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”JMX agent” un `username` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz ”Simple check”, ”SSH agent”, ”TELNET agent”, ”Database monitor” vai ”HTTP agent” - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz ”HTTP agent”) vai atklātajiem objektiem

Īpašība	Tips	Apraksts
post_type	integer	`posts` īpašībā glabātā POST datu ķermeņa tips. Iespējamās vērtības: 0 - *(noklusējums)* Neapstrādāti dati; 2 - JSON dati; 3 - XML dati. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
posts	string	HTTP(S) pieprasījuma ķermeņa dati. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "HTTP agent" un `post_type` ir iestatīts uz "JSON data" vai "XML data" - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" un `post_type` ir iestatīts uz "Raw data" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
prevvalue	string	Vienuma iepriekšējā vērtība. Pēc noklusējuma tiek parādītas tikai tās vērtības, kas ietilpst pēdējās 24 stundās. Jūs varat pagarināt šo laika periodu, mainot parametra *Max history display period* vērtību izvēlnes sadaļā *Administration → General*. Īpašības darbība: - *tikai lasāms*
privatekey	string	Privātās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent" un `authtype` ir iestatīts uz "public key" - *tikai lasāms* atklātajiem objektiem
publickey	string	Publiskās atslēgas faila nosaukums. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent" un `authtype` ir iestatīts uz "public key" - *tikai lasāms* atklātajiem objektiem
query_fields	array	Vaicājuma lauku masīvs, kas tiks nosūtīts, izpildot HTTP pieprasījumu. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
request_method	integer	Pieprasījuma metodes tips. Iespējamās vērtības: 0 - *(noklusējums)* GET; 1 - POST; 2 - PUT; 3 - HEAD. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "HTTP agent" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem

Īpašība	Tips	Apraksts
retrieve_mode	integer	Kura atbildes daļa jāglabā. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”GET”, ”POST” vai ”PUT”: 0 - *(noklusējums)* Ķermenis; 1 - Galvenes; 2 - Tiks glabāts gan ķermenis, gan galvenes. Iespējamās vērtības, ja `request_method` ir iestatīts uz ”HEAD”: 1 - Galvenes. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
snmp_oid	string	SNMP OID. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz ”SNMP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
ssl_cert_file	string	Publiskās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
ssl_key_file	string	Privātās SSL atslēgas faila ceļš. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
ssl_key_password	string	SSL atslēgas faila parole. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
state	integer	Vienuma stāvoklis. Iespējamās vērtības: 0 - *(noklusējums)* normāls; 1 - neatbalstīts. Īpašības darbība: - *tikai lasāms*
status	integer	Vienuma statuss. Iespējamās vērtības: 0 - *(noklusējums)* iespējots vienums; 1 - atspējots vienums.
status_codes	string	Nepieciešamo HTTP statusa kodu diapazoni, atdalīti ar komatiem. Atbalsta arī lietotāja makrosus kā daļu no ar komatiem atdalīta saraksta. Piemērs: 200,200-{$M},{$M},200-400 Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem

Īpašība	Tips	Apraksts
templateid	ID	Vecākveidnes vienuma ID. *Padoms*: Izmantojiet `hostid` īpašību, lai norādītu veidni, kurai vienums pieder. Īpašības darbība: - *tikai lasāms*
timeout	string	Vienuma datu aptaujas pieprasījuma noildze. Pieņem sekundes vai laika vienību ar sufiksu (piemēram, 30s, 1m). Pieņem arī lietotāja makrosus. Iespējamo vērtību diapazons: 1-600s. Noklusējums: "" - izmantot starpniekservera/globālos iestatījumus. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix agent" (0), "Simple check" (3) un `key_` nesākas ar "`vmware.`" un "`icmpping`", "Zabbix agent (active)" (7), "External check" (10), "Database monitor" (11), "SSH agent" (13), "TELNET agent" (14), "HTTP agent" (19), "SNMP agent" (20) un `snmp_oid` sākas ar "`walk[`" vai "`get[`", "Script" (21), "Browser" (22) - *tikai lasāms* mantotajiem un atklātajiem objektiem
trapper_hosts	string	Atļautie hosti. Īpašības darbība: - *tikai lasāms* atklātajiem objektiem - *atbalstīts*, ja `type` ir iestatīts uz "Zabbix trapper", vai ja `type` ir iestatīts uz "HTTP agent" un `allow_traps` ir iestatīts uz "Allow to accept incoming data"
trends	string	Laika vienība, cik ilgi jāglabā tendenču dati. Pieņem arī lietotāja makrosu. Noklusējums: 365d. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz "numeric float" vai "numeric unsigned" - *tikai lasāms* atklātajiem objektiem
units	string	Vērtības mērvienības. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz "numeric float" vai "numeric unsigned" - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
username	string	Lietotājvārds autentifikācijai. Īpašības darbība: - *obligāts*, ja `type` ir iestatīts uz "SSH agent", "TELNET agent", vai ja `type` ir iestatīts uz "JMX agent" un `password` ir iestatīts - *atbalstīts*, ja `type` ir iestatīts uz "Simple check", "Database monitor" vai "HTTP agent" - *tikai lasāms* mantotajiem objektiem (ja `type` ir iestatīts uz "HTTP agent") vai atklātajiem objektiem
uuid	string	Universāls unikāls identifikators, ko izmanto importēta vienuma sasaistīšanai ar jau esošajiem. Ja nav norādīts, tiek ģenerēts automātiski. Īpašības darbība: - *atbalstīts*, ja vienums pieder veidnei

Īpašība	Tips	Apraksts
valuemapid	ID	Saistītās vērtību kartes ID. Īpašības darbība: - *atbalstīts*, ja `value_type` ir iestatīts uz ”numeric float”, ”character” vai ”numeric unsigned” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
verify_host	integer	Vai pārbaudīt, ka savienojuma hosta nosaukums atbilst hosta sertifikātā norādītajam. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
verify_peer	integer	Vai pārbaudīt, ka hosta sertifikāts ir autentisks. Iespējamās vērtības: 0 - *(noklusējums)* Nepārbaudīt; 1 - Pārbaudīt. Īpašības darbība: - *atbalstīts*, ja `type` ir iestatīts uz ”HTTP agent” - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem

HTTP galvene

Galvenes objektam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	HTTP galvenes nosaukums. Īpašības darbība: - *obligāts*
value	string	Galvenes vērtība. Īpašības darbība: - *obligāts*

HTTP vaicājuma lauks

Vaicājuma lauka objekts definē nosaukumu un vērtību, ko izmanto URL parametra norādīšanai. Tam ir šādas īpašības:

Īpašība	Tips	Apraksts
name	string	Parametra nosaukums. Īpašības darbība: - *obligāta*
value	string	Parametra vērtība. Īpašības darbība: - *obligāta*

Vienuma tags

Vienuma taga objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
tag	string	Vienuma taga nosaukums. Īpašības darbība: - *obligāts*
value	string	Vienuma taga vērtība. Īpašības darbība: - *tikai lasāms*.
object	integer	Objekta tips, no kura tags tika mantots. Iespējamās vērtības: 0 - veidne; 1 - hosts. Īpašības darbība: - *tikai lasāms*.
objectid	ID	Objekta ID, no kura tags tika mantots. Īpašības darbība: - *tikai lasāms*.

Vienuma priekšapstrāde

Vienuma priekšapstrādes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
type	integer	Priekšapstrādes opcijas tips. Iespējamās vērtības: 1 - Pielāgots reizinātājs; 2 - Apgriezt pa labi; 3 - Apgriezt pa kreisi; 4 - Apgriezt; 5 - Regulārā izteiksme; 6 - Būla vērtība uz decimāldaļu; 7 - Oktāls uz decimāldaļu; 8 - Heksadecimāls uz decimāldaļu; 9 - Vienkārša izmaiņa; 10 - Izmaiņa sekundē; 11 - XML XPath; 12 - JSONPath; 13 - Diapazonā; 14 - Atbilst regulārajai izteiksmei; 15 - Neatbilst regulārajai izteiksmei; 16 - Pārbaudīt kļūdu JSON; 17 - Pārbaudīt kļūdu XML; 18 - Pārbaudīt kļūdu, izmantojot regulāro izteiksmi; 19 - Atmest, ja nav mainījies; 20 - Atmest, ja nav mainījies, ar sirdspukstu; 21 - JavaScript; 22 - Prometheus raksts; 23 - Prometheus uz JSON; 24 - CSV uz JSON; 25 - Aizstāt; 26 - Pārbaudīt neatbalstītu; 27 - XML uz JSON; 28 - SNMP walk vērtība; 29 - SNMP walk uz JSON; 30 - SNMP get vērtība. Īpašības darbība: - *obligāta*
params	string	Papildu parametri, ko izmanto priekšapstrādes opcija. Vairāki parametri tiek atdalīti ar jaunās rindas (`\n`) rakstzīmi. Ja `type` ir iestatīts uz ”Pārbaudīt neatbalstītu”, parametri izmanto sintaksi `<scope>[\n<pattern>]`, kur *pattern* ir regulārā izteiksme un *scope* ir viens no šiem: -1 - atbilst jebkurai kļūdai; 0 - pārbaudīt, vai kļūdas ziņojums atbilst *pattern*; 1 - pārbaudīt, vai kļūdas ziņojums neatbilst *pattern*. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Apgriezt pa labi” (2), ”Apgriezt pa kreisi” (3), ”Apgriezt” (4), ”Regulārā izteiksme” (5), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Pārbaudīt kļūdu, izmantojot regulāro izteiksmi” (18), ”Atmest, ja nav mainījies, ar sirdspukstu” (20), ”JavaScript” (21), ”Prometheus raksts” (22), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Aizstāt” (25), Check unsupported (26), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)

Īpašība	Tips	Apraksts
error_handler	integer	Darbības tips, ko izmanto priekšapstrādes soļa kļūmes gadījumā. Iespējamās vērtības: 0 - Kļūdas ziņojumu iestata Zabbix serveris; 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Iespējamās vērtības, ja `type` ir iestatīts uz ”Pārbaudīt neatbalstītu”: 1 - Atmest vērtību; 2 - Iestatīt pielāgotu vērtību; 3 - Iestatīt pielāgotu kļūdas ziņojumu. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”Pielāgots reizinātājs” (1), ”Regulārā izteiksme” (5), ”Būla vērtība uz decimāldaļu” (6), ”Oktāls uz decimāldaļu” (7), ”Heksadecimāls uz decimāldaļu” (8), ”Vienkārša izmaiņa” (9), ”Izmaiņa sekundē” (10), ”XML XPath” (11), ”JSONPath” (12), ”Diapazonā” (13), ”Atbilst regulārajai izteiksmei” (14), ”Neatbilst regulārajai izteiksmei” (15), ”Pārbaudīt kļūdu JSON” (16), ”Pārbaudīt kļūdu XML” (17), ”Pārbaudīt kļūdu, izmantojot regulāro izteiksmi” (18), ”Prometheus raksts” (22), ”Prometheus uz JSON” (23), ”CSV uz JSON” (24), ”Pārbaudīt neatbalstītu” (26), ”XML uz JSON” (27), ”SNMP walk vērtība” (28), ”SNMP walk uz JSON” (29) vai ”SNMP get vērtība” (30)
error_handler_params	string	Kļūdu apstrādātāja parametri. Īpašības darbība: - *obligāta*, ja `error_handler` ir iestatīts uz ”Iestatīt pielāgotu vērtību” vai ”Iestatīt pielāgotu kļūdas ziņojumu”

Katram priekšapstrādes tipam tiek atbalstīti šādi parametri un kļūdu apstrādātāji.

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
1	Pielāgots reizinātājs	skaitlis^1, 6^			0, 1, 2, 3
2	Apgriezt pa labi	raksts^2^			
3	Apgriezt pa kreisi	raksts^2^			
4	Apgriezt	raksts^2^			
5	Regulārā izteiksme	raksts^3^	izvade^2^		0, 1, 2, 3
6	Būla vērtība uz decimāldaļu				0, 1, 2, 3
7	Oktāls uz decimāldaļu				0, 1, 2, 3

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
8	Heksadecimāls uz decimāldaļu				0, 1, 2, 3
9	Vienkārša izmaiņa				0, 1, 2, 3
10	Izmaiņa sekundē				0, 1, 2, 3
11	XML XPath	ceļš[4]			0, 1, 2, 3
12	JSONPath	ceļš[4]			0, 1, 2, 3
13	Diapazona	min^1, 6^	max^1, 6^		0, 1, 2, 3
14	Atbilst regulārajai izteiksmei	raksts[3]			0, 1, 2, 3
15	Neatbilst regulārajai izteiksmei	raksts[3]			0, 1, 2, 3
16	Pārbaudīt kļūdu JSON	ceļš[4]			0, 1, 2, 3
17	Pārbaudīt kļūdu XML	ceļš[4]			0, 1, 2, 3
18	Pārbaudīt kļūdu, izmantojot regulāro izteiksmi	raksts[3]	izvade[2]		0, 1, 2, 3
19	Atmest, ja nav mainījies				
20	Atmest, ja nav mainījies, ar sirdspukstu	sekundes^5, 6^			
21	JavaScript	skripts[2]			
22	Prometheus rakstz	raksts^6, 7^	`value`, `label`, `function`	izvade^8, 9^	0, 1, 2, 3
23	Prometheus uz JSON	raksts^6, 7^			0, 1, 2, 3
24	CSV uz JSON	rakstzīme[2]	rakstzīme[2]	0,1	0, 1, 2, 3
25	Aizstāt	meklējamā virkne[2]	aizvietojums[2]		

Priekšapstrādes tips	Nosaukums	1. parametrs	2. parametrs	3. parametrs	Atbalstītie kļūdu apstrādātāji
26	Pārbaudēt neatbalstītu	Scope[1]	raksts^3, 6^		1, 2, 3
27	XML uz JSON				0, 1, 2, 3
28	SNMP walk vērtība	OID[2]	Formāts: 0 - Bez izmaiņām 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS		0, 1, 2, 3
29	SNMP walk uz JSON[10]	Lauka nosaukums[2]	OID prefikss[2]	Formāts: 0 - Bez izmaiņām 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS	0, 1, 2, 3
30	SNMP get vērtība	Formāts: 1 - UTF-8 no Hex-STRING 2 - MAC no Hex-STRING 3 - Vesels skaitlis no BITS			0, 1, 2, 3

[1] peldošā komata skaitlis (veselie skaitļi tiek netieši pārveidoti par float vērtībām)
 [2] virkne
 [3] regulārā izteiksme
 [4] JSONPath vai XML XPath
 [5] pozitīvs vesels skaitlis (ar laika sufiksu atbalstu, piemēram, 30s, 1m, 2h, 1d)
 [6] lietotāja makross
 [7] Prometheus raksts ar šādu sintaksi: `<metric name>{<label name>="<label value>", ...} == <value>`. Katrs Prometheus raksta komponents (metrika, etiķetes nosaukums, etiķetes vērtība un metrikas vērtība) var būt lietotāja makross.
 [8] Prometheus izvade ar šādu sintaksi: `<label name>` (var būt lietotāja makross), ja kā otrais parametrs ir atlasīts `label`.
 [9] Viena no agregācijas funkcijām: `sum`, `min`, `max`, `avg`, `count`, ja kā otrais parametrs ir atlasīts `function`.
 [10] Atbalsta vairākus "Lauka nosaukums,OID prefikss,Formāts ierakstus", kas atdalīti ar jaunās rindas rakstzīmi.

item.get

Apraksts

`integer/array item.get(object parameters)`

Metode ļauj izgūt vienumus atbilstoši norādītajiem parametriem.

> **Note:**
> Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
itemids	ID/array	Atgriezt tikai vienumus ar norādītajiem ID.

Parametrs	Tips	Apraksts
groupids	ID/array	Atgriezt tikai vienumus, kas pieder hostiem no norādītajām grupām.
templateids	ID/array	Atgriezt tikai vienumus, kas pieder norādītajām veidnēm.
hostids	ID/array	Atgriezt tikai vienumus, kas pieder norādītajiem hostiem.
proxyids	ID/array	Atgriezt tikai vienumus, kurus uzrauga norādītie starpniekserveri.
interfaceids	ID/array	Atgriezt tikai vienumus, kas izmanto norādītās hosta saskarnes.
graphids	ID/array	Atgriezt tikai vienumus, kas tiek izmantoti norādītajos grafikos.
triggerids	ID/array	Atgriezt tikai vienumus, kas tiek izmantoti norādītajos trigeros.
webitems	flag	Iekļaut rezultātā tīmekļa vienumus.
inherited	boolean	Ja iestatīts uz `true`, atgriezt tikai vienumus, kas mantoti no veidnes.
inheritedTags	boolean	Atgriezt vienumus, kuriem norādītie `tags` ir arī hostā/veidnē/piesaistītajās veidnēs. Iespējamās vērtības: `true` - veidnei/hostam/piesaistītajām veidnēm arī jābūt norādītajiem tagiem; `false` - *(noklusējums)* tagi no veidnes/hosta/piesaistītajām veidnēm tiek ignorēti.
templated	boolean	Ja iestatīts uz `true`, atgriezt tikai vienumus, kas pieder veidnēm.
monitored	boolean	Ja iestatīts uz `true`, atgriezt tikai iespējotus vienumus, kas pieder uzraudzītiem hostiem.
group	string	Atgriezt tikai vienumus, kas pieder grupai ar norādīto nosaukumu.
host	string	Atgriezt tikai vienumus, kas pieder hostam ar norādīto nosaukumu.
evaltype	integer	Tagu novērtēšanas metode. Iespējamās vērtības: 0 - *(noklusējums)* Un/Vai; 2 - Vai.
tags	array	Atgriezt tikai vienumus ar norādītajiem tagiem. Formāts: `[{"tag": "<tag>", "value": "<value>", "operator": "<operator>"}, ...]`. Tukšs masīvs atgriež visus vienumus. Iespējamās operator vērtības: 0 - *(noklusējums)* Satur; 1 - Vienāds ar; 2 - Nesatur; 3 - Nav vienāds ar; 4 - Eksistē; 5 - Neeksistē.
with_triggers	boolean	Ja iestatīts uz `true`, atgriezt tikai vienumus, kas tiek izmantoti trigeros.
selectHosts	query	Atgriezt `hosts` īpašību ar hostu masīvu, kuriem vienums pieder.
selectInterfaces	query	Atgriezt `interfaces` īpašību ar hosta saskarņu masīvu, ko izmanto vienums.
selectTriggers	query	Atgriezt `triggers` īpašību ar trigeriem, kuros vienums tiek izmantots. Atbalsta `count`.
selectGraphs	query	Atgriezt `graphs` īpašību ar grafikiem, kuros ir iekļauts vienums. Atbalsta `count`.

Parametrs	Tips	Apraksts
selectDiscoveryData	query	Atgriezt `discoveryData` īpašību ar vienuma atklāšanas objekta datiem. Vienuma atklāšanas objekts sasaista atklātu vienumu ar vienuma prototipu, no kura tas tika atklāts. Tam ir šādas īpašības: `parent_itemid` - `(string)` vienuma prototipa ID, no kura vienums ir izveidots; `key_` - `(string)` vienuma prototipa atslēga; `status` - `(int)` vienuma atklāšanas statuss: 0 - *(noklusējums)* vienums ir atklāts, 1 - vienums vairs netiek atklāts; `ts_delete` - `(timestamp)` laiks, kad vienums, kas vairs netiek atklāts, tiks dzēsts; `ts_disable` - `(timestamp)` laiks, kad vienums, kas vairs netiek atklāts, tiks atspējots; `disable_source` - `(int)` indikators, vai vienumu atspējoja LLD kārtula vai manuāli: 0 - *(noklusējums)* atspējots automātiski, 1 - atspējots ar LLD kārtulu.
selectDiscoveryRule	query	Atgriezt `discoveryRule` īpašību ar LLD kārtulu, kas izveidoja vienumu.
selectInheritedTags	query	Atgriezt `inheritedTags` īpašību ar tagiem, kas atrodas veidnē/hostā/piesaistītajās veidnēs.
selectPreprocessing	query	Atgriezt `preprocessing` īpašību ar vienuma priekšapstrādes opcijām.
selectTags	query	Atgriezt vienuma tagus `tags` īpašībā.
selectValueMap	query	Atgriezt `valuemap` īpašību ar vienuma vērtību kartējumu.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir vai nu viena vērtība, vai vērtību masīvs salīdzināšanai. Neatbalsta `text` datu tipa īpašības. Atbalsta papildu īpašības: `host` - hosta tehniskais nosaukums, kuram vienums pieder.
limitSelects	integer	Ierobežo apakšatlases vaicājumos atgriezto ierakstu skaitu. Attiecas uz šādām apakšatlasēm: `selectGraphs` - rezultāti tiks kārtoti pēc `name`; `selectTriggers` - rezultāti tiks kārtoti pēc `description`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `name`, `key_`, `delay`, `history`, `trends`, `type`, `status`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	
selectItemDiscovery	query	Atgriezt `itemDiscovery` īpašību ar vienuma atklāšanas objektu. Vienuma atklāšanas objekts sasaista vienumu ar vienuma prototipu, no kura tas tika izveidots. Šis vaicājums ir **novecojis**, lūdzu, tā vietā izmantojiet `selectDiscoveryData`.

Atgrieztās vērtības

`(integer/array)` Atgriež vai nu:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Vienumu atrašana pēc atslēgas

Iegūstiet visus vienumus, kas tiek izmantoti trigeros norādītajam hosta ID, kuru vienuma atslēga satur vārdu ”system.cpu”, un kārtojiet rezultātus pēc nosaukuma.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10084",
        "with_triggers": true,
        "search": {
            "key_": "system.cpu"
        },
        "sortfield": "name"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Atkarīgo vienumu atrašana pēc atslēgas

Iegūstiet visus atkarīgos vienumus no hosta ar ID ”10116”, kuru atslēga satur vārdu ”apache”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": "extend",
        "hostids": "10116",
        "search": {
            "key_": "apache"
        },
        "filter": {
            "type": 18
        }
    },
```

```
    "id": 1
}
```

Atbilde:


```
{
    "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
}
```

Atrast HTTP aģenta vienumu

Atrast HTTP aģenta vienumu ar XML ziņojuma pamatteksta tipu konkrētam hosta ID.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "hostids": "10255",
        "filter": {
            "type": 19,
            "post_type": 3
        }
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Vienumu izgūšana ar priekšapstrādes noteikumiem

Izgūstiet visus vienumus un to priekšapstrādes noteikumus norādītajam hosta ID.

Pieprasījums:


```
{
    "jsonrpc": "2.0",
    "method": "item.get",
    "params": {
        "output": ["itemid", "name", "key_"],
        "selectPreprocessing": "extend",
        "hostids": "10254"
    },
    "id": 1
}
```


Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemid": "23865",
        "name": "HTTP aģenta JSON piemērs",
        "key_": "json",
        "preprocessing": [
            {
                "type": "12",
                "params": "$.random",
                "error_handler": "1",
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Skatiet arī

- Atklāšanas kārtula
- Grafiks
- Hosts
- Hosta saskarne
- Trigeris

Avots

CItem::get() failā *ui/include/classes/api/services/CItem.php*.

vienums.create

Apraksts

`object item.create(object/array items)`

Šī metode ļauj izveidot jaunus vienumus.

Note:
Tīmekļa vienumus nevar izveidot, izmantojot Zabbix API.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var atsaukt lietotāju lomu iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Vienumi, ko izveidot.

Papildus standarta vienuma īpašībām metode pieņem šādus parametrus.

Parametrs	Tips	Apraksts
preprocessing	array	Vienuma priekšapstrādes opcijas.
tags	array	Vienuma birkas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto vienumu ID īpašībā `itemids`. Atgriezto ID secība atbilst nodoto vienumu secībai.

Piemēri

Vienuma izveide

Izveidojiet skaitlisku Zabbix aģenta vienumu ar 2 vienuma tagiem, lai uzraudzītu brīvo diska vietu hostā ar ID ”30074”.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24758"
        ]
    },
    "id": 1
}
```

Hosts inventarizācijas vienuma izveide

Izveidojiet Zabbix aģents vienumu, lai aizpildītu hosta inventarizācijas lauku "OS".

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "24759"
```

```
        ]
    },
    "id": 1
}
```

Vienuma izveide ar priekšapstrādi

Izveidojiet vienumu, izmantojot pielāgotu reizinātāju.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Ierīces darbības laiks",
        "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44210"
        ]
    },
    "id": 1
}
```

Atkarīgā vienuma izveide

Izveidojiet atkarīgo vienumu galvenajam vienumam ar ID 24759. Ir atļautas tikai atkarības uz tā paša hosta, tādēļ galvenajam un atkarīgajam vienumam jābūt ar vienādu hostid.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "44211"
        ]
    },
    "id": 1
}
```

Izveidot HTTP aģenta vienumu

Izveidojiet POST pieprasījuma metodes vienumu ar JSON atbildes priekšapstrādi.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
```

```
    "id": 1
}
```

Izveidot skripta vienumu

Izveidojiet vienkāršu datu vākšanu, izmantojot skripta vienumu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.create",
    "params": {
        "name": "Skripta piemērs",
        "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
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Avots

CItem::create() failā *ui/include/classes/api/services/CItem.php*.

vienums.delete

Apraksts

`object item.delete(array itemIds)`

Šī metode ļauj dzēst vienumus.

Note:
Tīmekļa vienumus nevar dzēst, izmantojot Zabbix API.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` dzēšamo vienumu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto vienumu ID zem `itemids` īpašības.

Piemēri

Vairāku vienumu dzēšana

Dzēsiet divus vienumus.
 Atkarīgie vienumi un vienumu prototipi tiek noņemti automātiski, ja tiek dzēsts galvenais vienums.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.delete",
    "params": [
        "22982",
        "22986"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "22982",
            "22986"
        ]
    },
    "id": 1
}
```

Avots

CItem::delete() failā *ui/include/classes/api/services/CItem.php*.

vienums.update

Apraksts

`object item.update(object/array items)`

Šī metode ļauj atjaunināt esošos vienumus.

Note:
Tīmekļa vienumus nevar atjaunināt, izmantojot Zabbix API.

Note:
Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāju lomu iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object/array)` Atjaunināmo vienumu īpašības.

Īpašībai `itemid` jābūt definētai katram vienumam, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Papildus standarta vienuma īpašībām metode pieņem šādus parametrus.

Parametrs	Type	Apraksts
preprocessing	array	Vienuma priekšapstrādes opcijas, ar kurām aizstāt pašreizējās priekšapstrādes opcijas. Parametra darbība: - *tikai lasāms* mantotajiem objektiem vai atklātajiem objektiem
tags	array	Vienuma birkas. Parametra darbība: - *tikai lasāms* atklātajiem objektiem

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto vienumu ID zem `itemids` īpašības.

Piemēri

Vienuma iespējošana

Iespējojiet vienumu, tas ir, iestatiet tā statusu uz ”0”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "10092",
        "status": 0
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10092"
        ]
    },
    "id": 1
}
```

Atjaunināt atkarīgo vienumu

Atjauniniet atkarīgā vienuma nosaukumu un galvenā vienuma ID. Ir atļautas tikai atkarības vienā un tajā pašā hostā, tādēļ galvenajam un atkarīgajam vienumam jābūt ar vienādu hostid.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "name": "Dependent item updated name",
        "master_itemid": "25562",
        "itemid": "189019"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
```

```
        "itemids": [
            "189019"
        ]
    },
    "id": 1
}
```

Atjaunināt HTTP aģenta vienumu

Iespējot vienuma vērtību uztveršanu.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "allow_traps": 1
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
    "id": 1
}
```

Vienuma atjaunināšana ar priekšapstrādi

Atjauniniet vienumu ar vienuma priekšapstrādes noteikumu ”Diapazonā”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23856",
        "preprocessing": [
            {
                "type": 13,
                "params": "\n100",
                "error_handler": 1,
                "error_handler_params": ""
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23856"
        ]
    },
```

```
    "id": 1
}
```

Skripta vienuma atjaunināšana

Atjauniniet skripta vienumu ar citu skriptu un noņemiet nevajadzīgos parametrus, kas tika izmantoti iepriekšējā skriptā.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "item.update",
    "params": {
        "itemid": "23865",
        "parameters": [],
        "params": "Zabbix.log(3, 'Log test');\nreturn 1;"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "23865"
        ]
    },
    "id": 1
}
```

Avots

CItem::update() failā *ui/include/classes/api/services/CItem.php*.

Vērtību karte

Šī klase ir paredzēta darbam ar vērtību kartēm.

Objektu atsauces:

- Vērtību karte
  - Vērtību kartējumi

Pieejamās metodes:

- valuemap.create - izveidot jaunas vērtību kartes
- valuemap.delete - dzēst vērtību kartes
- valuemap.get - izgūt vērtību kartes
- valuemap.update - atjaunināt vērtību kartes

Vērtību kartes objekts

Tālāk norādītie objekti ir tieši saistīti ar `valuemap` API.

Vērtību karte

Vērtību kartes objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
valuemapid	ID	Vērtību kartes ID. Īpašības darbība: - *tikai lasāms* - *obligāts* atjaunināšanas darbībām

Īpašība	Tips	Apraksts
hostid	ID	Hosta vai veidnes ID, kam pieder vērtību karte. Īpašības darbība: - *nemainīgs* - *obligāts* izveides darbībām
name	string	Vērtību kartes nosaukums. Īpašības darbība: - *obligāts* izveides darbībām
mappings	array	Pašreizējās vērtību kartes vērtību atbilstības. Atbilstības objekts ir detalizēti aprakstīts tālāk. Īpašības darbība: - *obligāts* izveides darbībām
uuid	string	Universāls unikāls identifikators, ko izmanto, lai sasaistītu importētās vērtību kartes ar jau esošajām. Tiek ģenerēts automātiski, ja nav norādīts. Īpašības darbība: - *atbalstīts*, ja vērtību karte pieder veidnei

Vērtību kartējumi

Vērtību kartējumu objekts definē vērtību kartējumus vērtību kartei. Tam ir šādas īpašības.

Īpašība	Type	Apraksts
type	integer	Kartējuma atbilstības tips. Iespējamās vērtības: 0 - *(noklusējums)* kartējums tiks piemērots, ja vērtība ir vienāda; 1 - kartējums tiks piemērots, ja vērtība ir lielāka vai vienāda[1]; 2 - kartējums tiks piemērots, ja vērtība ir mazāka vai vienāda[1]; 3 - kartējums tiks piemērots, ja vērtība ir diapazonā (diapazoni ir iekļaujoši; var definēt vairākus diapazonus, atdalot tos ar komatu)[1]; 4 - kartējums tiks piemērots, ja vērtība atbilst regulārai izteiksmei[2]; 5 - ja atbilstības netiek atrastas, kartējums netiks piemērots un tiks izmantota noklusējuma vērtība. Ja `type` ir iestatīts uz ”0”, ”1”, ”2”, ”3”, ”4”, tad `value` nevar būt tukšs. Ja `type` ir iestatīts uz ”5”, tad `value` jābūt tukšam.
value	string	Sākotnējā vērtība. Īpašības darbība: - *obligāta*, ja `type` ir iestatīts uz ”1”, ”2”, ”3”, ”4” - *atbalstīta*, ja `type` ir iestatīts uz ”5”
newvalue	string	Vērtība, uz kuru tiek kartēta sākotnējā vērtība. Īpašības darbība: - *obligāta*

[1] tiek atbalstīts tikai vienumiem ar vērtības tipu ”numeric unsigned”, ”numeric float”.
 [2] tiek atbalstīts tikai vienumiem ar vērtības tipu ”character”.

valuemap.create

Apraksts

```
object valuemap.create(object/array valuemaps)
```

Šī metode ļauj izveidot jaunas vērtību kartes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Izveidojamās vērtību kartes.

Metode pieņem vērtību kartes ar standarta vērtību kartes īpašībām.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur izveidoto vērtību karšu ID īpašumā `valuemapids`. Atgriezto ID secība atbilst nodoto vērtību karšu secībai.

Piemēri

Vērtību kartes izveide

Izveidojiet vienu vērtību karti ar divām atbilstībām.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.create",
    "params": {
        "hostid": "50009",
        "name": "Pakalpojuma stāvoklis",
        "mappings": [
            {
                "type": "1",
                "value": "1",
                "newvalue": "Darbojas"
            },
            {
                "type": "5",
                "newvalue": "Nedarbojas"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1"
        ]
    },
    "id": 1
}
```

Avots

CValueMap::create() failā *ui/include/classes/api/services/CValueMap.php*.

valuemap.delete

Apraksts

`object valuemap.delete(array valuemapids)`

Šī metode ļauj dzēst vērtību kartes.

Note:
Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(array)` dzēšamo vērtību karšu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur dzēsto vērtību karšu ID īpašībā `valuemapids`.

Piemēri

Vairāku vērtību karšu dzēšana

Dzēsiet divas vērtību kartes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.delete",
    "params": [
        "1",
        "2"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "1",
            "2"
        ]
    },
    "id": 1
}
```

Avots

CValueMap::delete() failā *ui/include/classes/api/services/CValueMap.php*.

valuemap.get

Apraksts

`integer/array valuemap.get(object parameters)`

Šī metode ļauj izgūt vērtību kartes atbilstoši norādītajiem parametriem.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
valuemapids	ID/array	Atgriezt tikai vērtību kartes ar norādītajiem ID.

Parametrs	Tips	Apraksts
selectMappings	query	Atgriezt vērtību kartējumus pašreizējai vērtību kartei īpašībā `mappings`. Atbalsta `count`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `valuemapid`, `name`.
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
preservekeys	boolean	
search	object	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

`(integer/array)` Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- iegūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Vērtību karšu izgūšana

Izgūt visas konfigurētās vērtību kartes.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend"
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Izgūt vienu vērtību karti ar tās kartējumiem.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.get",
    "params": {
        "output": "extend",
        "selectMappings": "extend",
        "valuemapids": ["4"]
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Avots

CValueMap::get() failā *ui/include/classes/api/services/CValueMap.php*.

valuemap.update

Apraksts

`object valuemap.update(object/array valuemaps)`

Šī metode ļauj atjaunināt esošās vērtību kartes.

> **Note:**
> Šī metode ir pieejama tikai *Super admin* lietotāja tipam. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Skatiet Lietotāju lomas papildinformācijai.

Parametri

`(object/array)` Vērtību kartes īpašības, kas jāatjaunina.

Īpašībai `valuemapid` jābūt definētai katrai vērtību kartei, visas pārējās īpašības nav obligātas. Tiks atjauninātas tikai nodotās īpašības, visas pārējās paliks nemainītas.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur atjaunināto vērtību karšu ID īpašībā `valuemapids`.

Piemēri

Vērtību kartes nosaukuma maiņa

Mainiet vērtību kartes nosaukumu uz ”Device status”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "name": "Device status"
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Vienas vērtību kartes atbilsmju maiņa.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "valuemap.update",
    "params": {
        "valuemapid": "2",
        "mappings": [
            {
                "type": "0",
                "value": "0",
```

```
                "newvalue": "Online"
            },
            {
                "type": "0",
                "value": "1",
                "newvalue": "Offline"
            }
        ]
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "valuemapids": [
            "2"
        ]
    },
    "id": 1
}
```

Avots

CValueMap::update() failā *ui/include/classes/api/services/CValueMap.php*.

Vēsture

Šī klase ir paredzēta darbam ar vēstures datiem.

Objektu atsauces:

- Peldošā komata vēsture
- Veselu skaitļu vēsture
- Virknes vēsture
- Teksta vēsture
- Žurnāla vēsture

Pieejamās metodes:

- history.clear - notīrīt vēstures datus
- history.get - izgūt vēstures datus
- history.push - nosūtīt vēstures datus uz Zabbix serveri

Vēstures objekts

Tālāk norādītie objekti ir tieši saistīti ar `history` API.

Note:
Vēstures objekti atšķiras atkarībā no vienuma informācijas tipa. Tos izveido Zabbix serveris, un tos nevar modificēt, izmantojot API.

Peldošā komata vēsture

Peldošā komata vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	float	Saņemtā vērtība.

Veselu skaitļu vēsture

Veselu skaitļu vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	integer	Saņemtā vērtība.

Virknes vēsture

Virknes vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	string	Saņemtā vērtība.

Teksta vēsture

Teksta vēstures objektam ir šādas īpašības.

Īpašība	Type	Apraksts
id	ID	Vēstures ieraksta ID.
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	text	Saņemtā vērtība.

Žurnāla vēsture

Žurnāla vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
id	ID	Vēstures ieraksta ID.
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
logeventid	integer	Windows notikumu žurnāla ieraksta ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
severity	integer	Windows notikumu žurnāla ieraksta līmenis.
source	string	Windows notikumu žurnāla ieraksta avots.
timestamp	timestamp	Windows notikumu žurnāla ieraksta laiks.
value	text	Saņemtā vērtība.

Binārā vēsture

Binārās vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	text	Saņemtā vērtība.

JSON vēsture

JSON vēstures objektam ir šādas īpašības.

Īpašība	Tips	Apraksts
clock	timestamp	Laiks, kad šī vērtība tika saņemta.
itemid	ID	Saistītā vienuma ID.
ns	integer	Nanosekundes, kad vērtība tika saņemta.
value	text	Saņemtā vērtība.

history.clear

Apraksts

`object history.clear(array itemids)`

Šī metode ļauj notīrīt vienuma vēsturi.

> **Note:**
> Šī metode ir pieejama tikai *Admin* un *Super admin* lietotāju tipiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet Lietotāju lomas.

Parametri

`(array)` notīrāmo vienumu ID.

Atgrieztās vērtības

`(object)` Atgriež objektu, kas satur notīrīto vienumu ID īpašībā `itemids`.

Piemēri

Notīrīt vēsturi

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "history.clear",
    "params": [
        "10325",
        "13205"
    ],
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": {
        "itemids": [
            "10325",
            "13205"
        ]
    },
    "id": 1
}
```

Avots

CHistory::clear() failā *ui/include/classes/api/services/CHistory.php*.

history.get

Apraksts

`integer/array history.get(object parameters)`

Metode ļauj izgūt vēstures datus atbilstoši norādītajiem parametriem.

Attention:
Šī metode var atgriezt dzēstas entītijas vēsturiskos datus, ja housekeeper šos datus vēl nav noņēmis.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object)` Parametri, kas nosaka vēlamo izvadi.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
history	integer	Atgriežamo vēstures objektu tipi. Iespējamās vērtības: 0 - skaitlis ar peldošo komatu; 1 - rakstzīme; 2 - žurnāls; 3 - *(noklusējums)* skaitlis bez zīmes; 4 - teksts; 5 - binārs; 6 - JSON.
hostids	ID/array	Atgriezt tikai vēsturi no norādītajiem hostiem.
itemids	ID/array	Atgriezt tikai vēsturi no norādītajiem vienumiem.
maxValueSize	integer	Maksimālais baitu skaits, ko atgriezt `value` īpašībā. Iespējamo vērtību diapazons: 1-134217728 (1 baits līdz 128MiB). Ja iestatīts uz `null`, ierobežojums netiek piemērots. Noklusējums: 65536 (64KiB). Parametra darbība: - *atbalstīts*, ja `history` ir iestatīts uz ”binary” vai ”JSON”
time_from	timestamp	Atgriezt tikai vērtības, kas saņemtas pēc norādītā laika vai tieši tajā laikā.
time_till	timestamp	Atgriezt tikai vērtības, kas saņemtas pirms norādītā laika vai tieši tajā laikā.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Iespējamās vērtības: `itemid`, `clock`, `ns`.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kurā atslēgas ir īpašību nosaukumi, bet vērtības ir meklējamās virknes. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Neatbalsta meklēšanu pēc `value`, ja `history` ir iestatīts uz 6 (JSON).
countOutput	boolean	Šie parametri ir aprakstīti atsauces komentārā.
editable	boolean	
excludeSearch	boolean	
filter	object	
limit	integer	
output	query	
searchByAny	boolean	
searchWildcardsEnabled	boolean	
sortorder	string/array	
startSearch	boolean	

Atgrieztās vērtības

(integer/array) Atgriež vienu no šīm vērtībām:

- objektu masīvu;
- izgūto objektu skaitu, ja ir izmantots parametrs `countOutput`.

Piemēri

Vienuma vēstures datu izgūšana

Atgriezt 10 jaunākās vērtības, kas saņemtas no skaitliska (float) vienuma.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "history.get",
    "params": {
        "output": "extend",
        "history": 0,
        "itemids": "23296",
        "sortfield": "clock",
        "sortorder": "DESC",
        "limit": 10
    },
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```


Avots

CHistory::get() failā *ui/include/classes/api/services/CHistory.php*.

history.push

Apraksts

`object history.push(object/array itemHistoryData)`

Šī metode ļauj nosūtīt vienuma vēstures datus uz Zabbix serveri.

Note:
Šī metode ir pieejama jebkura tipa lietotājiem. Atļaujas izsaukt šo metodi var tikt atsauktas lietotāja lomas iestatījumos. Plašāku informāciju skatiet sadaļā Lietotāju lomas.

Parametri

`(object/array)` Nosūtāmie vienuma vēstures dati.

Metode atbalsta šādus parametrus.

Parametrs	Tips	Apraksts
itemid	ID	Saistītā vienuma ID. Parametra darbība: - *obligāts*, ja `host` un `key` nav iestatīti
host	string	Hosta tehniskais nosaukums. Parametra darbība: - *obligāts*, ja `itemid` nav iestatīts
key	string	Vienuma atslēga. Parametra darbība: - *obligāts*, ja `itemid` nav iestatīts

Parametrs	Tips	Apraksts
value	mixed	Vienuma vērtība. Parametra darbība: - *obligāts*
clock	timestamp	Laiks, kad vērtība tika saņemta.
ns	integer	Nanosekundes, kad vērtība tika saņemta.

Atgrieztās vērtības

`(object)` Atgriež datu nosūtīšanas operācijas rezultātu.

Piemēri

Nosūtīt vienuma vēstures datus

Nosūtīt vienuma vēstures datus uz Zabbix serveris vienumiem ”10600”, ”10601” un ”999999”.

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "history.push",
    "params": [
        {
            "itemid": 10600,
            "value": 0.5,
            "clock": 1690891294,
            "ns": 45440940
        },
        {
            "itemid": 10600,
            "value": 0.6,
            "clock": 1690891295,
            "ns": 312431
        },
        {
            "itemid": 10601,
            "value": "[Tue Aug 01 15:01:35 2023] [error] [client 1.2.3.4] File does not exist: /var/www/ht
        },
        {
            "itemid": 999999,
            "value": 123
        }
    ],
    "id": 1
}
```

Atbilde:

```
{
    "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
}
```

Skatīt arī

- Trapper vienumi
- HTTP aģenta vienumi
- Hosts
- Vienums

Avots

CHistory::push() failā *ui/include/classes/api/services/CHistory.php*.

1. pielikums. Atsauces komentāri

Apzīmējumi Datu tipi

Zabbix API atbalsta šādus datu tipus kā ievadi:

Type	Description
ID	Unikāls identifikators, ko izmanto, lai atsauktos uz entītiju.
boolean	Būla vērtība (`true` vai `false`).
flag	Vērtība, kas tiek uzskatīta par `true`, ja tā ir nodota un nav vienāda ar `null`; pretējā gadījumā vērtība tiek uzskatīta par `false`.
integer	Vesels skaitlis.
float	Peldošā komata skaitlis.
string	Teksta virkne.
text	Garāka teksta virkne.
timestamp	Unix laikspiedols.
array	Sakārtota vērtību secība (parasts masīvs).
object	Asociatīvs masīvs.
query	Vērtība, kas nosaka atgriežamos datus. Vērtību var definēt kā īpašību nosaukumu masīvu (lai atgrieztu tikai noteiktas īpašības) vai kā vienu no iepriekš definētajām vērtībām: `extend` - atgriež visas objekta īpašības; `count` - atgriež iegūto ierakstu skaitu; to atbalsta tikai noteikti apakšatlases vaicājumi.

Attention:
Zabbix API vienmēr atgriež vērtības tikai kā virknes vai masīvus.

Īpašību darbība

Dažas objekta īpašības ir apzīmētas ar īsām etiķetēm, lai raksturotu to darbību. Tiek izmantotas šādas etiķetes:

- ***tikai lasāms*** - īpašības vērtība tiek iestatīta automātiski un lietotājs to nevar definēt vai mainīt pat dažos īpašos apstākļos (piemēram, *tikai lasāms* mantotiem objektiem vai atklātiem objektiem);
- ***tikai rakstāms*** - īpašības vērtību var iestatīt, bet pēc tam tai nevar piekļūt;
- ***konstants*** - īpašības vērtību var iestatīt, veidojot objektu, bet pēc tam to nevar mainīt;
- ***atbalstīts*** - īpašības vērtība nav obligāti jāiestata, taču to ir atļauts iestatīt dažos īpašos apstākļos (piemēram, *atbalstīts*, ja `type` ir iestatīts uz "Simple check", "External check", "SSH agent", "TELNET agent" vai "HTTP agent"); tomēr ņemiet vērā, ka *atbalstītas* īpašības joprojām var tikt iestatītas uz to noklusējuma vērtībām neatkarīgi no apstākļiem;
- ***obligāts*** - īpašības vērtība ir obligāti jāiestata visām operācijām (izņemot iegūšanas operācijas) vai dažos īpašos apstākļos (piemēram, *obligāts* izveides operācijām; *obligāts*, ja `operationtype` ir iestatīts uz "global script" un `opcommand_hst` nav iestatīts).

Note:
Atjaunināšanas operācijām īpašība tiek uzskatīta par "iestatītu", ja tā tiek iestatīta atjaunināšanas operācijas laikā.

Īpašības, kas nav apzīmētas ar etiķetēm, nav obligātas.

Parametru uzvedība

Daži operācijas parametri ir apzīmēti ar īsām etiķetēm, lai raksturotu to uzvedību operācijā. Tiek izmantotas šādas etiķetes:

- ***tikai lasāms*** - parametra vērtība tiek iestatīta automātiski un lietotājs to nevar definēt vai mainīt, pat noteiktos īpašos apstākļos (piemēram, *tikai lasāms* mantotiem objektiem vai atklātiem objektiem);
- ***tikai rakstāms*** - parametra vērtību var iestatīt, bet pēc tam tai nevar piekļūt;
- ***atbalstīts*** - parametra vērtību nav obligāti jāiestata, taču to ir atļauts iestatīt noteiktos īpašos apstākļos (piemēram, *atbalstīts*, ja Proxy objekta `operating_mode` ir iestatīts uz "passive proxy"); tomēr ņemiet vērā, ka *atbalstītie* parametri joprojām var tikt iestatīti uz to noklusējuma vērtībām neatkarīgi no apstākļiem;
- ***obligāts*** - parametra vērtība ir obligāti jāiestata.

Parametri, kas nav apzīmēti ar etiķetēm, nav obligāti.

Rezervētā ID vērtība "0" Rezervēto ID vērtību "0" var izmantot, lai filtrētu elementus un noņemtu atsauktos objektus. Piemēram, lai noņemtu atsaukto starpniekserveri no hosta, proxyid ir jāiestata uz 0 ("proxyid": "0") vai, lai filtrētu servera uzraudzītos hostus, opcijai proxyids ir jābūt iestatītai uz 0 ("proxyids": "0").

Biežāk izmantotie "get" metožu parametri Tālāk norādītie parametri tiek atbalstīti visās `get` metodēs:

Parametrs	Tips	Apraksts
countOutput	boolean	Atgriezt rezultātā esošo ierakstu skaitu faktiskās datu izvades vietā.
editable	boolean	Ja iestatīts uz `true`, atgriezt tikai tos objektus, kuriem lietotājam ir rakstīšanas atļaujas. Noklusējums: `false`.
excludeSearch	boolean	Atgriezt rezultātus, kas neatbilst `search` parametrā norādītajiem kritērijiem.
filter	object	Atgriezt tikai tos rezultātus, kas precīzi atbilst norādītajam filtram. Pieņem objektu, kurā atslēgas ir īpašību nosaukumi (piemēram, Host objekta īpašības `host.get`, Item objekta īpašības `item.get` utt.), bet vērtības ir vai nu viena vērtība, vai vērtību masīvs, ar ko salīdzināt. Neatbalsta `text` datu tipa īpašības. Ņemiet vērā, ka dažām metodēm šim parametram ir specifiska funkcionalitāte, kas ir aprakstīta metodes lapā (piemēram, parametrs `filter` metodē host.get atbalsta arī Host interfeisa īpašības).
limit	integer	Ierobežot atgriezto ierakstu skaitu.
output	query	Atgriežamās objekta īpašības. Ņemiet vērā, ka objekta ID (t.i., `hostid`, `itemid` utt.) vienmēr ir iekļauts atbildē, pat ja tas nav norādīts parametrā `output`. Noklusējums: `extend`.
preservekeys	boolean	Izmantot ID kā atslēgas rezultējošajā masīvā.
search	object	Atgriezt rezultātus, kas atbilst norādītajam paraugam (reģistrnejutīgi). Pieņem objektu, kurā atslēgas ir īpašību nosaukumi (piemēram, Host objekta īpašības `host.get`, Item objekta īpašības `item.get` utt.), bet vērtības ir virknes, kuras meklēt. Ja nav norādītas papildu opcijas, tiks veikta `LIKE "%...%"` meklēšana. Atbalsta tikai `string` un `text` datu tipa īpašības. Ņemiet vērā, ka dažām metodēm šim parametram ir specifiska funkcionalitāte, kas ir aprakstīta metodes lapā (piemēram, parametrs `search` metodē host.get atbalsta arī Host interfeisa īpašības).

Parametrs	Tips	Apraksts
searchByAny	boolean	Ja iestatīts uz `true`, atgriezt rezultātus, kas atbilst jebkuram no `filter` vai `search` parametrā norādītajiem kritērijiem, nevis visiem. Noklusējums: `false`.
searchWildcardsEnabled	boolean	Ja iestatīts uz `true`, parametrā `search` tiek iespējota "*" izmantošana kā aizstājējzīme. Noklusējums: `false`.
sortfield	string/array	Kārtot rezultātu pēc norādītajām īpašībām. Skatiet konkrētās API get metodes aprakstu, lai iegūtu to īpašību sarakstu, kuras var izmantot kārtošanai. Makrosi pirms kārtošanas netiek izvērsti. Ja vērtība nav norādīta, dati tiks atgriezti bez kārtošanas.
sortorder	string/array	Kārtošanas secība. Ja tiek nodots masīvs, katra vērtība tiks piesaistīta atbilstošajai īpašībai, kas norādīta parametrā `sortfield`. Iespējamās vērtības: `ASC` - *(noklusējums)* augošā secībā; `DESC` - dilstošā secībā.
startSearch	boolean	Parametrs `search` salīdzinās lauku sākumu, tas ir, tā vietā veiks `LIKE "…%"` meklēšanu. Tiek ignorēts, ja `searchWildcardsEnabled` ir iestatīts uz `true`.

Entītiju izcelsmes karodziņi Get metodes atgriež `flags` īpašību entītijām, kas saistītas ar zema līmeņa atklāšanu (LLD kārtula/LLD kārtulas prototips, vienums/vienuma prototips utt.). Šī īpašība ir noderīga, lai norādītu, vai entītija ir atklāta vai nav, jo atklāto entītiju rediģēšana ir ierobežota.

Īpašība `flags` atgriež rezultātu, pamatojoties uz šo vērtību kombināciju (operācija "+"):

Value	Description
0	Pamata entītija (vienums, trigeris, grafiks, hosts)
1	Zema līmeņa atklāšanas kārtula
2	Jebkurš prototips (vienuma prototips, trigera prototips, LLD kārtulas prototips utt.)
4	Atklāta entītija (atklāts vienums, trigeris, grafiks, hosts, LLD kārtula)

Ar īpašību `flags` atgrieztā **kombinētā** vērtība var būt:

Value	Combination of	Description
0	0	Parasta entītija (vienums, trigeris, grafiks, hosts).
2	2	Entītijas prototips (vienuma prototips, trigera prototips utt.).
6	2+4	Atklāts vienums, trigeris, grafiks, hosts (pārveidots no prototipa).
1	1	Zema līmeņa atklāšanas kārtula.
3	1+2	Zema līmeņa atklāšanas kārtulas prototips.
5	1+4	Atklāta zema līmeņa atklāšanas kārtula (pārveidota no prototipa).
7	1+2+4	Atklāts zema līmeņa atklāšanas kārtulas prototips.

Piemēri Lietotāja atļauju pārbaude

Vai lietotājam ir atļauja rakstīt hostos, kuru nosaukumi sākas ar "MySQL" vai "Linux" ?

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": ["MySQL", "Linux"]
        },
        "editable": true,
```

```
        "startSearch": true,
        "searchByAny": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": "0",
    "id": 1
}
```

Note:
Nulles rezultāts nozīmē, ka nav neviena hosta ar lasīšanas/rakstīšanas atļaujām.

Neatbilstību skaitīšana

Saskaitiet hostus, kuru nosaukumi nesatur apakšvirkni ”ubuntu”

Pieprasījums:

```
{
    "jsonrpc": "2.0",
    "method": "host.get",
    "params": {
        "countOutput": true,
        "search": {
            "host": "ubuntu"
        },
        "excludeSearch": true
    },
    "id": 1
}
```

Atbilde:

```
{
    "jsonrpc": "2.0",
    "result": "44",
    "id": 1
}
```

Hostu meklēšana, izmantojot aizstājējzīmes

Atrodiet hostus, kuru nosaukums satur vārdu ”server” un kuriem ir saskarņu porti ”10050” vai ”10071”. Sakārtojiet rezultātu pēc hosta nosaukuma dilstošā secībā un ierobežojiet to līdz 5 hostiem.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "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
}
```

Hostu meklēšana, izmantojot aizstājējzīmes ar ”preservekeys”

Ja iepriekšējam pieprasījumam pievienojat parametru ”preservekeys”, rezultāts tiek atgriezts kā asociatīvs masīvs, kur atslēgas ir objektu ID.

Pieprasījums:

```
{
    "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
}
```

Atbilde:

```
{
    "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
}
```

2. pielikums. Izmaiņas no 7.4 uz 8.0

Nesaderīgas izmaiņas vēsture

ZBXNEXT-10069 Metodei history.get pievienots jauns parametrs `maxValueSize`. Ja pieprasāt datus vienumiem ar bināru vērtību (`history=5`), iestatiet `maxValueSize` uz `null`; pretējā gadījumā `history.get` atgriezīs vienuma vērtību, kas saīsināta līdz 64KiB.

hosts

ZBXNEXT-10387 `host.massupdate`: Metode noņemta. Lūdzu, tā vietā izmantojiet `host.update`.

hostu grupa

ZBXNEXT-10387 `hostgroup.massupdate`: Metode noņemta. Lūdzu, tā vietā izmantojiet `hostgroup.update`, `hostgroup.massadd` vai `hostgroup.massremove`.

hostinterface

ZBXNEXT-10387 `hostinterface.replacehostinterfaces`: Metode noņemta. Lūdzu, tā vietā izmantojiet `hostinterface.update`, `hostinterface.massadd` vai `hostinterface.massremove`.

veidne

ZBXNEXT-10387 `template.massupdate`: Metode noņemta. Lūdzu, tā vietā izmantojiet `template.update`, `template.massadd` vai `template_massremove`.

templategroup

ZBXNEXT-10387 `templategroup.massupdate`: Metode noņemta. Lūdzu, tā vietā izmantojiet `templategroup.update`, `templategroup.massadd` vai `templategroup.massremove`.

Citas izmaiņas un kļūdu labojumi configuration

ZBXNEXT-5419 Added new parameter `returnMissingObjects` to the configuration.importcompare method.
ZBXNEXT-5419 Added new parameter `dashboards` to the rules object (`configuration.importcompare` method).
ZBXNEXT-5419 Added new parameter `returnMissingObjects` to the configuration.import method.
ZBXNEXT-5419 Added new parameter `dashboards` to the rules object (`configuration.import` method).

savienotājs

ZBXNEXT-10069 Objektam connector pievienota jauna `item_value_type` vērtība (64 - JSON).

panelis

ZBXNEXT-4769 Pievienots jauns paneļa logrīka lauks `show_hostnames` logrīkam graph.
 ZBXNEXT-9546 Pievienoti jauni paneļa logrīka lauki `ds.0.item_tags_evaltype`, `ds.0.item_tags.0.tag`, `ds.0.item_tags.0.operator` un `ds.0.item_tags.0.value` logrīkiem svggraph un piechart.
 ZBXNEXT-10158 Pievienoti jauni paneļa logrīka lauki `columns.0.aggregate_columns`, `columns.0.combined_column_name` un `columns.0.column_aggregate_function` logrīkam topitems.
 ZBXNEXT-9424, ZBXNEXT-4769 Pievienots jauns paneļa logrīka tips scatterplot.
 ZBXNEXT-8160 Pievienoti jauni paneļa logrīka lauki `clustering_mode` un `clustering_zoom_level` logrīkam geomap.
 ZBXNEXT-943 Pievienots jauns paneļa logrīka lauks `ds.0.invert_values` logrīkam graph.

veidnes panelis

ZBXNEXT-9424 Pievienots jauns veidnes paneļa logrīka tips izkliedes diagramma.

vēsture

ZBXNEXT-10069 Metodei history.get pievienota jauna `history` vērtība (6 - JSON).
 ZBXNEXT-10069 Pievienots jauns `JSON history` objekts.

hostprototype

ZBXNEXT-9546 Metodei hostprototype.get pievienots jauns parametrs `selectInheritedTags`.

httptest

ZBXNEXT-9546 Metodei httptest.get pievienoti jauni parametri `inheritedTags` un `selectInheritedTags`.

vienums

ZBXNEXT-10069 Pievienota jauna `value_type` vērtība (6 - JSON) vienums objektam.
 ZBXNEXT-9546 Metodei item.get pievienoti jauni parametri `inheritedTags` un `selectInheritedTags`.

vienuma prototips

ZBXNEXT-10069 Objektam itemprototype pievienota jauna `value_type` vērtība (6 - JSON).
 ZBXNEXT-9546 Metodei itemprototype.get pievienots jauns parametrs `selectInheritedTags`.

veidne

ZBXNEXT-9546 Pievienoti jauni parametri `inheritedTags` un `selectInheritedTags` metodei template.get.

trigeris

ZBXNEXT-9546 Metodei trigger.get pievienoti jauni parametri `inheritedTags` un `selectInheritedTags`.

trigera prototips

ZBXNEXT-9546 Metodei triggerprototype.get pievienots jauns parametrs `selectInheritedTags`.

trigera tags

ZBXNEXT-5467 Pievienota īpašība `automatic` trigera taga objektam.

userdirectory

ZBXNEXT-9834 Pievienotas īpašības `idp_certificate`, `sp_private_key`, `sp_certificate`, `idp_certificate_hash`, `sp_private_key_hash` un `sp_certificate_hash` lietotāju direktorija objektam.

3. pielikums. Izmaiņas 8.0 versijā

8.0.1 Šī versija vēl nav izlaista.

19 Paplašinājumi

Pārskats Lai gan Zabbix piedāvā plašu funkciju klāstu, vienmēr ir vieta papildu funkcionalitātei. Paplašinājumi ir ērts veids, kā modificēt un uzlabot Zabbix uzraudzības iespējas, nemainot tā pirmkodu.

Jūs varat paplašināt Zabbix funkcionalitāti, izmantojot vai nu iebūvētās paplašināšanas iespējas (trapper vienumi, lietotāja parametri u.c.), vai arī izmantojot vai veidojot pielāgotus paplašinājumus (ielādējamus moduļus, spraudņus u.c.).

Šajā sadaļā ir sniegts pārskats ar atsaucēm uz visām Zabbix paplašināšanas iespējām.

Datu vākšana ar pielāgotām komandām Trapper vienumi

Trapper vienumi ir vienumi, kas pieņem ienākošos datus, nevis tos pieprasa. Trapper vienumi ir noderīgi, lai nosūtītu konkrētus datus uz Zabbix serveri vai starpniekserveri, piemēram, periodiskus pieejamības un veiktspējas datus ilgstoši darbojošos lietotāja skriptu gadījumā. Datu nosūtīšana uz Zabbix serveri vai starpniekserveri ir iespējama, izmantojot utilītu Zabbix sender vai Zabbix sender protocol. Datu nosūtīšana uz Zabbix serveri ir iespējama arī, izmantojot API metodi `history.push`.

Ārējās pārbaudes

Ārējā pārbaude ir vienums, kas paredzēts pārbaužu izpildei, palaižot izpildāmu failu, piemēram, čaulas skriptu vai bināro failu.

Ārējās pārbaudes izpilda Zabbix serveris vai starpniekserveris (ja hosts tiek uzraudzīts ar starpniekserveri), un tām nav nepieciešams aģents, kas darbojas uz uzraugāmā hosta.

Lietotāja parametri

User parameter ir lietotāja definēta komanda (saistīta ar lietotāja definētu atslēgu), kura, izpildot to, var iegūt jums nepieciešamos datus no hosta, kur darbojas Zabbix aģents. Lietotāja parametri ir noderīgi, lai konfigurētu aģenta vai aģenta 2 vienumus, kas nav iepriekš definēti Zabbix.

`system.run[]` Zabbix aģenta vienumi

`system.run[]` Zabbix aģenta vienums ir vienums lietotāja definētai komandai (saistīts ar iepriekš definētu atslēgu `system.run[]`, piemēram, `system.run[myscript.sh]`), ko var izpildīt hostā, kur darbojas Zabbix aģents.

Piezīme: `system.run[]` vienumi pēc noklusējuma ir atspējoti, un, ja tie tiek izmantoti, tie ir jāiespējo (atļauti) un jādefinē Zabbix aģenta vai aģenta 2 konfigurācijas failā (konfigurācijas parametrs `AllowKey`).

Attention:
Lietotāja definētas komandas vienumos, piemēram, ārējās pārbaudēs, lietotāja parametros un `system.run[]` Zabbix aģenta vienumos, tiek izpildītas no OS lietotāja, kas tiek izmantots Zabbix komponentu darbināšanai. Lai izpildītu šīs komandas, šim lietotājam jābūt nepieciešamajām atļaujām.

HTTP aģenta vienumi

HTTP aģenta vienums ir vienums datu pieprasījumu izpildei, izmantojot HTTP/HTTPS. HTTP aģenta vienumi ir noderīgi pieprasījumu sūtīšanai uz HTTP galapunktiem, lai izgūtu datus no tādiem pakalpojumiem kā *Elasticsearch* un *OpenWeatherMap*, pārbaudītu Zabbix API statusu vai Apache vai Nginx tīmekļa servera statusu utt. HTTP aģenta vienumi (ar iespējotu uztveršanu) var darboties arī kā trapper vienumi.

Skripta vienumi

Skripta vienums ir vienums lietotāja definēta JavaScript koda izpildei, kas iegūst datus, izmantojot HTTP/HTTPS. Skripta vienumi ir noderīgi, ja ar HTTP aģenta vienumu nodrošināto funkcionalitāti nepietiek. Piemēram, sarežģītos datu vākšanas scenārijos, kuros nepieciešami vairāki soļi vai sarežģīta loģika, skripta vienumu var konfigurēt tā, lai tas veiktu HTTP izsaukumu, pēc tam apstrādātu saņemtos datus un pēc tam nodotu pārveidoto vērtību otram HTTP izsaukumam.

Note:
HTTP aģenta vienumus un skriptu vienumus atbalsta Zabbix serveris un starpniekserveris, un tiem nav nepieciešams aģents, kas darbojas uz uzraugāmā hosta.

Paplašinātie paplašinājumi Ielādējamie moduļi

C valodā rakstīti ielādējamie moduļi ir daudzpusīga un uz veiktspēju orientēta iespēja paplašināt Zabbix komponentu (serveris, starpniekserveris, aģents) funkcionalitāti UNIX platformās. Ielādējamais modulis būtībā ir koplietojama bibliotēka, ko izmanto Zabbix dēmons un kas tiek ielādēta palaišanas laikā. Bibliotēkai jāsatur noteiktas funkcijas, lai Zabbix process varētu noteikt, ka fails patiešām ir modulis, ko tas var ielādēt un izmantot.

Ielādējamiem moduļiem ir vairākas priekšrocības, tostarp iespēja pievienot jaunus metriku veidus vai ieviest jebkādu citu loģiku (piemēram, Zabbix vēstures datu eksportu), augsta veiktspēja, kā arī iespēja izstrādāt, izmantot un koplietot to nodrošināto funkcionalitāti. Tas veicina bezproblēmu uzturēšanu un palīdz vieglāk ieviest jaunu funkcionalitāti neatkarīgi no Zabbix kodola koda bāzes.

Ielādējamie moduļi ir īpaši noderīgi sarežģītā uzraudzības vidē. Uzraugot iegultās sistēmas, ja ir liels skaits uzraugāmo parametru vai tiek izmantoti apjomīgi skripti ar sarežģītu loģiku vai ilgu palaišanas laiku, tādi paplašinājumi kā lietotāja parametri, `system.run[]` Zabbix aģenta vienumi un ārējās pārbaudes ietekmēs veiktspēju. Ielādējamie moduļi piedāvā veidu, kā paplašināt Zabbix funkcionalitāti, nezaudējot veiktspēju.

Spraudņi

Spraudņi paplašina Zabbix aģents 2 uzraudzības iespējas. Tie ir rakstīti Go programmēšanas valodā un nodrošina alternatīvu ielādējamiem moduļiem (rakstītiem C valodā).

Ir iespējams izstrādāt savu spraudni (neatkarīgu datu savācēju) Zabbix aģents 2. Skatiet Izstrādātāju centru.

Brīdinājumu pielāgošana Webhook

Webhook ir Zabbix medija tips, kas nodrošina iespēju paplašināt Zabbix brīdināšanas iespējas, integrējot to ar ārējo programmatūru, piemēram, palīdzības dienesta sistēmām, tērzēšanas rīkiem vai ziņapmaiņas lietotnēm. Līdzīgi kā skriptu vienumi, webhook ir noderīgi HTTP izsaukumu veikšanai, izmantojot pielāgotu JavaScript kodu, piemēram, lai nosūtītu paziņojumus uz dažādām platformām, piemēram, Microsoft Teams, Discord un Jira. Ir iespējams arī atgriezt noteiktus datus (piemēram, par izveidotajiem palīdzības dienesta pieteikumiem), kas pēc tam tiek attēloti Zabbix.

Esošie webhook ir pieejami Zabbix Git repozitorijā. Lai izstrādātu pielāgotus webhook, skatiet Webhook izstrādes vadlīnijas.

Brīdinājumu skripti

Brīdinājumu skripts ir Zabbix medija tips, kas nodrošina iespēju izveidot alternatīvu veidu (skriptu) Zabbix brīdinājumu apstrādei. Brīdinājumu skripti ir noderīgi, ja jūs neapmierina esošie mediju tipi brīdinājumu nosūtīšanai Zabbix.

Lietotāja saskarnes pielāgošana Pielāgotas tēmas

Ir iespējams mainīt Zabbix lietotāja saskarne vizuālo izskatu, izmantojot pielāgotas tēmas. Skatiet instructions par savu tēmu izveidi un piemērošanu.

Lietotāja saskarnes moduļi

Lietotāja saskarnes moduļi nodrošina iespēju paplašināt Zabbix lietotāja saskarnes funkcionalitāti, pievienojot trešo pušu moduļus vai izstrādājot savus. Ar lietotāja saskarnes moduļiem varat pievienot jaunus izvēlnes vienumus, tiem atbilstošos skatus, darbības u.c.

Globālie skripti Globālais skripts ir lietotāja definēta komandu kopa, ko var izpildīt uzraudzības mērķī (ar čaulas (/bin/sh) interpretētāju), atkarībā no konfigurētās darbības jomas un lietotāja atļaujām. Globālos skriptus var konfigurēt šādām darbībām:

- Darbības operācija
- Manuāla hosta darbība
- Manuāla notikuma darbība

Globālie skripti ir noderīgi daudzos gadījumos. Piemēram, ja tie ir konfigurēti darbību operācijām vai manuālām hosta darbībām, varat izmantot globālos skriptus, lai automātiski vai manuāli izpildītu attālinātās komandas, piemēram, lietojumprogrammas restartēšanu (tīmekļa serveris, starpprogrammatūra, CRM u.c.) vai diska vietas atbrīvošanu (vecāku failu dzēšana, `/tmp` tīrīšana u.c.). Vai arī, cits piemērs, ja tie ir konfigurēti manuālām notikumu darbībām, varat izmantot globālos skriptus, lai pārvaldītu problēmu pieteikumus ārējās sistēmās.

Globālos skriptus var izpildīt Zabbix serveris, starpniekserveris vai aģents.

> **Attention:**
> Lietotāja definētās komandas tiek izpildītas no OS lietotāja, kas tiek izmantots Zabbix komponentu darbināšanai. Lai izpildītu šīs komandas, šim lietotājam jābūt nepieciešamajām atļaujām.

Zabbix API Zabbix API ir uz HTTP balstīts API, kas ir daļa no Zabbix lietotāja saskarnes. Izmantojot Zabbix API, jūs varat veikt jebkuru no šādām darbībām:

- Programmatiski izgūt un mainīt Zabbix konfigurāciju.
- Importēt un eksportēt Zabbix konfigurāciju.
- Piekļūt Zabbix vēsturiskajiem un tendenču datiem.
- Konfigurēt lietojumprogrammas darbam ar Zabbix.
- Integrēt Zabbix ar trešo pušu programmatūru.
- Automatizēt ikdienas uzdevumus.

Zabbix API sastāv no daudzām metodēm, kas nosacīti ir sagrupētas atsevišķos API. Katra metode veic konkrētu uzdevumu. Par pieejamajām metodēm, kā arī Zabbix API nodrošināto funkciju pārskatu skatiet Zabbix API Method reference.

1 Ielādējamie moduļi

Pārskats

Ielādējamie moduļi piedāvā uz veiktspēju orientētu iespēju paplašināt Zabbix funkcionalitāti.

Jūs varat paplašināt Zabbix funkcionalitāti daudzos veidos, piemēram, ar lietotāja parametriem, ārējām pārbaudēm un Zabbix aģents vienumiem system.run. Tie darbojas ļoti labi, taču tiem ir viens būtisks trūkums, proti, fork(). Zabbix ir jāizveido jauns

process katru reizi, kad tas apstrādā lietotāja metriku, un tas nav labi veiktspējai. Parasti tā nav liela problēma, tomēr tā var kļūt par nopietnu jautājumu, uzraugot iegultās sistēmas, ja ir liels skaits uzraugāmo parametru vai tiek izmantoti smagnēji skripti ar sarežģītu loģiku vai ilgu palaišanas laiku.

Ielādējamo moduļu atbalsts piedāvā veidus, kā paplašināt Zabbix aģents, serveris un starpniekserveris, nezaudējot veiktspēju.

Ielādējamais modulis būtībā ir koplietojama bibliotēka, ko izmanto Zabbix dēmons un kas tiek ielādēta palaišanas laikā. Bibliotēkai ir jāsatur noteiktas funkcijas, lai Zabbix process varētu noteikt, ka fails patiešām ir modulis, ko tas var ielādēt un izmantot.

Ielādējamiem moduļiem ir vairākas priekšrocības. Liela veiktspēja un iespēja īstenot jebkādu loģiku ir ļoti svarīgas, taču, iespējams, vissvarīgākā priekšrocība ir iespēja izstrādāt, izmantot un koplietot Zabbix moduļus. Tas veicina bezproblēmu uzturēšanu un palīdz vieglāk nodrošināt jaunu funkcionalitāti neatkarīgi no Zabbix pamata koda bāzes.

Moduļu licencēšanu un izplatīšanu binārā formā regulē AGPL-3.0 licence (moduļi izpildlaikā tiek sasaistīti ar Zabbix un izmanto Zabbix galvenes; viss Zabbix kods ir licencēts saskaņā ar AGPL-3.0 licenci kopš Zabbix 7.0). Binārā saderība Zabbix netiek garantēta.

Moduļu API stabilitāte tiek garantēta viena Zabbix LTS (Long Term Support) izlaiduma cikla laikā. Zabbix API stabilitāte netiek garantēta (tehniski no moduļa ir iespējams izsaukt Zabbix iekšējās funkcijas, taču nav garantijas, ka šādi moduļi darbosies).

Moduļa API

Lai koplietojamo bibliotēku varētu uzskatīt par Zabbix moduli, tai jāievieš un jāeksportē vairākas funkcijas. Pašlaik Zabbix moduļa API ir sešas funkcijas, no kurām tikai viena ir obligāta, bet pārējās piecas ir neobligātas.

Obligātā saskarne

Vienīgā obligātā funkcija ir **zbx_module_api_version()**:

```
int zbx_module_api_version(void);
```

Šai funkcijai jāatgriež šī moduļa ieviestā API versija, un, lai moduli varētu ielādēt, šai versijai jāsakrīt ar Zabbix atbalstīto moduļa API versiju. Zabbix atbalstītā moduļa API versija ir ZBX_MODULE_API_VERSION. Tāpēc šai funkcijai jāatgriež šī konstante. Vecā konstante ZBX_MODULE_API_VERSION_ONE, kas tika izmantota šim nolūkam, tagad ir definēta kā vienāda ar ZBX_MODULE_API_VERSION, lai saglabātu pirmkoda saderību, taču tās lietošana nav ieteicama.

Neobligātā saskarne

Neobligātās funkcijas ir **zbx_module_init()**, **zbx_module_item_list()**, **zbx_module_item_timeout()**, **zbx_module_history_write_cbs()** un **zbx_module_uninit()**:

```
int zbx_module_init(void);
```

Šai funkcijai jāveic modulim nepieciešamā inicializācija (ja tāda ir). Veiksmes gadījumā tai jāatgriež ZBX_MODULE_OK. Pretējā gadījumā tai jāatgriež ZBX_MODULE_FAIL. Pēdējā gadījumā Zabbix nestartēsies.

```
ZBX_METRIC  *zbx_module_item_list(void);
```

Šai funkcijai jāatgriež moduļa atbalstīto vienumu saraksts. Katrs vienums ir definēts ZBX_METRIC struktūrā; sīkāku informāciju skatiet tālāk esošajā sadaļā. Saraksts beidzas ar ZBX_METRIC struktūru, kuras lauka "key" vērtība ir NULL.

```
void    zbx_module_item_timeout(int timeout);
```

Ja modulis eksportē **zbx_module_item_list()**, tad Zabbix izmanto šo funkciju, lai norādītu Zabbix konfigurācijas failā iestatīto noildzi, kas jāievēro moduļa realizētajām vienumu pārbaudēm. Šeit parametrs "timeout" ir sekundēs.

```
ZBX_HISTORY_WRITE_CBS   zbx_module_history_write_cbs(void);
```

Šai funkcijai jāatgriež atgriezeniskā izsaukuma funkcijas, ko Zabbix serveris izmantos dažādu datu tipu vēstures eksportēšanai. Atgriezeniskā izsaukuma funkcijas ir norādītas kā ZBX_HISTORY_WRITE_CBS struktūras lauki; lauki var būt NULL, ja moduli neinteresē noteikta tipa vēsture.

```
int zbx_module_uninit(void);
```

Šai funkcijai jāveic nepieciešamā deinicializācija (ja tāda ir), piemēram, jāatbrīvo piešķirtie resursi, jāaizver failu deskriptori utt.

Visas funkcijas tiek izsauktas vienu reizi Zabbix palaišanas laikā, kad modulis tiek ielādēts, izņemot zbx_module_uninit(), kas tiek izsaukta vienu reizi Zabbix apturēšanas laikā, kad modulis tiek izlādēts.

Vienumu definēšana

Katrs vienums tiek definēts ZBX_METRIC struktūrā:

```
typedef struct
{
    char        *key;
```

```
    unsigned    flags;
    int     (*function)();
    char        *test_param;
}
ZBX_METRIC;
```

Šeit **key** ir vienuma atslēga (piemēram, ”dummy.random”), **flags** ir vai nu CF_HAVEPARAMS, vai 0 (atkarībā no tā, vai vienums pieņem parametrus), **function** ir C funkcija, kas realizē vienumu (piemēram, ”zbx_module_dummy_random”), un **test_param** ir parametru saraksts, kas jāizmanto, kad Zabbix aģents tiek palaists ar karodziņu ”-p” (piemēram, ”1,1000”, var būt NULL). Definīcijas piemērs var izskatīties šādi:

```
static ZBX_METRIC keys[] =
{
    { "dummy.random", CF_HAVEPARAMS, zbx_module_dummy_random, "1,1000" },
    { NULL }
}
```

Katrai funkcijai, kas realizē vienumu, jāpieņem divi rādītāja parametri: pirmais ar tipu AGENT_REQUEST un otrais ar tipu 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;
}
```

Šīm funkcijām jāatgriež SYSINFO_RET_OK, ja vienuma vērtība tika veiksmīgi iegūta. Pretējā gadījumā tām jāatgriež SYSINFO_RET_FAIL. Sīkāku informāciju par to, kā iegūt informāciju no AGENT_REQUEST un kā iestatīt informāciju AGENT_RESULT, skatiet tālāk esošajā moduļa ”dummy” piemērā.

Vēstures eksportēšanas atgriezenisko izsaukumu nodrošināšana

Attention:
Vēstures eksportēšana, izmantojot moduli, vairs netiek atbalstīta Zabbix starpniekserverī.

Modulis var norādīt funkcijas vēstures datu eksportēšanai pēc tipa: Numeric (float), Numeric (unsigned), Character, Text un 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;
```

Katrai no tām kā argumenti jāpieņem ”history” masīvs ar ”history_num” elementiem. Atkarībā no eksportējamā vēstures datu tipa ”history” ir attiecīgi šādu struktūru masīvs:

```
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;
```

Atgriezeniskie izsaukumi tiks izmantoti Zabbix servera vēstures sinhronizētāja procesos vēstures sinhronizācijas procedūras beigās pēc tam, kad dati ir ierakstīti Zabbix datubāzē un saglabāti vērtību kešatmiņā.

Attention:
Ja vēstures eksportēšanas modulī rodas iekšēja kļūda, ieteicams moduli izstrādāt tā, lai tas nebloķētu visu uzraudzību līdz atkopšanai, bet tā vietā atmestu datus un ļautu Zabbix serverim turpināt darbību.

Moduļu veidošana

Pašlaik moduļus ir paredzēts veidot Zabbix pirmkoda kokā, jo moduļa API ir atkarīgs no dažām datu struktūrām, kas ir definētas Zabbix galvenēs.

Svarīgākā galvene ielādējamiem moduļiem ir **include/module.h**, kas definē šīs datu struktūras. Citas nepieciešamās sistēmas galvenes, kas palīdz **include/module.h** darboties pareizi, ir **stdlib.h** un **stdint.h**.

Ņemot vērā šo informāciju, viss ir gatavs moduļa veidošanai. Modulim jāiekļauj **stdlib.h**, **stdint.h** un **module.h**, un veidošanas skriptam jānodrošina, ka šie faili ir iekļaušanas ceļā. Sīkāku informāciju skatiet tālāk redzamajā "dummy" moduļa piemērā.

Vēl viena noderīga galvene ir **include/zbxcommon.h**, kas definē funkciju **zabbix_log()**, kuru var izmantot žurnalēšanai un atkļūdošanai.

Konfigurācijas parametri

Zabbix aģents, serveris un starpniekserveris atbalsta divus parametrus, lai strādātu ar moduļiem:

- LoadModulePath – pilns ceļš uz ielādējamo moduļu atrašanās vietu
- LoadModule – modulis(-ļi), ko ielādēt startēšanas laikā. Moduļiem jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma. Ja iepriekš norādītais ceļš ir absolūts (sākas ar '/'), tad LoadModulePath tiek ignorēts. Ir atļauts iekļaut vairākus LoadModule parametrus.

Piemēram, lai paplašinātu Zabbix aģentu, mēs varētu pievienot šādus parametrus:

```
LoadModulePath=/usr/local/lib/zabbix/agent/
LoadModule=mariadb.so
LoadModule=apache.so
LoadModule=kernel.so
LoadModule=/usr/local/lib/zabbix/dummy.so
```

Aģenta startēšanas laikā tas ielādēs moduļus mariadb.so, apache.so un kernel.so no direktorija /usr/local/lib/zabbix/agent, savukārt dummy.so tiks ielādēts no /usr/local/lib/zabbix. Aģentu neizdosies palaist, ja kāds modulis trūkst, ir nepareizas piekļuves tiesības vai ja koplietojamā bibliotēka nav Zabbix modulis.

Lietotāja saskarnes konfigurācija

Ielādējamie moduļi tiek atbalstīti Zabbix aģentā, serverī un starpniekserverī. Tāpēc vienuma tips Zabbix lietotāja saskarnē ir atkarīgs no tā, kur modulis ir ielādēts. Ja modulis ir ielādēts aģentā, tad vienuma tipam jābūt "Zabbix aģents" vai "Zabbix aģents (aktīvs)". Ja modulis ir ielādēts serverī vai starpniekserverī, tad vienuma tipam jābūt "Vienkārša pārbaude".

Vēstures eksportam caur Zabbix moduļiem nav nepieciešama nekāda lietotāja saskarnes konfigurācija. Ja moduli serveris ir veiksmīgi ielādējis un tas nodrošina funkciju **zbx_module_history_write_cbs()**, kas atgriež vismaz vienu callback funkciju, kas nav NULL, tad vēstures eksports tiks iespējots automātiski.

Demonstrācijas modulis

Zabbix ietver C valodā rakstītu moduļa paraugu. Modulis atrodas mapē 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
```

Modulis ir labi dokumentēts, to var izmantot kā veidni saviem moduļiem.

Pēc tam, kad Zabbix pirmkoda koka saknē ir izpildīts ./configure, kā aprakstīts iepriekš, vienkārši palaidiet **make**, lai izveidotu **dummy.so**.

```
/*
** Zabbix
** Copyright (C) 2001-2020 Zabbix SIA
**
** This program is free software; you can redistribute it and/or modify
** it under the terms of the GNU General Public License as published by
** the Free Software Foundation; either version 2 of the License, or
** (at your option) any later version.
**
** This program is distributed in the hope that it will be useful,
** but WITHOUT ANY WARRANTY; without even the implied warranty of
** MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
** GNU General Public License for more details.
**
** You should have received a copy of the GNU General Public License
** along with this program; if not, write to the Free Software
** Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  02110-1301, USA.
**/

###include <stdlib.h>
###include <string.h>
###include <time.h>
###include <stdint.h>

###include "module.h"

/* the variable keeps timeout setting for item processing */
static int  item_timeout = 0;

/* module SHOULD define internal functions as static and use a naming pattern different from Zabbix intern
/* symbols (zbx_*) and loadable module API functions (zbx_module_*) to avoid conflicts
static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result);
```

```
static int  dummy_echo(AGENT_REQUEST *request, AGENT_RESULT *result);
static int  dummy_random(AGENT_REQUEST *request, AGENT_RESULT *result);

static ZBX_METRIC keys[] =
/*  KEY         FLAG        FUNCTION    TEST PARAMETERS */
{
    {"dummy.ping",      0,      dummy_ping, NULL},
    {"dummy.echo",      CF_HAVEPARAMS,  dummy_echo, "a message"},
    {"dummy.random",    CF_HAVEPARAMS,  dummy_random,   "1,1000"},
    {NULL}
};

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_api_version                                           *
 *                                                                            *
 * Purpose: returns version number of the module interface                    *
 *                                                                            *
 * Return value: ZBX_MODULE_API_VERSION - version of module.h module is       *
 *               compiled with, in order to load module successfully Zabbix   *
 *               MUST be compiled with the same version of this header file   *
 *                                                                            *
 ******************************************************************************/
int zbx_module_api_version(void)
{
    return ZBX_MODULE_API_VERSION;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_timeout                                          *
 *                                                                            *
 * Purpose: set timeout value for processing of items                         *
 *                                                                            *
 * Parameters: timeout - timeout in seconds, 0 - no timeout set               *
 *                                                                            *
 ******************************************************************************/
void    zbx_module_item_timeout(int timeout)
{
    item_timeout = timeout;
}

/******************************************************************************
 *                                                                            *
 * Function: zbx_module_item_list                                             *
 *                                                                            *
 * Purpose: returns list of item keys supported by the module                 *
 *                                                                            *
 * Return value: list of item keys                                            *
 *                                                                            *
 ******************************************************************************/
ZBX_METRIC  *zbx_module_item_list(void)
{
    return keys;
}

static int  dummy_ping(AGENT_REQUEST *request, AGENT_RESULT *result)
{
    SET_UI64_RESULT(result, 1);

    return SYSINFO_RET_OK;
}
```

```c
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;
}
```

Modulis eksportē trīs jaunus vienumus:

- `dummy.ping` - vienmēr atgriež '1'
- `dummy.echo[param1]` - atgriež pirmo parametru tādu, kāds tas ir, piemēram, `dummy.echo[ABC]` atgriezīs ABC
- `dummy.random[param1, param2]` - atgriež nejaušu skaitli param1-param2 diapazonā, piemēram, `dummy.random[1,1000000]`

Ierobežojumi

Ielādējamo moduļu atbalsts ir ieviests tikai Unix platformai. Tas nozīmē, ka tas nedarbojas Windows aģentiem.

Dažos gadījumos modulim var būt nepieciešams nolasīt ar moduli saistītus konfigurācijas parametrus no *zabbix_agentd.conf*. Pašlaik tas netiek atbalstīts. Ja jums ir nepieciešams, lai jūsu modulis izmantotu kādus konfigurācijas parametrus, jums, iespējams, vajadzētu ieviest modulim specifiska konfigurācijas faila parsēšanu.

2 Lietotāja saskarnes moduļi

Pārskats

Ir iespējams paplašināt Zabbix lietotāja saskarne funkcionalitāti, pievienojot trešo pušu moduļus vai izstrādājot savus moduļus, nemainot Zabbix pirmkodu.

Ņemiet vērā, ka moduļa kods tiks izpildīts ar tādām pašām privilēģijām kā Zabbix pirmkods. Tas nozīmē:

- trešo pušu moduļi var būt kaitīgi. Jums ir jāuzticas moduļiem, kurus instalējat;
- kļūdas trešās puses moduļa kodā var izraisīt lietotāja saskarne darbības pārtraukumu. Ja tas notiek, vienkārši noņemiet moduļa kodu no lietotāja saskarne. Tiklīdz atkārtoti ielādēsiet Zabbix lietotāja saskarne, redzēsiet paziņojumu, ka daži moduļi nav pieejami. Dodieties uz Moduļu administrēšanu (*Administrēšana* → *Vispārīgi* → *Moduļi*) un vēlreiz noklikšķiniet uz *Skenēt direktoriju*, lai no datubāzes noņemtu neeksistējošos moduļus.

Instalēšana

Lūdzu, vienmēr izlasiet konkrētā moduļa instalēšanas rokasgrāmatu. Ir ieteicams instalēt jaunus moduļus pa vienam, lai kļūmes būtu vieglāk pamanīt.

Tieši pirms moduļa instalēšanas:

- Pārliecinieties, ka modulis ir lejupielādēts no uzticama avota. Kaitīga koda instalēšana var izraisīt sekas, piemēram, datu zudumu
- Viena un tā paša moduļa dažādas versijas (ar vienādu ID) var tikt instalētas paralēli, taču vienlaikus var būt iespējota tikai viena versija

Moduļa instalēšanas darbības:

- Izpakojiet moduli tā atsevišķā mapē Zabbix lietotāja saskarnees mapē `modules`
- Pārliecinieties, ka moduļa mapē ir vismaz fails manifest.json
- Atveriet Module administration un noklikšķiniet uz pogas *Scan directory*
- Jaunais modulis parādīsies sarakstā kopā ar tā versiju, autoru, aprakstu un statusu
- Iespējojiet moduli, noklikšķinot uz tā statusa

Problēmu novēršana:

Problem	Solution
Modulis neparādījās sarakstā	Ja modulis neparādījās sarakstā, pārliecinieties, ka modules/your-module/ pastāv fails manifest.json. Ja tā nav, iespējams, modulis tika izpakots nepareizajā direktorijā. Ja tas pastāv, modulis var nebūt saderīgs ar jūsu Zabbix versiju. Pārbaudiet arī, vai tīmekļa servera lietotājam ir vismaz lasīšanas un meklēšanas piekļuve (`r-x`) moduļa mapei un visām apakšdirektorijām, kā arī lasīšanas piekļuve (`r--`) visiem tajā esošajiem failiem.
Lietotāja saskarne avarēja	Moduļa kods nav saderīgs ar pašreizējo Zabbix versiju vai servera konfigurāciju. Lūdzu, izdzēsiet moduļa failus un pārlādējiet lietotāja saskarni. Jūs redzēsiet paziņojumu, ka daži moduļi nav pieejami. Atveriet Module administration un vēlreiz noklikšķiniet uz *Scan directory*, lai no datubāzes noņemtu neeksistējošos moduļus.
Parādās kļūdas ziņojums par identisku nosaukumvietu, ID vai darbībām	Jaunais modulis mēģināja reģistrēt nosaukumvietu, ID vai darbības, kas jau ir reģistrētas citos iespējotos moduļos. Pirms jaunā moduļa iespējošanas atspējojiet konfliktējošo moduli (minēts kļūdas ziņojumā).
Parādās tehniski kļūdu ziņojumi	Ziņojiet par kļūdām moduļa izstrādātājam.

Moduļu izstrāde

Informāciju par pielāgotu moduļu izstrādi skatiet Izstrādātāju centrā.

20 Pielikumi

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam sadaļā "Pielikumi".

1 Instalēšana un iestatīšana

Lūdzu, izmantojiet sānjoslu, lai piekļūtu šīs sadaļas saturam.

1 Datubāzes izveide

Pārskats

Zabbix datubāze ir jāizveido Zabbix servera vai starpniekservera instalēšanas laikā.

Šajā sadaļā ir sniegtas instrukcijas Zabbix datubāzes izveidei. Katrai atbalstītajai datubāzei ir pieejams atsevišķs instrukciju kopums.

Note:
Lai uzlabotu datubāzes drošību, izveidojot datubāzes lomas/lietotājus ar minimālajām privilēģijām, skatiet datubāzes izveides paraugpraksi katrai atbalstītajai datubāzei:

- MySQL/MariaDB
- PostgreSQL/TimescaleDB

Lai konfigurētu drošus TLS savienojumus, skatiet Drošs savienojums ar datubāzi.

UTF-8 ir vienīgais Zabbix atbalstītais kodējums. Ir zināms, ka tas darbojas bez drošības ievainojamībām. Lietotājiem jāņem vērā, ka, izmantojot dažus citus kodējumus, pastāv zināmas drošības problēmas. Lai pārslēgtos uz UTF-8, skatiet Zabbix datubāzes rakstzīmju kopas un kolācijas labošana. Skatiet arī Filtrēšanas ierobežojumi ar utf8mb4 kolācijām.

Note:
Ja instalējat no Zabbix Git repository, pirms pārejas uz nākamajām darbībām ir jāizpilda šāda komanda:

 `make dbschema`

MySQL/MariaDB

Rakstzīmju kopas utf8 (jeb utf8mb3) un utf8mb4 tiek atbalstītas (attiecīgi ar utf8_bin un utf8mb4_bin kolāciju), lai Zabbix serveris/starpniekserveris varētu pareizi darboties ar MySQL datubāzi. Jaunām instalācijām ieteicams izmantot utf8mb4.

Deterministiski trigeri ir jāizveido shēmas importēšanas laikā. MySQL un MariaDB gadījumā tas prasa iestatīt GLOBAL `log_bin_trust_function_creators = 1`, ja ir iespējota binārā žurnalēšana, nav superlietotāja privilēģiju un MySQL konfigurācijas failā nav iestatīts `log_bin_trust_function_creators = 1`.

Ja instalējat no Zabbix **pakotnēm**, pārejiet uz savas platformas instrukcijām.

Ja instalējat Zabbix no pirmkoda:

- Izveidojiet un konfigurējiet datubāzi un lietotāju.

```
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;
```

- Importējiet datus datubāzē un iestatiet utf8mb4 rakstzīmju kopu kā noklusējuma vērtību (pieņemot, ka atrodaties Zabbix pirmkoda saknes direktorijā). Zabbix starpniekservera datubāzei jāimportē tikai `schema.sql` (nevis images.sql vai data.sql).

```
cd database/mysql
mysql -uzabbix -p<password> zabbix < schema.sql
#### stop here if you are creating database for Zabbix proxy
mysql -uzabbix -p<password> zabbix < images.sql
mysql -uzabbix -p<password> --default-character-set=utf8mb4 zabbix < data.sql
```

Pēc veiksmīgas shēmas importēšanas `log_bin_trust_function_creators` var atspējot:

```
mysql -uroot -p<password>

mysql> SET GLOBAL log_bin_trust_function_creators = 0;
mysql> quit;
```

PostgreSQL

Jums ir nepieciešams datubāzes lietotājs ar atļaujām izveidot datubāzes objektus.

Ja instalējat no Zabbix **pakotnēm**, pārejiet uz savas platformas instrukcijām.

Ja instalējat Zabbix no pirmkoda:

- Izveidojiet datubāzes lietotāju.

Šī shell komanda izveidos lietotāju `zabbix`. Kad tiks prasīts, norādiet paroli un atkārtojiet to (ņemiet vērā, ka vispirms jums var tikt prasīta sudo parole):

```
sudo -u postgres createuser --pwprompt zabbix
```

- Izveidojiet datubāzi.

Šī shell komanda izveidos datubāzi `zabbix` (pēdējais parametrs), izmantojot iepriekš izveidoto lietotāju kā īpašnieku (`-O zabbix`).

```
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

- Importējiet sākotnējo shēmu un datus (pieņemot, ka atrodaties Zabbix pirmkoda saknes direktorijā). Zabbix starpniekserveris datubāzei jāimportē tikai `schema.sql` (nevis images.sql vai data.sql).

```
cd database/postgresql
cat schema.sql | sudo -u zabbix psql zabbix
#### stop here if you are creating database for Zabbix proxy
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

Attention:
Iepriekš minētās komandas ir sniegtas kā piemērs, kas darbosies lielākajā daļā GNU/Linux instalāciju. Atkarībā no tā, kā ir konfigurēta jūsu sistēma/datubāze, varat izmantot arī citas komandas, piemēram:

 `psql -U <username>`

 Ja jums rodas jebkādas problēmas ar datubāzes iestatīšanu, lūdzu, sazinieties ar savu datubāzes administratoru.

TimescaleDB

Norādījumi TimescaleDB izveidei un konfigurēšanai ir sniegti atsevišķā sadaļā.

SQLite

SQLite izmantošana tiek atbalstīta tikai **Zabbix starpniekserverim**!

Datubāze tiks automātiski izveidota, ja tā nepastāv.

Atgriezieties uz instalēšanas sadaļu.

2 Zabbix datubāzes rakstzīmju kopas un kolācijas labošana

MySQL/MariaDB

Vēsturiski MySQL un tā atvasinājumi izmantoja 'utf8' kā utf8mb3 aizstājvārdu — MySQL paša 3 baitu standarta UTF8 implementāciju, kas patiesībā ir 4 baitu. Sākot ar MySQL 8.0.28 un MariaDB 10.6.1, rakstzīmju kopa 'utf8mb3' ir novecojusi, un kādā brīdī tās atbalsts tiks pārtraukts, savukārt 'utf8' kļūs par atsauci uz 'utf8mb4'. Kopš Zabbix 6.0 tiek atbalstīts 'utf8mb4'. Lai izvairītos no problēmām nākotnē, ir ļoti ieteicams izmantot 'utf8mb4'. Vēl viena pārejas uz 'utf8mb4' priekšrocība ir papildu Unicode rakstzīmju atbalsts.

Warning:
Tā kā versijas pirms Zabbix 6.0 neatpazīst utf8mb4, pirms utf8mb4 konvertēšanas izpildes vispirms noteikti jauniniet Zabbix serveris un DB shēmu uz 6.0.x vai jaunāku versiju.

1. Pārbaudiet datubāzes rakstzīmju kopu un kolāciju.

Piemēram:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| latin2                   | latin2 _general_ci   |
+--------------------------+----------------------+
```

Vai:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8                     | utf8_bin             |
+--------------------------+----------------------+
```

Kā redzams, rakstzīmju kopa šeit nav 'utf8mb4', un kolācija nav 'utf8mb4_bin', tāpēc tās ir jāizlabo.

2. Apturiet Zabbix.

3. Izveidojiet datubāzes rezerves kopiju!

4. Izlabojiet rakstzīmju kopu un kolāciju datubāzes līmenī:

```
alter database <your DB name> character set utf8mb4 collate utf8mb4_bin;
```

Izlabotās vērtības:

```
mysql> SELECT @@character_set_database, @@collation_database;
+--------------------------+----------------------+
| @@character_set_database | @@collation_database |
+--------------------------+----------------------+
| utf8mb4                  | utf8mb4_bin          |
+--------------------------+----------------------+
```

5. Ielādējiet skriptu , lai izlabotu rakstzīmju kopu un kolāciju tabulu un kolonnu līmenī:

```
mysql <your DB name> < utf8mb4_convert.sql
```

6. Izpildiet skriptu:

```
                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;
```

Lūdzu, ņemiet vērā, ka 'utf8mb4' var patērēt nedaudz vairāk diska vietas.

7. Ja nav kļūdu — iespējams, vēlēsieties izveidot datubāzes rezerves kopiju ar izlaboto datubāzi.

8. Palaidiet Zabbix.

3 Database upgrade to primary keys

Pārskats

Šajā sadaļā ir sniegti norādījumi, kā manuāli jaunināt tabulas esošajās instalācijās uz primārajām atslēgām.

Jaunināšana uz primārajām atslēgām optimizē datu indeksēšanu un piekļuvi tiem, kas var paātrināt vaicājumus un ietaupīt vietu. Tā arī uzlabo datu pārvaldību un sinhronizāciju klasteru iestatījumos, palīdzot mērogošanā un nodrošinot, ka sistēma saglabājas uzticama pat tad, ja daži serveri nedarbojas.

> **Attention:**
> Šajā lapā sniegtie norādījumi ir paredzēti pieredzējušiem lietotājiem, un tos var būt nepieciešams pielāgot jūsu konkrētajai konfigurācijai. Jaunināšana uz primārajām atslēgām var būt laikietilpīga un resursietilpīga. Pārliecinieties, ka ir pieejams pietiekami daudz brīvas diska vietas; atkarībā no jūsu datubāzes lieluma un saglabāto datu apjoma procesam var būt nepieciešams līdz pat 2,5 reizēm vairāk vietas nekā pašlaik izmanto vēstures tabulas.

Primārās atslēgas tiek izmantotas visām tabulām jaunās instalācijās kopš Zabbix 6.0.

Nav automātiskas datubāzes jauninašanas uz primārajām atslēgām; tomēr esošās instalācijas var tikt jauninātas manuāli **pēc** Zabbix serveris jauninašanas uz 6.0 vai jaunāku versiju.

> **Attention:**
> Kopš Zabbix 7.0 tabulu jaunināšana uz primārajām atslēgām vienlaikus jaunina arī tabulas, lai tās izmantotu dubultās precizitātes datu tipus.

 Ja jums ir Zabbix 7.0 (vai jaunāka versija), tabulas jau izmanto dubulto precizitāti. Tomēr šajā lapā sniegtos norādījumus joprojām var izmantot, lai jauninātu tabulas uz primārajām atslēgām, neietekmējot tabulas, kas jau izmanto dubulto precizitāti.

 Ja jums ir Zabbix 6.4 (vai vecāka versija), apsveriet iespēju vispirms jaunināt tabulas uz dubulto precizitāti. Plašāku informāciju skatiet Jaunināšana uz paplašināta diapazona skaitliskajām vērtībām Zabbix 7.0 dokumentācijā.

Norādījumi ir pieejami šādām sistēmām:

- MySQL
- PostgreSQL
- TimescaleDB

Svarīgas piezīmes

Lai veiktu datubāzes jauninašanu:

1. Apturiet Zabbix serveri.

Ir stingri ieteicams apturēt Zabbix serveri uz jauninašanas laiku. Tomēr, ja tas ir absolūti nepieciešams, jauninašanu var veikt arī laikā, kad serveris darbojas (tikai MySQL, MariaDB un PostgreSQL bez TimescaleDB gadījumā).

2. Izveidojiet savas datubāzes dublējumkopiju.
3. Instalējiet jaunāko zabbix-sql-scripts pakotni, kas ir saderīga ar jūsu Zabbix versiju (piemēram, RHEL gadījumā: `dnf install zabbix-sql-scripts`).
4. Palaidiet skriptus savai datubāzei.
5. Palaidiet Zabbix serveri.

> **Warning:**
> Palaidiet skriptus tikai servera datubāzei. starpniekserveris no šīs jauninašanas negūs labumu.

Ja datubāzē tiek izmantotas partīcijas, sazinieties ar DB administratoru vai Zabbix atbalsta dienestu, lai saņemtu palīdzību.

CSV failus pēc veiksmīgas jauninašanas uz primārajām atslēgām var noņemt.

Pēc izvēles Zabbix lietotāja saskarne var tikt pārslēgta uzturēšanas režīmā.

MySQL

Eksportēšana un importēšana jāveic tmux/screen vidē, lai nodrošinātu, ka sesija netiek pārtraukta.

Skatiet arī: Svarīgas piezīmes

MySQL 8.4+ ar mysqlsh

Šo metodi var izmantot ar darbojošos Zabbix serveris, taču ieteicams serveris apturēt uz jaunināšanas laiku. MySQL Shell (*mysqlsh*) jābūt instalētam un jāspēj izveidot savienojumu ar DB.

- Piesakieties MySQL konsolē kā root (ieteicams) vai kā jebkurš lietotājs ar FILE privilēģijām.
- Palaidiet MySQL ar iespējotu local_infile mainīgo.
- Pārdēvējiet vecās tabulas un izveidojiet jaunas tabulas, palaižot `history_upgrade_prepare.sql`.

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

- Eksportējiet un importējiet datus.

Izveidojiet savienojumu, izmantojot mysqlsh. Ja izmantojat savienojumu caur ligzdu, var būt nepieciešams norādīt ceļu.

```
sudo mysqlsh -uroot -S /run/mysqld/mysqld.sock --no-password -Dzabbix
```

Pārslēdzieties uz JavaScript režīmu, izmantojot:

```
\js
```

Pēc tam palaidiet tālāk norādīto kodu (CSVPATH var mainīt pēc nepieciešamības):

```
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" });
```

Ja saņemat ziņojumu "JavaScript is not supported", jūsu MySQL Shell instalācijai nav JS atbalsta. Šādā gadījumā instalējiet oficiālo MySQL Shell pakotni no Oracle (vai izveidojiet to no pirmkoda), lai JavaScript režīms būtu iespējots.

- Izpildiet pēcmigrācijas norādījumus, lai dzēstu vecās tabulas.

MariaDB/MySQL 8.4+ bez mysqlsh

Šī jaunināšanas metode aizņem vairāk laika, un tā jāizmanto tikai tad, ja jaunināšana ar *mysqlsh* nav iespējama.

Tabulu jaunināšana

- Piesakieties MySQL konsolē kā root (ieteicams) vai kā jebkurš lietotājs ar FILE privilēģijām.
- Ja migrāciju veicat ar darbojošos Zabbix serveri, palaidiet MySQL ar iespējotu local_infile mainīgo.
- Pārdēvējiet vecās tabulas un izveidojiet jaunas tabulas, palaižot `history_upgrade_prepare.sql`:

```
mysql -uzabbix -p<password> zabbix < /usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_pr
```

Migrācija ar apturētu serveri

Pirms datu migrācijas ir jāatspējo `max_execution_time` (MySQL) vai `max_statement_time` (MariaDB), lai migrācijas laikā izvairītos no noildzes.

MySQL gadījumā:

```
SET @@max_execution_time=0;
```

MariaDB gadījumā:

```
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;
```

Lai dzēstu vecās tabulas, izpildiet pēcmigrācijas norādījumus.

Migrācija ar darbojošos serveri

Pārbaudiet, kuriem ceļiem ir iespējota importēšana/eksportēšana:

```
mysql> SELECT @@secure_file_priv;
+-----------------------+
| @@secure_file_priv    |
+-----------------------+
| /var/lib/mysql-files/ |
+-----------------------+
```

Ja *secure_file_priv* vērtība ir ceļš uz direktoriju, eksportēšana/importēšana tiks veikta failiem šajā direktorijā. Šajā gadījumā attiecīgi rediģējiet failu ceļus vaicājumos vai iestatiet *secure_file_priv* vērtību uz tukšu virkni jaunināšanas laikam.

Ja *secure_file_priv* vērtība ir tukša, eksportēšanu/importēšanu var veikt no jebkuras atrašanās vietas.

Ja *secure_file_priv* vērtība ir NULL, iestatiet to uz ceļu, kurā atrodas eksportētie tabulu dati (iepriekš minētajā piemērā — '/var/lib/mysql-files/').

Papildinformāciju skatiet MySQL dokumentācijā vai MariaDB dokumentācijā.

Pirms datu eksportēšanas ir jāatspējo `max_execution_time` (MySQL) vai `max_statement_time` (MariaDB), lai eksportēšanas laikā izvairītos no noildzes.

MySQL gadījumā:

```
SET @@max_execution_time=0;
```

MariaDB gadījumā:

```
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
```

Lai dzēstu vecās tabulas, izpildiet pēcmigrācijas norādījumus.

PostgreSQL

Eksportēšana un importēšana jāveic tmux/screen vidē, lai nodrošinātu, ka sesija netiek pārtraukta. Instalācijām ar TimescaleDB izlaidiet šo sadaļu un pārejiet uz PostgreSQL + TimescaleDB.

Skatiet arī: Svarīgas piezīmes

Tabulu jaunināšana

- Pārdēvējiet tabulas, izmantojot `history_upgrade_prepare.sql`:

```
sudo -u zabbix psql zabbix < /usr/share/zabbix/sql-scripts/postgresql/option-patches/history_upgrade_prepa
```

Migrēšana ar apturētu serveri

- Eksportējiet pašreizējos vēstures datus, importējiet tos pagaidu tabulā un pēc tam ievietojiet datus jaunajās tabulās, ignorējot dublikātus:

```
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;
```

Skatiet ieteikumus INSERT veiktspējas uzlabošanai: PostgreSQL: Bulk Loading Huge Amounts of Data, Checkpoint Distance and Amount of WAL.

- Izpildiet pēcmigrācijas norādījumus, lai dzēstu vecās tabulas.

Migrēšana ar darbojošos serveri

- Eksportējiet pašreizējo vēsturi, importējiet to pagaidu tabulā un pēc tam ievietojiet datus jaunajās tabulās, ignorējot dublikātus:

```
\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;
```

- Izpildiet pēcmigrācijas norādījumus, lai dzēstu vecās tabulas.

PostgreSQL + TimescaleDB

Eksportēšana un importēšana jāveic tmux/screen vidē, lai nodrošinātu, ka sesija netiek pārtraukta. Jaunināšanas laikā Zabbix serverim jābūt apturētam.

Skatiet arī: Svarīgas piezīmes

- Pārdēvējiet tabulas, izmantojot `history_upgrade_prepare.sql`.
  - Ja saspiešana ir iespējota (noklusējuma instalācijā), palaidiet skriptu no /usr/share/zabbix/sql-scripts/postgresql/tim
    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/with-compression/history_
    ```
  - Ja saspiešana ir atspējota, palaidiet skriptu no /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-pat
    ```
    cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-patches/without-compression/histo
    ```
- Palaidiet TimescaleDB hypertable migrācijas skriptus atbilstoši saspiešanas iestatījumiem:
  - Ja saspiešana ir iespējota (noklusējuma instalācijā), palaidiet skriptus no /usr/share/zabbix/sql-scripts/postgresql/ti
    ```
    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
    ```
  - Ja saspiešana ir atspējota, palaidiet skriptus no /usr/share/zabbix/sql-scripts/postgresql/timescaledb/option-pa
    ```
    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
    ```

Skatiet arī: Padomi INSERT veiktspējas uzlabošanai.

- Lai dzēstu vecās tabulas, izpildiet pēcmigrācijas norādījumus.

Pēc migrācijas

Visām datubāzēm, kad migrācija ir pabeigta, veiciet tālāk norādītās darbības:

- Pārbaudiet, vai viss darbojas, kā paredzēts.
- Dzēsiet vecās tabulas:

```
DROP TABLE history_old;
DROP TABLE history_uint_old;
DROP TABLE history_str_old;
DROP TABLE history_log_old;
DROP TABLE history_text_old;
```

- TimescaleDB gadījumā dzēsiet arī šo veco tabulu:

```
DROP TABLE trends_old;
```

Skatiet arī

- Auditlog tabulas sagatavošana particionēšanai

4 Auditlog tabulas sagatavošana sadalīšanai nodalījumos

Pārskats

Dažām datubāzēm (piemēram, MySQL) ir nepieciešams, lai particionēšanas kolonna būtu daļa no tabulas unikālā ierobežojuma. Tādēļ, lai particionētu `auditlog` tabulu pēc laika, primārā atslēga ir jāmaina no `auditid` uz salikto atslēgu `auditid` + `clock`.

Šajā sadaļā ir sniegti norādījumi, kā mainīt `auditlog` tabulas primāro atslēgu.

Attention:
Šajā lapā sniegtie norādījumi ir paredzēti pieredzējušiem lietotājiem. Ņemiet vērā, ka šie norādījumi var būt jāpielāgo jūsu konkrētajai konfigurācijai. Primārās atslēgas maiņa var arī nebūt saderīga ar turpmākajiem jauninājumiem, tāpēc var būt nepieciešama turpmāko jauninājumu manuāla apstrāde.

 Primārās atslēgas maiņa var būt resursietilpīga darbība, kas atkarībā no `auditlog` tabulas lieluma var aizņemt daudz laika. Ieteicams uz izmaiņu veikšanas laiku apturēt Zabbix serveris un pārslēgt Zabbix lietotāja saskarne apkopes režīmā. Tomēr, ja tas ir absolūti nepieciešams, ir veids, kā mainīt primāro atslēgu bez dīkstāves (skatiet tālāk).

`auditlog` tabulas sadalīšana particijās var uzlabot, piemēram, novecojušo datu tīrīšanu lielās vidēs. Lai gan Zabbix novecojušo datu tīrīšana pašlaik nevar izmantot particionētu tabulu priekšrocības (izņemot TimescaleDB), varat atspējot Zabbix novecojušo datu tīrīšanu un dzēst particijas, izmantojot skriptus.

Sākot ar Zabbix 7.0, TimescaleDB `auditlog` tabula ir pārveidota par hipertabulu, kas ļauj novecojušo datu tīrīšanas procesam dzēst datus pa gabaliem. Lai jaunināt esošo `auditlog` tabulu uz hipertabulu, skatiet TimescaleDB shēmas jaunināšana.

MySQL

Svarīgas piezīmes par indeksu pārbūvi

MySQL automātiski pārbūvē primārās atslēgas indeksus `ALTER TABLE` darbības laikā. Tomēr ir ļoti ieteicams arī manuāli pārbūvēt indeksus, izmantojot `OPTIMIZE TABLE` priekšrakstu, lai nodrošinātu optimālu datubāzes veiktspēju.

Indeksu pārbūvei īslaicīgi var būt nepieciešams tikpat daudz papildu diska vietas, cik izmanto pati tabula. Lai iegūtu pašreizējo datu un indeksu lielumu, varat izpildīt šādus priekšrakstus:

```
ANALYZE TABLE auditlog;
SHOW TABLE STATUS LIKE 'auditlog';
```

Ja pieejamā diska vieta rada bažas, ievērojiet norādījumus sadaļā Primārās atslēgas maiņa bez dīkstāves. Ir pieejamas arī citas iespējas:

- MySQL parametra `sort_buffer_size` palielināšana var palīdzēt samazināt diska vietas izmantošanu, manuāli pārbūvējot indeksus. Tomēr šī mainīgā modificēšana var ietekmēt kopējo datubāzes atmiņas izmantošanu.
- Apsveriet iespēju atbrīvot vietu, dzēšot potenciāli nevajadzīgus datus.
- Apsveriet iespēju samazināt housekeeper parametru *Datu glabāšanas periods* pirms housekeeper izpildes.

Primārās atslēgas mainīšana ar dīkstāvi

1. Noņemiet pašreizējo `auditlog` tabulas primāro atslēgu un pievienojiet jauno primāro atslēgu.

```
ALTER TABLE auditlog DROP PRIMARY KEY, ADD PRIMARY KEY (auditid, clock);
```

2. Pārbūvējiet indeksus (neobligāti, bet ļoti ieteicams; skatiet Svarīgas piezīmes par indeksu pārbūvēšanu).

```
OPTIMIZE TABLE auditlog;
```

Primārās atslēgas maiņa bez dīkstāves

Šeit ir aprakstīta primārās atslēgas manuāla maiņas metode. Alternatīvi varat izmantot Percona pt-online-schema-change rīkkopu. Šī rīkkopa automātiski veic tālāk norādītās darbības, vienlaikus samazinot `auditlog` tabulas pārveidošanai izmantotās vietas apjomu.

1. Izveidojiet jaunu tabulu ar jauno primāro atslēgu un izveidojiet indeksus.

```
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. Samainiet tabulas.

```
RENAME TABLE auditlog TO auditlog_old, auditlog_new TO auditlog;
```

3. Kopējiet datus no vecās tabulas uz jauno tabulu.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

To var izdarīt pa daļām (vairāki `INSERT INTO` priekšraksti ar `WHERE clock` klauzulām pēc nepieciešamības), lai izvairītos no pārmērīga resursu patēriņa.

4. Dzēsiet veco tabulu.

```
DROP TABLE auditlog_old;
```

PostgreSQL

Svarīgas piezīmes par indeksu pārbūvēšanu

PostgreSQL automātiski pārbūvē primārās atslēgas indeksus `ALTER TABLE` darbības laikā. Tomēr ir ļoti ieteicams arī manuāli pārbūvēt indeksus ar `REINDEX TABLE CONCURRENTLY` priekšrakstu, lai nodrošinātu optimālu datubāzes veiktspēju.

Indeksu pārbūvēšanai īslaicīgi var būt nepieciešams līdz pat trīs reizēm vairāk diska vietas nekā pašlaik izmanto indeksi. Lai iegūtu pašreizējo indeksu izmēru, varat izpildīt šādu vaicājumu:

```
SELECT pg_size_pretty(pg_indexes_size('auditlog'));
```

Ja pieejamā diska vieta rada bažas, sekojiet norādījumiem sadaļā Primārās atslēgas maiņa bez dīkstāves. Ir pieejamas arī citas iespējas:

- PostgreSQL parametra `maintenance_work_mem` palielināšana var palīdzēt samazināt diska vietas izmantošanu manuālas indeksu pārbūves laikā. Tomēr šī mainīgā modificēšana var ietekmēt kopējo datubāzes atmiņas izmantošanu.
- Ja jums ir cits disks vai tabulvieta ar vairāk pieejamas vietas, varat apsvērt pagaidu glabāšanas vietas maiņu indeksu pārbūvei. Varat iestatīt PostgreSQL parametru `temp_tablespaces`, lai norādītu citu tabulvietu pagaidu objektiem.
- Apsveriet iespēju atbrīvot vietu, dzēšot potenciāli nevajadzīgus datus.
- Apsveriet iespēju samazināt housekeeper parametru *Datu glabāšanas periods* pirms housekeeper izpildes.

Primārās atslēgas mainīšana ar dīkstāvi

1. Noņemiet pašreizējo `auditlog` tabulas primāro atslēgu un pievienojiet jauno primāro atslēgu.

```
ALTER TABLE auditlog DROP CONSTRAINT auditlog_pkey;
ALTER TABLE auditlog ADD PRIMARY KEY (auditid,clock);
```

2. Pārbūvējiet indeksus (neobligāti, bet ļoti ieteicams, skatiet Svarīgas piezīmes par indeksu pārbūvi).

```
REINDEX TABLE CONCURRENTLY auditlog;
```

Primārās atslēgas maiņa bez dīkstāves

Šeit ir aprakstīta primārās atslēgas manuāla maiņas metode. Kā alternatīvu var apsvērt `pg_repack` paplašinājumu, lai izveidotu jaunu tabulu, kopētu datus un samainītu tabulas.

1. Izveidojiet jaunu tabulu ar jauno primāro atslēgu un izveidojiet indeksus.

```
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. Samainiet tabulas.

```
ALTER TABLE auditlog RENAME TO auditlog_old;
ALTER TABLE auditlog_new RENAME TO auditlog;
```

3. Kopējiet datus no vecās tabulas uz jauno tabulu.

```
INSERT INTO auditlog SELECT * FROM auditlog_old;
```

To var izdarīt pa daļām (vairāki `INSERT INTO` priekšraksti ar `WHERE clock` klauzulām pēc nepieciešamības), lai izvairītos no pārmērīga resursu patēriņa.

4. Dzēsiet veco tabulu.

```
DROP TABLE auditlog_old;
```

Skatīt arī

- Datubāzes jaunināšana uz primārajām atslēgām

5 Drošs savienojums ar datubāzi

Pārskats

Šajā sadaļā ir sniegti Zabbix iestatīšanas soļi un konfigurācijas piemēri drošiem TLS savienojumiem starp:

Datubāze	Zabbix komponenti
MySQL	Zabbix lietotāja saskarne, Zabbix serveris, Zabbix starpniekserveris
PostgreSQL	Zabbix lietotāja saskarne, Zabbix serveris, Zabbix starpniekserveris

Lai iestatītu savienojuma šifrēšanu DBVS ietvaros, skatiet oficiālo piegādātāja dokumentāciju, lai iegūtu sīkāku informāciju:

- MySQL: avota un replikas replikācijas datubāzu serveri.
- MySQL: grupu replikācijas u.c. datubāzu serveri.
- PostgreSQL šifrēšanas iespējas.

Visi piemēri ir balstīti uz MySQL CE (8.0) un PgSQL (13) GA laidieniem, kas pieejami oficiālajās repozitorijās un izmanto CentOS 8.

Prasības

Lai iestatītu šifrēšanu, ir nepieciešams:

- Izstrādātāja atbalstīta operētājsistēma ar OpenSSL >=1.1.X vai alternatīvu.

> **Note:**
> Ieteicams izvairīties no operētājsistēmām, kuru dzīves cikls ir beidzies, īpaši jaunu instalāciju gadījumā

- Datubāzes dzinis (RDBMS), kas ir instalēts un uzturēts no oficiālā repozitorija, ko nodrošina izstrādātājs. Operētājsistēmas bieži tiek piegādātas ar novecojušām datubāzu programmatūras versijām, kurām šifrēšanas atbalsts nav ieviests, piemēram, uz RHEL 7 balstītas sistēmas un PostgreSQL 9.2, MariaDB 5.5 bez šifrēšanas atbalsta.

Terminoloģija

Iestatot šo opciju, tiek piespiests izmantot TLS savienojumu ar datubāzi no Zabbix servera/starpniekservera un lietotāja saskarnes uz datubāzi:

- `required` - izveidot savienojumu, izmantojot TLS kā transporta režīmu bez identitātes pārbaudēm
- `verify_ca` - izveidot savienojumu, izmantojot TLS, un pārbaudīt sertifikātu
- `verify_full` - izveidot savienojumu, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, vai datubāzes identitāte (CN), kas norādīta DBHost, atbilst tās sertifikātam

Zabbix konfigurācija

Lietotāja saskarne uz datubāzi

Drošu savienojumu ar datubāzi var konfigurēt lietotāja saskarnes instalēšanas laikā:

- Atzīmējiet izvēles rūtiņu *Database TLS encryption* solī Configure DB connection, lai iespējotu transporta šifrēšanu.
- Atzīmējiet izvēles rūtiņu *Verify database certificate*, kas parādās, kad ir atzīmēts lauks *TLS encryption*, lai iespējotu šifrēšanu ar sertifikātiem.

Note:
MySQL gadījumā izvēles rūtiņa *Database TLS encryption* ir atspējota, ja *Database host* ir iestatīts uz localhost, jo savienojumu, kas izmanto ligzdas failu (Unix sistēmās) vai koplietojamo atmiņu (Windows sistēmās), nevar šifrēt.
 PostgreSQL gadījumā izvēles rūtiņa *TLS encryption* ir atspējota, ja lauka *Database host* vērtība sākas ar slīpsvītru vai lauks ir tukšs.

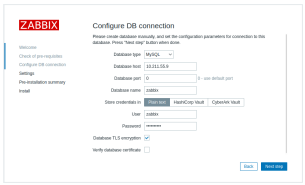
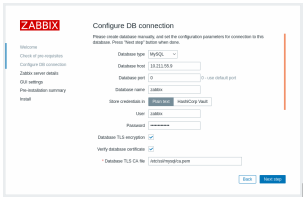
TLS šifrēšanas režīmā ar sertifikātiem kļūst pieejami šādi parametri (ja ir atzīmētas abas izvēles rūtiņas):

Parameter	Description
Database TLS CA file	Norādiet pilnu ceļu uz derīgu TLS sertifikācijas iestādes (CA) failu.
Database TLS key file	Norādiet pilnu ceļu uz derīgu TLS atslēgas failu.
Database TLS certificate file	Norādiet pilnu ceļu uz derīgu TLS sertifikāta failu.
Database host verification	Atzīmējiet šo izvēles rūtiņu, lai aktivizētu hosta verifikāciju. Atspējots MYSQL, jo PHP MySQL bibliotēka neļauj izlaist partnera sertifikāta validācijas soli.
Database TLS cipher list	Norādiet pielāgotu derīgo šifru sarakstu. Šifru saraksta formātam jāatbilst OpenSSL standartam. Pieejams tikai MySQL.

Attention:
TLS parametriem jānorāda uz derīgiem failiem. Ja tie norāda uz neeksistējošiem vai nederīgiem failiem, tas izraisīs autorizācijas kļūdu.
 Ja sertifikātu faili ir rakstāmi, lietotāja saskarne ģenerē brīdinājumu atskaitē System information, ka "TLS certificate files must be read-only." (tiek parādīts tikai tad, ja PHP lietotājs ir sertifikāta īpašnieks).
Ar parolēm aizsargāti sertifikāti netiek atbalstīti.

Lietošanas gadījumi

Zabbix lietotāja saskarne izmanto GUI saskarni, lai definētu iespējamās opcijas: `required`, `verify_ca`, `verify_full`. Norādiet nepieciešamās opcijas instalēšanas vedņa solī *Configure DB connections*. Šīs opcijas tiek attēlotas konfigurācijas failā (zabbix.conf.php) šādi:

GUI iestatījumi	Konfigurācijas fails	Apraksts	Rezultāts
	... // Izmanto TLS savienojumam. $DB['ENCRYPTION'] = true; $DB['KEY_FILE'] = ''; $DB['CERT_FILE'] = ''; $DB['CA_FILE'] = ''; $DB['VERIFY_HOST'] = false; $DB['CIPHER_LIST'] = ''; ...	Atzīmējiet *Database TLS encryption* Atstājiet *Verify database certificate* neatzīmētu	Iespējo `required` režīmu.
	... $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. Atzīmējiet *Database TLS encryption* un *Verify database certificate* 2. Norādiet ceļu uz *Database TLS CA file*	Iespējo `verify_ca` režīmu.

GUI iestatījumi	Konfigurācijas fails	Apraksts	Rezultāts

	...
	// Izmanto TLS savienojumam ar stingri noteiktu Cipher sarakstu.
	$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>';
	...
	
	Vai:
	
	...
	// Izmanto TLS savienojumam bez definēta Cipher saraksta - to izvēlas MySQL serveris
	$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. Atzīmējiet *Database TLS encryption* un *Verify database certificate*
		2. Norādiet ceļu uz *Database TLS key file*
		3. Norādiet ceļu uz *Database TLS CA file*
		4. Norādiet ceļu uz *Database TLS certificate file*
		5. Norādiet *Database TLS cipher list* (neobligāti)
			Iespējo `verify_full` režīmu 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. Atzīmējiet *Database TLS encryption* un *Verify database certificate* 2. Norādiet ceļu uz *Database TLS key file* 3. Norādiet ceļu uz *Database TLS CA file* 4. Norādiet ceļu uz *Database TLS certificate file* 5. Atzīmējiet *Database host verification*	Iespējo `verify_full` režīmu PostgreSQL.

Skatiet arī: Šifrēšanas konfigurācijas piemēri MySQL, Šifrēšanas konfigurācijas piemēri PostgreSQL.

Zabbix servera/starpniekservera konfigurācija

Drošus savienojumus ar datubāzi var konfigurēt ar attiecīgajiem parametriem Zabbix servera un/vai starpniekservera konfigurācijas failā.

Konfigurācija	Rezultāts
Nav	Savienojums ar datubāzi bez šifrēšanas.
1. Iestatiet `DBTLSConnect=required`	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi. Nešifrēts savienojums nav atļauts.
1. Iestatiet `DBTLSConnect=verify_ca` 2. Iestatiet `DBTLSCAFile` - norādiet TLS sertifikācijas iestādes failu	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi pēc datubāzes sertifikāta pārbaudes.
1. Iestatiet `DBTLSConnect=verify_full` 2. Iestatiet `DBTLSCAFile` - norādiet TLS sertifikācijas iestādes failu	Serveris/starpniekserveris izveido TLS savienojumu ar datubāzi pēc datubāzes sertifikāta un datubāzes hosta identitātes pārbaudes.
1. Iestatiet `DBTLSCAFile` - norādiet TLS sertifikācijas iestādes failu 2. Iestatiet `DBTLSCertFile` - norādiet klienta publiskās atslēgas sertifikāta failu 3. Iestatiet `DBTLSKeyFile` - norādiet klienta privātās atslēgas failu	Serveris/starpniekserveris, veidojot savienojumu ar datubāzi, nodrošina klienta sertifikātu.
1. Iestatiet `DBTLSCipher` - šifrēšanas šifru saraksts, ko klients atļauj savienojumiem, izmantojot TLS protokolus līdz TLS 1.2 vai `DBTLSCipher13` - šifrēšanas šifru saraksts, ko klients atļauj savienojumiem, izmantojot TLS 1.3 protokolu	(MySQL) TLS savienojums tiek izveidots, izmantojot šifru no norādītā saraksta. (PostgreSQL) Šīs opcijas iestatīšana tiks uzskatīta par kļūdu.

1 MySQL šifrēšanas konfigurācija

Pārskats

Šajā sadaļā ir sniegti vairāki šifrēšanas konfigurācijas piemēri CentOS 8.2 un MySQL 8.4.0 vidēm, un to var izmantot kā īso sākuma ceļvedi savienojuma ar datubāzi šifrēšanai.

Attention:
Ja MySQL hosts ir iestatīts kā localhost, šifrēšanas opcijas nebūs pieejamas. Šajā gadījumā savienojums starp Zabbix lietotāja saskarne un datubāzi izmanto ligzdas failu (Unix vidē) vai koplietojamo atmiņu (Windows vidē), un to nevar šifrēt.

Note:
Šifrēšanas kombināciju saraksts neaprobežojas tikai ar šajā lapā norādītajām. Ir pieejams daudz vairāk kombināciju.

Priekšnosacījumi

Instalējiet MySQL datubāzi no oficiālās repozitorija.

Skatiet MySQL dokumentāciju, lai iegūtu detalizētu informāciju par MySQL repozitorija izmantošanu.

MySQL serveris ir gatavs pieņemt drošus savienojumus, izmantojot pašparakstītu sertifikātu.

Lai redzētu, kuri lietotāji izmanto šifrētu savienojumu, izpildiet šādu vaicājumu (Performance Schema jābūt ieslēgtai):

```
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;
```

Tikai transporta šifrēšana

MySQL konfigurācija

Mūsdienu datubāzes versijas ir gatavas darbam uzreiz pēc instalēšanas ar `required` šifrēšanas režīmu. Pēc sākotnējās iestatīšanas un palaišanas tiks izveidots servera puses sertifikāts.

Izveidojiet lietotājus un lomas galvenajiem komponentiem:

```
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'@'%';
```

Ņemiet vērā, ka identitātes pārbaudei netiek izmantots X.509 protokols, taču lietotājs ir konfigurēts izmantot tikai šifrētus savienojumus. Plašāku informāciju par lietotāju konfigurēšanu skatiet MySQL dokumentācijā.

Palaidiet šo komandu, lai pārbaudītu savienojumu (drošu savienojumu testēšanai nevar izmantot socket savienojumu):

```
mysql -u zbx_srv -p -h 10.211.55.9 --ssl-mode=REQUIRED
```

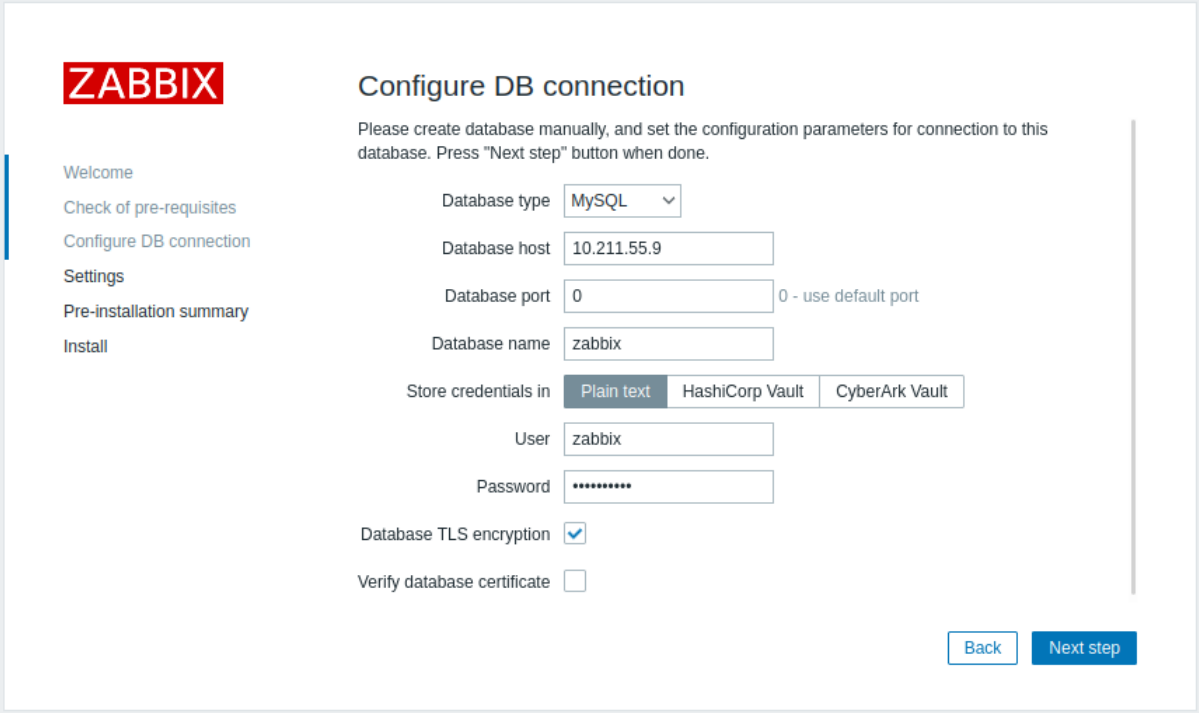
Pārbaudiet pašreizējo statusu un pieejamos šifru komplektus:

```
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
```

Lietotāja saskarne

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption*
- Atstājiet *Verify database certificate* neatzīmētu



Serveris

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp serveri un datubāzi, konfigurējiet */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Šifrēšana ar sertifikācijas iestādes verifikāciju

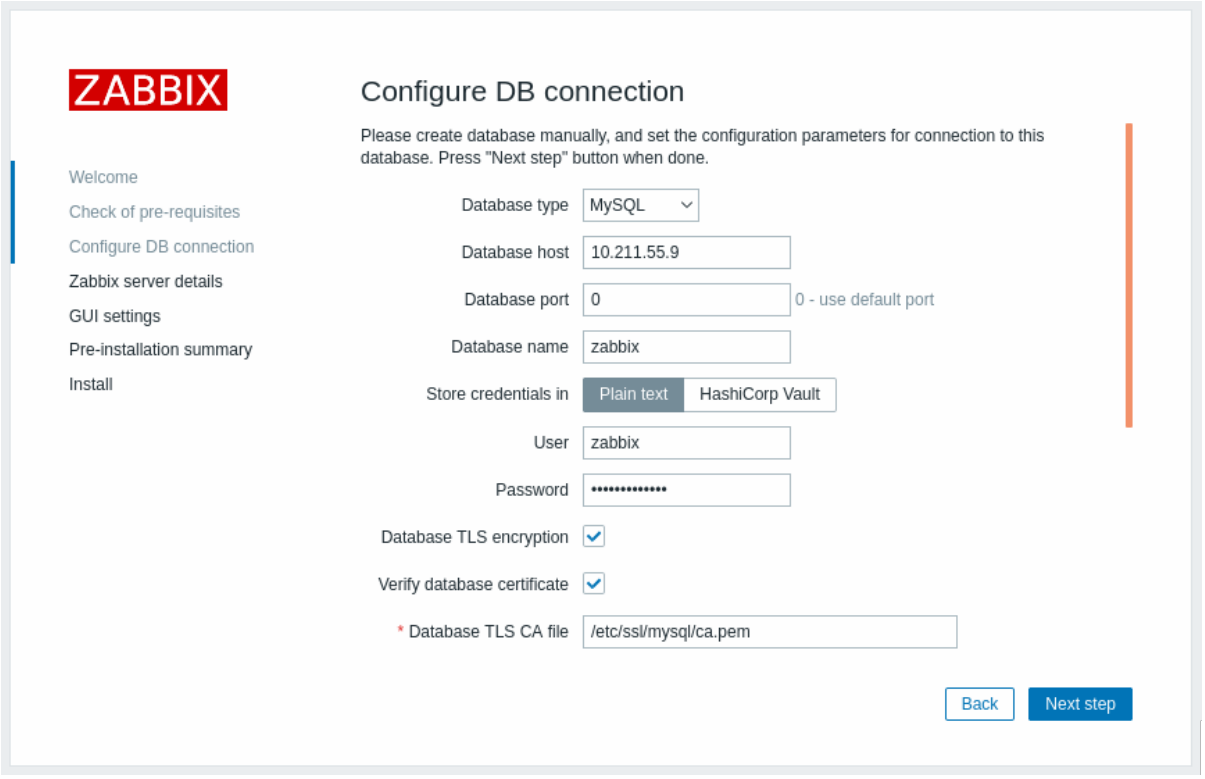
Nokopējiet nepieciešamo MySQL CA uz Zabbix lietotāja saskarnes serveri un piešķiriet atbilstošas atļaujas, lai tīmekļa serveris varētu nolasīt šo failu.

Note:
Šis režīms nedarbojas uz RHEL 7 vecāku MySQL bibliotēku dēļ.

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertifikāta pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet ceļu uz Database TLS CA failu



Alternatīvi, to var iestatīt failā */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'] = '';
...
```

Problēmu novēršanai lietotājs var izmantot komandrindas rīku, lai pārbaudītu, vai savienojums ir iespējams nepieciešamajam lietotājam:

```
    mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=REQUIRED --ssl-ca=/var/lib/mysql/ca.pem
```

Serveris

Lai iespējotu šifrēšanu ar sertifikāta verifikāciju savienojumiem starp Zabbix serveri un datubāzi, konfigurējiet */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
...
```

Šifrēšana ar pilnu verifikāciju

MySQL konfigurācija

Iestatiet MySQL CE servera konfigurācijas opciju (*/etc/my.cnf.d/server-tls.cnf*) šādi:

```
[mysqld]
...
##### šajos piemēros atslēgas atrodas MySQL CE datadir direktorijā
ssl_ca=ca.pem
ssl_cert=server-cert.pem
ssl_key=server-key.pem

require_secure_transport=ON
```

```
tls_version=TLSv1.3
...
```

MySQL CE servera un klienta (Zabbix lietotāja saskarne) atslēgas jāizveido manuāli saskaņā ar MySQL CE dokumentāciju: SSL un RSA sertifikātu un atslēgu izveide, izmantojot MySQL vaiSSL sertifikātu un atslēgu izveide, izmantojot openssl

Attention:
MySQL servera sertifikātā jābūt laukam Common Name, kas iestatīts uz FQDN nosaukumu, jo Zabbix lietotāja saskarne izmantos DNS nosaukumu saziņai ar datubāzi vai datubāzes hosta IP adresi.

Izveidojiet MySQL lietotāju:

```
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;
```

Pārbaudiet, vai ir iespējams pieteikties ar šo lietotāju:

```
mysql -u zbx_web -p -h 10.211.55.9 --ssl-mode=VERIFY_IDENTITY --ssl-ca=/var/lib/mysql/ca.pem --ssl-cert=/v
```

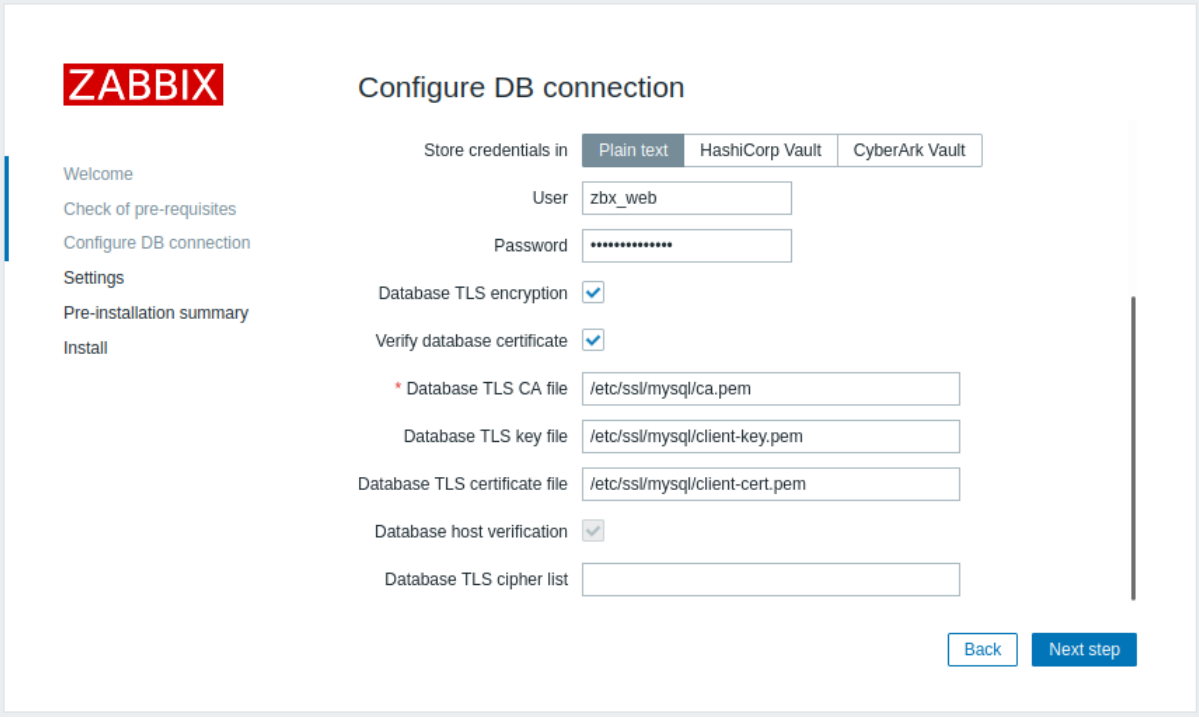
Lietotāja saskarne

Lai iespējotu šifrēšanu ar pilnu verifikāciju savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet ceļu uz *Database TLS key file*
- Norādiet ceļu uz *Database TLS CA file*
- Norādiet ceļu uz *Database TLS certificate file*

Ņemiet vērā, ka *Database host verification* ir atzīmēts un iekrāsots pelēks — šo soli MySQL gadījumā nevar izlaist.

Warning:
Ja lauks *Database TLS cipher list* ir atstāts tukšs, tiks iespējoti kopīgie šifri, ko atļauj gan lietotāja saskarne (klients), gan serveris. Alternatīvi, šifrus var iestatīt tieši, ievērojot šifru konfigurācijas prasības.



Alternatīvi, to var iestatīt failā */etc/zabbix/web/zabbix.conf.php*:

```
...
// Used for TLS connection with strictly defined Cipher list.
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
```

```
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256:TLS_AES_1
...
// or
...
// Used for TLS connection without Cipher list defined - selected by MySQL server
$DB['ENCRYPTION'] = true;
$DB['KEY_FILE'] = '/etc/ssl/mysql/client-key.pem';
$DB['CERT_FILE'] = '/etc/ssl/mysql/client-cert.pem';
$DB['CA_FILE'] = '/etc/ssl/mysql/ca.pem';
$DB['VERIFY_HOST'] = true;
$DB['CIPHER_LIST'] = '';
...
```

Serveris

Lai iespējotu šifrēšanu ar pilnu verifikāciju savienojumiem starp Zabbix serveri un datubāzi, konfigurējiet */etc/zabbix/zabbix_server.conf*:

```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_full
DBTLSCAFile=/etc/ssl/mysql/ca.pem
DBTLSCertFile=/etc/ssl/mysql/client-cert.pem
DBTLSKeyFile=/etc/ssl/mysql/client-key.pem
...
```

2 PostgreSQL šifrēšanas konfigurācija

Pārskats

Šajā sadaļā ir sniegti vairāki šifrēšanas konfigurācijas piemēri CentOS 8.2 un PostgreSQL 13.

> **Note:**
> Savienojumu starp Zabbix lietotāja saskarne un PostgreSQL nevar šifrēt (parametri GUI ir atspējoti), ja lauka *Database host* vērtība sākas ar slīpsvītru vai lauks ir tukšs.

Priekšnosacījumi

Instalējiet PostgreSQL datubāzi, izmantojot oficiālo repozitoriju.

PostgreSQL pēc noklusējuma nav konfigurēts TLS savienojumu pieņemšanai. Lūdzu, sekojiet PostgreSQL dokumentācijas norādījumiem par sertifikātu sagatavošanu ar postgresql.conf un arī par lietotāju piekļuves kontroli, izmantojot pg_hba.conf.

Pēc noklusējuma PostgreSQL ligzda ir piesaistīta localhost; lai atļautu attālinātus tīkla savienojumus, iespējojiet klausīšanos reālajā tīkla saskarnē.

PostgreSQL iestatījumi visiem režīmiem var izskatīties šādi:

/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'
...
```

Piekļuves kontrolei pielāgojiet */var/lib/pgsql/13/data/pg_hba.conf*:

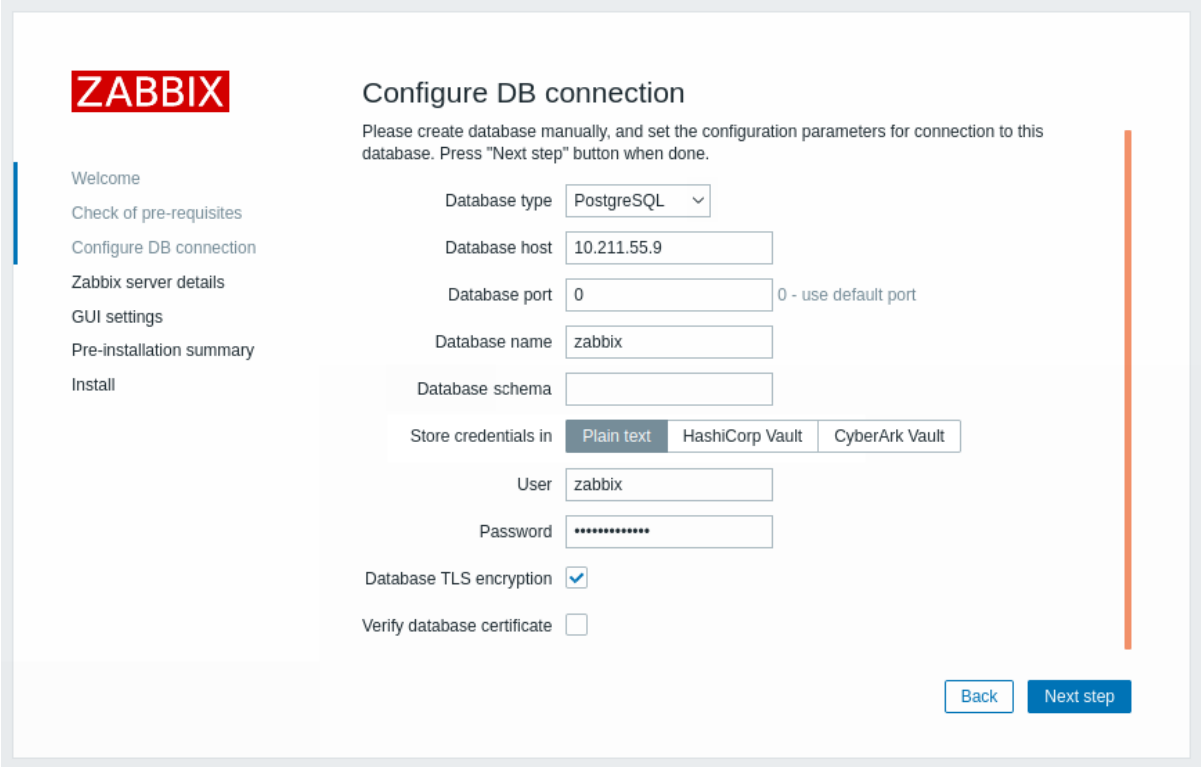
```
...
### require
hostssl all all 0.0.0.0/0 md5

### 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
...
```

Tikai transporta šifrēšana

Lietotāja saskarne

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Datubāzes TLS šifrēšana*
- Atstājiet *Pārbaudīt datubāzes sertifikātu* neatzīmētu



Serveris

Lai iespējotu tikai transporta šifrēšanu savienojumiem starp serveri un datu bāzi, konfigurējiet */etc/zabbix/zabbix_server.conf*:

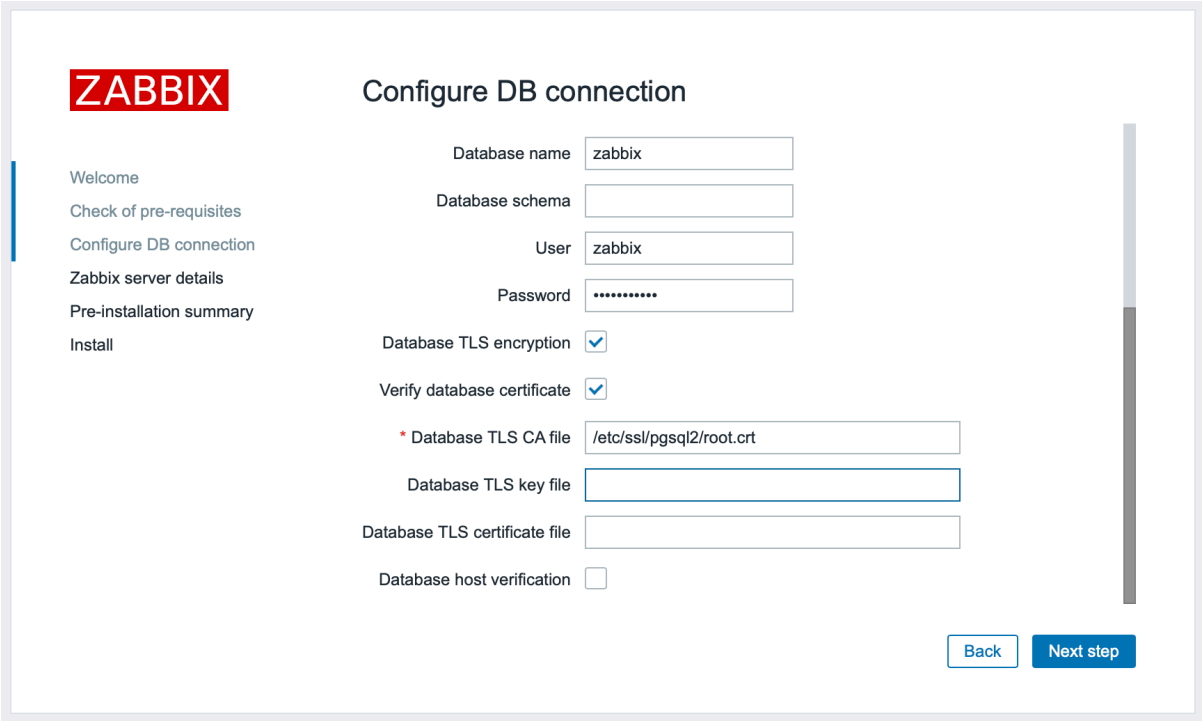
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=required
...
```

Šifrēšana ar sertifikātu iestādes verifikāciju

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertifikācijas iestādes verifikāciju savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet ceļu uz *Database TLS CA file*



Alternatīvi, to var iestatīt failā */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'] = '';
...
```

Serveris

Lai iespējotu šifrēšanu ar sertifikāta verifikāciju savienojumiem starp Zabbix serveri un datubāzi, konfigurējiet */etc/zabbix/zabbix_server.conf:*

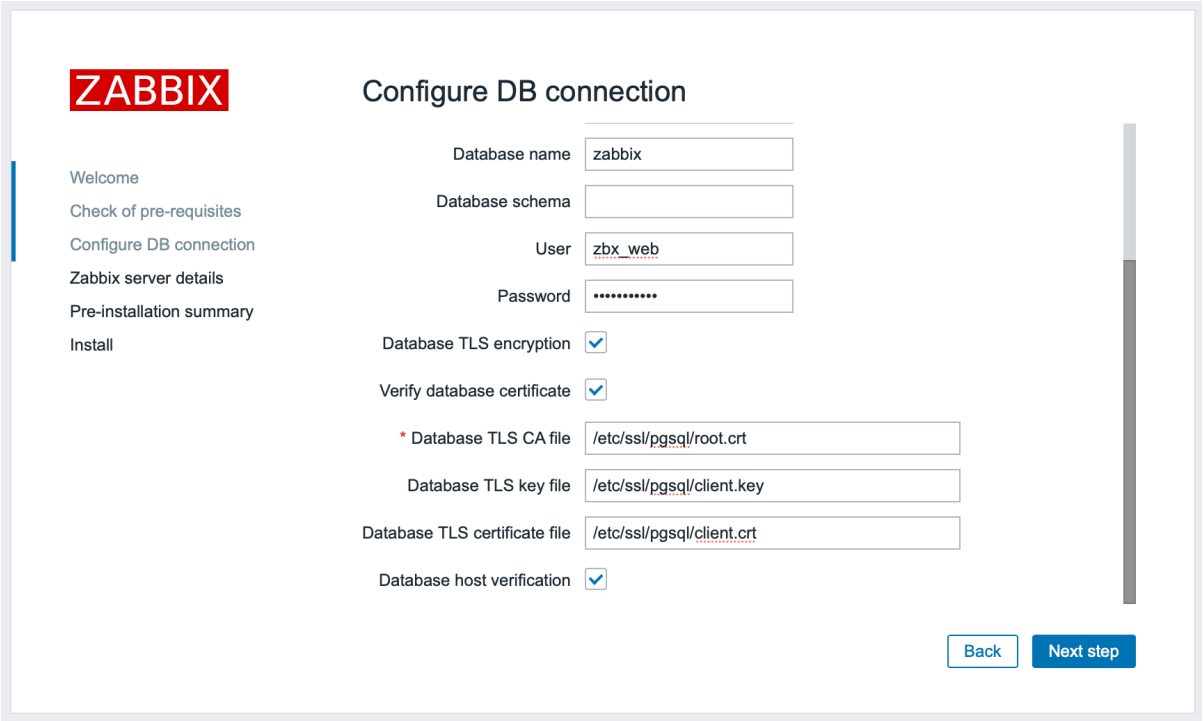
```
...
DBHost=10.211.55.9
DBName=zabbix
DBUser=zbx_srv
DBPassword=<strong_password>
DBTLSConnect=verify_ca
DBTLSCAFile=/etc/ssl/pgsql/root.crt
...
```

Šifrēšana ar pilnu verifikāciju

Lietotāja saskarne

Lai iespējotu šifrēšanu ar sertifikātu un datubāzes hosta identitātes pārbaudi savienojumiem starp Zabbix lietotāja saskarni un datubāzi:

- Atzīmējiet *Database TLS encryption* un *Verify database certificate*
- Norādiet ceļu uz *Database TLS key file*
- Norādiet ceļu uz *Database TLS CA file*
- Norādiet ceļu uz *Database TLS certificate file*
- Atzīmējiet *Database host verification*



Alternatīvi to var iestatīt failā */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'] = '';
...
```

Serveris

Lai iespējotu šifrēšanu ar sertifikātu un datubāzes hosta identitātes pārbaudi savienojumiem starp Zabbix serveri un datubāzi, konfigurējiet */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 Drošs savienojums ar lietotāja saskarni

Pārskats

Šajā sadaļā ir sniegti Zabbix iestatīšanas soļi un konfigurācijas piemēri drošiem TLS savienojumiem starp Zabbix lietotāja saskarne un Zabbix serveri.

Konfigurācija

Pēc noklusējuma saziņa starp Zabbix lietotāja saskarne un Zabbix serveris nav šifrēta. Lai uzlabotu drošību, iespējojiet TLS abās pusēs. Tālāk ir sniegts piemērs vienkāršākajam veidam, kā to izdarīt.

1. Ģenerējiet sertifikātus un atslēgas.

Izveidojiet darba direktoriju:

```
sudo mkdir -p /etc/zabbix/ssl && cd /etc/zabbix/ssl
```

Izveidojiet CA sertifikātu (pielāgojiet `MyZabbixCA` vērtību, lai tā atbilstu faktiskajam vispārpieņemtajam nosaukumam):

```
sudo openssl genrsa -out ca.key 4096
sudo openssl req -new -x509 -days 3650 -key ca.key -out ca.crt -subj "/CN=MyZabbixCA/"
```

Ģenerējiet privāto atslēgu un sertifikātu Zabbix serverim (pielāgojiet `zabbix-server.example.com` vērtību, lai tā atbilstu faktiskajam vispārpieņemtajam nosaukumam):

```
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
```

Ģenerējiet privāto atslēgu un sertifikātu Zabbix lietotāja saskarnei (pielāgojiet `zabbix-frontend-node` vērtību, lai tā atbilstu faktiskajam vispārpieņemtajam nosaukumam):

```
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. Iestatiet atbilstošas atļaujas.

Zabbix serverim (pielāgojiet īpašnieku/grupu atbilstoši jūsu distribūcijas Zabbix servera dēmona lietotājam):

```
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
```

Lietotāja saskarnei (pielāgojiet īpašnieku/grupu atbilstoši jūsu distribūcijas tīmekļa servera lietotājam):

```
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. Konfigurējiet Zabbix serveri.

Failā `zabbix_server.conf` pievienojiet:

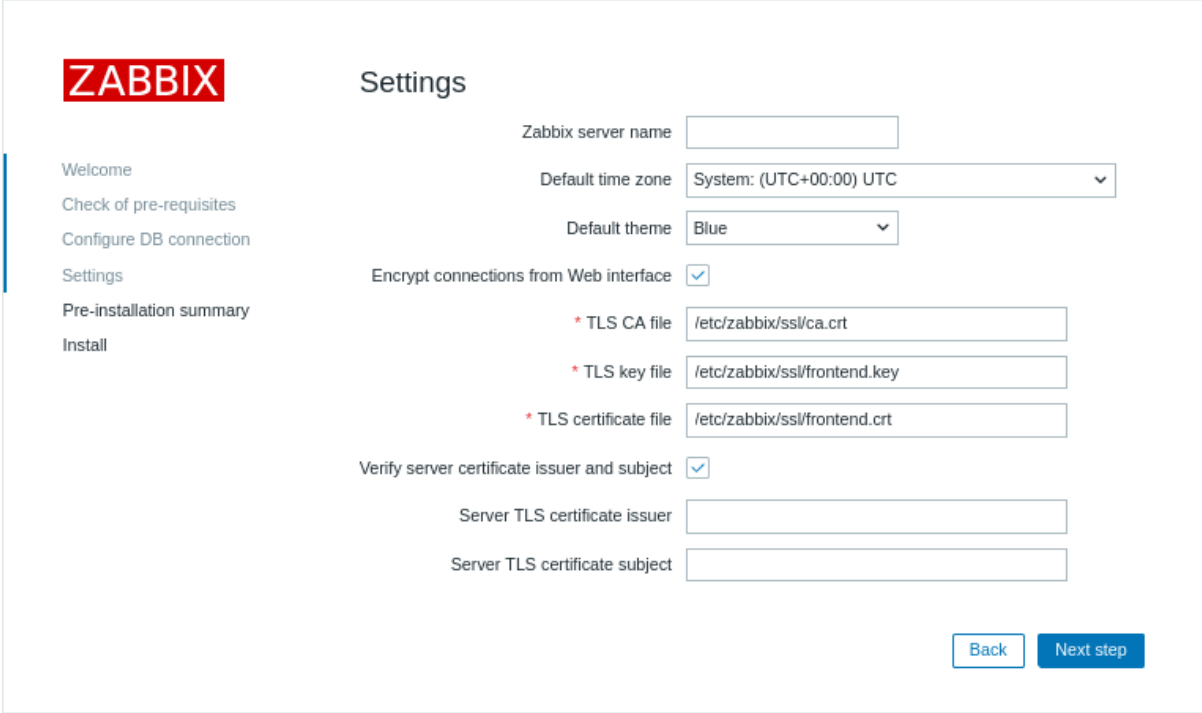
```
TLSFrontendAccept=cert
TLSCertFile=/etc/zabbix/ssl/server.crt
TLSKeyFile=/etc/zabbix/ssl/server.key
TLSCAFile=/etc/zabbix/ssl/ca.crt
#### Pēc izvēles:
#### TLSFrontendCertIssuer=/CN=MyZabbixCA/
#### TLSFrontendCertSubject=/CN=zabbix-frontend-node/
```

Pēc tam restartējiet serveri:

```
sudo systemctl restart zabbix-server
```

4. Konfigurējiet Zabbix lietotāja saskarni.

Web interfeisa instalēšanas laikā iespējojiet opciju *Encrypt connections from Web interface* (un opciju *Verify server certificate issuer and subject*, ja nepieciešams) un aizpildiet laukus *TLS CA file*, *TLS key file*, *TLS certificate file* (un laukus *Server TLS certificate issuer* un *Server TLS certificate subject*, ja nepieciešams):



Parametrs	Apraksts
TLS CA file	Norādiet pilnu ceļu uz sertifikācijas iestādes (CA) sertifikāta failu, ko izmanto servera sertifikāta pārbaudei.
TLS key file	Norādiet pilnu ceļu uz klienta privātās atslēgas failu, kas atbilst klienta sertifikātam.
TLS certificate file	Norādiet pilnu ceļu uz klienta sertifikāta failu, ja ir nepieciešama savstarpēja TLS autentifikācija.
Server TLS certificate issuer	Norādiet izdevēja atšķirto nosaukumu (DN), ar ko salīdzināt servera sertifikātu.
Server TLS certificate subject	Norādiet subjekta atšķirto nosaukumu (DN), ar ko salīdzināt servera sertifikātu.

Esošās instalācijās rediģējiet šādus laukus failā `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';
// Pēc izvēles:
// $ZBX_SERVER_TLS['CERTIFICATE_ISSUER']  = '/CN=MyZabbixCA/';
// $ZBX_SERVER_TLS['CERTIFICATE_SUBJECT'] = '/CN=zabbix-server.example.com/';
```

5. Pārbaudiet šifrēšanu, pārliecinoties, ka Zabbix lietotāja saskarne vai Zabbix servera žurnālfailā nav kļūdu ziņojumu:

```
tail -f /var/log/zabbix/zabbix_server.log
```

7 TimescaleDB iestatīšana

Pārskats

Zabbix atbalsta TimescaleDB — uz PostgreSQL balstītu datubāzes risinājumu, kas automātiski sadala datus laika intervālos balstītos gabalos, lai nodrošinātu ātrāku veiktspēju lielā mērogā.

> **Warning:**
> Pašlaik TimescaleDB netiek atbalstīts Zabbix starpniekserverī.

Šajā lapā sniegtās instrukcijas var izmantot šādos scenārijos:

- TimescaleDB datubāzes izveidei vai migrēšanai no esošajām PostgreSQL tabulām uz TimescaleDB (skatiet Konfigurācija).
- Esošās TimescaleDB datubāzes shēmas jaunināšanai, ja tiek jaunināts Zabbix (skatiet TimescaleDB shēmas jaunināšana).

Konfigurācija

Priekšnosacījumi: datubāzes serverī instalēts atbalstītas versijas TimescaleDB paplašinājums supported version. Instalēšanas norādījumus skatiet TimescaleDB documentation.

> **Warning:**
> Pirms TimescaleDB instalēšanas instalējiet atbalstītu PostgreSQL laidienu no official PostgreSQL repozitorija.

Iespējojiet TimescaleDB paplašinājumu konkrētajai datubāzei, izpildot:

```
echo "CREATE EXTENSION IF NOT EXISTS timescaledb CASCADE;" | sudo -u postgres psql zabbix
```

Šīs komandas izpildei ir nepieciešamas datubāzes administratora privilēģijas.

> **Note:**
> Ja izmantojat datubāzes shēmu, kas nav 'public', iepriekš norādītajai komandai jāpievieno SCHEMA klauzula. Piemēram:
 `echo "CREATE EXTENSION IF NOT EXISTS timescaledb SCHEMA yourschema CASCADE;" | sudo -u postgres psql zabbix`

Pēc tam palaidiet skriptu `postgresql/timescaledb/schema.sql`. Jaunām instalācijām skripts jāpalaiž pēc tam, kad parastā PostgreSQL datubāze ir izveidota ar sākotnējo shēmu/datiem (skatiet database creation).

```
cat /usr/share/zabbix/sql-scripts/postgresql/timescaledb/schema.sql | sudo -u zabbix psql zabbix
```

> **Attention:**
> Lūdzu, ignorējiet brīdinājuma ziņojumus, kas informē, ka, palaižot `schema.sql` skriptu ar TimescaleDB versiju 2.9.0 un jaunāku, netiek ievērota labākā prakse. Neskatoties uz šo brīdinājumu, konfigurēšana tiks veiksmīgi pabeigta.

Esošo vēstures, tendenču un audita žurnāla datu migrēšana var aizņemt daudz laika. Migrācijas laikā Zabbix serverim un lietotāja saskarnei jābūt apturētiem.

Skripts `schema.sql` iestata šādus housekeepinga parametrus:

- Pārrakstīt vienuma vēstures periodu
- Pārrakstīt vienuma tendenču periodu

Lai izmantotu sadalītu housekeeping vēsturei un tendencēm, abām šīm opcijām jābūt iespējotām. Ir iespējams arī iespējot pārrakstīšanu atsevišķi — tikai vēsturei vai tikai tendencēm.

PostgreSQL un TimescaleDB gadījumā skripts `postgresql/timescaledb/schema.sql` iestata vēl divus papildu parametrus:

- Iespējot saspiešanu
- Saspiest ierakstus, kas ir vecāki par 7 dienām

Lai housekeeper veiksmīgi noņemtu saspiestos datus, jābūt iespējotām abām opcijām — *Override item history period* un *Override item trend period*. Ja pārrakstīšana ir atspējota un tabulās ir saspiesti gabali, housekeeper nenoņems datus no šīm tabulām, un sadaļās *Housekeeping* un *System information* tiks parādīti brīdinājumi par nepareizu konfigurāciju.

Visus šos parametrus pēc instalēšanas var mainīt sadaļā *Administrēšana* > *Housekeeping*.

> **Note:**
> Iespējams, vēlēsieties palaist TimescaleDB nodrošināto rīku timescaledb-tune, lai optimizētu PostgreSQL konfigurācijas parametrus savā `postgresql.conf`.

TimescaleDB shēmas jaunināšana

Jauninot Zabbix uz versiju, kurā ir iekļautas jaunas TimescaleDB hipertabulas, Zabbix serveris šīs hipertabulas automātiski nekonfigurē (piemēram, jauninot no Zabbix 6.4 uz 8.0, jo versijās 7.0.0 un 7.0.2 ir ieviestas jaunas hipertabulas).

Lai konfigurētu jaunās TimescaleDB hipertabulas, izpildiet šādas darbības:

1. Palaidiet Zabbix serveri; tā tiks jaunināta esošā datubāze.
2. Pārbaudiet servera žurnālfailā, vai datubāzes jaunināšana ir pabeigta; kad tā ir pabeigta, apturiet Zabbix serveri. Ņemiet vērā, ka serveris reģistrē brīdinājumu, ja tas mēģina iespējot saspiešanu tabulai, kas nav hipertabula.
3. Palaidiet skriptu `postgresql/timescaledb/schema.sql`; tas konfigurē jaunās TimescaleDB hipertabulas. Ņemiet vērā, ka kopš Zabbix 7.0.0 skripta atrašanās vieta un nosaukums ir mainīti no `postgresql/timescaledb.sql` uz `postgresql/timescaledb/schema.sql`.

Attention:
Lūdzu, ignorējiet brīdinājuma ziņojumus, kas informē, ka, palaižot `schema.sql` skriptu TimescaleDB versijā 2.9.0 un jaunākās versijās, netiek ievērota labākā prakse. Neatkarīgi no šī brīdinājuma konfigurēšana tiks veiksmīgi pabeigta.

TimescaleDB saspiešana

Vietējā TimescaleDB saspiešana tiek atbalstīta visām Zabbix tabulām, kas ir TimescaleDB hipertabulas. Jaunināšanas vai migrācijas laikā uz TimescaleDB lielo tabulu sākotnējā saspiešana var aizņemt daudz laika.

Ņemiet vērā, ka saspiešana tiek atbalstīta ar "timescale" Timescale Community licenci, un tā netiek atbalstīta ar "apache" Apache 2.0 licenci. Ja Zabbix konstatē, ka saspiešana netiek atbalstīta, Zabbix servera žurnālā tiek ierakstīts brīdinājuma ziņojums, un lietotāji nevar iespējot saspiešanu lietotāja saskarnē.

Note:
Lietotāji tiek aicināti pirms saspiešanas izmantošanas iepazīties ar saspiešanu TimescaleDB dokumentācijā.

Ņemiet vērā, ka saspiešanai ir noteikti ierobežojumi, konkrēti:

- Saspiestu gabalu modificēšana (ievietošana, dzēšana, atjaunināšana) nav atļauta
- Shēmas izmaiņas saspiestām tabulām nav atļautas.

Saspiešanas iestatījumus var mainīt blokā *Vēstures, tendenču un audita žurnāla saspiešana* sadaļā *Administrēšana > Housekeeping* Zabbix lietotāja saskarnē.

Parameter	Default	Comments
Enable compression	Enabled	Atzīmējot vai noņemot atzīmi izvēles rūtiņā, saspiešana netiek aktivizēta/deaktivizēta nekavējoties. Tā kā saspiešanu apstrādā Housekeeper, izmaiņas stāsies spēkā ne vēlāk kā pēc 2 × `HousekeepingFrequency` stundām (iestatīts zabbix_server.conf) Pēc saspiešanas atspējošanas jauni gabali, kas ietilpst saspiešanas periodā, netiks saspiesti. Tomēr visi iepriekš saspiestie dati paliks saspiesti. Lai atspiestu iepriekš saspiestos gabalus, sekojiet norādījumiem TimescaleDB dokumentācijā.
Compress records older than	7d	Jauninot no vecākām Zabbix versijām ar TimescaleDB atbalstu, saspiešana pēc noklusējuma netiks iespējota. Šis parametrs nevar būt mazāks par 7 dienām. Saspiestu gabalu nemainīguma dēļ visi novēlotie dati (piemēram, starpniekservera aizkavēti dati), kas ir vecāki par šo vērtību, tiks atmesti.

Lai uzlabotu tendenču atjaunināšanas veiktspēju, iespējams, vēlēsieties samazināt "chunk_time_interval" tabulām `trends` un `trends_uint` no 30 dienām līdz 7 dienām vai mazāk atkarībā no tā, cik daudz vienumu izmanto tendences. Šī iestatījuma mērķis ir ievērot TimescaleDB labāko praksi un nodrošināt, ka gabalu izmērs nepārsniedz sistēmā pieejamos resursus.

8 Elasticsearch iestatīšana

Zabbix var glabāt vēstures datus Elasticsearch kā alternatīvu relāciju datubāzei.

Attention:
Elasticsearch atbalsts pašlaik ir eksperimentāls.

Šajā rokasgrāmatā ir aprakstīta Elasticsearch 7.X iestatīšana. Ja izmantojat citu versiju, daļa funkcionalitātes var nedarboties, kā paredzēts.

Iestatīšana ietver Elasticsearch glabātuves atrašanās vietas izveidi katram vērtības tipam, priekšapstrādes iestatīšanu (ja nepieciešams) un Zabbix savienošanu ar jūsu Elasticsearch instanci.

Elasticsearch var glabāt šādus vērtību tipus:

Vienuma vērtības tips	Datubāzes tabula	Elasticsearch tips
Skaitlisks (bez zīmes)	history_uint	uint
Skaitlisks (peldošā komata)	history	dbl
Rakstzīmes	history_str	str
Žurnāls	history_log	log
Teksts	history_text	text
Binārs	history_bin	Zabbix neatbalsta
JSON	history_json	json

Svarīgas piezīmes

- Elasticsearch nepieciešams libcurl. Sīkāku informāciju skatiet prasībās.
- Housekeeper nedzēš datus no Elasticsearch.
- Ja visi vēstures dati tiek glabāti Elasticsearch, tendences datubāzē **netiek** aprēķinātas vai saglabātas. Apsveriet iespēju pagarināt vēstures glabāšanas periodu.
- Ja tiek izmantots Elasticsearch, diapazona vaicājumus, kas izgūst vērtības no datubāzes, ierobežo datu glabāšanas perioda laikspiedols.
- Elasticsearch netiek atbalstīts Zabbix starpniekserverim; tā vietā, lūdzu, izmantojiet SQLite.

Ja Elasticsearch vēl nav instalēts, pirms turpināt skatiet oficiālo instalēšanas rokasgrāmatu.

Elasticsearch konfigurēšana

Lai glabātu vēstures datus Elasticsearch, jums ir nepieciešams:

- Izveidot indeksu katram vērtības tipam, ko vēlaties glabāt — šeit Elasticsearch glabā datus, līdzīgi kā tabulā relāciju datubāzē.
- Definēt kartējumu katram indeksam — tas nosaka datu struktūru, līdzīgi kā tabulas shēma.
- Iestatīt uzņemšanas konveijeru, lai apstrādātu vērtības pirms glabāšanas (nepieciešams JSON vērtībām un uz datumu balstītiem indeksiem).

Elasticsearch var glabāt datus vienā indeksā katram vērtības tipam vai vairākos uz datumu balstītos indeksos. Abas pieejas ir aprakstītas tālāk.

Vēstures glabāšana vienā indeksā

Šajā pieejā visi vēstures dati konkrētam vērtības tipam tiek ierakstīti vienā indeksā (piemēram, `uint` vai `text`).

Lai izveidotu indeksu vērtības tipam *Numeric (unsigned)*, nosūtiet šādu pieprasījumu (ar `/uint` URL adresē) uz savu Elasticsearch instanci:

```
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 atbildēs ar apstiprinājumu, ka indekss ir izveidots:

```
{"acknowledged": true, "shards_acknowledged": true, "index": "uint"}
```

Līdzīgi pieprasījumi jānosūta katram papildu vērtības tipam, kuru vēlaties glabāt Elasticsearch.

> **Note:**
> Kartējumi visiem vērtību tipiem ir pieejami Zabbix avota repozitorijā.

Piemēram, lai izveidotu indeksu vērtības tipam *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"
         }
      }
   }
}'
```

JSON vērtības tips

Atšķirībā no citiem vērtību tipiem JSON vērtībām pirms saglabāšanas ir nepieciešama papildu apstrāde.

Tālāk norādītais indekss izmanto atsevišķus laukus parsētām un neapstrādātām vērtībām, tāpēc ir nepieciešams ingest pipeline, lai parsētu katru vērtību kā JSON un saglabātu to pareizajā laukā.

Lai izveidotu indeksu *JSON* vērtības tipam, nosūtiet šādu pieprasījumu (ar `/json` URL adresē) uz savu Elasticsearch instanci.

```
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 }
    }
  }
}'
```

Pēc tam izveidojiet ingest pipeline:

```
curl -X PUT \
 http://localhost:9200/_ingest/pipeline/json \
 -H 'content-type:application/json' \
 -d '{
   "processors": [
      {
         "json": {
            "field": "value",
            "target_field": "value_parsed",
```

```
            "ignore_failure": true
         }
      },
      {
         "set": {
            "if": "ctx.value_parsed == null",
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ],
   "on_failure": [
      {
         "set": {
            "field": "value_raw",
            "value": "{{{ value }}}"
         }
      }
   ]
}'
```

Elasticsearch atbildēs ar apstiprinājumu, ka ingest pipeline tika izveidots:

```
{"acknowledged": true}
```

Vēstures glabāšana datumos balstītos indeksos

Tā vietā, lai visus vēstures datus rakstītu vienā indeksā (piemēram, `uint`), Elasticsearch var sadalīt šos datus vairākos datumos balstītos indeksos (piemēram, `uint-2026-01-01`, `uint-2026-01-02`). Tas atvieglo datu apjoma un glabāšanas ilguma pārvaldību laika gaitā.

Lai to iespējotu, jums ir jāveic tālāk norādītais:

- Izveidojiet indeksa veidni katram vērtību tipam, ko vēlaties glabāt — tā norāda Elasticsearch, kādu kartējumu piemērot, kad tas automātiski izveido jaunu datumā balstītu indeksu.
- Izveidojiet uzņemšanas konveijeru katram vērtību tipam — tas apstrādā katru ienākošo vērtību un novirza to uz pareizo datumā balstīto indeksu.
- Konfigurējiet parametru `HistoryStorageDateIndex` Zabbix servera konfigurācijas failā — tas iespējo vērtību glabāšanu vairākos datumos balstītos indeksos.

Indeksu veidnes

Lai izveidotu veidni `text` indeksam, nosūtiet pieprasījumu ar šādu informāciju:

- Izmantojiet `_template/text_template` sava Elasticsearch instances URL.
- Izmantojiet `"text*"` laukā `"index_patterns"`, lai atbilstu indeksa nosaukumam.
- Izmantojiet kartējumu `text` vērtības tipam (skatiet kartējumus Zabbix avota repozitorijā).

```
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"
         }
      }
   }
}'
```

Veidne `json` indeksam:

```
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 }
      }
   }
}'
```

Uzņemšanas konveijeri

Lai izveidotu uzņemšanas konveijeru `text` indeksam:

- Izmantojiet `_ingest/pipeline/text-pipeline` savas Elasticsearch instances URL.
- Iekļaujiet `date_index_name` procesoru, lai novirzītu katru vērtību uz pareizo uz datumu balstīto indeksu, pamatojoties uz tās laikspiedolu.

```
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"
         }
      }
   ]
}'
```

`json` indeksam konveijeram pirms novirzīšanas uz pareizo indeksu ir arī jāparsē JSON vērtība:

```
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"
         }
      }
   ]
}'
```

Zabbix servera konfigurēšana

Jūsu Zabbix servera konfigurācijas failā (`zabbix_server.conf`) iestatiet šādus parametrus:

- `HistoryStorageURL` - jūsu Elasticsearch instances URL.
- `HistoryStorageTypes` - ar komatiem atdalīts vērtību tipu saraksts, ko glabāt Elasticsearch.

Piemēram, lai Elasticsearch glabātu *Character*, *Log*, *Text* un *JSON* tipa vērtības (vienlaikus saglabājot *Numeric* vērtības datubāzē):

```
HistoryStorageURL=http://localhost:9200
HistoryStorageTypes=str,log,text,json
```

Ja izmantojat uz datumu balstītus indeksus **visām** Elasticsearch glabātajām vērtībām, iestatiet arī parametru `HistoryStorageDateIndex`:

```
HistoryStorageDateIndex=1
```

Pēc izmaiņu veikšanas restartējiet Zabbix serveri:

```
systemctl restart zabbix-server
```

Zabbix lietotāja saskarnes konfigurēšana

Zabbix lietotāja saskarnes konfigurācijas failā (`zabbix.conf.php`) deklarējiet `$HISTORY` kā globālu mainīgo un iestatiet tā `url` un `types` vērtības atbilstoši servera konfigurācijai:

```
// Zabbix GUI configuration file.
global $DB, $HISTORY;

$HISTORY['url']   = 'http://localhost:9200';
$HISTORY['types'] = ['str', 'log', 'text', 'json'];
```

Problēmu novēršana

Tālāk norādītās darbības var palīdzēt novērst problēmas ar jūsu Elasticsearch iestatījumu:

1. Pārbaudiet, vai `auto_create_index` ir iespējots:

```
curl -X GET \
 "http://localhost:9200/_cluster/settings?include_defaults=true&filter_path=**.auto_create_index"

#### {"defaults": {"action": {"auto_create_index": "false"} } }
```

Lai to iespējotu, nosūtiet šādu pieprasījumu:

```
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. Pārbaudiet, vai kartējumi, veidnes un ingest cauruļvadi ir pareizi, nosūtot GET pieprasījumus uz to attiecīgajiem 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*
```

Saņemtās atbildes varat salīdzināt ar sagaidāmajām atbildēm Elasticsearch API dokumentācijā.

3. Pārbaudiet, vai kādas šķembas nav kļūmes stāvoklī; Elasticsearch restartēšana var palīdzēt to novērst.
4. Pārbaudiet, vai jūsu Elasticsearch konfigurācija atļauj piekļuvi no Zabbix serveris un Zabbix lietotāja saskarne.
5. Izmantojiet Zabbix serveris konfigurācijas parametru `LogSlowQueries`, lai noteiktu lēnus vaicājumus.
6. Pārbaudiet Elasticsearch žurnālus, vai tajos nav kļūdu.
7. Ja nepieciešams atiestatīt Elasticsearch iestatījumu un sākt no jauna, varat dzēst visus indeksus, veidnes un ingest cauruļvadus:

```
curl -X DELETE "http://localhost:9200/_all"
curl -X DELETE "http://localhost:9200/_template/*"
curl -X DELETE "http://localhost:9200/_ingest/pipeline/*"
```

9 Izplatīšanai specifiskas piezīmes par Nginx iestatīšanu Zabbix vajadzībām

RHEL

RHEL 8, 9 un 10 versijās Nginx ir iekļauts AppStream — nav nepieciešamas papildu repozitorijas. Vienkārši iespējojiet un instalējiet to ar:

```
dnf module list nginx
dnf -y install nginx
```

RHEL 7 versijā Nginx var instalēt no CentOS 7 Extras repozitorijas (ja izmantojat CentOS), izmantojot:

```
yum -y install nginx
```

Ja vēlaties pašas jaunākās augšupplūsmas būves, konfigurējiet oficiālo NGINX repozitoriju failā `/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

SUSE Linux Enterprise Server 15 sistēmā jums jākonfigurē `php-fpm` (ceļš uz konfigurācijas failu var nedaudz atšķirties atkarībā no servisa pakotnes):

```
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 Aģenta palaišana kā root

Kopš Zabbix **5.0.0** versijas systemd servisa fails Zabbix aģentam oficiālajās pakotnēs tieši ietver `User` un `Group` direktīvas. Abām ir iestatīta vērtība `zabbix`.

Vairs nav iespējams konfigurēt, ar kuru lietotāju Zabbix aģents tiek palaists, izmantojot `zabbix_agentd.conf` failu, jo aģents ignorēs šo konfigurāciju un tiks palaists ar lietotāju, kas norādīts systemd servisa failā. Lai palaistu Zabbix aģentu kā root, ir jāveic tālāk aprakstītās izmaiņas.

Zabbix aģents

Lai ignorētu noklusējuma lietotāju un grupu Zabbix aģentam, izpildiet:

```
systemctl edit zabbix-agent
```

Pēc tam pievienojiet šādu saturu:

```
[Service]
User=root
Group=root
```

Pārlādējiet dēmonus un restartējiet zabbix-agent servisu:

```
systemctl daemon-reload
systemctl restart zabbix-agent
```

Zabbix aģentam tas atkārtoti iespējo lietotāja konfigurēšanas funkcionalitāti failā `zabbix_agentd.conf`. Tagad aģenta konfigurācijas failā ir jāiestata konfigurācijas parametri `User=root` un `AllowRoot=1`.

Zabbix aģents 2

Lai pārrakstītu noklusējuma lietotāju un grupu Zabbix aģentam 2, izpildiet:

```
systemctl edit zabbix-agent2
```

Pēc tam pievienojiet šādu saturu:

```
[Service]
User=root
Group=root
```

Pārlādējiet dēmonus un restartējiet zabbix-agent2 servisu:

```
systemctl daemon-reload
systemctl restart zabbix-agent2
```

Zabbix agent2 gadījumā tas pilnībā nosaka lietotāju, ar kuru tas tiek darbināts. Papildu izmaiņas nav nepieciešamas.

11 Zabbix aģents Microsoft Windows vidē

Aģenta konfigurēšana

Abas Zabbix aģentu paaudzes darbojas kā Windows pakalpojums. Zabbix aģentam 2 tālāk sniegtajās instrukcijās aizstājiet *agentd* ar *agent2*.

Microsoft Windows hostā varat darbināt vienu Zabbix aģenta instanci vai vairākas aģenta instances. Viena instance var izmantot vienu no šīm iespējām:

- noklusējuma konfigurācijas failu, kas atrodas tajā pašā direktorijā, kur aģenta binārais fails;
- komandrindā norādītu konfigurācijas failu.

Vairāku instanču gadījumā katrai aģenta instancei jābūt savam konfigurācijas failam (viena no instancēm var izmantot noklusējuma konfigurācijas failu).

Konfigurācijas faila piemērs ir pieejams Zabbix pirmkoda arhīvā:

- `conf/zabbix_agentd.conf` Zabbix aģentam;
- `conf/zabbix_agent2.conf` Zabbix aģentam 2.

Ja vēlaties instalēt Zabbix aģentu/aģentu 2 operētājsistēmai Windows kā pakalpojumu no arhīva, skaidri nenorādot konfigurācijas failu, tad pirms aģenta instalēšanas:

- `conf/zabbix_agentd.conf` manuāli jākopē direktorijā, kurā tiks instalēts zabbix_agentd.exe;
- `conf/zabbix_agent2.conf` un direktorija `conf/zabbix_agent2.d` manuāli jākopē direktorijā, kurā tiks instalēts zabbix_agent2.exe.

Plašāku informāciju par Zabbix Windows aģenta konfigurēšanu skatiet konfigurācijas faila opcijās.

Hostname parametrs

Lai veiktu aktīvās pārbaudes hostā, Zabbix aģentam jābūt definētam hosta nosaukumam. Turklāt aģenta pusē iestatītajai hosta nosaukuma vērtībai precīzi jāsakrīt ar hostam lietotāja saskarnē konfigurēto "Host name".

Hosta nosaukuma vērtību aģenta pusē var definēt ar parametru **Hostname** vai **HostnameItem** aģenta konfigurācijas failā — vai arī tiek izmantotas noklusējuma vērtības, ja kāds no šiem parametriem nav norādīts.

Parametra **HostnameItem** noklusējuma vērtība ir vērtība, ko atgriež aģenta atslēga "system.hostname". Operētājsistēmā Windows tā atgriež gethostname() funkcijas rezultātu, kas vaicā nosaukumvietu nodrošinātājiem, lai noteiktu lokālā hosta nosaukumu. Ja neviens nosaukumvietu nodrošinātājs neatbild, tiek atgriezts NetBIOS nosaukums.

Hostname noklusējuma vērtība ir vērtība, ko atgriež parametrs HostnameItem. Tādējādi, ja abi šie parametri nav norādīti, faktiskais hosta nosaukums būs hosta NetBIOS nosaukums; Zabbix aģents izmantos NetBIOS hosta nosaukumu, lai saņemtu aktīvo pārbaužu sarakstu no Zabbix servera un nosūtītu tam rezultātus.

Atslēga "system.hostname" atbalsta divus neobligātus parametrus — *type* un *transform*.

Type nosaka nosaukuma tipu, ko vienumam jāatgriež:

- *netbios* (noklusējums) — atgriež NetBIOS hosta nosaukumu, kas ir ierobežots līdz 15 simboliem un ir tikai LIELAJOS BURTOS;
- *host* — reģistrjutīgs, atgriež pilnu, reālo Windows hosta nosaukumu (bez domēna);
- *shorthost* — atgriež hosta nosaukuma daļu pirms pirmā punkta. Tiks atgriezta pilna virkne, ja nosaukums nesatur punktu.
- *fqdn* — atgriež pilno kvalificēto domēna nosaukumu (bez beigu punkta).

Transform ļauj norādīt papildu hosta nosaukuma pārveidošanas noteikumu:

- *none* (noklusējums) — izmantot oriģinālo burtu reģistru;
- *lower* — pārveidot tekstu uz mazajiem burtiem.

Tātad, lai vienkāršotu zabbix_agentd.conf faila konfigurāciju un padarītu to vienotu, var izmantot trīs dažādas pieejas:

1. Atstāt parametrus **Hostname** vai **HostnameItem** nedefinētus, un Zabbix aģents izmantos NetBIOS hosta nosaukumu kā hosta nosaukumu.
2. Atstāt parametru **Hostname** nedefinētu un definēt **HostnameItem** šādi:
 HostnameItem=system.hostname[host] — lai Zabbix aģents izmantotu pilnu, reālo (reģistrjutīgo) Windows hosta nosaukumu kā hosta nosaukumu.
 HostnameItem=system.hostname[shorthost,lower] — lai Zabbix aģents izmantotu tikai hosta nosaukuma daļu pirms pirmā punkta, pārveidotu uz mazajiem burtiem.
 HostnameItem=system.hostname[fqdn] — lai Zabbix aģents izmantotu pilno kvalificēto domēna nosaukumu kā hosta nosaukumu.

Hosta nosaukums tiek izmantots arī kā daļa no Windows servisa nosaukuma, ko izmanto Windows servisa instalēšanai, palaišanai, apturēšanai un atinstalēšanai. Piemēram, ja Zabbix aģenta konfigurācijas failā ir norādīts `Hostname=Windows_db_server`, tad aģents tiks instalēts kā Windows serviss "`Zabbix Agent [Windows_db_server]`". Tāpēc, lai katrai Zabbix aģenta instancei būtu atšķirīgs Windows servisa nosaukums, katrai instancei jāizmanto atšķirīgs hosta nosaukums.

Aģenta instalēšana kā Windows pakalpojumu

Pirms aģenta instalēšanas manuāli nokopējiet `conf/zabbix_agentd.conf` uz direktoriju, kurā tiks instalēts zabbix_agentd.exe.

Lai instalētu vienu Zabbix aģenta instanci ar noklusējuma konfigurācijas failu:

```
zabbix_agentd.exe --install
```

> **Attention:**
> 64 bitu sistēmā ir nepieciešama 64 bitu Zabbix aģenta versija, lai visas pārbaudes, kas saistītas ar palaistiem 64 bitu procesiem, darbotos pareizi.

Ja vēlaties izmantot konfigurācijas failu, kas nav noklusējuma fails, pakalpojuma instalēšanai jāizmanto šāda komanda:

```
zabbix_agentd.exe --config <your_configuration_file> --install
```

Jānorāda pilns ceļš uz konfigurācijas failu.

Vairākas Zabbix aģenta instances var instalēt kā pakalpojumus šādi:

```
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
```

Instalētajam pakalpojumam tagad jābūt redzamam vadības panelī.

Aģenta palaišana

Lai palaistu aģenta servisu, varat izmantot vadības paneli vai izdarīt to no komandrindas.

Lai palaistu vienu Zabbix aģenta instanci ar noklusējuma konfigurācijas failu:

```
zabbix_agentd.exe --start
```

Lai palaistu vienu Zabbix aģenta instanci ar citu konfigurācijas failu:

```
zabbix_agentd.exe --config <your_configuration_file> --start
```

Lai palaistu vienu no vairākām Zabbix aģenta instancēm:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --start --multiple-agents
```

Aģenta apturēšana

Lai apturētu aģenta servisu, varat izmantot Vadības paneli vai izdarīt to no komandrindas.

Lai apturētu vienu Zabbix aģenta instanci, kas palaista ar noklusējuma konfigurācijas failu:

```
zabbix_agentd.exe --stop
```

Lai apturētu vienu Zabbix aģenta instanci, kas palaista ar citu konfigurācijas failu:

```
zabbix_agentd.exe --config <your_configuration_file> --stop
```

Lai apturētu vienu no vairākām Zabbix aģenta instancēm:

```
zabbix_agentd.exe --config <configuration_file_for_this_instance> --stop --multiple-agents
```

Zabbix aģenta Windows servisa atinstalēšana

Lai atinstalētu vienu Zabbix aģenta instanci, izmantojot noklusējuma konfigurācijas failu:

```
zabbix_agentd.exe --uninstall
```

Lai atinstalētu vienu Zabbix aģenta instanci, izmantojot nenoklusējuma konfigurācijas failu:

```
zabbix_agentd.exe --config <your_configuration_file> --uninstall
```

Lai atinstalētu vairākas Zabbix aģenta instances no Windows servisiem:

```
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
```

Ierobežojumi

Zabbix aģents operētājsistēmai Windows neatbalsta nestandarta Windows konfigurācijas, kurās CPU ir sadalīti nevienmērīgi starp NUMA mezgliem. Ja loģiskie CPU ir sadalīti nevienmērīgi, tad CPU veiktspējas metrikas dažiem CPU var nebūt pieejamas. Piemēram, ja ir 72 loģiskie CPU un 2 NUMA mezgli, abos mezglos jābūt pa 36 CPU katrā.

12 SAML iestatīšana ar Microsoft Entra ID

Pārskats

Šajā sadaļā ir sniegtas vadlīnijas vienotās pierakstīšanās un lietotāju nodrošināšanas konfigurēšanai Zabbix no Microsoft Entra ID (iepriekš Microsoft Azure Active Directory), izmantojot SAML 2.0 autentifikāciju.

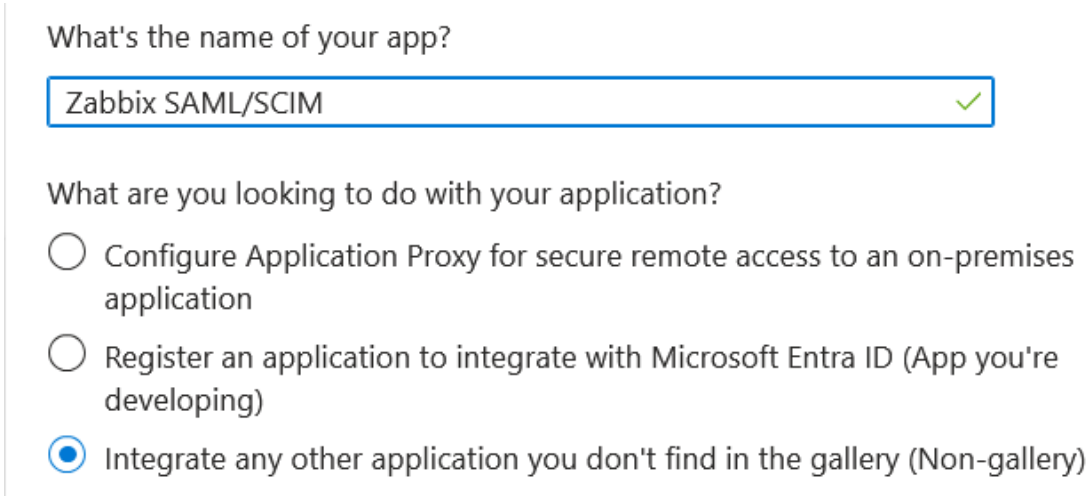
Microsoft Entra ID konfigurācija

Lietojumprogrammas izveide

1. Piesakieties Microsoft Entra administrēšanas centrā vietnē Microsoft Entra ID. Testēšanas nolūkos varat izveidot bezmaksas izmēģinājuma kontu Microsoft Entra ID.

2. Microsoft Entra administrēšanas centrā atlasiet *Applications -> Enterprise applications -> New application -> Create your own application*.

3. Pievienojiet savas lietojumprogrammas nosaukumu un atlasiet opciju *Integrate any other application...*. Pēc tam noklikšķiniet uz *Create*.

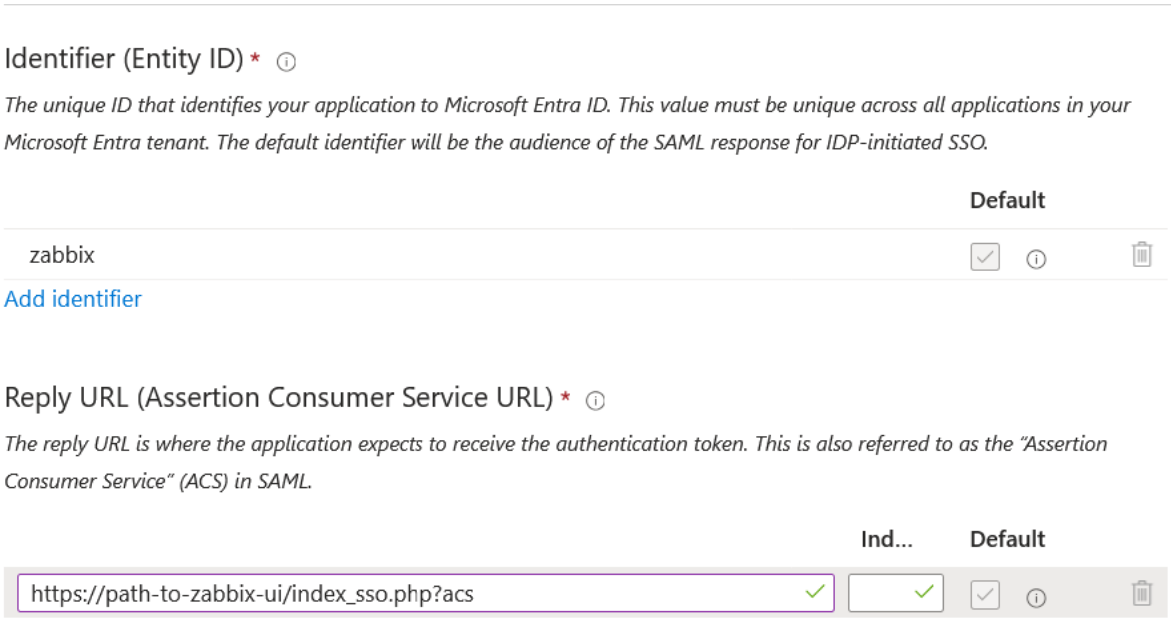


Vienotās pierakstīšanās iestatīšana

1. Lietojumprogrammas lapā atveriet *Set up single sign on* un noklikšķiniet uz *Get started*. Pēc tam atlasiet *SAML*.

2. Rediģējiet *Basic SAML Configuration*:

- Laukā *Identifier (Entity ID)* iestatiet unikālu nosaukumu, lai identificētu savu lietotni Microsoft Entra ID, piemēram, `zabbix`;
- Laukā *Reply URL (Assertion Consumer Service URL)* iestatiet Zabbix vienotās pierakstīšanās galapunktu: `https://<path-to-zabbix`



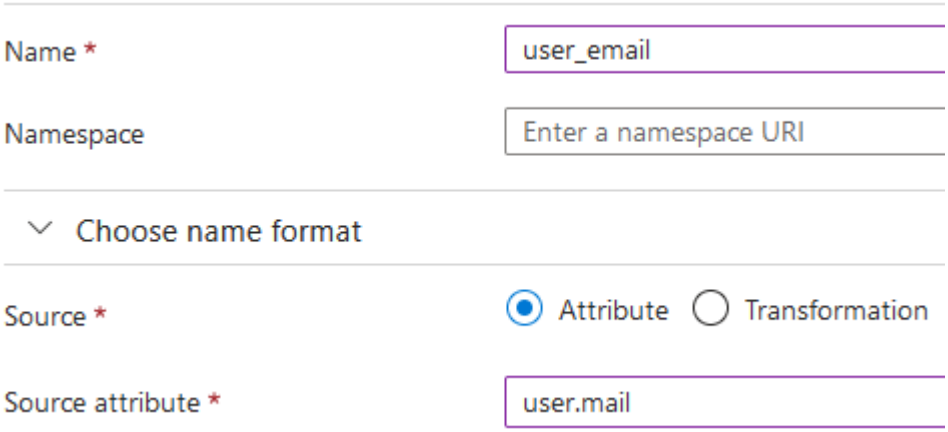
Ņemiet vērā, ka ir nepieciešams "https". Lai tas darbotos ar Zabbix, failam `conf/zabbix.conf.php` ir jāpievieno šāda rinda:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

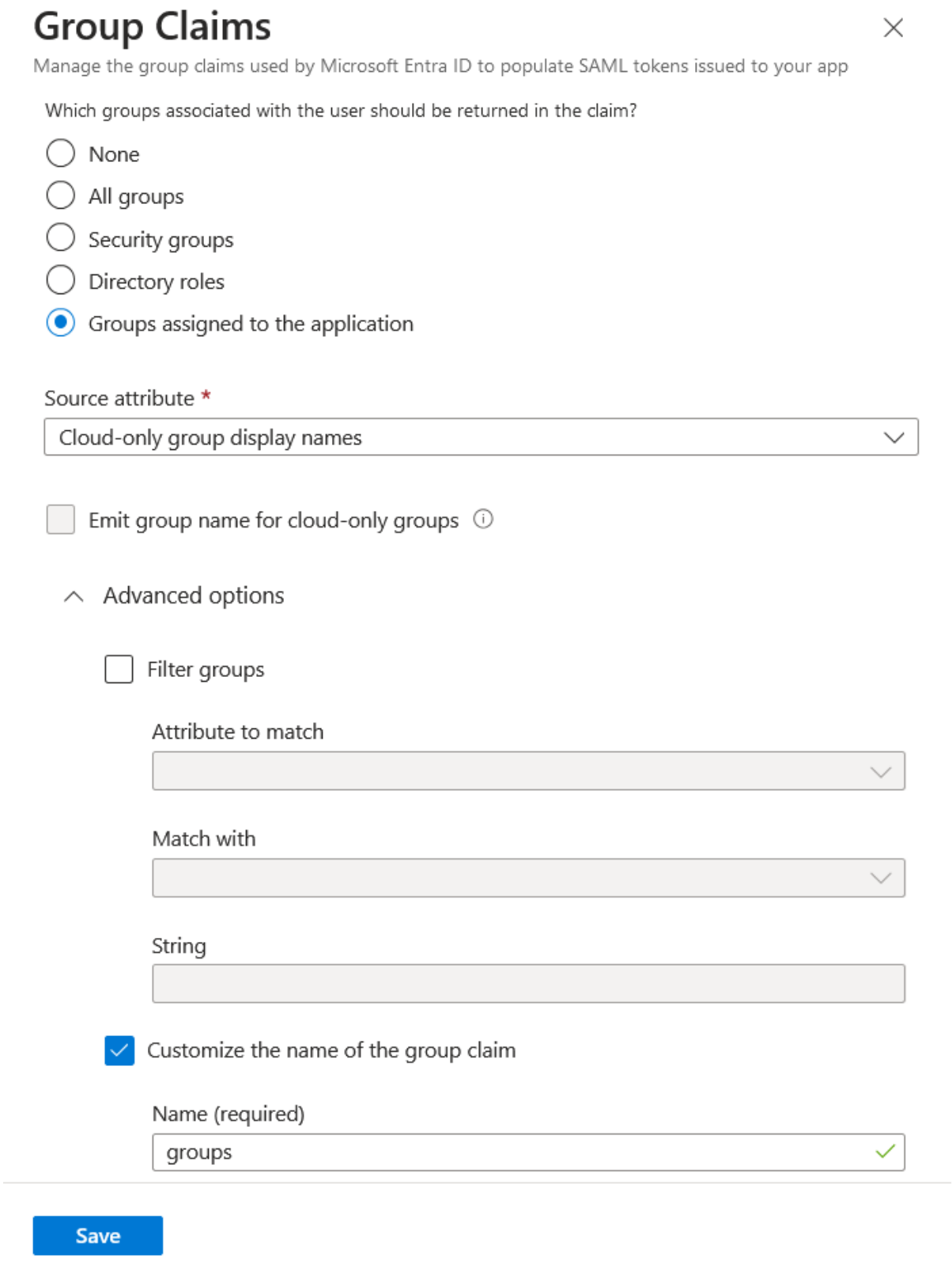
3. Rediģējiet *Attributes & Claims*. Jums jāpievieno visi atribūti, kurus vēlaties nodot Zabbix (user_name, user_lastname, user_email, user_mobile, groups).

Atribūtu nosaukumi ir patvaļīgi. Var izmantot dažādus atribūtu nosaukumus, tomēr ir nepieciešams, lai tie atbilstu attiecīgā lauka vērtībai Zabbix SAML iestatījumos.

- Noklikšķiniet uz *Add new claim*, lai pievienotu atribūtu:



- Noklikšķiniet uz *Add a group claim*, lai pievienotu atribūtu grupu nodošanai Zabbix:



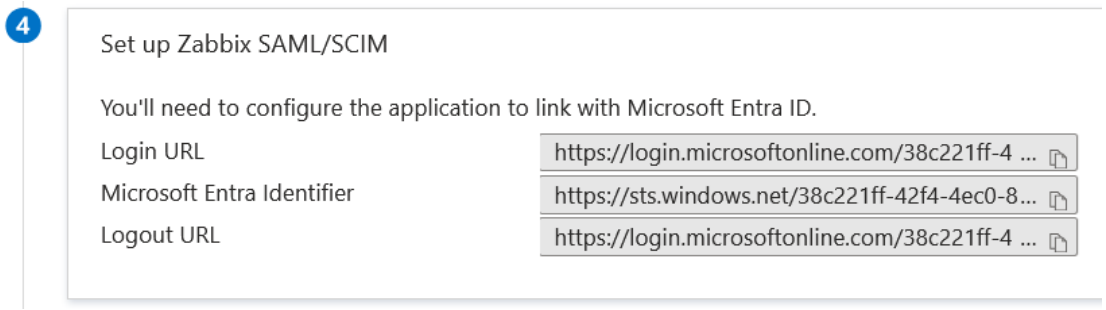
Šajā prasībā ir svarīgi, lai uz Zabbix tiktu nodoti grupu nosaukumi (nevis grupu ID), izmantojot atlasīto *Source attribute*. Pretējā gadījumā JIT lietotāju nodrošināšana nedarbosies pareizi.

4. Sadaļā *SAML Certificates* lejupielādējiet Entra ID nodrošināto Base64 sertifikātu un ievietojiet to Zabbix lietotāja saskarne instalācijas mapē `conf/certs`.

Iestatiet tam 644 atļaujas, izpildot:

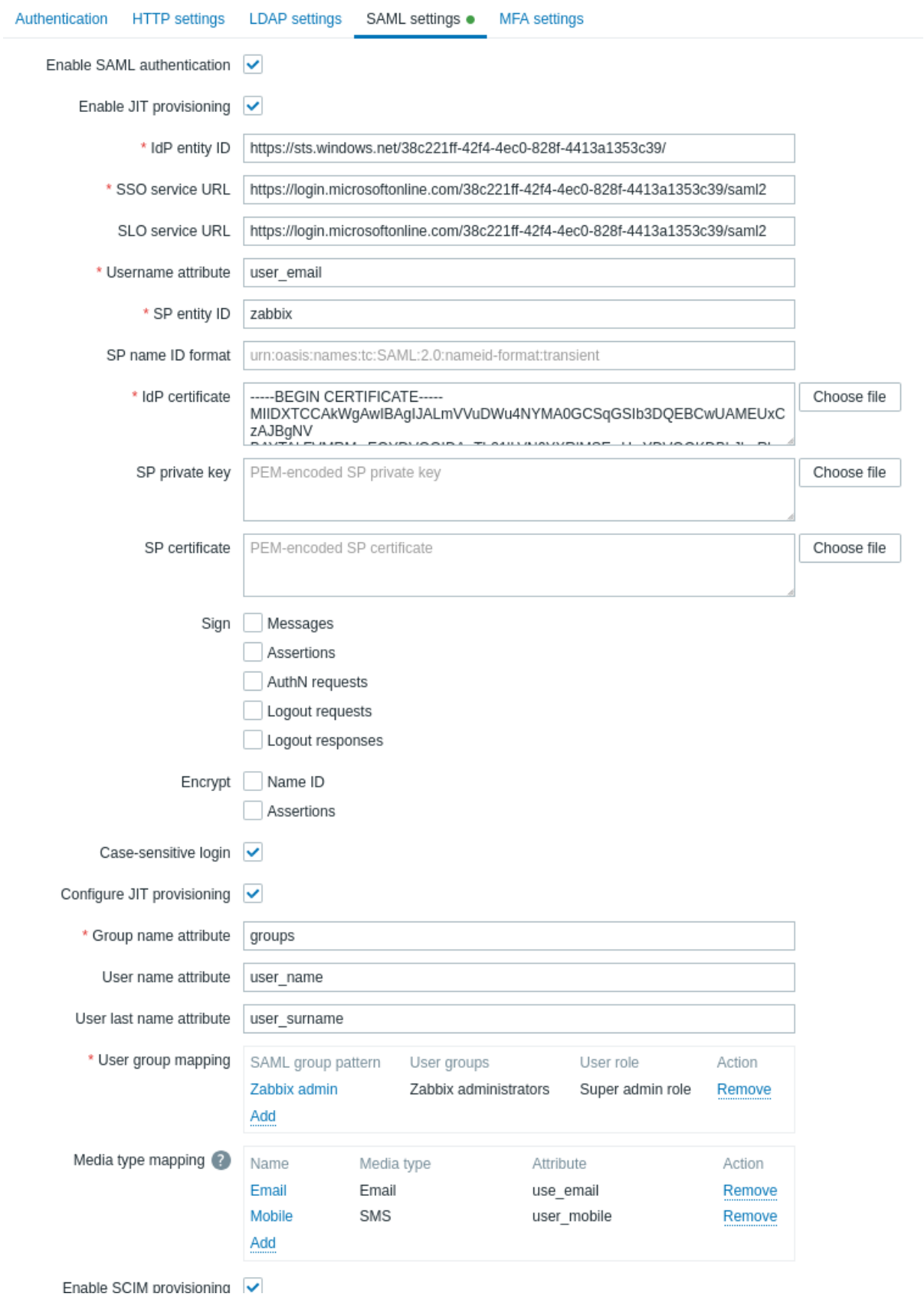
```
chmod 644 entra.cer
```

5. Izmantojiet vērtības no *Set up <your app name>* sadaļas Entra ID, lai konfigurētu Zabbix SAML autentifikāciju (skatiet nākamo sadaļu):



Zabbix konfigurācija

1. Zabbix vidē atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas atbilstoši Entra ID konfigurācijai:



Zabbix lauks	Iestatīšanas lauks Entra ID	Piemēra vērtība
IdP entītijas ID	Microsoft Entra identifikators	
SSO pakalpojuma URL	Pieteikšanās URL	
SLO pakalpojuma URL	Atslēgšanās URL	
SP entītijas ID	Identifikators (Entity ID)	
Lietotājvārda atribūts	Pielāgots atribūts (claim)	`user_email`
Grupas nosaukuma atribūts	Pielāgots atribūts (claim)	`groups`
Lietotāja vārda atribūts	Pielāgots atribūts (claim)	`user_name`

Zabbix lauks	Iestatīšanas lauks Entra ID	Piemēra vērtība
Lietotāja uzvārda atribūts	Pielāgots atribūts (claim)	`user_lastname`

2. Pievienojiet Entra ID nodrošināto Base64 sertifikātu.

Ja zabbix.conf.php failā ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā lietotāja saskarnē varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu — faili failu sistēmā nav nepieciešami.

Ja zabbix.conf.php failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē ui/conf/certs vai ceļā, kas konfigurēts zabbix.conf.php), un lietotāja saskarne nesaglabās sertifikātus datubāzē. Ņemiet vērā, ka, ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir aizkomentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no ui/conf/certs.

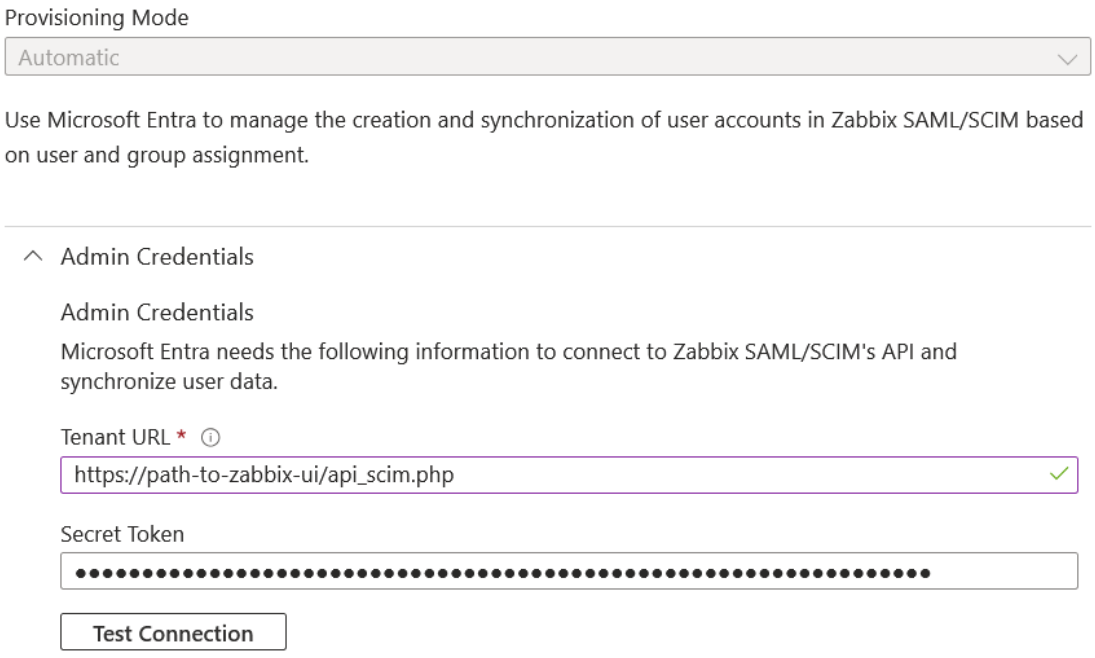
Nepieciešams arī konfigurēt lietotāju grupu kartēšanu. Mediju kartēšana nav obligāta.

3. Nospiediet pogu *Update*, lai saglabātu šos iestatījumus.

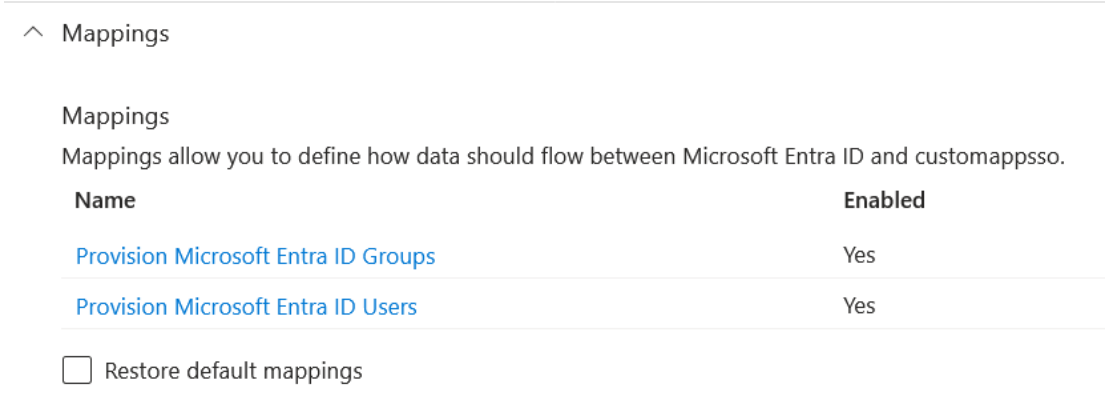
SCIM lietotāju nodrošināšana

1. Entra ID lietotnes lapā galvenajā izvēlnē atveriet lapu *Provisioning*. Noklikšķiniet uz *Get started* un pēc tam atlasiet automātiskās nodrošināšanas režīmu:

- Laukā *Tenant URL* iestatiet šādu vērtību: `https://<path-to-zabbix-ui>/api_scim.php`
- Laukā *Secret token* ievadiet Zabbix API tokenu ar Super admin atļaujām.
- Noklikšķiniet uz *Test connection*, lai pārbaudītu, vai savienojums ir izveidots.

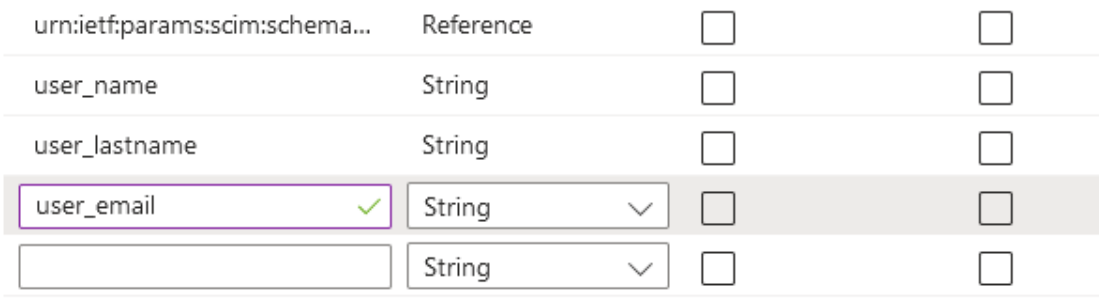


2. Tagad varat pievienot visus atribūtus, kas tiks nodoti no SCIM uz Zabbix. Lai to izdarītu, noklikšķiniet uz *Mappings* un pēc tam uz *Provision Microsoft Entra ID Users*.



Atribūtu kartēšanas saraksta apakšdaļā iespējojiet *Show advanced options* un pēc tam noklikšķiniet uz *Edit attribute list for customappsso*.

Atribūtu saraksta apakšdaļā pievienojiet savus atribūtus ar tipu 'String':



Saglabājiet sarakstu.

3. Tagad varat pievienot kartējumus pievienotajiem atribūtiem. Atribūtu kartēšanas saraksta apakšdaļā noklikšķiniet uz *Add New Mapping* un izveidojiet kartējumus, kā parādīts zemāk:

Kad visi kartējumi ir pievienoti, saglabājiet kartējumu sarakstu.

4. Kā priekšnosacījums lietotāju nodrošināšanai Zabbix, Entra ID ir jābūt konfigurētiem lietotājiem un grupām.

Lai to izdarītu, atveriet *Microsoft Entra admin center* un pēc tam attiecīgajās lapās Users un Groups pievienojiet lietotājus/grupas.

5. Kad lietotāji un grupas ir izveidoti Entra ID, varat atvērt savas lietotnes izvēlni *Users and groups* un pievienot tos lietotnei.

6. Atveriet savas lietotnes izvēlni *Provisioning* un noklikšķiniet uz *Start provisioning*, lai nodrošinātu lietotājus Zabbix.

Ņemiet vērā, ka Entra ID lietotāju PATCH pieprasījums neatbalsta izmaiņas medijos.

Autentifikācijas pieprasījuma parakstīšana

Ir iespējams konfigurēt Entra ID, lai validētu parakstu parakstītiem autentifikācijas pieprasījumiem.

Lai tas darbotos, izveidojiet publisko/privāto atslēgu pāri:

```
openssl req -x509 -newkey rsa:4096 -keyout /usr/share/zabbix/conf/certs/request-sign.key -out /usr/share/z
```

Piešķiriet atļaujas:

```
chown apache /usr/share/zabbix/conf/certs/request-sign.key
chmod 400 /usr/share/zabbix/conf/certs/request-sign.key
```

Atjauniniet Zabbix lietotāja saskarne konfigurāciju, pievienojot:

```
$SSO['SP_KEY'] = 'conf/certs/request-sign.key';
$SSO['SP_CERT'] = 'conf/certs/request-sign.crt';
```

Problēmu novēršana

Autentifikācijas problēmas var rasties ar MicroSoft Edge pārlūkprogrammām, ja lietotājs, mēģinot pieteikties Zabbix, izmantojot SAML, jau ir pieteicies ar MicroSoft Edge profilu. Kā šādas problēmas pazīme lietotājs var spēt pieteikties Zabbix, izmantojot MicroSoft Edge privātajā režīmā.

Lai šajā gadījumā izvairītos no autentifikācijas problēmām, var būt nepieciešams iestatīt `requestedAuthnContext` uz "false" Zabbix lietotāja saskarnes konfigurācijas failā (*zabbix.conf.php*).

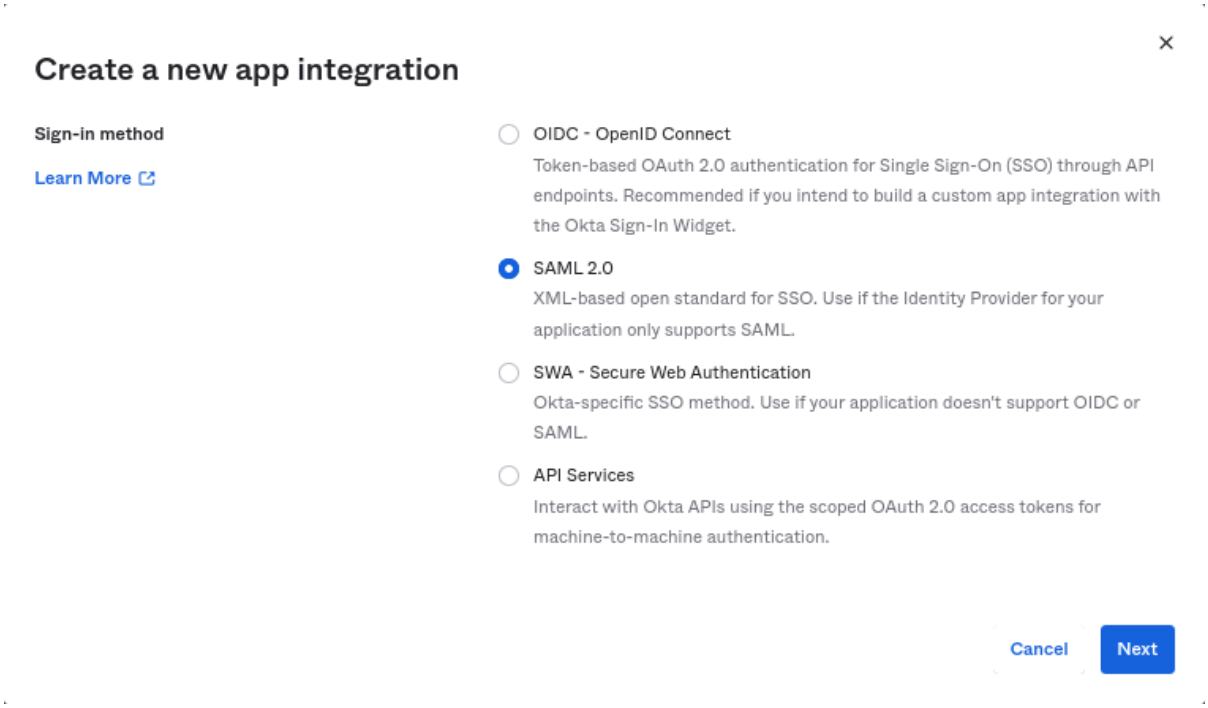
```
$SSO['SETTINGS'] = [
    'security' => [
        'requestedAuthnContext' => false
    ]
];
```

13 SAML iestatīšana ar Okta

Šajā sadaļā ir sniegtas vadlīnijas Okta konfigurēšanai, lai iespējotu SAML 2.0 autentifikāciju un lietotāju nodrošināšanu Zabbix.

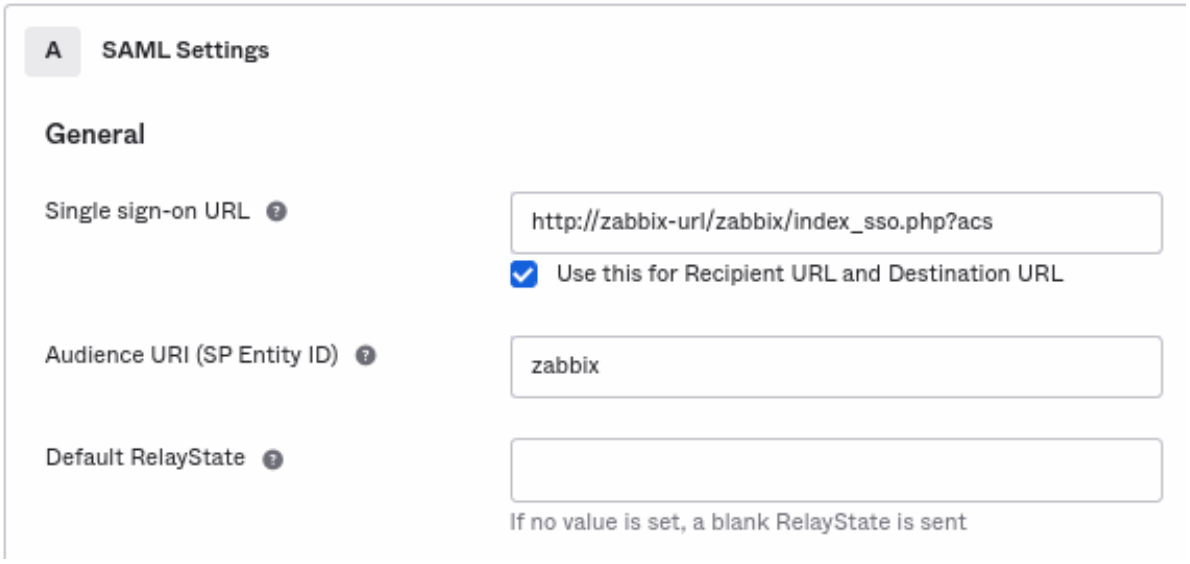
Okta konfigurācija

1. Dodieties uz https://developer.okta.com/signup/ un reģistrējieties/pierakstieties savā kontā.
2. Okta tīmekļa saskarnē atveriet *Applications* → *Applications*.
3. Noklikšķiniet uz *Create App Integration*.

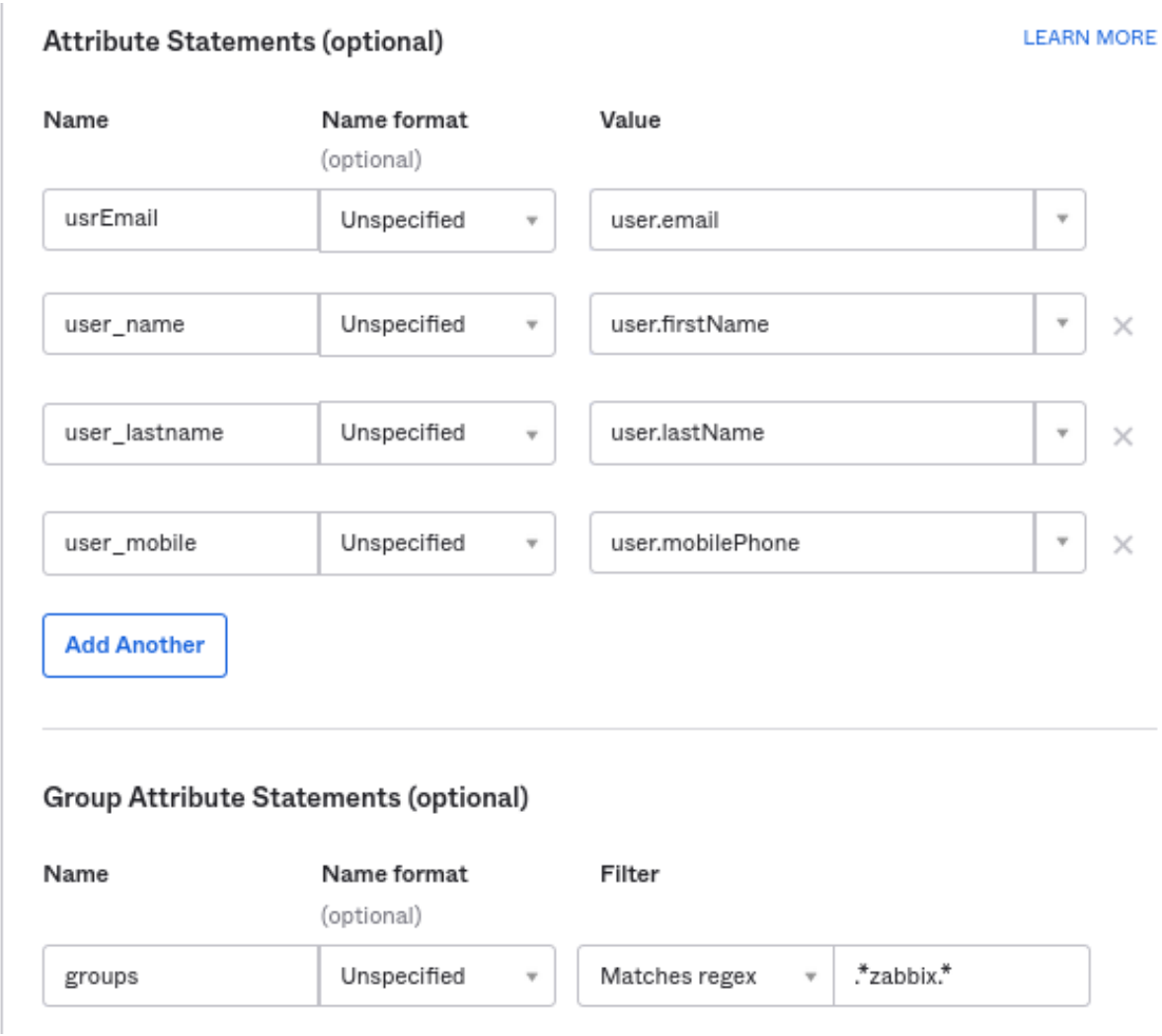


Kā pierakstīšanās metodi izvēlieties "SAML 2.0" un noklikšķiniet uz *Next*.

4. Vispārīgajos iestatījumos ievadiet lietotnes nosaukumu un noklikšķiniet uz *Next*.
5. SAML konfigurācijā ievadiet tālāk norādītās vērtības, pēc tam noklikšķiniet uz *Next*.



- Sadaļā **General** pievienojiet:
  - *Single sign-on URL*: `http://<your-zabbix-url>/zabbix/index_sso.php?acs`
 Ņemiet vērā, ka jāizmanto "http", nevis "https", lai pieprasījumā netiktu izgriezts parametrs `acs`. Jāatzīmē arī izvēles rūtiņa *Use this for Recipient URL and Destination URL*.
  - *Audience URI (SP Entity ID)*: `zabbix`
 Ņemiet vērā, ka šī vērtība tiks izmantota SAML apliecinājumā kā unikāls pakalpojuma sniedzēja identifikators (ja tā nesakritīs, darbība tiks noraidīta). Šajā laukā iespējams norādīt URL vai jebkuru datu virkni.
  - *Default RelayState*:
 Atstājiet šo lauku tukšu; ja nepieciešama pielāgota pāradresācija, to var pievienot Zabbix sadaļas *Users* > *Users* iestatījumos.
  - Aizpildiet pārējos laukus atbilstoši savām vajadzībām.
- Sadaļā **Attribute Statements/Group Attribute Statements** pievienojiet:



Šie atribūtu paziņojumi tiek ievietoti SAML apliecinājumos, kas tiek kopīgoti ar Zabbix.

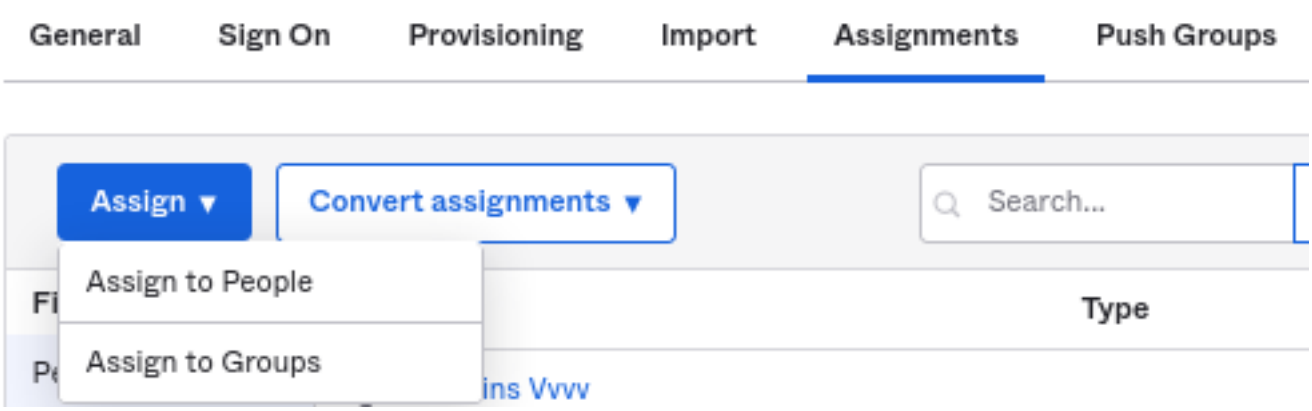
Šeit izmantotie atribūtu nosaukumi ir patvaļīgi piemēri. Varat izmantot citus atribūtu nosaukumus, taču tiem jāsakrīt ar attiecīgā lauka vērtību Zabbix SAML iestatījumos.

Ja vēlaties konfigurēt SAML pierakstīšanos Zabbix *bez* JIT lietotāju nodrošināšanas, tad ir nepieciešams tikai e-pasta atribūts.

Note:
Ja plānojat izmantot šifrētu savienojumu, ģenerējiet privāto un publisko šifrēšanas sertifikātu, pēc tam augšupielādējiet publisko sertifikātu uz Okta. Sertifikāta augšupielādes forma parādās, kad *Assertion Encryption* ir iestatīts uz "Encrypted" (noklikšķiniet uz *Show Advanced Settings*, lai atrastu šo parametru).

6. Nākamajā cilnē izvēlieties "I'm a software vendor. I'd like to integrate my app with Okta" un noklikšķiniet uz *Finish*.

7. Atveriet jaunizveidotās lietotnes cilni "Assignments" un noklikšķiniet uz pogas *Assign*, pēc tam nolaižamajā izvēlnē atlasiet "Assign to People".



8. Parādītajā uznirstošajā logā piešķiriet lietotni lietotājiem, kuri izmantos SAML 2.0 autentifikācijai ar Zabbix, pēc tam noklikšķiniet uz *Save and go back*.

9. Lejupielādējiet IdP sertifikātu. Iestatiet tam 644 atļaujas, izpildot:

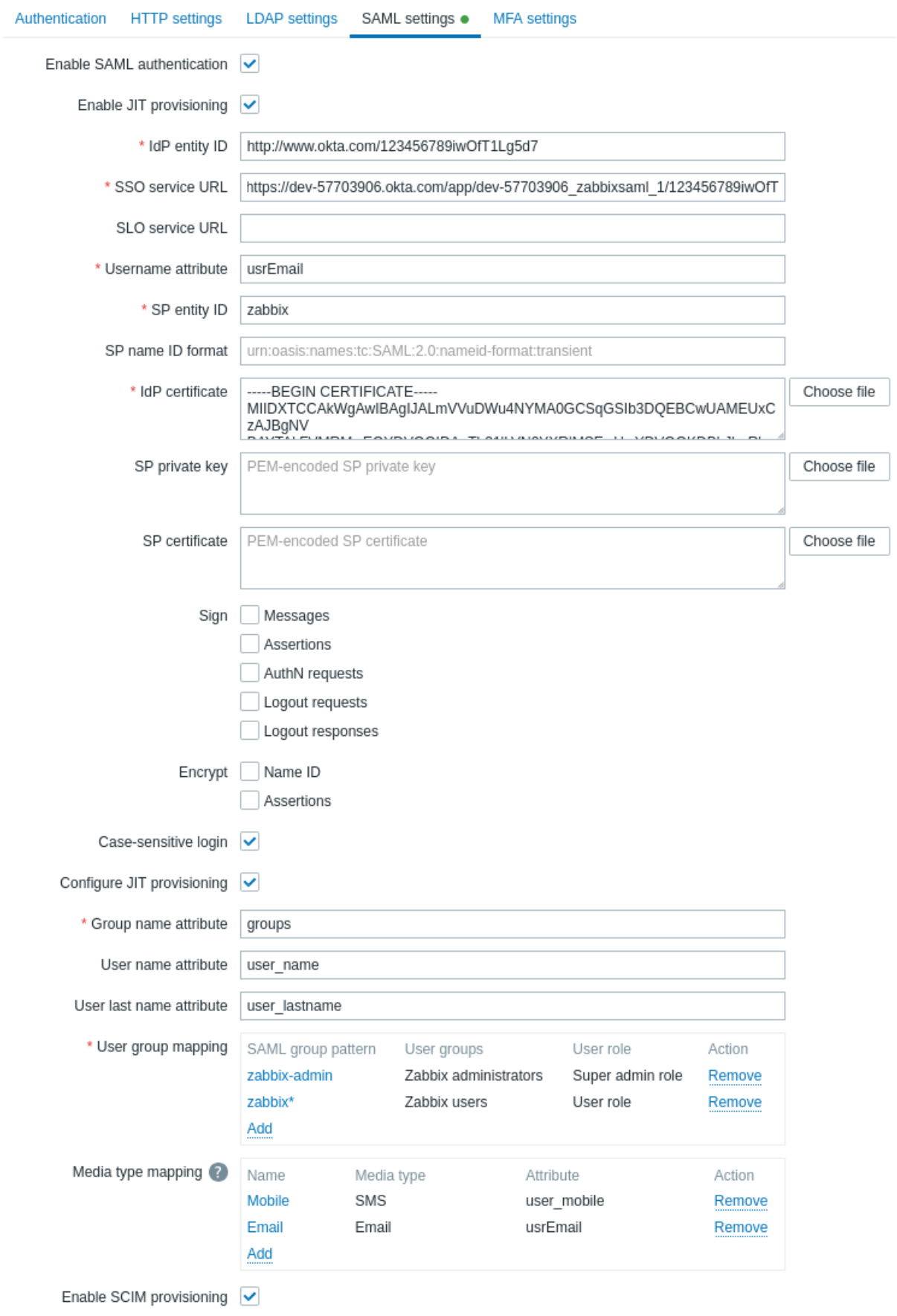
```
chmod 644 idp.crt
```

10. Atveriet cilni "Sign On" un noklikšķiniet uz pogas *View Setup Instructions*.

Iestatīšanas **norādījumi** tiks atvērti jaunā cilnē; konfigurējot Zabbix, atstājiet šo cilni atvērtu.

Zabbix konfigurācija

1. Zabbix lietotāja saskarnē atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas, pamatojoties uz Okta iestatīšanas norādījumiem:



Zabbix lauks	Iestatīšanas lauks Okta	Piemēra vērtība
IdP entītijas ID	Identity Provider Issuer	
SSO pakalpojuma URL	Identity Provider Single Sign-On URL	
Lietotājvārda atribūts	Atribūta nosaukums	`usrEmail`
SP entītijas ID	Audience URI	`zabbix`
Grupas nosaukuma atribūts	Atribūta nosaukums	`groups`

Zabbix lauks	Iestatīšanas lauks Okta	Piemēra vērtība
Lietotāja vārda atribūts	Atribūta nosaukums	`user_name`
Lietotāja uzvārda atribūts	Atribūta nosaukums	`user_lastname`

Nepieciešams arī konfigurēt lietotāju grupu un datu nesēju kartēšanu.

2. Pievienojiet Base64 sertifikātu, kas norādīts Okta SAML iestatīšanas instrukcijās.

Ja zabbix.conf.php ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā lietotāja saskarnē varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu — faili failu sistēmā nav nepieciešami.

Ja zabbix.conf.php ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē ui/conf/certs vai ceļā, kas konfigurēts zabbix.conf.php), un lietotāja saskarne nesaglabās sertifikātus datubāzē. Ņemiet vērā, ka, ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir aizkomentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no ui/conf/certs.

3. Ja Okta iestatījumā *Assertion Encryption* ir norādīts "Encrypted", arī Zabbix parametrā *Encrypt* jāatzīmē izvēles rūtiņa "Assertions".

4. Nospiediet pogu *Update*, lai saglabātu šos iestatījumus.

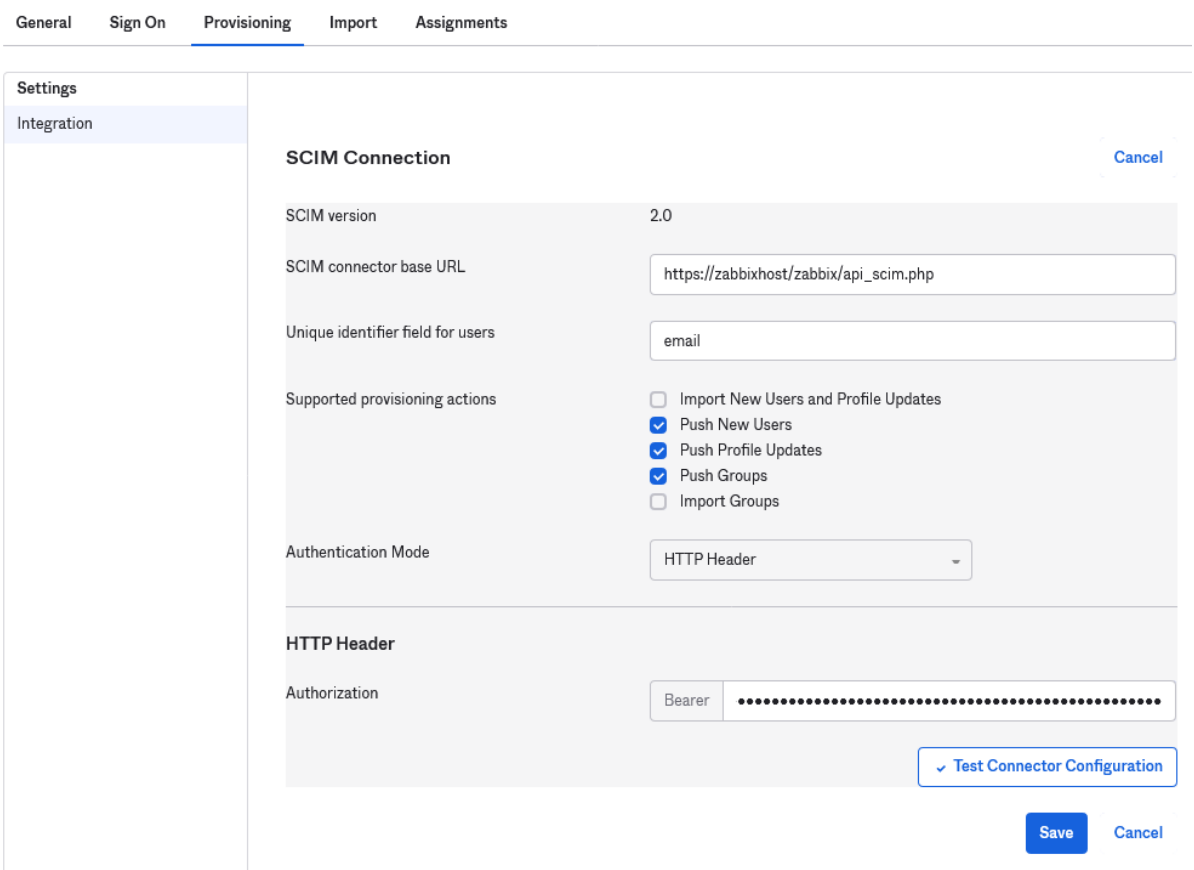
SCIM nodrošināšana

1. Lai ieslēgtu SCIM nodrošināšanu, atveriet lietotnes sadaļu "General" -> "App Settings" pakalpojumā Okta.

Atzīmējiet izvēles rūtiņu *Enable SCIM provisioning*. Rezultātā parādīsies jauna cilne *Provisioning*.

2. Atveriet cilni "Provisioning", lai iestatītu SCIM savienojumu:

- Laukā *SCIM connector base URL* norādiet ceļu uz Zabbix lietotāja saskarne un pievienojiet tam `api_scim.php`, piemēram:
 `https://<your-zabbix-url>/zabbix/api_scim.php`
- *Unique identifier field for users*: `email`
- *Authentication mode*: `HTTP header`
- Laukā *Authorization* ievadiet derīgu API tokenu ar Super admin tiesībām



Attention:
Ja rodas autentifikācijas problēmas, skatiet Authorization header forwarding.

3. Noklikšķiniet uz *Test Connector Configuration*, lai pārbaudītu savienojumu. Ja viss ir pareizi, tiks parādīts veiksmes ziņojums.

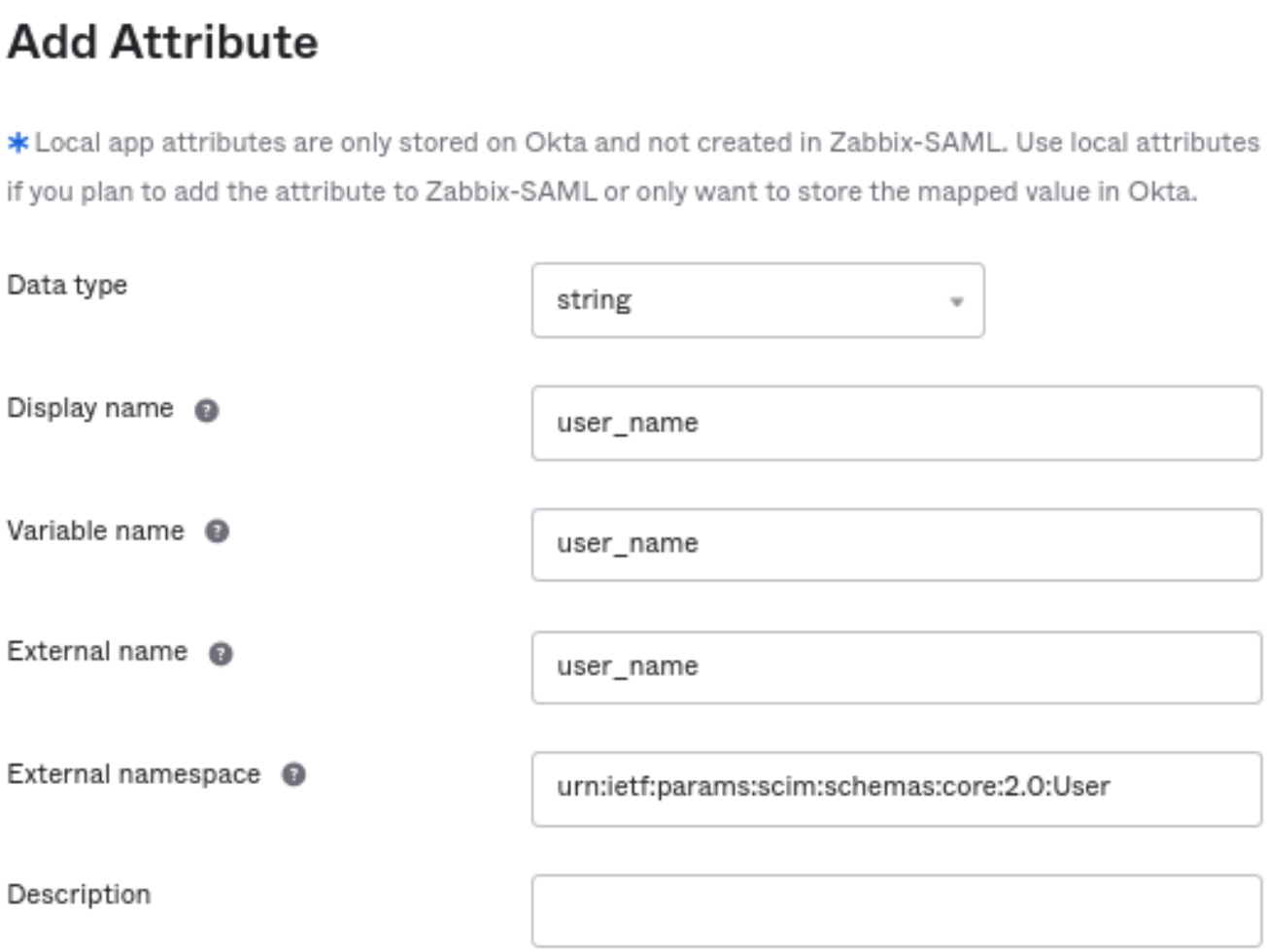
4. Sadaļā ”Provisioning” -> ”To App” pārliecinieties, ka ir atzīmētas šādas izvēles rūtiņas:

- Create Users
- Update User Attributes
- Deactivate Users

Tas nodrošinās, ka šie pieprasījumu tipi tiks nosūtīti uz Zabbix.

5. Pārliecinieties, ka visi SAML definētie atribūti ir definēti arī SCIM. Savas lietotnes profila redaktoru varat atvērt sadaļā ”Provisioning” -> ”To App”, noklikšķinot uz *Go to Profile Editor*.

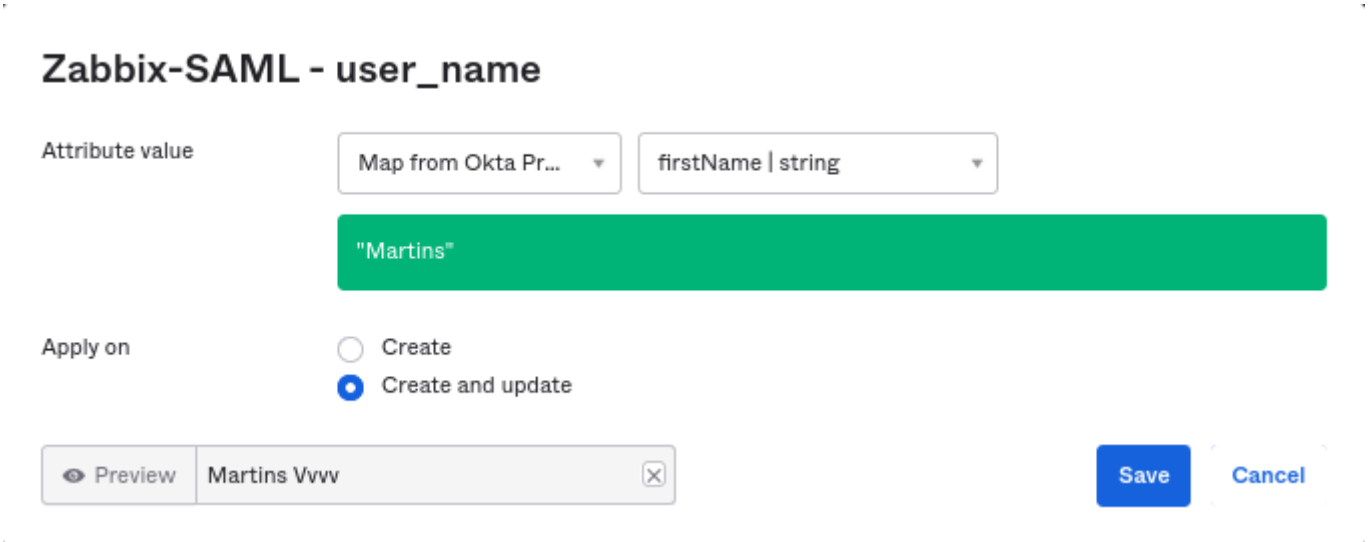
Noklikšķiniet uz *Add Attribute*. Laukus *Display name*, *Variable name*, *External name* aizpildiet ar SAML atribūta nosaukumu, piemēram, `user_name`.



External namespace jābūt tādai pašai kā lietotāja shēmai: `urn:ietf:params:scim:schemas:core:2.0:User`

6. Atveriet savas lietotnes sadaļu ”Provisioning” -> ”To App” -> ”Attribute Mappings”. Apakšā noklikšķiniet uz *Show Unmapped Attributes*. Parādīsies jaunpievienotie atribūti.

7. Kartējiet katru pievienoto atribūtu.



8. Pievienojiet lietotājus cilnē ”Assignments”. Lietotājiem pirms tam jābūt pievienotiem sadaļā *Directory -> People*. Visi šie piešķīrumi tiks nosūtīti uz Zabbix kā pieprasījumi.

9. Pievienojiet grupas cilnē ”Push Groups”. Lietotāju grupu kartēšanas šablonam Zabbix SAML iestatījumos jāsakrīt ar šeit norādīto grupu. Ja sakritības nav, lietotāju nevar izveidot Zabbix.

Informācija par grupu dalībniekiem tiek nosūtīta katru reizi, kad tiek veikta kāda izmaiņa.

14 SAML iestatīšana ar OneLogin

Pārskats

Šajā sadaļā ir sniegtas vadlīnijas vienotās pierakstīšanās un lietotāju nodrošināšanas konfigurēšanai Zabbix no OneLogin, izmantojot SAML 2.0 autentifikāciju.

OneLogin konfigurācija

Lietotnes izveide

1. Piesakieties savā OneLogin kontā. Testēšanas nolūkos varat izveidot bezmaksas izstrādātāja kontu pakalpojumā OneLogin.

2. OneLogin tīmekļa saskarnē dodieties uz *Applications → Applications*.

3. Noklikšķiniet uz ”Add App” un meklējiet atbilstošo lietotni. Šajā lapā sniegtās vadlīnijas ir balstītas uz lietotnes *SCIM Provisioner with SAML (SCIM v2 Enterprise, full SAML)* piemēru.

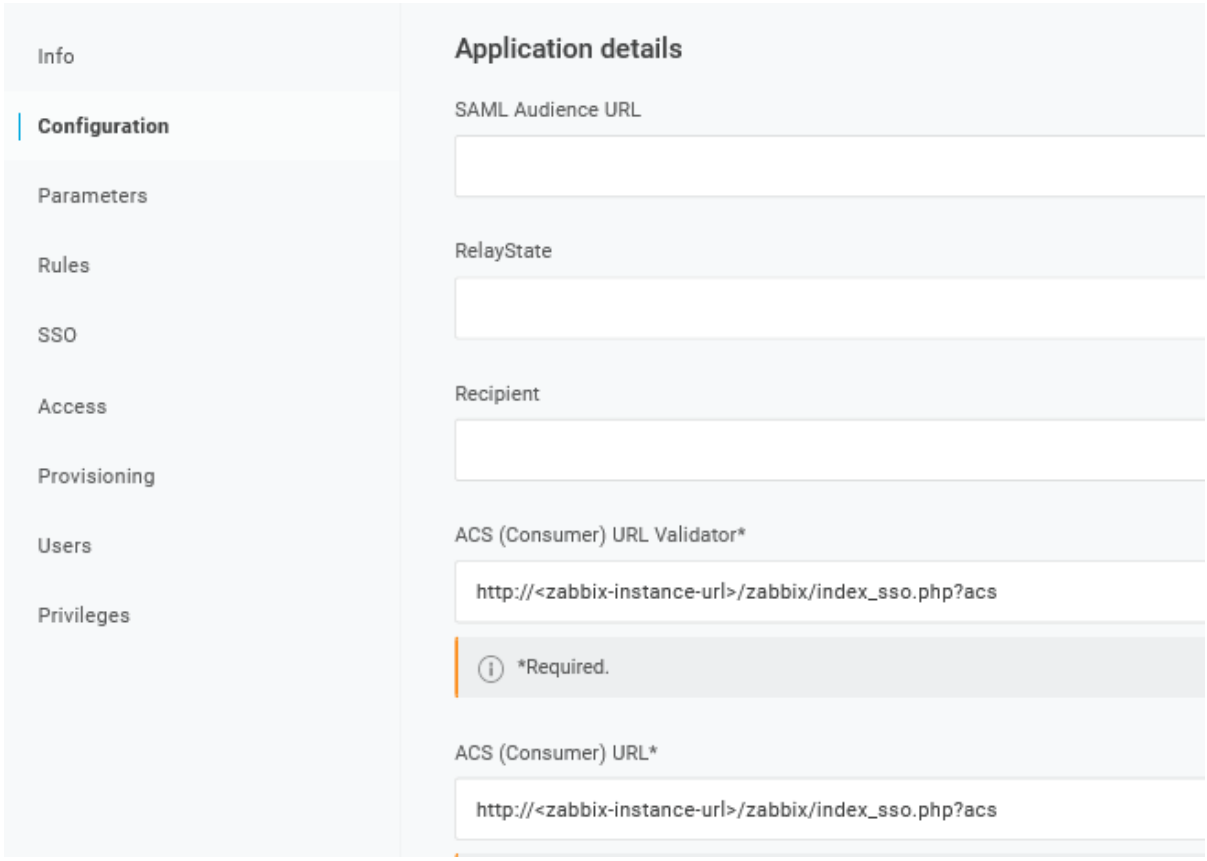
4. Sākumā, iespējams, vēlēsieties pielāgot savas lietotnes attēlojamo nosaukumu. Iespējams, vēlēsieties pievienot arī ikonu un lietotnes informāciju. Pēc tam noklikšķiniet uz *Save*.

SSO/SCIM nodrošināšanas iestatīšana

1. Sadaļā *Configuration -> Application details* iestatiet Zabbix vienotās pierakstīšanās galapunktu `http://<zabbix-instance-url>/zabb` kā šo lauku vērtību:

- *ACS (Consumer) URL Validator*
- *ACS (Consumer) URL*

Ņemiet vērā, ka jāizmanto ”http”, nevis ”https”, lai pieprasījumā netiktu izgriezts parametrs `acs`.



Ir iespējams izmantot arī "https". Lai tas darbotos ar Zabbix, failam `conf/zabbix.conf.php` nepieciešams pievienot šādu rindu:

```
$SSO['SETTINGS'] = ['use_proxy_headers' => true];
```

Pārējām opcijām atstājiet noklusējuma vērtības.

2. Sadaļā *Configuration -> API connection* iestatiet šādas vērtības:

- *SCIM Base URL*: `https://<zabbix-instance-url>/zabbix/api_scim.php`
- *SCIM JSON Template*: jāietver visi pielāgotie atribūti, kurus vēlaties nodot Zabbix, izmantojot SCIM, piemēram, `user_name`, `user_lastname`, `user_email` un `user_mobile`:

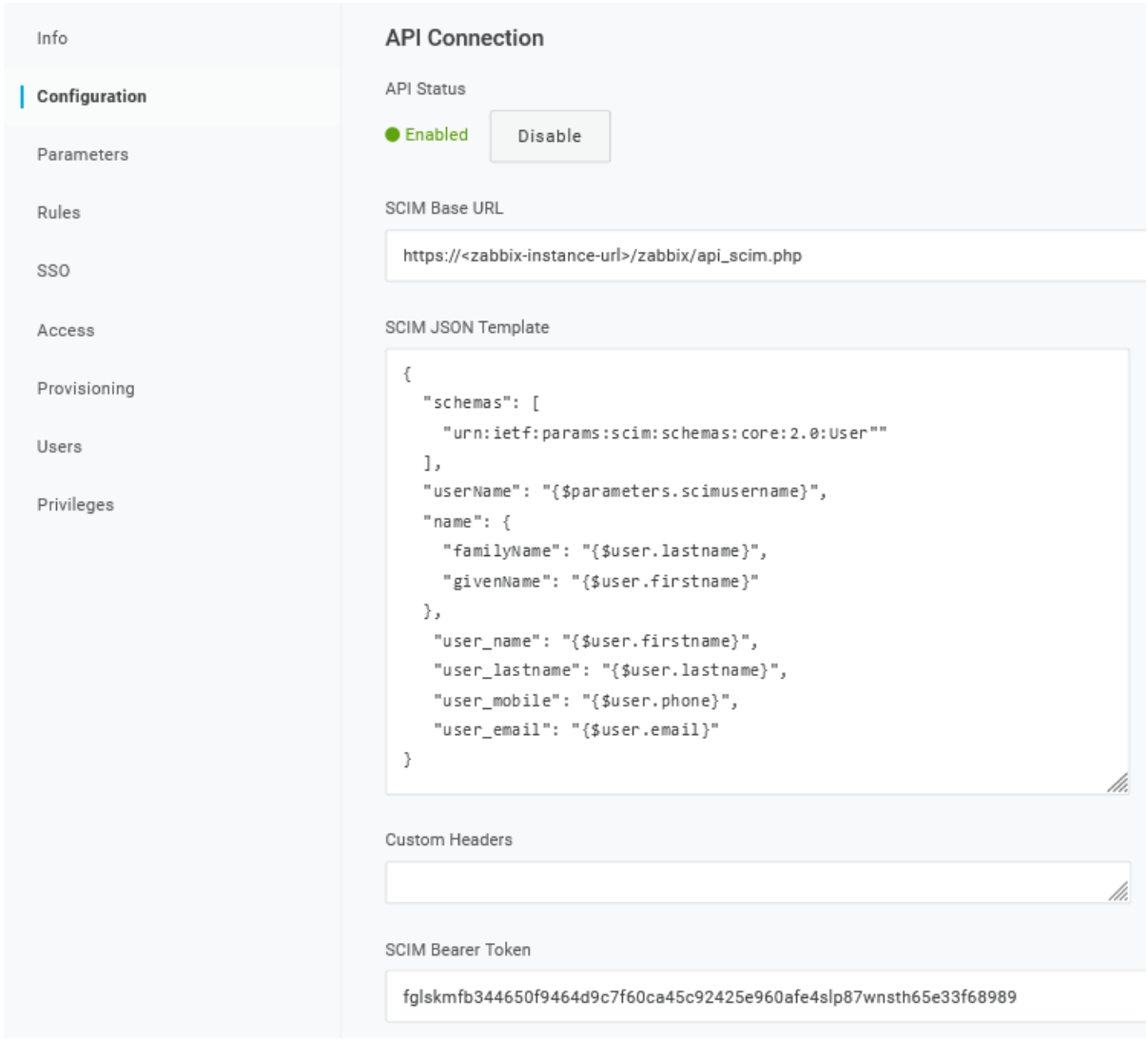
```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName": "{$parameters.scimusername}",
  "name": {
    "familyName": "{$user.lastname}",
    "givenName": "{$user.firstname}"
  },
   "user_name": "{$user.firstname}",
   "user_lastname": "{$user.lastname}",
   "user_mobile": "{$user.phone}",
   "user_email": "{$user.email}"
}
```

Atribūtu nosaukumi ir patvaļīgi. Var izmantot arī citus atribūtu nosaukumus, taču tiem jāsakrīt ar attiecīgā lauka vērtību Zabbix SAML iestatījumos.

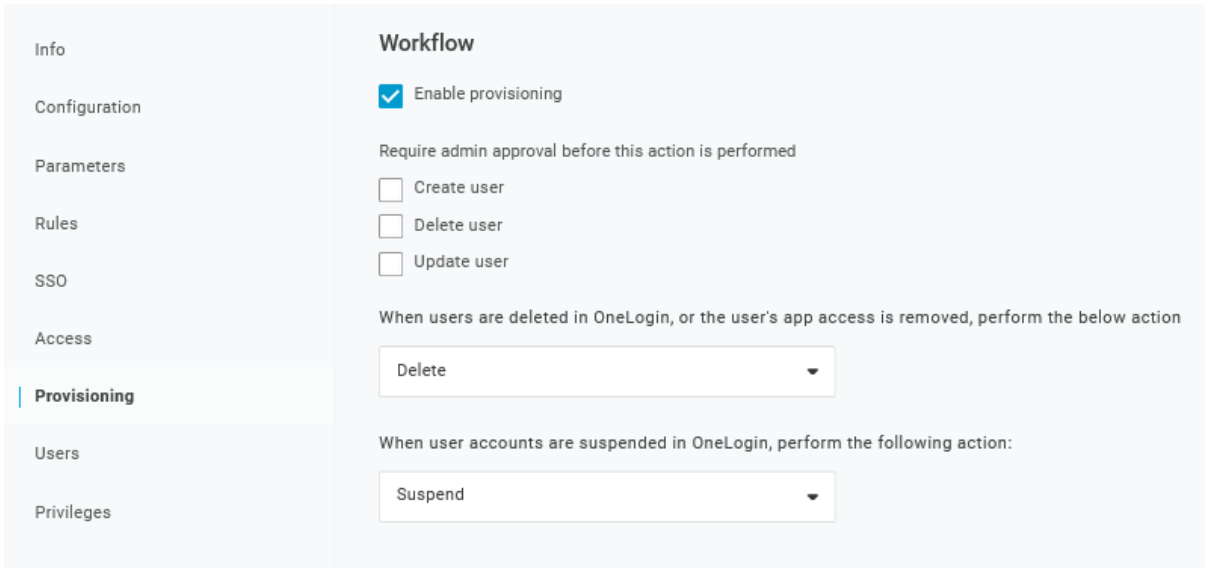
Ņemiet vērā, ka, lai lietotāju nodrošināšana darbotos, OneLogin atbildē jāsaņem atribūts 'name' ar 'givenName' un 'familyName', pat ja pakalpojuma sniedzējs to nepieprasa. Tādēļ tas ir jānorāda shēmā lietotnes konfigurācijas daļā.

- *SCIM Bearer Token*: ievadiet Zabbix API tokenu ar Super admin atļaujām.

Noklikšķiniet uz *Enable*, lai aktivizētu savienojumu.

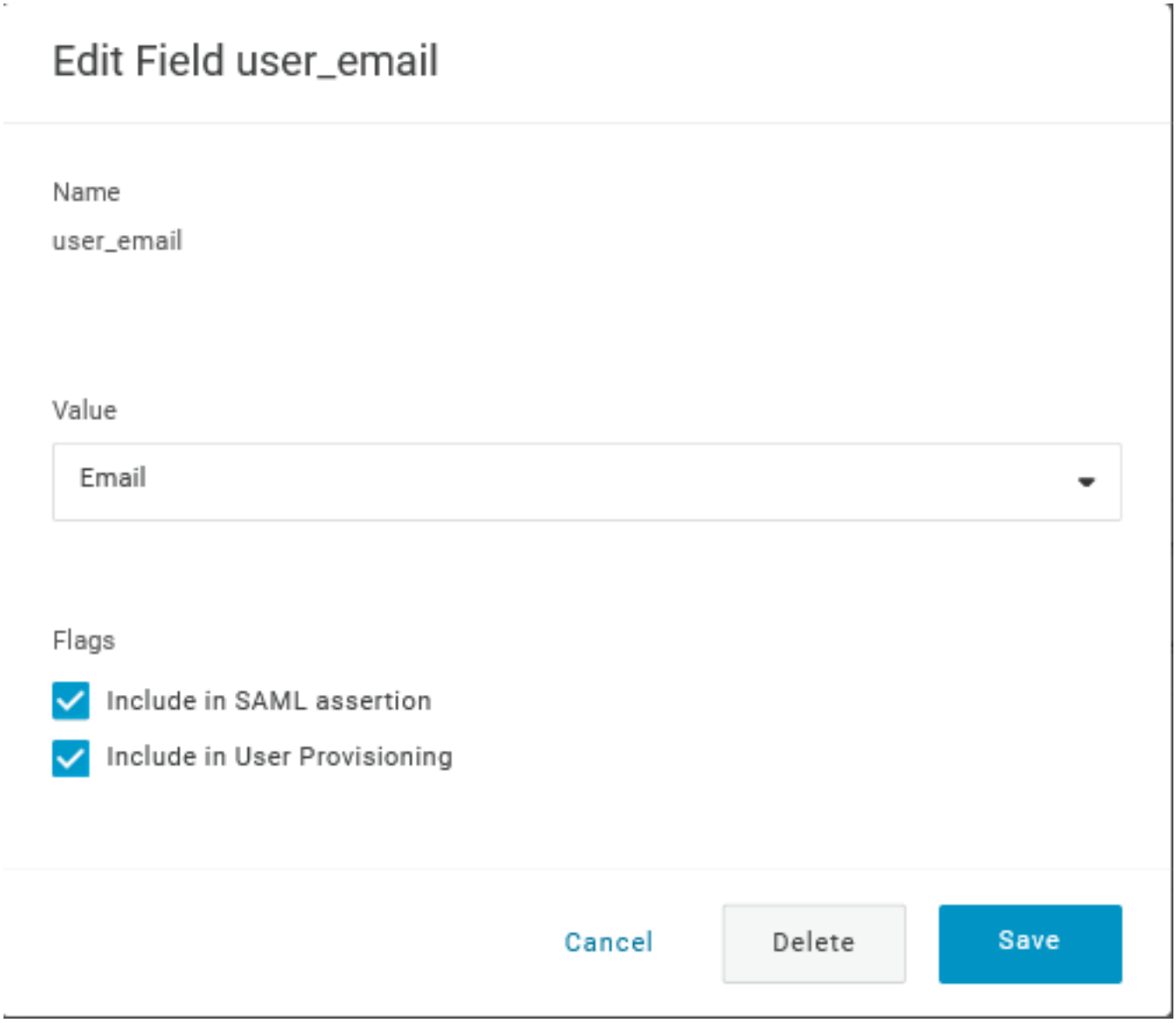


3. Lapā *Provisioning* iespējojiet opciju Provisioning:



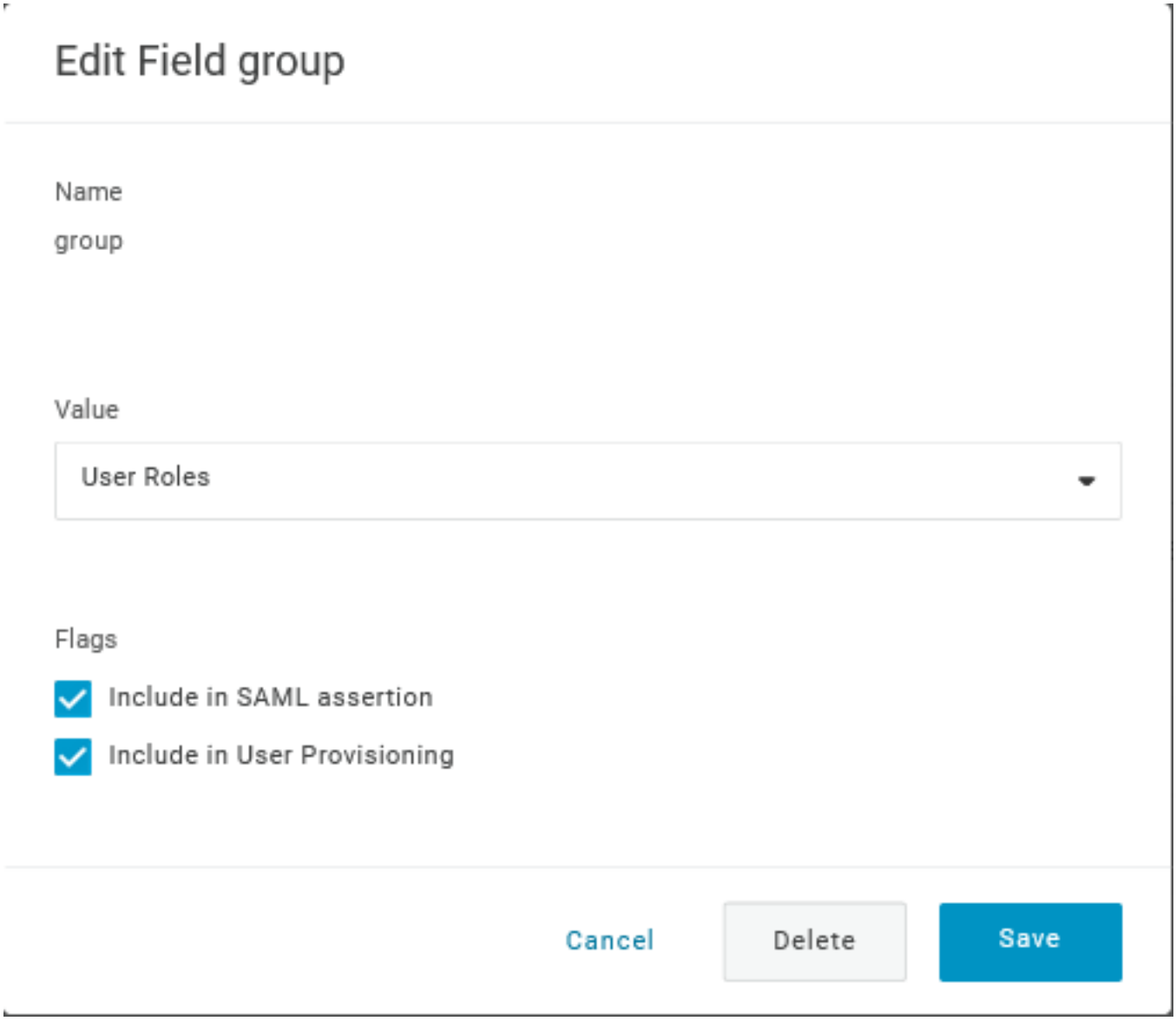
4. Lapā *Parameters* ir noklusējuma parametru saraksts:

- Pārliecinieties, ka 'scimusername' atbilst lietotāja pieteikšanās vērtībai OneLogin (piemēram, e-pasta adresei);
- Parametram 'Groups' atzīmējiet opciju *Include in User Provisioning*;
- Noklikšķiniet uz "+", lai izveidotu pielāgotos parametrus, kas nepieciešami SAML apgalvojumiem un lietotāju nodrošināšanai, piemēram, `user_name`, `user_lastname`, `user_email` un `user_mobile`:

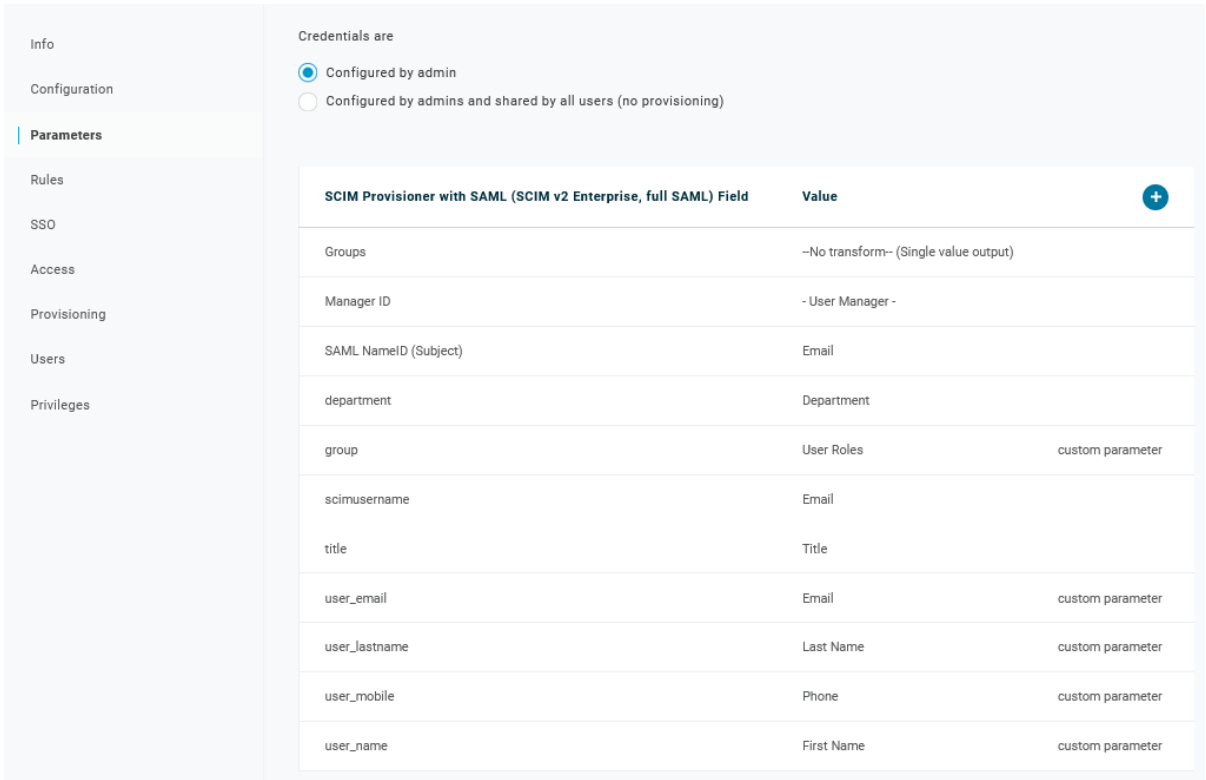


Pievienojot parametru, noteikti atzīmējiet gan opciju *Include in SAML assertion*, gan *Include in User Provisioning*.

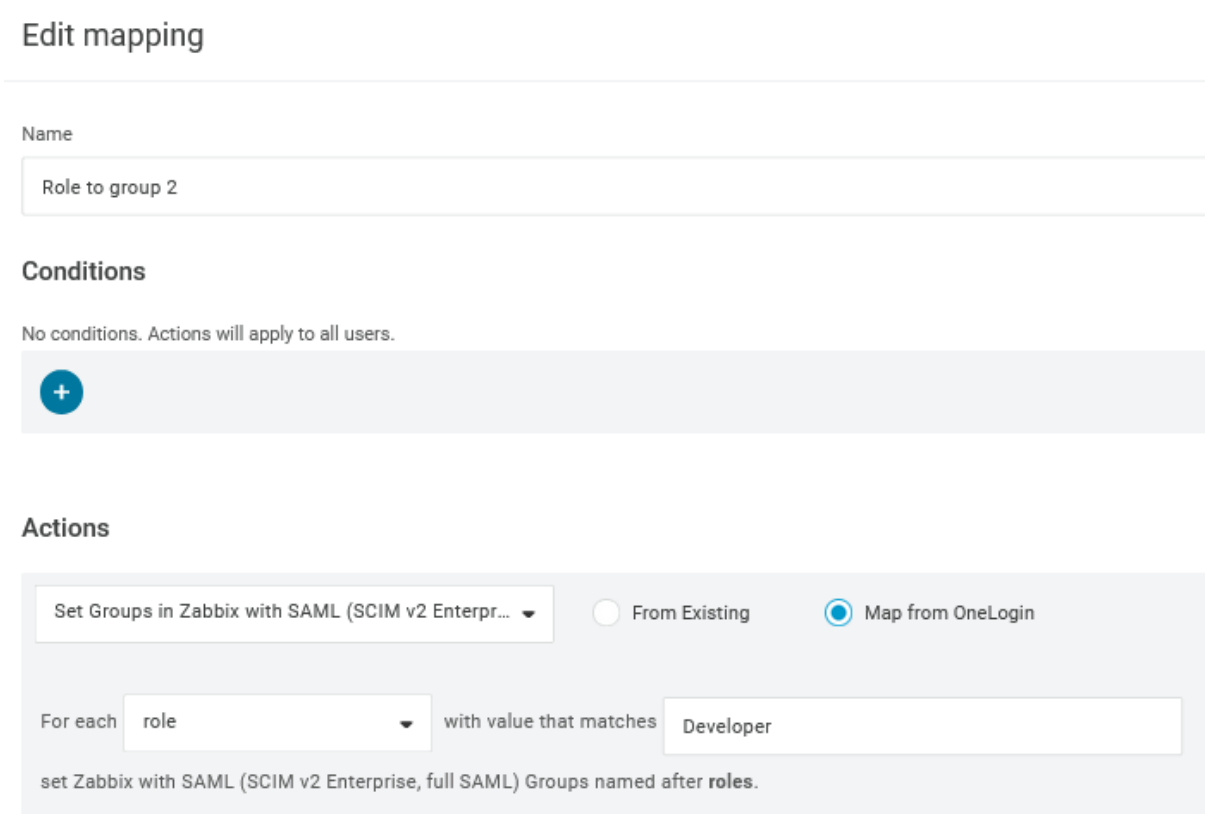
- Pievienojiet parametru 'group', kas atbilst lietotāju lomām OneLogin. Lietotāju lomas tiks nodotas kā virkne, atdalīta ar semikolu ;. OneLogin lietotāju lomas tiks izmantotas lietotāju grupu izveidei Zabbix:



Pārbaudiet parametru sarakstu:



5. Lapā *Rules* izveidojiet lietotāju lomu kartējumus noklusējuma parametram Groups.



Varat izmantot regulāro izteiksmi, lai nodotu konkrētas lomas kā grupas. Lomu nosaukumi nedrīkst saturēt `;`, jo OneLogin to izmanto kā atdalītāju, sūtot atribūtu ar vairākām lomām.

6. Lejupielādējiet IdP sertifikātu. Iestatiet tam 644 atļaujas, izpildot:

```
chmod 644 idp.crt
```

Zabbix konfigurācija

1. Zabbix lietotāja saskarnē atveriet SAML iestatījumus un aizpildiet konfigurācijas opcijas atbilstoši OneLogin konfigurācijai:

Zabbix lauks	Iestatīšanas lauks OneLogin	Vērtības piemērs
IdP entītijas ID	Issuer URL (skatiet savas lietotnes cilni SSO pakalpojumā OneLogin)	

Zabbix lauks	Iestatīšanas lauks OneLogin	Vērtības piemērs
SSO pakalpojuma URL	SAML 2.0 Endpoint (HTTP) (skatiet savas lietotnes cilni SSO pakalpojumā OneLogin)	
SLO pakalpojuma URL	SLO Endpoint (HTTP) (skatiet savas lietotnes cilni SSO pakalpojumā OneLogin)	
Lietotājvārda atribūts	Pielāgots parametrs	`user_email`
Grupas nosaukuma atribūts	Pielāgots parametrs	`group`
Lietotāja vārda atribūts	Pielāgots parametrs	`user_name`
Lietotāja uzvārda atribūts	Pielāgots parametrs	`user_lastname`

Nepieciešams arī konfigurēt lietotāju grupu kartēšanu. Mediju kartēšana nav obligāta. Noklikšķiniet uz *Update*, lai saglabātu šos iestatījumus.

2. Pievienojiet OneLogin nodrošināto Base64 sertifikātu.

Ja zabbix.conf.php failā ir iestatīts `$SSO['CERT_STORAGE'] = 'database'`, SAML konfigurēšanas laikā lietotāja saskarnē varat ielīmēt sertifikāta tekstu vai augšupielādēt sertifikāta failu — faili failu sistēmā nav nepieciešami.

Ja zabbix.conf.php failā ir iestatīts `$SSO['CERT_STORAGE'] = 'file'`, sertifikātam jābūt pieejamam failu sistēmā (pēc noklusējuma mapē ui/conf/certs vai ceļā, kas konfigurēts zabbix.conf.php), un lietotāja saskarne nesaglabās sertifikātus datubāzē. Ņemiet vērā, ka, ja `$SSO['CERT_STORAGE']` nav iestatīts vai ir aizkomentēts, tiek pieņemta glabāšana failā un sertifikāti tiek nolasīti no ui/conf/certs.

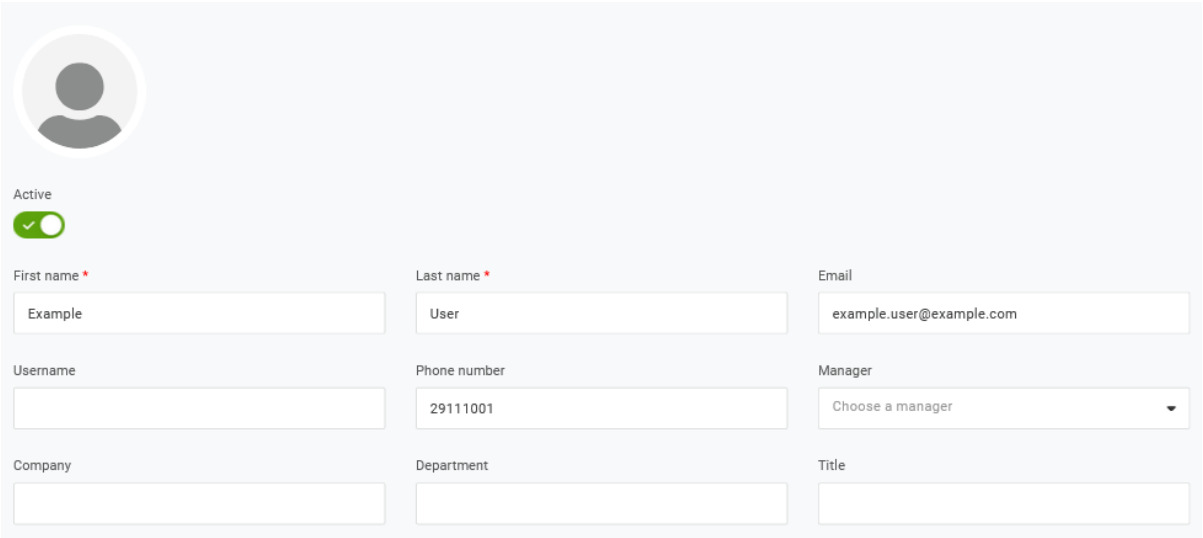
OneLogin sertifikāta lejupielādi varat atrast sadaļā *Applications* -> *SSO* -> noklikšķiniet uz *View details* zem pašreizējā sertifikāta.

3. Nospiediet pogu *Update*, lai saglabātu šos iestatījumus.

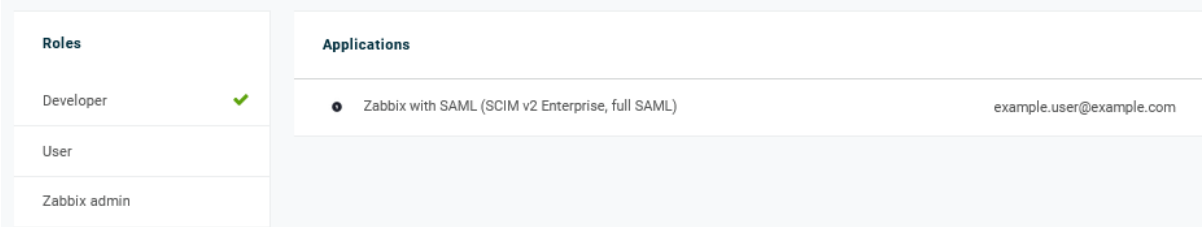
SCIM lietotāju nodrošināšana

Ja lietotāju nodrošināšana ir iespējota, tagad ir iespējams pievienot/atjaunināt lietotājus un viņu lomas OneLogin vidē un panākt, ka tie tiek nekavējoties nodrošināti Zabbix.

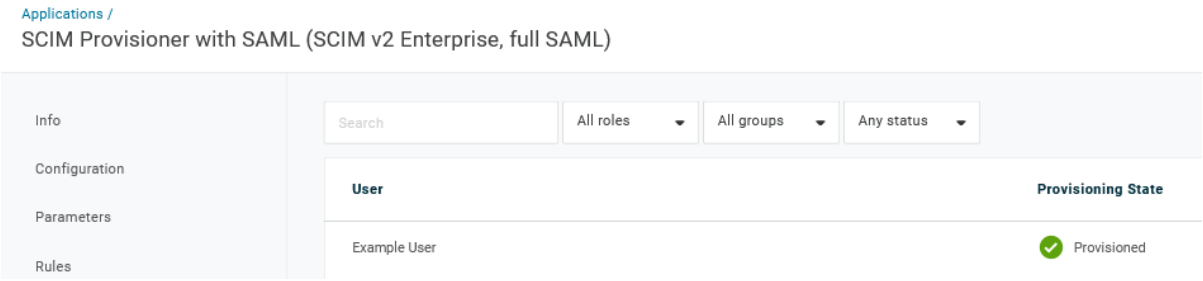
Piemēram, varat izveidot jaunu lietotāju:



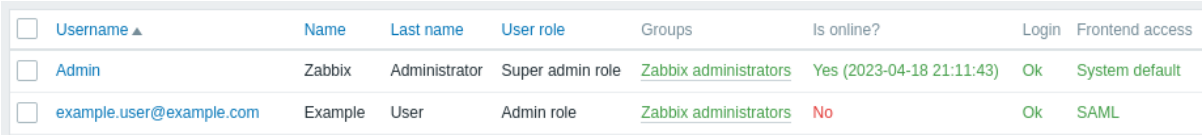
Pievienojiet to lietotāja lomai un lietotnei, kas nodrošinās lietotāju:



Saglabājot lietotāju, tas tiks nodrošināts Zabbix. Sadaļā Application -> Users varat pārbaudīt pašreizējo lietotnes lietotāju nodrošināšanas statusu:



Ja nodrošināšana ir veiksmīga, lietotājs būs redzams Zabbix lietotāju sarakstā.



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 Ieplānoto atskaišu iestatīšana

Pārskats

Šajā sadaļā ir sniegti norādījumi par Zabbix tīmekļa pakalpojuma instalēšanu un Zabbix konfigurēšanu, lai iespējotu plānoto atskaišu ģenerēšanu.

Instalēšana

Lai iespējotu ieplānoto atskaišu ģenerēšanu, jāinstalē jauns Zabbix tīmekļa pakalpojuma process un Google Chrome pārlūks. Tīmekļa pakalpojumu var instalēt tajā pašā datorā, kur ir instalēts Zabbix serveris, vai arī citā datorā. Google Chrome pārlūks jāinstalē tajā pašā datorā, kur ir instalēts tīmekļa pakalpojums.

Oficiālā zabbix-web-service pakotne ir pieejama Zabbix repozitorijā. Google Chrome pārlūks šajās pakotnēs nav iekļauts, un tas ir jāinstalē atsevišķi.

Lai kompilētu Zabbix tīmekļa servisu no pirmkoda, skatiet Installing Zabbix web service.

Pēc instalēšanas palaidiet zabbix_web_service datorā, kur ir instalēts tīmekļa serviss:

```
zabbix_web_service
```

Konfigurācija

Lai nodrošinātu pareizu saziņu starp visiem iesaistītajiem elementiem, pārliecinieties, ka servera konfigurācijas fails un lietotāja saskarnes konfigurācijas parametri ir pareizi konfigurēti.

Zabbix serveris

Zabbix servera konfigurācijas failā ir jāatjaunina šādi parametri: *WebServiceURL* un *StartReportWriters*.

WebServiceURL

Šis parametrs ir nepieciešams, lai iespējotu saziņu ar tīmekļa pakalpojumu. URL jābūt formātā `http[s]://host:port/report`.

- Pēc noklusējuma tīmekļa pakalpojums klausās 10053. portā. Citu portu var norādīt tīmekļa pakalpojuma konfigurācijas failā.
- Ceļa `/report` norādīšana ir obligāta (ceļš ir stingri ieprogrammēts un to nevar mainīt).

Piemērs:

```
WebServiceURL=http://localhost:10053/report
```

Attention:
Ir stingri ieteicams iestatīt šifrēšanu starp Zabbix serveri un Zabbix tīmekļa pakalpojumu, izmantojot sertifikātus. Pēc noklusējuma dati, kas tiek pārsūtīti starp Zabbix serveri un Zabbix tīmekļa pakalpojumu, netiek šifrēti, kas var izraisīt nesankcionētu piekļuvi.

StartReportWriters

Šis parametrs nosaka, cik daudz atskaišu rakstītāja procesu ir jāpalaiž. Ja tas nav iestatīts vai ir vienāds ar 0, atskaišu ģenerēšana ir atspējota. Atkarībā no nepieciešamo atskaišu skaita un biežuma var iespējot no 1 līdz 100 atskaišu rakstītāja procesiem.

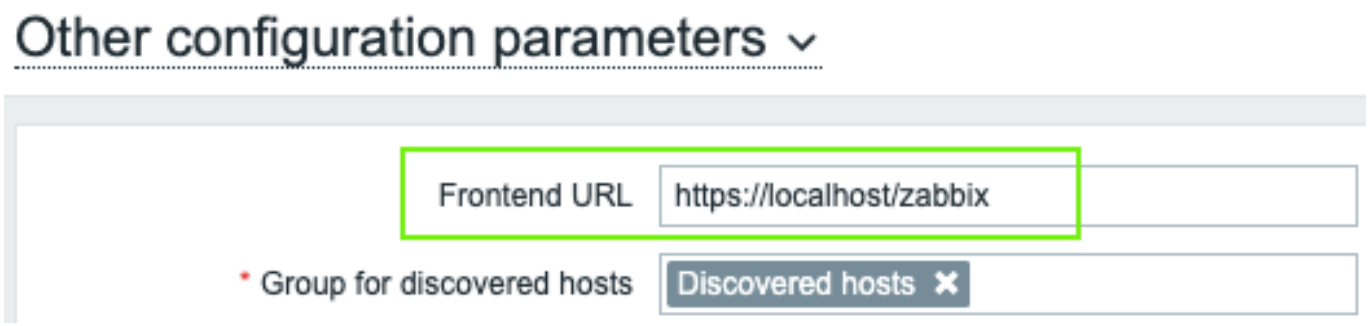
Piemērs:

```
StartReportWriters=3
```

Zabbix lietotāja saskarne

Lai iespējotu saziņu starp Zabbix lietotāja saskarni un Zabbix tīmekļa servisu, jāiestata parametrs *Frontend URL*:

- Atveriet lietotāja saskarnes izvēlnes sadaļu *Administration* > *General* > *Other*
- Parametrā *Frontend URL* norādiet pilnu Zabbix tīmekļa saskarnes URL



Note:
Kad iestatīšanas procedūra ir pabeigta, ieteicams konfigurēt un nosūtīt testa atskaiti, lai pārliecinātos, ka viss darbojas pareizi.

16 Papildu lietotāja saskarnes valodas

Pārskats

Lai Zabbix tīmekļa lietotāja saskarnē izmantotu jebkuru citu valodu, nevis angļu, attiecīgajai lokalizācijai jābūt instalētai tīmekļa serverī. Turklāt, lai tulkojumi darbotos, ir nepieciešams PHP gettext paplašinājums.

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.

Zabbix instalēšana

Ja instalējat Zabbix tieši no Zabbix git repository, tulkojumu faili ir jāģenerē manuāli. Lai ģenerētu tulkojumu failus, izpildiet:

```
make gettext
locale/make_mo.sh
```

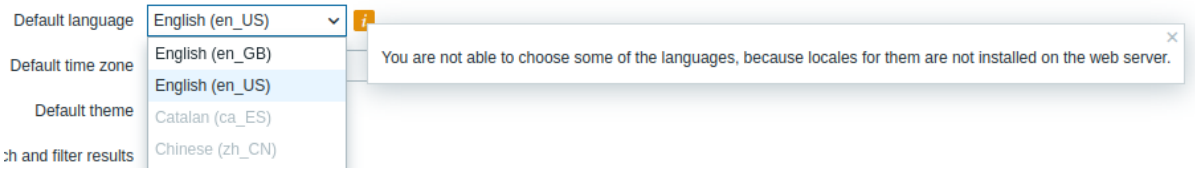
Šis solis nav nepieciešams, ja instalējat Zabbix no pakotnēm vai avota tar.gz failiem.

Valodas izvēle

Ir vairāki veidi, kā izvēlēties valodu Zabbix tīmekļa lietotāja saskarnē:

- Instalējot tīmekļa lietotāja saskarni — lietotāja saskarnes instalēšanas vednī. Izvēlētā valoda tiks iestatīta kā sistēmas noklusējuma valoda.
- Pēc instalēšanas sistēmas noklusējuma valodu var mainīt *Administration→General→GUI* izvēlnes sadaļā.
- Konkrēta lietotāja valodu var mainīt lietotāja profilā.

Ja valodas lokalizācija nav instalēta datorā, šī valoda Zabbix valodu atlasītājā būs pelēkota. Blakus valodu atlasītājam tiek parādīta oranža ikona, ja trūkst vismaz vienas lokalizācijas. Nospiežot šo ikonu, tiks parādīts šāds ziņojums: ”Jūs nevarat izvēlēties dažas no valodām, jo to lokalizācijas nav instalētas tīmekļa serverī.”



2 Procesa konfigurācija

Lūdzu, izmantojiet sānjoslu, lai piekļūtu šīs sadaļas saturam.

1 Zabbix serveris

Pārskats

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Zabbix serveris konfigurācijas failā (zabbix_server.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parametrs	Apraksts
AlertScriptsPath	Pielāgotu brīdinājumu skriptu atrašanās vieta.
AllowRoot	Atļaut serverim darboties kā 'root'.
AllowSoftwareUpdateCheck	Atļaut Zabbix UI saņemt informāciju par programmatūras atjauninājumiem no zabbix.com.
AllowUnsupportedDBVersions	Atļaut serverim darboties ar neatbalstītām datubāzes versijām.
CacheSize	Konfigurācijas kešatmiņas lielums.
CacheUpdateFrequency	Šis parametrs nosaka, cik bieži Zabbix veiks konfigurācijas kešatmiņas atjaunināšanu sekundēs.
DBHost	Datubāzes hosta nosaukums.
DBName	Datubāzes nosaukums.
DBPassword	Datubāzes parole.
DBPort	Datubāzes ports, ja netiek izmantota lokālā ligzda.
DBSchema	Datubāzes shēmas nosaukums. Tiek izmantots PostgreSQL.
DBSocket	Ceļš uz MySQL ligzdas failu.
DBUser	Datubāzes lietotājs.
DBTLSConnect	Iestatot šo opciju uz norādīto vērtību, tiek piespiedu kārtā izmantots TLS savienojums ar datubāzi.
DBTLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus datubāzes sertifikāta pārbaudei.
DBTLSCertFile	Pilns ceļš uz failu, kas satur Zabbix servera sertifikātu autentificēšanai datubāzē.
DBTLSKeyFile	Pilns ceļš uz failu, kas satur privāto atslēgu autentificēšanai datubāzē.
DBTLSCipher	Šifrēšanas šifru saraksts, ko Zabbix serveris atļauj TLS protokoliem līdz TLS v1.2. Atbalstīts tikai MySQL.
DBTLSCipher13	Šifrēšanas komplektu saraksts, ko Zabbix serveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot no versijas 8.0.16.
DebugLevel	Norādīt atkļūdošanas līmeni.
EnableGlobalScripts	Iespējot globālos skriptus Zabbix serverī.
ExportDir	Direktorija notikumu, vēstures un tendenču reāllaika eksportam newline-delimited JSON formātā. Ja iestatīts, iespējo reāllaika eksportu.
ExportFileSize	Maksimālais viena eksporta faila lielums baitos.
ExportType	Ar komatiem atdalīts entītiju tipu saraksts (events, history, trends) reāllaika eksportam (pēc noklusējuma visi tipi).
ExternalScripts	Ārējo skriptu atrašanās vieta.
Fping6Location	fping6 atrašanās vieta.
FpingLocation	fping atrašanās vieta.
FrontendAllowedIP	Ar komatiem atdalīts IP adrešu vai CIDR diapazonu saraksts, kuriem ir atļauts pieslēgties no lietotāja saskarnes.
HANodeName	Augstas pieejamības klastera mezgla nosaukums.
HistoryCacheSize	Vēstures kešatmiņas lielums.
HistoryIndexCacheSize	Vēstures indeksu kešatmiņas lielums.
HistoryStorageDateIndex	Iespējot vēstures vērtību priekšapstrādi vēstures krātuvē, lai saglabātu vērtības dažādos indeksos atkarībā no datuma.
HistoryStorageURL	Vēstures krātuves HTTP[S] URL.
HistoryStorageTypes	Ar komatiem atdalīts vērtību tipu saraksts, kas tiks nosūtīti uz vēstures krātuvi.

Parametrs	Apraksts
HousekeepingFrequency	Šis parametrs nosaka, cik bieži Zabbix veiks uzturēšanas procedūru stundās.
Include	Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
JavaGateway	Zabbix Java gateway IP adrese (vai hosta nosaukums).
JavaGatewayPort	Ports, kurā klausās Zabbix Java gateway.
ListenBacklog	Maksimālais gaidošo savienojumu skaits TCP rindā.
ListenIP	Ar komatiem atdalīts IP adrešu saraksts, uz kurām trapperim jāklausās.
ListenPort	Trapper klausīšanās ports.
LoadModule	Modulis, ko ielādēt servera palaišanas laikā.
LoadModulePath	Pilns ceļš uz servera moduļu atrašanās vietu.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais lielums.
LogSlowQueries	Nosaka, cik ilgi datubāzes vaicājums drīkst ilgt, pirms tas tiek ierakstīts žurnālā, milisekundēs.
LogType	Žurnāla izvades tips.
MaxConcurrentChecksPerPoller	Maksimālais asinhrono pārbaužu skaits, ko vienlaikus var izpildīt katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs.
MaxHousekeeperDelete	Ne vairāk kā `MaxHousekeeperDelete` rindas no tabulām `history`, `history_str`, `history_log`, `history_uint`, `history_text`, `history_bin`, `history_json`, `trends`, `trends_uint` un `problem` tiks dzēstas vienā uzdevumā viena uzturēšanas cikla laikā.
NodeAddress	IP adrese vai hosta nosaukums ar neobligātu portu, lai pārrakstītu to, kā lietotāja saskarnei jāpieslēdzas serverim.
PidFile	PID faila nosaukums.
ProblemHousekeepingFrequency	Nosaka, cik bieži Zabbix dzēsīs problēmas dzēstiem trigeriem.
ProxyConfigFrequency	Nosaka, cik bieži Zabbix serveris sūta konfigurācijas datus uz Zabbix starpniekserveri.
ProxyDataFrequency	Nosaka, cik bieži Zabbix serveris pieprasa vēstures datus no Zabbix starpniekservera.
ServiceManagerSyncFrequency	Nosaka, cik bieži Zabbix sinhronizēs pakalpojumu pārvaldnieka konfigurāciju.
SMSDevices	Ar komatiem atdalīts modemu failu saraksts, kurus atļauts izmantot Zabbix serverim.
SNMPTrapperFile	Pagaidu fails, ko izmanto datu nodošanai no SNMP trap dēmona uz serveri.
SocketDir	Direktorija IPC ligzdu glabāšanai, ko izmanto iekšējie Zabbix pakalpojumi.
SourceIP	Avota IP adrese.
SSHKeyLocation	Publisko un privāto atslēgu atrašanās vieta SSH pārbaudēm un darbībām.
SSLCertLocation	SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai.
SSLKeyLocation	SSL privāto atslēgu failu atrašanās vieta klienta autentifikācijai.
SSLCALocation	Pārrakstīt sertifikācijas iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei.
StartAgentPollers	Iepriekš atzarotu asinhrono Zabbix aģenta aptaujātāju instanču skaits.
StartAlerters	Iepriekš atzarotu brīdinātāju instanču skaits.
StartBrowserPollers	Iepriekš atzarotu pārlūka vienumu aptaujātāju instanču skaits.
StartConnectors	Iepriekš atzarotu savienotāju darbinieku instanču skaits.
StartDBSyncers	Iepriekš atzarotu vēstures sinhronizatoru instanču skaits.
StartDiscoverers	Iepriekš atzarotu atklāšanas darbinieku instanču skaits.
StartEscalators	Iepriekš atzarotu eskalatoru instanču skaits.
StartHistoryPollers	Iepriekš atzarotu vēstures aptaujātāju instanču skaits.
StartHTTPAgentPollers	Iepriekš atzarotu asinhrono HTTP aģenta aptaujātāju instanču skaits.
StartHTTPPollers	Iepriekš atzarotu HTTP aptaujātāju instanču skaits.
StartIPMIPollers	Iepriekš atzarotu IPMI aptaujātāju instanču skaits.
StartJavaPollers	Iepriekš atzarotu Java aptaujātāju instanču skaits.
StartLLDProcessors	Iepriekš atzarotu zema līmeņa atklāšanas (LLD) darbinieku instanču skaits.
StartODBCPollers	Iepriekš atzarotu ODBC aptaujātāju instanču skaits.
StartPingers	Iepriekš atzarotu ICMP pingotāju instanču skaits.
StartPollersUnreachable	Iepriekš atzarotu aptaujātāju instanču skaits nesasniedzamiem hostiem (ieskaitot IPMI un Java).
StartPollers	Iepriekš atzarotu aptaujātāju instanču skaits.
StartPreprocessors	Iepriekš palaistu priekšapstrādes darbinieku instanču skaits.
StartProxyPollers	Iepriekš atzarotu aptaujātāju instanču skaits pasīvajiem starpniekserveriem.
StartReportWriters	Iepriekš atzarotu atskaišu rakstītāju instanču skaits.
StartSNMPPollers	Iepriekš atzarotu asinhrono SNMP aptaujātāju instanču skaits.
StartSNMPTrapper	Ja iestatīts uz 1, tiks palaists SNMP trapper process.
StartTimers	Iepriekš atzarotu taimeru instanču skaits.
StartTrappers	Iepriekš atzarotu trapperu instanču skaits.
StartVMwareCollectors	Iepriekš atzarotu VMware savācēju instanču skaits.
StatsAllowedIP	Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai ārējo Zabbix instanču DNS nosaukumi. Statistikas pieprasījums tiks pieņemts tikai no šeit uzskaitītajām adresēm.

Parametrs	Apraksts
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri, aģentu, tīmekļa pakalpojumu, kā arī SNMP pārbaudēm (izņemot SNMP `walk[OID]` un `get[OID]` vienumus).
TLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus partnera sertifikāta pārbaudei, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kas satur servera sertifikātu vai sertifikātu ķēdi, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifra virkne. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherAll13	Šifra virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherCert	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifra virkne. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherCert13	Šifra virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifra virkne. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus PSK balstītai šifrēšanai.
TLSCipherPSK13	Šifra virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrēšanas komplektu atlases kritērijus PSK balstītai šifrēšanai.
TLSCRLFile	Pilns ceļš uz failu, kas satur atsauktus sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSFrontendAccept	Kādus ienākošos savienojumus pieņemt no lietotāja saskarnes.
TLSFrontendCertIssuer	Atļautais lietotāja saskarnes sertifikāta izdevējs.
TLSFrontendCertSubject	Atļautais lietotāja saskarnes sertifikāta subjekts.
TLSKeyFile	Pilns ceļš uz failu, kas satur servera privāto atslēgu, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSListen	Kontrolē TLS trappera ligzdā.
TmpDir	Pagaidu direktorija.
TrapperTimeout	Norāda noildzi sekundēs šādiem gadījumiem: - vēsturisko datu iegūšanai no Zabbix starpniekservera; - konfigurācijas datu nosūtīšanai uz Zabbix starpniekserveri; - globālo skriptu izpildei vai attālināto komandu izpildei Zabbix serverī.
TrendCacheSize	Tendenču kešatmiņas lielums.
TrendFunctionCacheSize	Tendenču funkciju kešatmiņas lielums.
UnavailableDelay	Nosaka, cik bieži hosts tiek pārbaudīts pieejamībai nepieejamības periodā.
UnreachableDelay	Nosaka, cik bieži hosts tiek pārbaudīts pieejamībai nesasniedzamības periodā.
UnreachablePeriod	Nosaka, pēc cik sekundēm nesasniedzamības hosts tiek uzskatīts par nepieejamu.
User	Atmest privilēģijas uz konkrētu, sistēmā esošu lietotāju.
ValueCacheSize	Vēstures vērtību kešatmiņas lielums.
Vault	Norāda vault nodrošinātāju.
VaultDBPath	Norāda atrašanās vietu, no kuras datubāzes akreditācijas dati jāiegūst pēc atslēgām.
VaultPrefix	Pielāgots prefikss vault ceļam vai vaicājumam.
VaultTLSCertFile	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai.
VaultTLSKeyFile	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai.
VaultToken	HashiCorp vault autentifikācijas marķieris.
VaultURL	vault servera HTTP[S] URL.
VMwareCacheSize	Koplietojamās atmiņas lielums VMware datu glabāšanai.
VMwareFrequency	Aizture sekundēs starp datu iegūšanu no viena VMware pakalpojuma.
VMwarePerfFrequency	Aizture sekundēs starp veiktspējas skaitītāju statistikas iegūšanu no viena VMware pakalpojuma.
VMwareTimeout	Maksimālais sekunžu skaits, cik ilgi vmware savācējs gaidīs atbildi no VMware pakalpojuma.
WebDriverURL	WebDriver saskarnes HTTP[S] URL.
WebServiceURL	HTTP[S] URL uz Zabbix tīmekļa pakalpojumu formātā <host:port>/report.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

AlertScriptsPath

Agojuma pielāgoto skriptu atrašanās vieta (atkarīgs no *datadir* kompilēšanas laika instalācijas mainīgā).

Noklusējums: `/usr/local/share/zabbix/alertscripts`

AllowRoot

Atļaut serverim darboties kā 'root'. Ja atspējots un serveris tiek palaists ar 'root', serveris mēģinās pārslēgties uz 'zabbix' lietotāju. Nav ietekmes, ja palaists ar parastu lietotāju.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

AllowSoftwareUpdateCheck

Atļaut Zabbix lietotāja saskarnei saņemt informāciju par programmatūras atjauninājumiem no zabbix.com.

Noklusējums: 1
 Vērtības: 0 - neatļaut; 1 - atļaut

AllowUnsupportedDBVersions

Atļaut serverim darboties ar neatbalstītām datubāzes versijām.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

CacheSize

Konfigurācijas kešatmiņas lielums baitos. Koplietojamās atmiņas lielums hostu, vienumu un trigeru datu glabāšanai.

Noklusējums: 32M
 Diapazons: 128K-64G

CacheUpdateFrequency

Šis parametrs nosaka, cik bieži Zabbix veiks konfigurācijas kešatmiņas atjaunināšanu sekundēs. Skatiet arī runtime control opcijas.

Noklusējums: 10
 Diapazons: 1-3600

DBHost

Datubāzes hosta nosaukums.
MySQL gadījumā `localhost` vai tukša virkne nozīmē, ka tiks izmantota ligzda. PostgreSQL gadījumā, ja iestatīta tukša virkne, tiek izmantota noklusējuma Unix domēna ligzda; ja iestatīts ceļš (piemēram, /var/run/pgbouncer), tiek izmantota norādītā Unix domēna ligzda.

Noklusējums: `localhost`

DBName

Datubāzes nosaukums.

Obligāts: Jā

DBPassword

Datubāzes parole. Komentējiet šo rindu, ja parole netiek izmantota.

DBPort

Datubāzes ports, ja netiek izmantota noklusējuma Unix domēna ligzda.[3]

Noklusējums MySQL: 3306

Noklusējums PostgreSQL: 5432

Diapazons: 1024-65535

DBSchema

Datubāzes shēmas nosaukums. Tiek izmantots PostgreSQL.

DBSocket

Ceļš uz MySQL ligzdas failu.[3]

DBUser

Datubāzes lietotājs.

DBTLSConnect

Iestatot šo opciju uz kādu no tālāk norādītajām vērtībām, tiek piespiesta TLS savienojuma izmantošana ar datubāzi:
required - izveidot savienojumu, izmantojot TLS
verify_ca - izveidot savienojumu, izmantojot TLS, un pārbaudīt sertifikātu
verify_full - izveidot savienojumu, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, vai DBHost norādītā datubāzes identitāte atbilst tās sertifikātam

MySQL gadījumā, sākot no versijas 5.7.11, un PostgreSQL gadījumā tiek atbalstītas šādas vērtības: `required`, `verify_ca`, `verify_full`.
MariaDB gadījumā, sākot no versijas 10.2.6, tiek atbalstītas vērtības `required` un `verify_full`.
Pēc noklusējuma nav iestatīta neviena opcija, un darbība ir atkarīga no datubāzes konfigurācijas.

DBTLSCAFile

Pilns faila ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus datubāzes sertifikāta verifikācijai.

Obligāts: nē (jā, ja DBTLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)

DBTLSCertFile

Pilns ceļš uz failu, kas satur Zabbix serveris sertifikātu autentifikācijai datubāzē.

DBTLSKeyFile

Pilns ceļš uz failu, kas satur privāto atslēgu autentificēšanai datubāzē.

DBTLSCipher

Šifrēšanas šifru saraksts, ko Zabbix serveris atļauj TLS protokoliem līdz TLS v1.2 ieskaitot. Atbalstīts tikai MySQL.

DBTLSCipher13

Šifrēšanas šifru kopu saraksts, ko Zabbix serveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot ar versiju 8.0.16.

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).
Skatiet arī izpildlaika vadības opcijas.

Noklusējums: 3
 Diapazons: 0-5

EnableGlobalScripts

Iespējot globālos skriptus Zabbix serverī.
 Piezīme: Globālo skriptu izpilde pēc noklusējuma ir iespējota. Tomēr jaunām instalācijām, sākot ar Zabbix 7.0, EnableGlobalScripts ir skaidri iestatīts uz 0 (atspējots).

Noklusējums: 1
 Vērtības: 0 - atspējot; 1 - iespējot

ExportDir

Direktorija reāllaika eksportam notikumiem, vēsturei un tendencēm JSON formātā ar rindām atdalītiem ierakstiem. Ja iestatīts, iespējo reāllaika eksportu.

ExportFileSize

Maksimālais izmērs katram eksporta failam baitos. Tiek izmantots rotācijai, ja ir iestatīts `ExportDir`.

Noklusējums: 1G
 Diapazons: 1M-1G

ExportType

Ar komatiem atdalītu entītiju tipu saraksts (notikumi, vēsture, tendences) reāllaika eksportam (pēc noklusējuma visi tipi). Derīgs tikai tad, ja ir iestatīts ExportDir.
Ņemiet vērā, ka, ja ir norādīts ExportType, bet ExportDir nav iestatīts, tā ir konfigurācijas kļūda un serveris netiks palaists.

Piemērs vēstures un tendenču eksportam:

```
ExportType=history,trends
```

Piemērs tikai notikumu eksportam:

```
ExportType=events
```

ExternalScripts

Ārējo skriptu atrašanās vieta (atkarīga no `datadir` kompilēšanas laika instalācijas mainīgā).

Noklusējums: `/usr/local/share/zabbix/externalscripts`

Fping6Location

fping6 atrašanās vieta. Pārliecinieties, ka fping6 binārajam failam ir root īpašnieks un iestatīts SUID karodziņš. Atstājiet tukšu (”Fping6Location=”), ja jūsu fping utilīta spēj apstrādāt IPv6 adreses.

Noklusējums: `/usr/sbin/fping6`

FpingLocation

fping atrašanās vieta. Pārliecinieties, ka fping binārajam failam ir root īpašnieks un iestatīts SUID karodziņš.

Noklusējums: `/usr/sbin/fping`

FrontendAllowedIP

Ar komatiem atdalīts IP adrešu vai CIDR diapazonu saraksts, kuriem ir atļauts pieslēgties no lietotāja saskarnes. Savienojums ar lietotāja saskarni tiks pieņemts tikai no šeit norādītajām adresēm, ja šis parametrs ir iestatīts. Pēc noklusējuma lietotāja saskarnes pieprasījumiem tiek pieņemti visi savienojumi. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi.

Piemērs:

```
FrontendAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

HANodeName

Augstas pieejamības klastera mezgla nosaukums. Ja tas ir tukšs, serveris darbojas savrupā režīmā un tiek izveidots mezgls ar tukšu nosaukumu.

HistoryCacheSize

Vēstures kešatmiņas lielums baitos. Koplietojamās atmiņas lielums vēstures datu glabāšanai.

Noklusējums: 16M
 Diapazons: 128K-16G

HistoryIndexCacheSize

Vēstures indeksa kešatmiņas izmērs baitos. Koplietojamās atmiņas izmērs vēstures kešatmiņā glabāto vēstures datu indeksēšanai. Indeksa kešatmiņas izmēram ir nepieciešami aptuveni 100 baiti, lai kešatmiņā saglabātu vienu vienumu.

Noklusējums: 4M
 Diapazons: 128K-16G

HistoryStorageDateIndex

Iespējot vēstures vērtību priekšapstrādi vēstures krātuvē, lai saglabātu vērtības dažādos indeksos atkarībā no datuma.

Noklusējums: 0
 Vērtības: 0 - atspējot; 1 - iespējot

HistoryStorageURL

Vēstures krātuves HTTP[S] URL. Šis parametrs tiek izmantots Elasticsearch iestatīšanai.

HistoryStorageTypes

Ar komatiem atdalīts vērtību tipu saraksts, kas jānosūta uz vēstures krātuvi. Šis parametrs tiek izmantots Elasticsearch iestatīšanai.

Noklusējums: `uint,dbl,str,log,text,json`

HousekeepingFrequency

Šis parametrs nosaka, cik bieži Zabbix veiks uzturēšanas procedūru stundās. Uzturēšana ir novecojušas informācijas noņemšana no datubāzes.
Piezīme: Lai novērstu uzturētāja pārslodzi (piemēram, ja vēstures un tendenču periodi ir būtiski samazināti), vienā uzturēšanas ciklā katram vienumam tiek dzēsta ne vairāk kā 4 reizes HousekeepingFrequency stundas ilga novecojušas informācijas daļa. Tādējādi, ja HousekeepingFrequency ir 1, vienā ciklā tiks dzēstas ne vairāk kā 4 stundas novecojušas informācijas (sākot no vecākā ieraksta).
Piezīme: Lai samazinātu slodzi servera startēšanas laikā, uzturēšana tiek atlikta uz 30 minūtēm pēc servera starta. Tādējādi, ja HousekeepingFrequency ir 1, pati pirmā uzturēšanas procedūra pēc servera starta tiks izpildīta pēc 30 minūtēm un pēc tam atkārtosies ar vienas stundas aizkavi.
Automātisko uzturēšanu ir iespējams atspējot, iestatot HousekeepingFrequency uz 0. Šajā gadījumā uzturēšanas procedūru var palaist tikai ar *housekeeper_execute* izpildlaika vadības opciju, un vienā uzturēšanas ciklā dzēstās novecojušās informācijas periods ir 4 reizes lielāks par periodu kopš pēdējā uzturēšanas cikla, bet ne mazāks par 4 stundām un ne lielāks par 4 dienām.
Skatiet arī izpildlaika vadības opcijas un informāciju par uzturēšanas procedūru.

Noklusējums: 1
 Diapazons: 0-24

Iekļaut

Jūs varat iekļaut atsevišķus failus vai visus failus direktorijā konfigurācijas failā. Lai iekļautu tikai atbilstošos failus norādītajā direktorijā, rakstzīmju aizstājējzīme zvaigznīte tiek atbalstīta šablonu atbilstības noteikšanai. Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java gateway IP adrese (vai resursdatora nosaukums). Nepieciešams tikai tad, ja ir palaisti Java aptaujātāji.

JavaGatewayPort

Ports, kuru klausās Zabbix Java gateway.

Noklusējums: 10052
 Diapazons: 1024-32767

ListenBacklog

Maksimālais gaidošo savienojumu skaits TCP rindā.
Noklusējuma vērtība ir stingri kodēta konstante, kas ir atkarīga no sistēmas.
Arī maksimālā atbalstītā vērtība ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz "implementācijas noteiktajam maksimumam".

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās trapper jāgaida savienojumi.
Ja šis parametrs nav norādīts, trapper klausīsies visās tīkla saskarnēs.

Noklusējums: 0.0.0.0

ListenPort

Trapper klausīšanās ports.

Noklusējums: 10051
 Diapazons: 1024-32767

LoadModule

Modulis, ko ielādēt servera startēšanas laikā. Moduļi tiek izmantoti, lai paplašinātu servera funkcionalitāti. Modulim jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma. Ja iepriekš norādītais ceļš ir absolūts (sākas ar '/'), tad LoadModulePath tiek ignorēts.
Formāti:
LoadModule=<module.so>
LoadModule=<path/module.so><b atļauts iekļaut vairākus LoadModule parametrus.

LoadModulePath

Pilns ceļš uz servera moduļu atrašanās vietu. Noklusējuma vērtība ir atkarīga no kompilēšanas opcijām.

LogFile

Žurnālfaila nosaukums.

Obligāti: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija kāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024
 Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogSlowQueries

Nosaka, cik ilgi datubāzes vaicājums drīkst izpildīties, pirms tas tiek reģistrēts žurnālā, milisekundēs.
0 - nereģistrēt lēnus vaicājumus.
Šī opcija tiek iespējota, sākot ar DebugLevel=3.

Noklusējums: 0
 Diapazons: 0-3600000

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar parametru LogFile;
system - rakstīt žurnālu syslog;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: file

MaxConcurrentChecksPerPoller

Maksimālais asinhrono pārbaužu skaits, ko vienlaikus var izpildīt katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs. Skatiet StartHTTPAgentPollers, StartAgentPollers un StartSNMPPollers.

Noklusējums: 1000
 Diapazons: 1-1000

MaxHousekeeperDelete

Tabulā "housekeeper" ir ietverti mājturības procedūras "uzdevumi" šādā formātā: [housekeeperid], [object], [objectid].
 Ne vairāk kā MaxHousekeeperDelete rindas no tabulām history, history_str, history_log, history_uint, history_text, history_bin, history_json, trends, trends_uint un problem tiks dzēstas vienam uzdevumam vienā mājturības ciklā.
 Ja iestatīts uz 0, tad ierobežojums vispār netiek piemērots. Šajā gadījumā jums ir jāzina, ko darāt, lai nepārslogotu datubāzi.[2]
Šis parametrs attiecas tikai uz tādu datu dzēšanu, kas palikuši pēc jau izdzēstiem vienumiem.
 Skatiet arī sīkāku informāciju par mājturības procedūru.

Noklusējums: 5000
 Diapazons: 0-1000000

NodeAddress

IP adrese vai resursdatora nosaukums ar neobligātu portu, lai ignorētu to, kā lietotāja saskarnei jāpieslēdzas serverim.
Formāts: <address>[:<port>]

Ja IP adrese vai resursdatora nosaukums nav iestatīts, tiks izmantota ListenIP vērtība. Ja ListenIP nav iestatīts, tiks izmantota vērtība localhost.
Ja ports nav iestatīts, tiks izmantota ListenPort vērtība. Ja ListenPort nav iestatīts, tiks izmantota vērtība 10051.

Šo opciju var ignorēt ar lietotāja saskarnes konfigurācijā norādīto adresi.

Skatiet arī: parametru HANodeName; Augstas pieejamības iespējošana.

Noklusējums: localhost:10051

PidFile

PID faila nosaukums.

Noklusējums: /tmp/zabbix_server.pid

ProblemHousekeepingFrequency

Nosaka, cik bieži Zabbix dzēsīs problēmas dzēstiem trigeriem sekundēs.
 Skatiet arī detalizētu informāciju par uzkopšanas procedūru.

Noklusējums: 60
 Diapazons: 1-3600

ProxyConfigFrequency

Nosaka, cik bieži Zabbix serveris nosūta konfigurācijas datus Zabbix starpniekserverim sekundēs. Tiek izmantots tikai starpniekserveriem pasīvajā režīmā.

Noklusējums: 10
 Diapazons: 1-604800

ProxyDataFrequency

Nosaka, cik bieži Zabbix serveris pieprasa vēstures datus no Zabbix starpniekservera sekundēs. Tiek izmantots tikai starpniekserveriem pasīvajā režīmā.

Noklusējums: 1
 Diapazons: 1-3600

ServiceManagerSyncFrequency

Nosaka, cik bieži Zabbix sinhronizēs pakalpojumu pārvaldnieka konfigurāciju sekundēs.

Noklusējums: 60
 Diapazons: 1-3600

SMSDevices

Ar komatiem atdalīts modema failu saraksts, kuriem ir atļauts izmantot Zabbix serveris.
SMS sūtīšana nav iespējama, ja šis parametrs nav iestatīts.

Piemērs:

```
SMSDevices=/dev/ttyUSB0,/dev/ttyUSB1
```

SNMPTrapperFile

Pagaidu fails, ko izmanto datu nodošanai no SNMP trapu dēmona uz serveri.
Tam jābūt tādam pašam kā zabbix_trap_receiver.pl vai SNMPTT konfigurācijas failā.

Noklusējums: /tmp/zabbix_traps.tmp

SocketDir

Direktorija, kurā glabāt IPC ligzdas, ko izmanto iekšējie Zabbix servisi.

Noklusējums: /tmp

SourceIP

Avota IP adrese šādiem savienojumiem:

- izejošajiem savienojumiem ar Zabbix starpniekserveris un Zabbix aģents

- bezaģenta savienojumiem (VMware, SSH, JMX, SNMP, Telnet un vienkāršajām pārbaudēm)
- HTTP aģents savienojumiem
- skripta vienums JavaScript HTTP pieprasījumiem
- priekšapstrādes JavaScript HTTP pieprasījumiem
- paziņojumu e-pastu sūtīšanai (savienojumiem ar SMTP serveri)
- webhook paziņojumiem (JavaScript HTTP savienojumiem)
- savienojumiem ar Vault

SSHKeyLocation

Publisko un privāto atslēgu atrašanās vieta SSH pārbaudēm un darbībām.

SSLCertLocation

SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai.
Šis parametrs tiek izmantots tikai tīmekļa uzraudzībā.

SSLKeyLocation

SSL privātās atslēgas failu atrašanās vieta klienta autentifikācijai.
Šis parametrs tiek izmantots tikai tīmekļa uzraudzībā.

SSLCALocation

Pārraksta sertifikātu iestādes (CA) failu atrašanās vietu SSL servera sertifikāta verifikācijai. Ja nav iestatīts, tiks izmantots sistēmas mēroga direktorijs.
Ņemiet vērā, ka šī parametra vērtība tiks iestatīta kā libcurl opcija CURLOPT_CAPATH. Libcurl versijām pirms 7.42.0 tas darbojas tikai tad, ja libcurl tika kompilēts lietošanai ar OpenSSL. Plašāku informāciju skatiet cURL tīmekļa lapā.
Šis parametrs tiek izmantots tīmekļa uzraudzībā un SMTP autentifikācijā.

StartAgentPollers

Iepriekš atzaroto Zabbix aģents aptaujātāju instanču skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartAlerters

Iepriekš atzaroto brīdinājumu procesu skaits.

Noklusējums: 3
 Diapazons: 1-100

StartBrowserPollers

Pirms atzarošanas izveidoto pārlūka vienumu aptaujātāju instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartConnectors

Iepriekš atzaroto savienotāju darbinieku instanču skaits. Savienotāju pārvaldnieka process tiek palaists automātiski, kad tiek palaists savienotāja darbinieks.

Noklusējums: 0
 Diapazons: 0-1000

StartDBSyncers

Iepriekš atzaroto vēstures sinhronizētāju instanču skaits.
Piezīme: Esiet piesardzīgi, mainot šo vērtību; tās palielināšana var nodarīt vairāk ļauna nekā laba. Aptuveni noklusējuma vērtībai vajadzētu būt pietiekamai, lai apstrādātu līdz 4000 NVPS.

Noklusējums: 4
 Diapazons: 1-100

StartDiscoverers

Iepriekš atzaroto atklāšanas procesu[1] skaits.

Noklusējums: 5
 Diapazons: 0-1000

StartEscalators

Iepriekš atzaroto eskalatoru instanču skaits.

Noklusējums: 1
 Diapazons: 1-100

StartHistoryPollers

Iepriekš atzaroto vēstures aptaujātāju instanču skaits.
Nepieciešams tikai aprēķinātajām pārbaudēm.

Noklusējums: 5
 Diapazons: 0-1000

StartHTTPAgentPollers

Iepriekš atzaroto HTTP aģenta aptaujātāju skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartHTTPPollers

Iepriekš atzaroto HTTP aptaujātāju[1] instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartIPMIPollers

Iepriekš atzaroto IPMI aptaujātāju instanču skaits.

Noklusējums: 0
 Diapazons: 0-1000

StartJavaPollers

Iepriekš atzaroto Java aptaujātāju[1] instanču skaits.

Noklusējums: 0
 Diapazons: 0-1000

StartLLDProcessors

Zema līmeņa atklāšanas (LLD) darbinieku[1] iepriekš atzaroto instanču skaits.
LLD pārvaldnieka process tiek automātiski palaists, kad tiek palaists LLD darbinieks.

Noklusējums: 2
 Diapazons: 1-100

StartODBCPollers

Iepriekš atdalītu ODBC aptaujātāju[1] instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartPingers

Iepriekš atzaroto ICMP pingeru[1] skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartPollersUnreachable

Iepriekš atzaroto nepieejamu hostu aptaujātāju instanču skaits (ieskaitot IPMI un Java)[1].
Ja ir palaisti parastie, IPMI vai Java aptaujātāji, jādarbojas vismaz vienam nepieejamu hostu aptaujātājam.

Noklusējums: 1
 Diapazons: 0-1000

StartPollers

Iepriekš atzaroto aptaujātāju[1] instanču skaits.

Noklusējums: 5
 Diapazons: 0-1000

StartPreprocessors

Iepriekš palaisto priekšapstrādes darbinieku[1] pavedienu skaits jāiestata ne mazāks par pieejamo CPU kodolu skaitu. Jāiestata vairāk darbinieku, ja priekšapstrādi neierobežo CPU un tai ir daudz tīkla pieprasījumu.

Noklusējums: 16
 Diapazons: 1-1000

StartProxyPollers

Iepriekš atzaroto pasīvo starpniekserveru aptaujātāju[1] instanču skaits.

Noklusējums: 1
 Diapazons: 0-250

StartReportWriters

Iepriekš atzaroto atskaišu rakstītāju instanču skaits.
Ja iestatīts uz 0, plānotā atskaišu ģenerēšana ir atspējota.
Atskaites pārvaldnieka process tiek automātiski palaists, kad tiek palaists atskaišu rakstītājs.

Noklusējums: 0
 Diapazons: 0-100

StartSNMPPollers

Iepriekš atzaroto SNMP aptaujātāju instanču skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartSNMPTrapper

Ja iestatīts uz 1, tiks palaists SNMP trapper process.

Noklusējums: 0
 Diapazons: 0-1

StartTimers

Iepriekš atzaroto taimeru instanču skaits.
Taimeri apstrādā uzturēšanas periodus.

Noklusējums: 1
 Diapazons: 1-1000

StartTrappers

Iepriekš atzaroto trapperu[1] instanču skaits.
Trapperi pieņem ienākošos savienojumus no Zabbix sender, aktīvajiem aģentiem un aktīvajiem starpniekserveriem.

Noklusējums: 5
 Diapazons: 0-1000

StartVMwareCollectors

Iepriekš atzaroto VMware kolektora instanču skaits.

Noklusējums: 0
 Diapazons: 0-250

StatsAllowedIP

Ar komatiem atdalīts ārējo Zabbix instanču IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Statistikas pieprasījumi tiks pieņemti tikai no šeit norādītajām adresēm. Ja šis parametrs nav iestatīts, netiks pieņemts neviens statistikas pieprasījums.
Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām adresēm, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi.

Piemērs:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Timeout

Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri, aģentu, tīmekļa pakalpojumu un mantotajām SNMP pārbaudēm (viens OID numurs vai virkne).

Šis parametrs nosaka ilgumu dažādām saziņas darbībām:

- attālo komandu izpildei Zabbix aģentā
- SSH/Telnet komandu izpildei
- pieprasījumiem uz Zabbix tīmekļa pakalpojumu
- saziņas noildzei mediju tipa testa pieprasījumam un `history.push` metodei
- vienumu pārplānošanai, kad IPMI saskarne kļūst nepieejama
- atbildes nosūtīšanai Zabbix starpniekserverim, ja datu apmaiņa neizdodas tiesību vai šifrēšanas problēmu dēļ
- termiņam IPC asinhronajām ligzdām un izpildlaika vadības opcijām
- JMX savienojumiem
- statistikas iegūšanai no attāla Zabbix starpniekservera vai servera
- atbilžu nosūtīšanai Zabbix lietotāja saskarnei
- asinhrono aptaujātāju DNS pieprasījumiem
- atbildei aktīvās pārbaudes sirdspukstam
- Zabbix aģenta datu (vērtību) iegūšanai no aktīvajiem aģentiem
- datu iegūšanai no Zabbix sender
- atbildei, kad trapper neizdodas apstrādāt pieprasījumu
- aktīvo pārbaužu saraksta nosūtīšanai Zabbix aģentam

Šī noildze **netiks** izmantota tām pārbaudēm, kurām lietotāja saskarnē ir konfigurēti elastīgās noildzes iestatījumi (globālajā, starpniekservera vai atsevišķa vienuma līmenī).
Piemēram, SNMP `walk[OID]` un `get[OID]` vienumi izmanto lietotāja saskarnē konfigurēto noildzi; mantotās SNMP pārbaudes joprojām izmanto servera noildzes vērtību.

Noklusējums: 3
 Diapazons: 1-30

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus(-u) vienādranga sertifikāta verifikācijai un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur servera sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCipherAll

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātos un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 protokolā. Pārraksta noklusējuma šifrkomplektu atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātiem balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrēšanas algoritmu kopas atlases kritērijus sertifikātos balstītai šifrēšanai.

TLSCipherPSK

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrkomplektu atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCRLFile

Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSFrontendAccept

Kādus ienākošos savienojumus pieņemt no lietotāja saskarnes.

Var norādīt vairākas vērtības, atdalot tās ar komatu:

- unencrypted - pieņemt savienojumus bez šifrēšanas.
- cert - pieņemt savienojumus, kas aizsargāti ar TLS un sertifikātu.

Noklusējums: `unencrypted`

TLSFrontendCertIssuer

Atļautais lietotāja saskarnes sertifikāta izdevējs.

TLSFrontendCertSubject

Atļautais lietotāja saskarnes sertifikāta subjekts.

TLSKeyFile

Pilns ceļš uz failu, kas satur servera privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSListen

Kontrolē TLS trapper ligzdā.

Atbalstītās vērtības:

- required - pieņemt tikai TLS savienojumus

TmpDir

Pagaidu direktorija.

Noklusējums: `/tmp`

TrapperTimeout

Norāda noildzi sekundēs šādām darbībām:

- vēsturisko datu saņemšanai no Zabbix starpniekservera
- konfigurācijas datu nosūtīšanai uz Zabbix starpniekserveri
- globālo skriptu izpildei vai attālo komandu izpildei uz Zabbix servera

Noklusējums: 300
 Diapazons: 1-300

TrendCacheSize

Tendenču kešatmiņas izmērs baitos.
Koplietojamās atmiņas izmērs tendenču datu glabāšanai.

Noklusējums: 4M
 Diapazons: 128K-16G

TrendFunctionCacheSize

Tendenču funkciju kešatmiņas lielums baitos.
Koplietojamās atmiņas lielums aprēķināto tendenču funkciju datu kešošanai.

Noklusējums: 4M
 Diapazons: 128K-2G

UnavailableDelay

Nosaka, cik bieži hosts tiek pārbaudīts pieejamības noteikšanai nepieejamības periodā sekundēs.

Noklusējums: 60
 Diapazons: 1-3600

UnreachableDelay

Nosaka, cik bieži hosts tiek pārbaudīts pieejamības noteikšanai nesasniedzamības periodā sekundēs.

Noklusējums: 15
 Diapazons: 1-3600

UnreachablePeriod

Nosaka, pēc cik sekundēm nesasniedzamības hosts tiek uzskatīts par nepieejamu.

Noklusējums: 45
 Diapazons: 1-3600

Lietotājs

Samazināt privilēģijas līdz konkrētam, sistēmā esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā 'root' un AllowRoot ir atspējots.

Noklusējums: `zabbix`

ValueCacheSize

Vēstures vērtību kešatmiņas lielums baitos.
Koplietojamās atmiņas lielums vienumu vēstures datu pieprasījumu kešošanai.
Iestatot uz 0, vērtību kešatmiņa tiek atspējota (nav ieteicams).
Kad vērtību kešatmiņai beidzas koplietojamā atmiņa, servera žurnālā ik pēc 5 minūtēm tiek ierakstīts brīdinājuma ziņojums.

Noklusējums: 8M
 Diapazons: 0,128K-64G

Vault

Norāda glabātuves nodrošinātāju:
HashiCorp - HashiCorp KV Secrets Engine versija 2
CyberArk - CyberArk Central Credential Provider
Jāsakrīt ar glabātuves nodrošinātāju, kas iestatīts lietotāja saskarnē.

Noklusējums: `HashiCorp`

VaultDBPath

Vault ceļš vai vaicājums atkarībā no Vault, no kura datubāzes akreditācijas dati tiks iegūti pēc atslēgām.

Atslēgas, kas tiek izmantotas HashiCorp, ir 'password' un 'username'.

Ceļa piemērs ar VaultPrefix=/v1/secret/data/zabbix/:

```
database
```

Ceļa piemērs bez VaultPrefix:

```
secret/zabbix/database
```

Atslēgas, kas tiek izmantotas CyberArk, ir 'Content' un 'UserName'.

Piemērs:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_server_database
```

Šo opciju var izmantot tikai tad, ja DBUser un DBPassword nav norādīti.

VaultPrefix

Pielāgots prefikss Vault ceļam vai vaicājumam atkarībā no Vault. Ja tas nav norādīts, tiks izmantotas vispiemērotākās noklusējuma vērtības.
Ņemiet vērā, ka HashiCorp gadījumā pēc montēšanas punkta automātiski tiek pievienots 'data', ja VaultPrefix nav norādīts.

HashiCorp prefiksa piemērs:

```
v1/secret/data/zabbix/
```

Cyberark prefiksa piemērs:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai
 Sertifikāta failam jābūt PEM1 formātā.
 Ja sertifikāta fails satur arī privāto atslēgu, atstājiet SSL atslēgas faila lauku tukšu.
 Direktorija, kurā atrodas šis fails, ir norādīta ar konfigurācijas parametru SSLCertLocation.
Šo opciju var izlaist, taču tā ir ieteicama CyberArkCCP glabātuvei.

VaultTLSKeyFile

SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai.
 Privātās atslēgas failam jābūt PEM1 formātā.
 Direktorijs, kurā atrodas šis fails, ir norādīts konfigurācijas parametrā SSLKeyLocation.
Šo opciju var izlaist, taču tā ir ieteicama CyberArkCCP glabātuvei.

VaultToken

HashiCorp Vault autentifikācijas marķieris, kam jābūt ģenerētam tikai Zabbix serverim ar tikai lasīšanas atļauju ceļiem, kas norādīti Vault makrosos, un tikai lasīšanas atļauju ceļam, kas norādīts neobligātajā VaultDBPath konfigurācijas parametrā.
Tā ir kļūda, ja VaultToken un vides mainīgais VAULT_TOKEN ir definēti vienlaikus.

Obligāts: Jā, ja Vault ir iestatīts uz *HashiCorp*; pretējā gadījumā nē

VaultURL

Glabātuves servera HTTP[S] URL. Ja SSLCALocation nav norādīts, tiks izmantots sistēmas mēroga CA sertifikātu direktorijs.

Noklusējums: `https://127.0.0.1:8200`

VMwareCacheSize

Koplietojamās atmiņas lielums VMware datu glabāšanai.
VMware iekšējo pārbaudi zabbix[vmware,buffer,...] var izmantot, lai uzraudzītu VMware kešatmiņas lietojumu (skatiet Iekšējās pārbaudes).
Ņemiet vērā, ka koplietojamā atmiņa netiek piešķirta, ja nav konfigurēta neviena palaižama vmware kolektora instance.

Noklusējums: 8M
 Diapazons: 256K-2G

VMwareFrequency

Aizkave sekundēs starp datu iegūšanu no viena VMware pakalpojuma.
Šī aizkave jāiestata atbilstoši mazākajam atjaunināšanas intervālam jebkuram VMware uzraudzības vienumam.

Noklusējums: 60
 Diapazons: 10-86400

VMwarePerfFrequency

Aizkave sekundēs starp veiktspējas skaitītāju statistikas izgūšanu no viena VMware pakalpojuma. Šī aizkave jāiestata uz mazāko atjaunināšanas intervālu jebkuram VMware uzraudzības vienumam, kas izmanto VMware veiktspējas skaitītājus.

Noklusējums: 60
 Diapazons: 10-86400

VMwareTimeout

Maksimālais sekunžu skaits, cik ilgi VMware kolektors gaidīs atbildi no VMware servisa (vCenter vai ESX hipervizora).

Noklusējums: 10
 Diapazons: 1-300

WebServiceURL

HTTP[S] URL uz Zabbix tīmekļa pakalpojumu formātā `http[s]://host:port/report`.

Piemērs:

```
WebServiceURL=http://localhost:10053/report
```

Piezīme: shēmu (`http://`) var izlaist tikai savienojumiem bez TLS (HTTP); ja ir konfigurēts TLS, jāizmanto `https://`.

WebDriverURL

WebDriver saskarnes HTTP[S] URL.

Piemērs (izmantošanai ar Selenium WebDriver autonomo serveri):

```
WebDriverURL=http://localhost:4444
```

Piezīmes

[1] Ņemiet vērā, ka pārāk liels datu iegūšanas procesu skaits (pollers, unreachable pollers, ODBC pollers, HTTP pollers, Java pollers, pingers, trappers, proxypollers) kopā ar IPMI manager, SNMP trapper, preprocessing workers un discovery workers var izsmelt preprocessing manager failu deskriptoru limita vērtību uz procesu.

> **Warning:**
> Failu deskriptoru limita izsmelšana izraisīs Zabbix serveris darbības apstāšanos, parasti neilgi pēc palaišanas, taču dažkārt tas var notikt vēlāk. Lai izvairītos no šādām problēmām, pārskatiet Zabbix serveris konfigurācijas failu, lai optimizētu vienlaicīgo pārbaužu un procesu skaitu. Papildus, ja nepieciešams, pārliecinieties, ka failu deskriptoru limits ir iestatīts pietiekami augsts, pārbaudot un pielāgojot sistēmas limitus.

[2] Ja tiek dzēsts liels skaits vienumu, tas palielina datubāzes noslodzi, jo housekeeper būs jānoņem visi šo vienumu vēstures dati. Piemēram, ja no veidne jānoņem tikai 1 vienuma prototips, bet šī veidne ir piesaistīta 50 hosts un katram hosts prototips tiek izvērsts līdz 100 reāliem vienumiem, kopumā būs jānoņem 5000 vienumu (1*50*100). Ja MaxHousekeeperDelete ir iestatīts uz 500 (MaxHousekeeperDelete=500), housekeeper procesam vienā ciklā būs jānoņem līdz 2500000 vērtību (5000*500) dzēstajiem vienumiem no history un trends tabulām.

[3] DBSocket un DBPort serveris konfigurācijā ir savstarpēji izslēdzoši. Norādiet tikai vienu no tiem vai atstājiet abus nedefinētus.

2 Zabbix starpniekserveris

Pārskats

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Zabbix starpniekservera konfigurācijas fails (zabbix_proxy.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parametrs	Apraksts
AllowRoot	Atļaut starpniekserverim darboties kā 'root'.
AllowUnsupportedDBVersions	Atļaut starpniekserverim darboties ar neatbalstītām datubāzes versijām.
CacheSize	Konfigurācijas kešatmiņas lielums.
ConfigFrequency	Šis parametrs ir **novecojis** (**deprecated**) (tā vietā izmantojiet ProxyConfigFrequency). Cik bieži starpniekserveris saņem konfigurācijas datus no Zabbix servera sekundēs.
DataSenderFrequency	Starpniekserveris sūtīs savāktos datus uz serveri ik pēc N sekundēm.
DBHost	Datubāzes hosta nosaukums.
DBName	Datubāzes nosaukums vai ceļš uz datubāzes failu SQLite3 gadījumā.
DBPassword	Datubāzes parole.
DBPort	Datubāzes ports, ja netiek izmantota lokālā ligzda.
DBSchema	Datubāzes shēmas nosaukums. Tiek izmantots PostgreSQL.
DBSocket	Ceļš uz MySQL ligzdas failu.

Parametrs	Apraksts
DBUser	Datubāzes lietotājs.
DBTLSConnect	Iestatot šo opciju uz norādīto vērtību, tiek piespiests izmantot TLS savienojumu ar datubāzi.
DBTLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus datubāzes sertifikāta pārbaudei.
DBTLSCertFile	Pilns ceļš uz failu, kas satur Zabbix starpniekservera sertifikātu autentificēšanai datubāzē.
DBTLSKeyFile	Pilns ceļš uz failu, kas satur privāto atslēgu autentificēšanai datubāzē.
DBTLSCipher	Šifrēšanas šifru saraksts, ko Zabbix starpniekserveris atļauj TLS protokoliem līdz TLS v1.2. Atbalstīts tikai MySQL.
DBTLSCipher13	Šifrēšanas komplektu saraksts, ko Zabbix starpniekserveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot no versijas 8.0.16.
DebugLevel	Atkļūdošanas līmenis.
EnableRemoteCommands	Vai ir atļautas attālās komandas no Zabbix servera.
ExternalScripts	Ārējo skriptu atrašanās vieta.
Fping6Location	fping6 atrašanās vieta.
FpingLocation	fping atrašanās vieta.
HistoryCacheSize	Vēstures kešatmiņas lielums.
HistoryIndexCacheSize	Vēstures indeksa kešatmiņas lielums.
Hostname	Unikāls, reģistrjutīgs starpniekservera nosaukums.
HostnameItem	Vienums, ko izmanto Hostname iestatīšanai, ja tas nav definēts.
HousekeepingFrequency	Cik bieži Zabbix veiks tīrīšanas procedūru stundās.
Include	Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
JavaGateway	Zabbix Java gateway IP adrese (vai hosta nosaukums).
JavaGatewayPort	Ports, kurā klausās Zabbix Java gateway.
ListenBacklog	Maksimālais gaidošo savienojumu skaits TCP rindā.
ListenIP	Ar komatiem atdalīts IP adrešu saraksts, kurās trapperim jāklausās.
ListenPort	Trapper klausīšanās ports.
LoadModule	Modulis, ko ielādēt starpniekservera palaišanas laikā.
LoadModulePath	Pilns ceļš uz starpniekservera moduļu atrašanās vietu.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais lielums.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogSlowQueries	Cik ilgi datubāzes vaicājums drīkst ilgt, pirms tas tiek reģistrēts žurnālā.
LogType	Žurnāla izvades tips.
MaxConcurrentChecksPerPoller	Maksimālais asinhrono pārbaužu skaits, ko vienlaikus var izpildīt katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs.
PidFile	PID faila nosaukums.
ProxyBufferMode	Norāda vēstures, atklāšanas un automātiskās reģistrācijas datu glabāšanas mehānismu (disks/atmiņa/hibrīds).
ProxyConfigFrequency	Cik bieži starpniekserveris saņem konfigurācijas datus no Zabbix servera sekundēs.
ProxyLocalBuffer	Starpniekserveris glabās datus lokāli N stundas, pat ja dati jau ir sinhronizēti ar serveri.
ProxyMemoryBufferAge	Maksimālais datu vecums starpniekservera atmiņas buferī sekundēs.
ProxyMemoryBufferSize	Koplietojamās atmiņas kešatmiņas lielums savāktajiem vēstures, atklāšanas un automātiskās reģistrācijas datiem.
ProxyMode	Starpniekservera darbības režīms (aktīvs/pasīvs).
ProxyOfflineBuffer	Starpniekserveris glabās datus N stundas, ja nav savienojamības ar Zabbix serveri.
Server	Ja ProxyMode ir iestatīts uz aktīvo režīmu: Zabbix servera IP adrese vai DNS nosaukums (adrese:ports) vai klasteris (adrese:ports;adrese2:ports), no kura saņemt konfigurācijas datus un uz kuru sūtīt datus. Ja ProxyMode ir iestatīts uz pasīvo režīmu: ar komatiem atdalīts Zabbix servera IP adrešu saraksts, pēc izvēles CIDR apzīmējumā, vai DNS nosaukumi.
SNMPTrapperFile	Pagaidu fails, ko izmanto datu nodošanai no SNMP trap dēmona uz starpniekserveri.
SocketDir	Direktorija, kur glabāt IPC ligzdas, ko izmanto iekšējie Zabbix servisi.
SourceIP	Avota IP adrese.
SSHKeyLocation	Publisko un privāto atslēgu atrašanās vieta SSH pārbaudēm un darbībām.
SSLCertLocation	SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai.
SSLKeyLocation	SSL privāto atslēgu failu atrašanās vieta klienta autentifikācijai.
SSLCALocation	Pārrakstīt sertifikātu iestādes (CA) failu atrašanās vietu SSL servera sertifikāta pārbaudei.
StartAgentPollers	Iepriekš atzaroto asinhrono Zabbix aģenta aptaujātāju instanču skaits.
StartBrowserPollers	Iepriekš atzaroto pārlūka vienumu aptaujātāju instanču skaits.
StartDBSyncers	Iepriekš atzaroto vēstures sinhronizētāju instanču skaits.
StartDiscoverers	Iepriekš atzaroto atklāšanas darbinieku instanču skaits.
StartHTTPAgentPollers	Iepriekš atzaroto asinhrono HTTP aģenta aptaujātāju instanču skaits.

Parametrs	Apraksts
StartHTTPPollers	Iepriekš atzaroto HTTP aptaujātāju instanču skaits.
StartIPMIPollers	Iepriekš atzaroto IPMI aptaujātāju instanču skaits.
StartJavaPollers	Iepriekš atzaroto Java aptaujātāju instanču skaits.
StartODBCPollers	Iepriekš atzaroto ODBC aptaujātāju instanču skaits.
StartPingers	Iepriekš atzaroto ICMP pingotāju instanču skaits.
StartPollersUnreachable	Iepriekš atzaroto aptaujātāju instanču skaits nesasniedzamiem hostiem (ieskaitot IPMI un Java).
StartPollers	Iepriekš atzaroto aptaujātāju instanču skaits.
StartPreprocessors	Iepriekš palaisto priekšapstrādes darbinieku instanču skaits.
StartSNMPPollers	Iepriekš atzaroto asinhrono SNMP aptaujātāju instanču skaits.
StartSNMPTrapper	Ja iestatīts uz 1, tiks palaists SNMP trapper process.
StartTrappers	Iepriekš atzaroto trapperu instanču skaits.
StartVMwareCollectors	Iepriekš atzaroto VMware savācēju instanču skaits.
StatsAllowedIP	Ar komatiem atdalīts ārējo Zabbix instanču IP adrešu saraksts, pēc izvēles CIDR apzīmējumā, vai DNS nosaukumi. Statistikas pieprasījums tiks pieņemts tikai no šeit norādītajām adresēm.
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri, aģentu, tīmekļa servisu, kā arī SNMP pārbaudēm (izņemot SNMP `walk[OID]` un `get[OID]` vienumus).
TLSAccept	Kādus ienākošos savienojumus pieņemt no Zabbix servera.
TLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus partnera sertifikāta pārbaudei, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kas satur servera sertifikātu vai sertifikātu ķēdi, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherAll13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru komplektu atlases kritērijus sertifikātu un PSK balstītai šifrēšanai.
TLSCipherCert	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru komplektu atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherCert13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru komplektu atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru komplektu atlases kritērijus PSK balstītai šifrēšanai.
TLSCipherPSK13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru komplektu atlases kritērijus PSK balstītai šifrēšanai.
TLSConnect	Kā starpniekserverim jāpieslēdzas Zabbix serverim.
TLSCRLFile	Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kas satur starpniekservera privāto atslēgu, ko izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSListen	Kontrolē TLS trappera ligzdā.
TLSPSKFile	Pilns ceļš uz failu, kas satur starpniekservera iepriekš koplietoto atslēgu, ko izmanto šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Atļautais servera sertifikāta izdevējs.
TLSServerCertSubject	Atļautais servera sertifikāta subjekts.
TmpDir	Pagaidu direktorija.
TrapperTimeout	Norāda noildzi sekundēs šādiem gadījumiem: - konfigurācijas datu saņemšanai no Zabbix servera; - globāla skripta izpildei vai attālās komandas izpildei.
UnavailableDelay	Cik bieži hosts tiek pārbaudīts pieejamībai nepieejamības periodā.
UnreachableDelay	Cik bieži hosts tiek pārbaudīts pieejamībai nesasniedzamības periodā.
UnreachablePeriod	Pēc cik sekundēm nesasniedzamības hosts tiek uzskatīts par nepieejamu.
User	Samazināt privilēģijas līdz konkrētam, sistēmā esošam lietotājam.
Vault	Vault nodrošinātājs.
VaultDBPath	Atrašanās vieta, no kuras datubāzes akreditācijas dati jāiegūst pēc atslēgām.
VaultPrefix	Pielāgots prefikss vault ceļam vai vaicājumam.
VaultTLSCertFile	SSL sertifikāta faila nosaukums, ko izmanto klienta autentifikācijai.
VaultTLSKeyFile	SSL privātās atslēgas faila nosaukums, ko izmanto klienta autentifikācijai.
VaultToken	HashiCorp vault autentifikācijas marķieris.
VaultURL	Vault servera HTTP[S] URL.
VMwareCacheSize	Koplietojamās atmiņas lielums VMware datu glabāšanai.

Parametrs	Apraksts
VMwareFrequency	Aizture sekundēs starp datu iegūšanu no viena VMware servisa.
VMwarePerfFrequency	Aizture sekundēs starp veiktspējas skaitītāju statistikas iegūšanu no viena VMware servisa.
VMwareTimeout	Maksimālais sekunžu skaits, cik ilgi VMware savācējs gaidīs atbildi no VMware servisa.
WebDriverURL	WebDriver saskarnes HTTP[S] URL.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

AllowRoot

Atļaut starpniekserverim darboties kā 'root'. Ja atspējots un starpniekserveri palaiž 'root', starpniekserveris mēģinās pārslēgties uz lietotāju 'zabbix'. Nav ietekmes, ja tas ir palaists ar parastu lietotāju.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

AllowUnsupportedDBVersions

Atļaut starpniekserverim darboties ar neatbalstītām datubāzes versijām.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

CacheSize

Konfigurācijas kešatmiņas lielums baitos. Koplietojamās atmiņas lielums hostu un vienumu datu glabāšanai.

Noklusējums: 32M
 Diapazons: 128K-64G

ConfigFrequency

Šis parametrs ir **novecojis** (tā vietā izmantojiet ProxyConfigFrequency).
Cik bieži starpniekserveris saņem konfigurācijas datus no Zabbix servera sekundēs.
Aktīvā starpniekservera parametrs. Ignorēts pasīvajiem starpniekserveriem (skatiet parametru ProxyMode).

Noklusējums: 3600
 Diapazons: 1-604800

DataSenderFrequency

Starpniekserveris nosūtīs savāktos datus serverim ik pēc N sekundēm. Ņemiet vērā, ka aktīvs starpniekserveris joprojām aptaujās Zabbix serveri katru sekundi, lai saņemtu attālo komandu uzdevumus.
Aktīva starpniekservera parametrs. Ignorē pasīviem starpniekserveriem (skatiet ProxyMode parametru).

Noklusējums: 1
 Diapazons: 1-3600

DBHost

Datubāzes hosta nosaukums.
MySQL gadījumā `localhost` vai tukša virkne nozīmē, ka tiks izmantota ligzda. PostgreSQL gadījumā tikai tukša virkne nozīmē mēģinājumu izmantot ligzdu.

Noklusējums: `localhost`

DBName

Datubāzes nosaukums vai ceļš uz datubāzes failu priekš SQLite3 (Zabbix vairāku procesu arhitektūra neļauj izmantot in-memory database, piemēram, `:memory:`, `file::memory:?cache=shared` vai `file:memdb1?mode=memory&cache=shared`).
Brīdinājums: Nemēģiniet izmantot to pašu datubāzi, ko izmanto Zabbix serveris.

Obligāts: Jā

DBPassword

Datubāzes parole. Komentējiet šo rindu, ja parole netiek izmantota. Ignorēts SQLite gadījumā.

DBPort

Datubāzes ports, ja netiek izmantota lokālā ligzda.[1] Tiek ignorēts SQLite gadījumā.

Noklusējuma vērtība MySQL: 3306

Noklusējuma vērtība PostgreSQL: 5432

Diapazons: 1024-65535

DBSchema

Datubāzes shēmas nosaukums. Izmanto PostgreSQL.

DBSocket

Ceļš uz MySQL ligzdas failu.[1]
Datu bāzes ports, ja netiek izmantota lokālā ligzda. Ignorēts SQLite gadījumā.

Noklusējums: 3306

DBUser

Datubāzes lietotājs. Ignorēts SQLite gadījumā.

DBTLSConnect

Šīs opcijas iestatīšana piespiež izmantot TLS savienojumu ar datubāzi:
required - izveidot savienojumu, izmantojot TLS
verify_ca - izveidot savienojumu, izmantojot TLS, un pārbaudīt sertifikātu
verify_full - izveidot savienojumu, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, vai datubāzes identitāte, kas norādīta DBHost, atbilst tās sertifikātam
MySQL, sākot no versijas 5.7.11, un PostgreSQL atbalsta šādas vērtības: ”required”, ”verify”, ”verify_full”.
MariaDB, sākot no versijas 10.2.6, atbalsta vērtības ”required” un ”verify_full”.
Pēc noklusējuma nav iestatīta neviena opcija, un darbība ir atkarīga no datubāzes konfigurācijas.

DBTLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus datubāzes sertifikāta verifikācijai.

Obligāts: nē (jā, ja DBTLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)

DBTLSCertFile

Pilns ceļš uz failu, kas satur Zabbix starpniekservera sertifikātu autentificēšanai datubāzē.

DBTLSKeyFile

Pilns ceļš uz failu, kas satur privāto atslēgu autentificēšanai datubāzē.

DBTLSCipher

Šifrēšanas šifru saraksts, ko Zabbix starpniekserveris atļauj TLS protokoliem līdz TLS v1.2 ieskaitot. Atbalstīts tikai MySQL.

DBTLSCipher13

Šifrēšanas šifru kopu saraksts, ko Zabbix starpniekserveris atļauj TLS v1.3 protokolam. Atbalstīts tikai MySQL, sākot ar versiju 8.0.16.

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - informācija par kļūdām;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

EnableRemoteCommands

Vai ir atļautas attālinātās komandas no Zabbix serveris.

Noklusējums: 0
 Vērtības: 0 - nav atļauts; 1 - atļauts

ExternalScripts

Ārējo skriptu atrašanās vieta (atkarīgs no `datadir` kompilēšanas laika instalācijas mainīgā).

Noklusējums: `/usr/local/share/zabbix/externalscripts`

Fping6Location

fping6 atrašanās vieta. Pārliecinieties, ka fping6 binārajam failam ir root īpašnieks un iestatīts SUID karodziņš. Atstājiet tukšu (”Fping6Location=”), ja jūsu fping utilīta spēj apstrādāt IPv6 adreses.

Noklusējums: `/usr/sbin/fping6`

FpingLocation

fping atrašanās vieta. Pārliecinieties, ka fping binārajam failam ir root īpašnieks un iestatīts SUID karodziņš.

Noklusējums: `/usr/sbin/fping`

HistoryCacheSize

Vēstures kešatmiņas lielums baitos. Koplietojamās atmiņas lielums vēstures datu glabāšanai.

Noklusējums: `16M`
 Diapazons: 128K-16G

HistoryIndexCacheSize

Vēstures indeksa kešatmiņas izmērs baitos. Koplietojamās atmiņas izmērs vēstures kešatmiņā glabāto vēstures datu indeksēšanai. Indeksa kešatmiņas izmēram ir nepieciešami aptuveni 100 baiti, lai kešatmiņā saglabātu vienu vienumu.

Noklusējums: `4M`
 Diapazons: 128K-16G

Hostname

Unikāls, reģistrjutīgs starpniekservera nosaukums. Pārliecinieties, ka starpniekservera nosaukums ir zināms serverim.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128

Noklusējums: iestata HostnameItem

HostnameItem

Vienums, ko izmanto Hostname iestatīšanai, ja tas nav definēts (tas tiks izpildīts uz starpniekservera līdzīgi kā uz aģenta). Tiek ignorēts, ja Hostname ir iestatīts.
Neatbalsta UserParameters, veiktspējas skaitītājus vai aizstājvārdus, bet atbalsta system.run[].

Noklusējums: system.hostname

HousekeepingFrequency

Cik bieži Zabbix veiks datubāzes tīrīšanas procedūru (stundās). Datubāzes tīrīšana noņem novecojušu informāciju no datubāzes.
Piezīme: Lai samazinātu slodzi starpniekservera palaišanas laikā, datubāzes tīrīšana tiek atlikta uz 30 minūtēm pēc starpniekservera starta. Tādējādi, ja HousekeepingFrequency ir 1, pati pirmā datubāzes tīrīšanas procedūra pēc starpniekservera starta tiks izpildīta pēc 30 minūtēm un pēc tam atkārtosies katru stundu.
Automātisko datubāzes tīrīšanu ir iespējams atspējot, iestatot HousekeepingFrequency uz 0. Šajā gadījumā datubāzes tīrīšanas procedūru var palaist tikai ar *housekeeper_execute* izpildlaika vadības opciju.
Skatiet arī izpildlaika vadības opcijas un informāciju par datubāzes tīrīšanas procedūru.

Noklusējums: 1
 Diapazons: 0-24

Iekļaušana

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
Lai norādītajā direktorijā iekļautu tikai atbilstošos failus, šablonu atbilstībai tiek atbalstīta aizstājējzīme zvaigznīte.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

JavaGateway

Zabbix Java gateway IP adrese (vai resursdatora nosaukums). Nepieciešams tikai tad, ja ir palaisti Java aptaujātāji.

JavaGatewayPort

Ports, kuru klausās Zabbix Java gateway.

Noklusējums: 10052
 Diapazons: 1024-32767

ListenBacklog

Maksimālais gaidošo savienojumu skaits TCP rindā.
Noklusējuma vērtība ir stingri kodēta konstante, kas ir atkarīga no sistēmas.
Arī maksimālā atbalstītā vērtība ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz "implementācijas noteiktajam maksimumam".

Noklusējums: `SOMAXCONN`
 Diapazons: 0 - INT_MAX

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās trapper jāgaida savienojumi.
Ja šis parametrs nav norādīts, trapper gaidīs savienojumus visās tīkla saskarnēs.

Noklusējums: `0.0.0.0`

ListenPort

Trapper klausīšanās ports.

Noklusējums: 10051
 Diapazons: 1024-32767

LoadModule

Starpniekservera palaišanas laikā ielādējamais modulis. Moduļi tiek izmantoti, lai paplašinātu starpniekservera funkcionalitāti. Modulim jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma. Ja iepriekš norādītais ceļš ir absolūts (sākas ar '/'), tad LoadModulePath tiek ignorēts.
Formāti:
LoadModule=<module.so>
LoadModule=<p atļauts iekļaut vairākus LoadModule parametrus.

LoadModulePath

Pilns ceļš uz starpniekservera moduļu atrašanās vietu. Noklusējuma vērtība ir atkarīga no kompilācijas opcijām.

LogFile

Žurnālfaila nosaukums.

Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfailu rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija kāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar LogFile parametru;
system - rakstīt žurnālu syslog;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

LogSlowQueries

Cik ilgi datubāzes vaicājums drīkst ilgt, pirms tas tiek reģistrēts žurnālā (milisekundēs).
0 - nereģistrēt lēnus vaicājumus.
Šī opcija tiek iespējota, sākot ar DebugLevel=3.

Noklusējums: 0
 Diapazons: 0-3600000

MaxConcurrentChecksPerPoller

Maksimālais asinhrono pārbaužu skaits, ko vienlaikus var izpildīt katrs HTTP aģenta aptaujātājs, aģenta aptaujātājs vai SNMP aptaujātājs. Skatiet StartHTTPAgentPollers, StartAgentPollers un StartSNMPPollers.

Noklusējums: 1000
 Diapazons: 1-1000

PidFile

PID faila nosaukums.

Noklusējums: `/tmp/zabbix_proxy.pid`

ProxyBufferMode

Norāda vēstures, tīkla atklāšanas un automātiskās reģistrācijas datu glabāšanas mehānismu: *disk* - dati tiek glabāti datubāzē un augšupielādēti no datubāzes; *memory* - dati tiek glabāti atmiņā un augšupielādēti no atmiņas. Ja buferī beidzas atmiņa, vecie dati tiks atmesti. Izslēgšanas laikā buferis tiek atmests. *hybrid* - starpniekservera buferis parasti darbojas kā *memory* režīmā, līdz tam beidzas atmiņa vai vecākais ieraksts pārsniedz konfigurēto vecumu. Ja tas notiek, buferis tiek izrakstīts datubāzē un darbojas kā *disk* režīmā, līdz visi dati ir augšupielādēti, pēc tam tas atkal sāk darboties atmiņā. Izslēgšanas laikā atmiņas buferis tiek izrakstīts datubāzē.

Skatiet arī: Starpniekservera atmiņas buferis.

Noklusējums: `disk`
 Vērtības: disk; memory; hybrid

ProxyConfigFrequency

Cik bieži starpniekserveris saņem konfigurācijas datus no Zabbix servera sekundēs.
Aktīvā starpniekservera parametrs. Ignorēts pasīvajiem starpniekserveriem (skatiet parametru ProxyMode).

Noklusējums: 10
 Diapazons: 1-604800

ProxyLocalBuffer

Starpniekserveris glabās datus lokāli N stundas, pat ja dati jau ir sinhronizēti ar serveri.
Šo parametru var izmantot, ja lokālos datus izmantos trešo pušu lietotnes.

Noklusējums: 0
 Diapazons: 0-720

ProxyMemoryBufferAge

Maksimālais datu vecums starpniekservera atmiņas buferī sekundēs. Ja tas ir iespējots (nav nulle) un ieraksti starpniekservera atmiņas buferī ir vecāki, tas piespiež starpniekservera buferi pārslēgties uz datubāzes režīmu, līdz visi ieraksti ir augšupielādēti serverī. Šim parametram jābūt mazākam vai vienādam ar parametru ProxyOfflineBuffer.

Noklusējums: 0
 Diapazons: 0;600-864000

ProxyMemoryBufferSize

Koplietojamās atmiņas kešatmiņas lielums savāktajiem vēstures, atklāšanas un automātiskās reģistrācijas datiem baitos. Ja iespējots (nav nulle), starpniekserveris glabās vēstures, atklāšanas un automātiskās reģistrācijas datus atmiņā, ja vien kešatmiņa nav pilna vai saglabātie ieraksti nav vecāki par noteikto ProxyMemoryBufferAge. Šo parametru nevar izmantot kopā ar parametru ProxyLocalBuffer.

Noklusējums: 0
 Diapazons: 0;128K-2G

ProxyMode

Starpniekservera darbības režīms.
0 - starpniekserveris aktīvajā režīmā
1 - starpniekserveris pasīvajā režīmā
Ņemiet vērā, ka, izmantojot aktīvu starpniekserveri, (sensitīvi) starpniekservera konfigurācijas dati var kļūt pieejami pusēm, kurām ir piekļuve Zabbix servera trapper portam. Tas ir iespējams, jo jebkurš var izlikties par aktīvu starpniekserveri un pieprasīt konfigurācijas datus; autentifikācija nenotiek.

Noklusējums: 0
 Diapazons: 0-1

ProxyOfflineBuffer

Starpniekserveris saglabās datus N stundas, ja nebūs savienojuma ar Zabbix serveri.
Vecāki dati tiks zaudēti.

Noklusējums: 1
 Diapazons: 1-720

Serveris

Ja ProxyMode ir iestatīts uz *aktīvo režīmu*:
Zabbix servera IP adrese vai DNS nosaukums (adrese:ports) vai klasteris (adrese:ports;adrese2:ports), no kura saņemt konfigurācijas datus un uz kuru sūtīt datus.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
Klastera mezgli jāatdala ar semikolu.

Ja ProxyMode ir iestatīts uz *pasīvo režīmu*:
Ar komatiem atdalīts Zabbix servera IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajām adresēm. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām.
'::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligāts: jā

SNMPTrapperFile

Pagaidu fails, ko izmanto datu nodošanai no SNMP trap dēmona uz starpniekserveri.
Tam jābūt tādam pašam kā zabbix_trap_receiver.pl vai SNMPTT konfigurācijas failā.

Noklusējums: `/tmp/zabbix_traps.tmp`

SocketDir

Direktorija, kurā glabāt IPC ligzdas, ko izmanto iekšējie Zabbix servisi.

Noklusējums: `/tmp`

SourceIP

Avota IP adrese šādiem savienojumiem:

- izejošajiem savienojumiem ar Zabbix serveri
- bezaģenta savienojumiem (VMware, SSH, JMX, SNMP, Telnet un vienkāršajām pārbaudēm)
- HTTP aģenta savienojumiem
- skripta vienuma JavaScript HTTP pieprasījumiem
- priekšapstrādes JavaScript HTTP pieprasījumiem
- savienojumiem ar Vault

SSHKeyLocation

Publisko un privāto atslēgu atrašanās vieta SSH pārbaudēm un darbībām.

SSLCertLocation

SSL klienta sertifikātu failu atrašanās vieta klienta autentifikācijai.
Šis parametrs tiek izmantots tikai tīmekļa uzraudzībā.

SSLKeyLocation

SSL privātās atslēgas failu atrašanās vieta klienta autentifikācijai.
Šis parametrs tiek izmantots tikai tīmekļa uzraudzībā.

SSLCALocation

Sertifikātu iestādes (CA) failu atrašanās vieta SSL servera sertifikāta verifikācijai.
Ņemiet vērā, ka šī parametra vērtība tiks iestatīta kā libcurl opcija CURLOPT_CAPATH. Libcurl versijām pirms 7.42.0 tas darbojas tikai tad, ja libcurl tika kompilēts lietošanai ar OpenSSL. Plašāku informāciju skatiet cURL tīmekļa lapā.
Šis parametrs tiek izmantots tīmekļa uzraudzībā un SMTP autentifikācijā.

StartAgentPollers

Iepriekš atzaroto Zabbix aģents aptaujātāju skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartBrowserPollers

Pārlūka vienumu aptaujātāju iepriekš atzaroto instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartDBSyncers

Iepriekš atzaroto vēstures sinhronizētāju instanču skaits.
Piezīme: Esiet piesardzīgi, mainot šo vērtību — tās palielināšana var nodarīt vairāk ļauna nekā laba.

Noklusējums: 4
 Diapazons: 1-100

StartDiscoverers

Iepriekš atzaroto atklāšanas darbinieku instanču skaits.

Noklusējums: 5
 Diapazons: 0-1000

StartHTTPAgentPollers

Iepriekš atzaroto HTTP aģenta aptaujātāju instanču skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartHTTPPollers

Iepriekš atzaroto HTTP aptaujātāju instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartIPMIPollers

Iepriekš atzaroto IPMI aptaujātāju instanču skaits.

Noklusējums: 0
 Diapazons: 0-1000

StartJavaPollers

Iepriekš atzaroto Java aptaujātāju instanču skaits.

Noklusējums: 0
 Diapazons: 0-1000

StartODBCPollers

Iepriekš atzaroto ODBC aptaujātāju instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartPingers

Iepriekš atzaroto ICMP pingeru instanču skaits.

Noklusējums: 1
 Diapazons: 0-1000

StartPollersUnreachable

Iepriekš atzaroto nepieejamu hostu aptaujātāju instanču skaits (ieskaitot IPMI un Java). Ja ir palaisti parastie, IPMI vai Java aptaujātāji, jādarbojas vismaz vienam nepieejamu hostu aptaujātājam.

Noklusējums: 1
 Diapazons: 0-1000

StartPollers

Iepriekš atzaroto aptaujātāju instanču skaits.

Noklusējums: 5
 Diapazons: 0-1000

StartPreprocessors

Iepriekš palaisto priekšapstrādes darbinieku pavedienu instanču skaits jāiestata ne mazāks par pieejamo CPU kodolu skaitu. Jāiestata vairāk darbinieku, ja priekšapstrāde nav CPU ierobežota un tajā ir daudz tīkla pieprasījumu.

Noklusējums: 16
 Diapazons: 1-1000

StartSNMPPollers

Iepriekš atzaroto SNMP aptaujātāju instanču skaits. Skatiet MaxConcurrentChecksPerPoller.

Noklusējums: 1
 Diapazons: 0-1000

StartSNMPTrapper

Ja iestatīts uz 1, tiks palaists SNMP trapper process.

Noklusējums: 0
 Diapazons: 0-1

StartTrappers

Iepriekš atzaroto trapperu instanču skaits.
Trapperi pieņem ienākošos savienojumus no Zabbix sender un aktīvajiem aģentiem.

Noklusējums: 5
 Diapazons: 0-1000

StartVMwareCollectors

Iepriekš atzaroto VMware kolektora instanču skaits.

Noklusējums: 0
 Diapazons: 0-250

StatsAllowedIP

Ar komatiem atdalīts ārējo Zabbix instanču IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Statistikas pieprasījums tiks pieņemts tikai no šeit norādītajām adresēm. Ja šis parametrs nav iestatīts, netiks pieņemts neviens statistikas pieprasījums.
Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi.

Piemērs:

```
StatsAllowedIP=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Noildze

Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix serveris, aģents, tīmekļa pakalpojumu un mantotajām SNMP pārbaudēm (viens OID numurs vai virkne).

Šis parametrs nosaka ilgumu dažādām saziņas darbībām:

- Attālo komandu izpilde Zabbix aģents
- SSH / Telnet komandu izpilde
- Vienumu pārplānošana, kad IPMI saskarne kļūst nepieejama
- Atbildes nosūtīšana Zabbix serveris, ja datu apmaiņa neizdodas tiesību vai šifrēšanas problēmu dēļ
- IPC asinhrono ligzdu un izpildlaika vadības opciju noildze
- Asinhronie aptaujātāja DNS pieprasījumi
- Atbilde aktīvās pārbaudes sirdspukstam
- Zabbix aģents datu (vērtību) iegūšana no aktīvajiem aģentiem
- Datu iegūšana no Zabbix sender
- Aktīvo pārbaužu saraksta nosūtīšana Zabbix aģents

Šī noildze **netiks** izmantota tām pārbaudēm, kurām lietotāja saskarne ir konfigurēti elastīgās noildzes iestatījumi (globālā, starpniekserveris vai katra vienums līmenī).
Piemēram, SNMP `walk[OID]` un `get[OID]` vienumi izmanto lietotāja saskarne konfigurēto noildzi; mantotās SNMP pārbaudes joprojām izmanto serveris noildzes vērtību.

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Kādus ienākošos savienojumus pieņemt no Zabbix servera. Izmanto pasīvam starpniekserverim, ignorē aktīvam starpniekserverim. Var norādīt vairākas vērtības, atdalot tās ar komatu:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
psk - pieņemt savienojumus ar TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieņemt savienojumus ar TLS un sertifikātu

Obligāts: jā pasīvam starpniekserverim, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta verifikācijai un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur starpniekservera sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCipherAll

GnuTLS prioritātes virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 protokolā. Pārraksta noklusējuma šifru kopas atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātos balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Šifrēšanas virknes iestatījums OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 protokolā. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātos balstītai šifrēšanai.

TLSCipherPSK

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrkomplektu atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Kā starpniekserverim vajadzētu izveidot savienojumu ar Zabbix serveri. Tiek izmantots aktīvam starpniekserverim, pasīvam starpniekserverim tiek ignorēts. Var norādīt tikai vienu vērtību:
unencrypted - izveidot savienojumu bez šifrēšanas (noklusējums)
psk - izveidot savienojumu, izmantojot TLS un iepriekš koplietotu atslēgu (PSK)
cert - izveidot savienojumu, izmantojot TLS un sertifikātu

Obligāts: jā aktīvam starpniekserverim, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCRLFile

Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur starpniekservera privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSListen

Kontrolē TLS trapper ligzdā.

Atbalstītās vērtības:

- required - pieņemt tikai TLS savienojumus

TLSPSKFile

Pilns ceļš uz failu, kas satur starpniekservera iepriekš koplietoto atslēgu, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Atļautais servera sertifikāta izdevējs.

TLSServerCertSubject

Atļautais servera sertifikāta subjekts.

TmpDir

Pagaidu direktorijs.

Noklusējums: `/tmp`

TrapperTimeout

Norāda noildzi sekundēs šādām darbībām:
 - konfigurācijas datu saņemšanai no Zabbix servera;
 - globālo skriptu izpildei vai attālo komandu izpildei.

Noklusējums: 300
 Diapazons: 1-300

UnavailableDelay

Cik bieži hosts tiek pārbaudīts attiecībā uz pieejamību nepieejamības periodā sekundēs.

Noklusējums: 60
 Diapazons: 1-3600

UnreachableDelay

Cik bieži hosts tiek pārbaudīts attiecībā uz pieejamību nesasniedzamības periodā sekundēs.

Noklusējums: 15
 Diapazons: 1-3600

UnreachablePeriod

Pēc cik sekundēm nesasniedzamības uzskatīt hostu par nepieejamu.

Noklusējums: 45
 Diapazons: 1-3600

Lietotājs

Samazināt privilēģijas līdz konkrētam, sistēmā esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā 'root' un AllowRoot ir atspējots.

Noklusējums: `zabbix`

Glabātuve

Glabātuves nodrošinātājs:
HashiCorp - HashiCorp KV Secrets Engine versija 2
CyberArk - CyberArk Central Credential Provider
Jāsakrīt ar glabātuves nodrošinātāju, kas iestatīts lietotāja saskarnē.

Noklusējums: `HashiCorp`

VaultDBPath

Vault ceļš vai vaicājums atkarībā no Vault, no kurienes datubāzes akreditācijas dati tiks iegūti pēc atslēgām.

Atslēgas, kas tiek izmantotas HashiCorp, ir 'password' un 'username'.

Ceļa piemērs ar VaultPrefix=/v1/secret/data/zabbix/:

```
database
```

Ceļa piemērs bez VaultPrefix:

```
secret/zabbix/database
```

Atslēgas, kas tiek izmantotas CyberArk, ir 'Content' un 'UserName'.

Piemērs:

```
AppID=zabbix_server&Query=Safe=passwordSafe;Object=zabbix_proxy_database
```

Šo opciju var izmantot tikai tad, ja DBUser un DBPassword nav norādīti.

VaultPrefix

Pielāgots prefikss Vault ceļam vai vaicājumam atkarībā no Vault. Ja tas nav norādīts, tiks izmantotas vispiemērotākās noklusējuma vērtības.
Ņemiet vērā, ka HashiCorp gadījumā pēc montēšanas punkta automātiski tiek pievienots `data`, ja VaultPrefix nav norādīts.
Ņemiet vērā, ka HashiCorp gadījumā pēc montēšanas punkta automātiski tiek pievienots `data`, ja VaultPrefix nav norādīts.

Hashicorp prefiksa piemērs:

```
v1/secret/data/zabbix/
```

Cyberark prefiksa piemērs:

```
/AIMWebService/api/Accounts?
```

VaultTLSCertFile

SSL sertifikāta faila nosaukums, kas tiek izmantots klienta autentifikācijai. Sertifikāta failam jābūt PEM1 formātā.
Ja sertifikāta fails satur arī privāto atslēgu, atstājiet SSL atslēgas faila lauku tukšu.
Direktorija, kurā atrodas šis fails, ir norādīta konfigurācijas parametrā SSLCertLocation.
Šo opciju var nenorādīt, taču tā ir ieteicama CyberArkCCP glabātuvei.

VaultTLSKeyFile

SSL privātās atslēgas faila nosaukums, kas tiek izmantots klienta autentifikācijai. Privātās atslēgas failam jābūt PEM1 formātā.
Direktorijs, kurā atrodas šis fails, ir norādīts ar konfigurācijas parametru SSLKeyLocation.
Šo opciju var izlaist, taču tā ir ieteicama CyberArkCCP glabātuvei.

VaultToken

HashiCorp vault autentifikācijas marķieris, kam jābūt ģenerētam tikai Zabbix starpniekserverim ar tikai lasīšanas atļauju ceļam, kas norādīts neobligātajā VaultDBPath konfigurācijas parametrā.
Kļūda rodas, ja VaultToken un vides mainīgais VAULT_TOKEN ir definēti vienlaikus.

Obligāts: Jā, ja Vault ir iestatīts uz *HashiCorp*; pretējā gadījumā nē

VaultURL

Glabātuves servera HTTP[S] URL. Ja SSLCALocation nav norādīts, tiks izmantots sistēmas mēroga CA sertifikātu direktorijs.

Noklusējums: `https://127.0.0.1:8200`

VMwareCacheSize

Koplietojamās atmiņas lielums VMware datu glabāšanai.
VMware iekšējo pārbaudi zabbix[vmware,buffer,...] var izmantot, lai uzraudzītu VMware kešatmiņas lietojumu (skatiet Iekšējās pārbaudes).
Ņemiet vērā, ka koplietojamā atmiņa netiek piešķirta, ja nav konfigurēts palaist nevienu VMware kolektora instanci.

Noklusējums: 8M
 Diapazons: 256K-2G

VMwareFrequency

Aizture sekundēs starp datu iegūšanu no viena VMware pakalpojuma.
Šī aizture jāiestata atbilstoši mazākajam atjaunināšanas intervālam jebkuram VMware uzraudzības vienumam.

Noklusējums: 60
 Diapazons: 10-86400

VMwarePerfFrequency

Aizkave sekundēs starp veiktspējas skaitītāju statistikas izgūšanu no viena VMware pakalpojuma.
Šī aizkave jāiestata atbilstoši mazākajam atjaunināšanas intervālam jebkuram VMware uzraudzības vienumam, kas izmanto VMware veiktspējas skaitītājus.

Noklusējums: 60
 Diapazons: 10-86400

VMwareTimeout

Maksimālais sekunžu skaits, cik ilgi VMware kolektors gaidīs atbildi no VMware servisa (vCenter vai ESX hipervizora).

Noklusējums: 10
 Diapazons: 1-300

WebDriverURL

WebDriver saskarnes HTTP[S] URL.

Piemērs (izmantošanai ar Selenium WebDriver standalone serveri):

```
WebDriverURL=http://localhost:4444
```

Piezīmes

[1] DBSocket un DBPort ir savstarpēji izslēdzoši starpniekservera konfigurācijā. Norādiet tikai vienu vai atstājiet abus nedefinētus.

3 Zabbix aģents (UNIX)

Pārskats

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Zabbix aģents konfigurācijas fails (zabbix_agentd.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parameter	Description
Alias	Iestata aizstājvārdu vienuma atslēgai.
AllowKey	Atļauj izpildīt tās vienumu atslēgas, kas atbilst šablonam.
AllowRoot	Atļauj aģentam darboties kā 'root'.
BufferSend	Neglabāt datus buferī ilgāk par N sekundēm.
BufferSize	Maksimālais vērtību skaits atmiņas buferī.
DebugLevel	Atkļūdošanas līmenis.
DenyKey	Aizliedz izpildīt tās vienumu atslēgas, kas atbilst šablonam.
EnableRemoteCommands	Vai ir atļautas attālās komandas no Zabbix servera.
HeartbeatFrequency	Sirdspuksta ziņojumu biežums sekundēs.
HostInterface	Neobligāts parametrs, kas nosaka hosta saskarni.
HostInterfaceItem	Neobligāts parametrs, kas nosaka vienumu, ko izmanto hosta saskarnes iegūšanai.
HostMetadata	Neobligāts parametrs, kas nosaka hosta metadatus.
HostMetadataItem	Neobligāts parametrs, kas nosaka Zabbix aģenta vienumu, ko izmanto hosta metadatu iegūšanai.
Hostname	Neobligāts parametrs, kas nosaka hosta nosaukumu.
HostnameItem	Neobligāts parametrs, kas nosaka Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai.
Include	Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
ListenBacklog	Maksimālais gaidošo savienojumu skaits TCP rindā.
ListenIP	Ar komatiem atdalīts IP adrešu saraksts, kurās aģentam jāgaida savienojumi.
ListenPort	Aģents šajā portā gaidīs savienojumus no servera.
LoadModule	Modulis, ko ielādēt aģenta palaišanas laikā.
LoadModulePath	Pilns ceļš uz aģenta moduļu atrašanās vietu.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais izmērs.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogType	Žurnāla izvades tips.
MaxLinesPerSecond	Maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveri vai starpniekserveri, apstrādājot 'log' un 'logrt' aktīvās pārbaudes.
PidFile	PID faila nosaukums.

Parameter	Description
RefreshActiveChecks	Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts.
Server	Ar komatiem atdalīts IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru un Zabbix starpniekserveru DNS nosaukumi.
ServerActive	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras iegūt aktīvās pārbaudes.
SourceIP	Avota IP adrese.
StartAgents	Iepriekš izveidoto zabbix_agentd instanču skaits, kas apstrādā pasīvās pārbaudes.
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherAll13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherCert	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherCert13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK	GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.
TLSCipherPSK13	Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.
TLSConnect	Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim.
TLSCRLFile	Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kas satur aģenta privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSPSKFile	Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu un tiek izmantots šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Iepriekš koplietotās atslēgas identitātes virkne, kas tiek izmantota šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Atļautais servera (starpniekservera) sertifikāta izdevējs.
TLSServerCertSubject	Atļautais servera (starpniekservera) sertifikāta subjekts.
UnsafeUserParameters	Atļauj visus simbolus nodot lietotāja definēto parametru argumentos.
User	Samazināt privilēģijas līdz konkrētam, sistēmā esošam lietotājam.
UserParameter	Lietotāja definēts parametrs uzraudzībai.
UserParameterDir	Noklusējuma meklēšanas ceļš UserParameter komandām.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta tikai konfigurācijas failus UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

Alias

Iestata aizstājvārdu vienuma atslēgai. To var izmantot, lai aizstātu garu un sarežģītu vienuma atslēgu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki *Alias* parametri. Vairāki parametri ar vienu un to pašu *Alias* atslēgu nav atļauti.
 Dažādas *Alias* atslēgas var atsaukties uz vienu un to pašu vienuma atslēgu.
 Aizstājvārdus var izmantot parametrā *HostMetadataItem*, bet ne parametrā *HostnameItem*.

1. piemērs: lietotāja 'zabbix' ID iegūšana.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Tagad saīsināto atslēgu **zabbix.userid** var izmantot datu iegūšanai.

2. piemērs: CPU noslodzes iegūšana ar noklusējuma un pielāgotiem parametriem.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Tas ļauj izmantot atslēgu **cpu.util**, lai iegūtu CPU noslodzes procentuālo vērtību ar noklusējuma parametriem, kā arī izmantot **cpu.util[all, idle, avg15]**, lai iegūtu konkrētus datus par CPU noslodzi.

3. piemērs: vairāku zemā līmeņa atklāšanas noteikumu palaišana, kas apstrādā vienus un tos pašus atklāšanas vienumus.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Tagad ir iespējams iestatīt vairākus atklāšanas noteikumus, izmantojot **vfs.fs.discovery** ar atšķirīgiem parametriem katram noteikumam, piemēram, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]** utt.

AllowKey

Atļaut izpildīt tos vienumu atslēgu modeļus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Var definēt vairākus atslēgu atbilstības noteikumus kombinācijā ar DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

AllowRoot

Atļauj aģentam darboties kā 'root'. Ja atspējots un aģents tiek palaists ar 'root', aģents mēģinās pārslēgties uz lietotāju 'zabbix'. Nav ietekmes, ja tas tiek palaists ar parastu lietotāju.

Noklusējums: 0
 Vērtības: 0 - neatļaut; 1 - atļaut

BufferSend

Neglabāt datus buferī ilgāk par N sekundēm.

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Maksimālais vērtību skaits atmiņas buferī. Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns.

Noklusējums: 100
 Diapazons: 2-65535

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

DenyKey

Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Vairāki atslēgu atbilstības noteikumi var tikt definēti kombinācijā ar AllowKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

EnableRemoteCommands

Vai ir atļautas attālās komandas no Zabbix servera. Šis parametrs ir **novecojis**, tā vietā izmantojiet AllowKey=system.run[*] vai DenyKey=system.run[*].
Tas ir iekšējs aizstājvārds parametriem AllowKey/DenyKey atkarībā no vērtības:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

HeartbeatFrequency

Sirdspukstu ziņojumu biežums sekundēs. Izmanto aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Neobligāts parametrs, kas nosaka hosta saskarni (IP adresi vai DNS nosaukumu), ko izmanto hosta automātiskās reģistrācijas procesa laikā. Šī vērtība tiks izmantota, lai aizpildītu saskarni jaunizveidotajā hostā, un ļauj skaidri konfigurēt vai nu IP, vai DNS adresi. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Ja nav definēts, vērtība tiks iegūta no HostInterfaceItem.

Ja vērtība pārsniedz 255 rakstzīmju ierobežojumu, aģents izdos kļūdu un nestartēsies.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Neobligāts parametrs, kas definē vienumu, ko izmanto, lai noteiktu hosta saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas procesa laikā. Šī vērtība tiek izmantota tikai tad, ja HostInterface nav definēts. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 255 rakstzīmju ierobežojumu.

Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

HostMetadata

Neobligāts parametrs, kas nosaka metadatus, ko izmanto, lai identificētu vai atšķirtu hostu hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadata ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Ja tas nav definēts, vērtība tiks iegūta no HostMetadataItem.

Aģents ziņos par kļūdu un nestartēs, ja norādītā vērtība pārsniedz 2034 baitu ierobežojumu vai nav UTF-8 virkne. Ja parametrs paredz IP adresi vai DNS nosaukumu, tiks noraidītas un atzīmētas kā nederīgas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Diapazons: 0-2034 baiti

HostMetadataItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta metadatu iegūšanai. Šī opcija tiek izmantota tikai tad, ja HostMetadata nav definēts.

HostMetadataItem vērtība tiek iegūta katrā automātiskās reģistrācijas mēģinājumā un tiek izmantota tikai hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadataItem ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai jābūt UTF-8 virknei, pretējā gadījumā tā tiks ignorēta. Ja parametrs paredz IP adresi vai DNS nosaukumu, vērtības, kas ir derīgas UTF-8, bet nav derīgas IP adreses vai DNS nosaukumi, arī tiks noraidītas un atzītas par nederīgām.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindas pārtraukuma.

Hostname

Ar komatiem atdalītu, unikālu, reģistrjutīgu hostu nosaukumu saraksts. Nepieciešams aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā konfigurēts serverī. Vērtība tiek iegūta no HostnameItem, ja tā nav definēta.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: Iestata HostnameItem

HostnameItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai. Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts. Lietotāja parametri vai aizstājvārdi netiek atbalstīti, bet system.run[] vienums tiek atbalstīts neatkarīgi no AllowKey/DenyKey vērtībām.

Noklusējums: `system.hostname`

Iekļaut

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā. Lai norādītajā direktorijā iekļautu tikai atbilstošos failus, šablonu atbilstībai tiek atbalstīta aizstājējzīme zvaigznīte.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenBacklog

Maksimālais gaidošo savienojumu skaits TCP rindā.
Noklusējuma vērtība ir stingri kodēta konstante, kas ir atkarīga no sistēmas.
Arī maksimālā atbalstītā vērtība ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz "implementācijas noteiktajam maksimumam".

Noklusējums: SOMAXCONN
 Diapazons: 0 - INT_MAX

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās aģents klausīsies.

Noklusējums: 0.0.0.0

ListenPort

Aģents klausīsies šajā portā savienojumiem no servera.

Noklusējums: 10050
 Diapazons: 1024-32767

LoadModule

Modulis, ko ielādēt aģents palaišanas laikā. Moduļi tiek izmantoti, lai paplašinātu aģents funkcionalitāti. Modulim jāatrodas direktorijā, kas norādīta ar LoadModulePath, vai arī ceļam jābūt norādītam pirms moduļa nosaukuma. Ja iepriekš norādītais ceļš ir absolūts (sākas ar '/'), tad LoadModulePath tiek ignorēts.
Formāti:
LoadModule=<module.so>
LoadModule=<path/module.so><b
atļauts iekļaut vairākus LoadModule parametrus.

LoadModulePath

Pilns ceļš uz aģenta moduļu atrašanās vietu. Noklusējuma vērtība ir atkarīga no kompilēšanas opcijām.

LogFile

Žurnālfaila nosaukums.

Obligāti: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija jebkāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus. Komandas tiks reģistrētas tikai tad, ja tās tiks izpildītas attālināti. Žurnāla ieraksti netiks izveidoti, ja system.run[] tiek palaists lokāli ar parametriem HostMetadataItem, HostInterfaceItem vai HostnameItem.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar parametru LogFile;
system - rakstīt žurnālu syslog;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: file

MaxLinesPerSecond

Maksimālais jauno rindu skaits, ko aģents nosūtīs sekundē uz Zabbix serveri vai starpniekserveri, apstrādājot aktīvās pārbaudes log un logrt. Norādīto vērtību pārrakstīs parametrs maxlines, kas norādīts log vai logrt vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu, nekā iestatīts *MaxLinesPerSecond*, lai žurnāla vienumos meklētu nepieciešamo virkni.

Noklusējums: 20
 Diapazons: 1-1000

PidFile

PID faila nosaukums.

Noklusējums: /tmp/zabbix_agentd.pid

RefreshActiveChecks

Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts, sekundēs. Ņemiet vērā, ka pēc neveiksmīga mēģinājuma atsvaidzināt aktīvās pārbaudes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Server

Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru un Zabbix starpniekserveru DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka "IPv4-compatible IPv6 addresses" (prefikss 0000::/96) tiek atbalstītas, bet RFC4291 tās ir atzītas par novecojušām. Atstarpes ir atļautas.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligāts: jā, ja StartAgents nav tieši iestatīts uz 0

ServerActive

Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes. Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums un neobligāts ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairākas servera vai starpniekserveru grupas dalībnieku adreses, kas atdalītas ar semikolu. Var norādīt vairākus Zabbix serverus/klasterus un Zabbix starpniekserverus, atdalot tos ar komatu. Ja netiek izmantotas starpniekserveru grupas, no katra Zabbix servera/klastera nevajadzētu norādīt vairāk par vienu Zabbix starpniekserveri. Ja ir norādīts Zabbix starpniekserveris, tad šī starpniekservera Zabbix serveri/klasteri nevajadzētu norādīt.
Var norādīt vairākas ar komatiem atdalītas adreses, lai paralēli izmantotu vairākus neatkarīgus Zabbix serverus. Atstarpes ir atļautas.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
IPv6 adresēm jābūt ietvertām kvadrātiekavās, ja šim hostam ir norādīts ports. Ja ports nav norādīts, kvadrātiekavas IPv6 adresēm nav obligātas.
Ja šis parametrs nav norādīts, aktīvās pārbaudes ir atspējotas.

Piemērs Zabbix starpniekserverim:

```
ServerActive=127.0.0.1:10051
```

Piemērs Zabbix starpniekserveru grupai:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Piemērs vairākiem serveriem:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Piemērs augstai pieejamībai:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Piemērs augstai pieejamībai ar diviem klasteriem un vienu serveri:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Avota IP adrese:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (web.page.get, net.tcp.port u.c.).

StartAgents

Iepriekš atzaroto zabbix_agentd instanču skaits, kas apstrādā pasīvās pārbaudes. Ja iestatīts uz 0, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.

Noklusējums: 10
 Diapazons: 0-100

Timeout

Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- atbildes gaidīšanu no Zabbix servera;
- pieprasījumu sūtīšanu uz Zabbix serveri, tostarp vienumu konfigurācijas pieprasījumus un vienumu datus aktīvajās pārbaudēs;
- žurnāla datu iegūšanu, izmantojot logfile;
- heartbeat ziņojumu sūtīšanu;
- maksimālo ilgumu `vfs.*` pārbaudēm;
- izmantošanu Zabbix aģenta moduļos;

- izmantošanu kā rezerves vērtību gadījumos, kad serveris vai starpniekserveris, kas ir vecāks par versiju 7.0, nosūta pārbaudes bez noildzes.

Šī noildze **netiks** izmantota tām aģenta pārbaudēm, kurām lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālā, starpniekservera vai atsevišķa vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Kādus ienākošos savienojumus pieņemt. Izmanto pasīvajām pārbaudēm. Var norādīt vairākas vērtības, atdalot tās ar komatu:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
psk - pieņemt savienojumus ar TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieņemt savienojumus ar TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai ar Zabbix komponentiem.

TLSCipherAll

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātos un PSK balstītai šifrēšanai.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherAll13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3. Pārraksta noklusējuma šifrkomplektu atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherCert

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātos balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-RSA:+RSA:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

TLSCipherCert13

Šifrēšanas algoritmu virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 protokolā. Pārraksta noklusējuma šifrēšanas algoritmu kopas atlases kritērijus sertifikātos balstītai šifrēšanai.

TLSCipherPSK

GnuTLS prioritāšu virkne vai OpenSSL (TLS 1.2) šifru virkne. Pārraksta noklusējuma šifru kopas atlases kritērijus PSK balstītai šifrēšanai.

Piemērs GnuTLS:

```
NONE:+VERS-TLS1.2:+ECDHE-PSK:+PSK:+AES-128-GCM:+AES-128-CBC:+AEAD:+SHA256:+SHA1:+CURVE-ALL:+COMP-NULL:+SIG
```

Piemērs OpenSSL:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 protokolā. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim. Izmanto aktīvajām pārbaudēm. Var norādīt tikai vienu vērtību:
unencrypted — pieslēgties bez šifrēšanas (noklusējums)
psk — pieslēgties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK)
cert — pieslēgties, izmantojot TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCRLFile

Pilns ceļš uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur aģents privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Atļautā servera (starpniekservera) sertifikāta izdevējs.

TLSServerCertSubject

Atļautā servera (starpniekservera) sertifikāta subjekts.

UnsafeUserParameters

Atļaut visas rakstzīmes nodot argumentos lietotāja definētiem parametriem. Šādas rakstzīmes nav atļautas: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Papildus tam nav atļautas arī jaunās rindas rakstzīmes.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

Lietotājs

Samazināt privilēģijas līdz konkrētam, sistēmā esošam lietotājam.
Stājas spēkā tikai tad, ja tiek palaists kā 'root' un AllowRoot ir atspējots.

Noklusējums: `zabbix`

UserParameter

Lietotāja definēts parametrs uzraudzībai. Var būt vairāki lietotāja definēti parametri.
Formāts: UserParameter=<key>,<shell command>
Ņemiet vērā, ka čaulas komanda nedrīkst atgriezt tukšu virkni vai tikai EOL. Čaulas komandām var būt relatīvi ceļi, ja ir norādīts parametrs UserParameterDir.

Piemērs:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Noklusējuma meklēšanas ceļš UserParameter komandām. Ja tiek izmantots, aģents pirms komandas izpildes nomainīs savu darba direktoriju uz šeit norādīto. Tādējādi UserParameter komandām pilna ceļa vietā var izmantot relatīvu prefiksu `./`.
Ir atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=/opt/myscripts
```

Skatiet arī

1. Atšķirības Zabbix aģenta konfigurācijā aktīvajām un pasīvajām pārbaudēm, sākot no versijas 2.0.0

4 Zabbix aģents 2 (UNIX)

Pārskats

Zabbix aģents 2 ir jaunas paaudzes Zabbix aģents, un to var izmantot Zabbix aģenta vietā.

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Zabbix aģenta 2 konfigurācijas fails (zabbix_agent2.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parameter	Description
Alias	Iestata vienuma atslēgai aizstājvārdu.
AllowKey	Atļaut to vienumu atslēgu izpildi, kas atbilst paraugam.
BufferSend	Neglabāt datus buferī ilgāk par N sekundēm.
BufferSize	Maksimālais vērtību skaits atmiņas buferī.
ControlSocket	Vadības ligzda, ko izmanto izpildlaika komandu sūtīšanai ar opciju '-R'.
DebugLevel	Atkļūdošanas līmenis.
DenyKey	Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam.
EnablePersistentBuffer	Iespējot lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
ForceActiveChecksOnStart	Veikt aktīvās pārbaudes uzreiz pēc restartēšanas pirmajai saņemtajai konfigurācijai.
HeartbeatFrequency	Sirdspuksta ziņojumu biežums sekundēs.
HostInterface	Neobligāts parametrs, kas definē hosta saskarni.
HostInterfaceItem	Neobligāts parametrs, kas definē vienumu, ko izmanto hosta saskarnes iegūšanai.
HostMetadata	Neobligāts parametrs, kas definē hosta metadatus.
HostMetadataItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta metadatu iegūšanai.
Hostname	Neobligāts parametrs, kas definē hosta nosaukumu.
HostnameItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai.
Include	Konfigurācijas failā var iekļaut atsevišķus failus vai visus failus direktorijā.
ListenIP	Ar komatiem atdalītu IP adrešu saraksts, kurās aģentam jāgaida savienojumi.
ListenPort	Aģents gaidīs savienojumus no servera šajā portā.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais izmērs.
LogType	Žurnāla izvades tips.
PersistentBufferFile	Fails, kurā Zabbix aģentam 2 jāglabā SQLite datubāze.
PersistentBufferPeriod	Laika periods, cik ilgi dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri.
PidFile	PID faila nosaukums.
Plugins.<PluginName>.System.Capacity	Vienlaikus izpildāmo pārbaužu limits katram spraudnim.
Plugins.Log.MaxLinesPerSecond	Maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs Zabbix serverim vai starpniekserverim, apstrādājot 'log' un 'logrt' aktīvās pārbaudes.
Plugins.SystemRun.LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
PluginSocket	Ceļš uz UNIX ligzdu ielādējamo spraudņu saziņai.
PluginTimeout	Noildze savienojumiem ar ielādējamiem spraudņiem sekundēs.
RefreshActiveChecks	Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts.
Server	Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru un Zabbix starpniekserveru DNS nosaukumi.
ServerActive	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras iegūt aktīvās pārbaudes.
SourceIP	Avota IP adrese.
StatusPort	Ja iestatīts, aģents gaidīs HTTP statusa pieprasījumus šajā portā (http://localhost:<port>/status).
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus partnera sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll	OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherAll13	OpenSSL šifru virkne (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherCert	OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātiem balstītai šifrēšanai.

Parameter	Description
TLSCipherCert13	OpenSSL šifru virkne (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK	OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.
TLSCipherPSK13	OpenSSL šifru virkne (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.
TLSConnect	Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim.
TLSCRLFile	Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kas satur aģenta privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSPSKFile	Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu un tiek izmantots šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Iepriekš koplietotās atslēgas identitātes virkne, kas tiek izmantota šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Atļautais servera (starpniekservera) sertifikāta izdevējs.
TLSServerCertSubject	Atļautais servera (starpniekservera) sertifikāta subjekts.
UnsafeUserParameters	Atļaut visām rakstzīmēm tikt nodotām lietotāja definēto parametru argumentos.
UserParameter	Lietotāja definēts parametrs uzraudzībai.
UserParameterDir	Noklusējuma meklēšanas ceļš UserParameter komandām.

Visi parametri ir neobligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

Alias

Iestata aizstājvārdu vienuma atslēgai. To var izmantot, lai aizstātu garu un sarežģītu vienuma atslēgu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki *Alias* parametri. Vairāki parametri ar vienu un to pašu *Alias* atslēgu nav atļauti.
 Dažādas *Alias* atslēgas var atsaukties uz vienu un to pašu vienuma atslēgu.
 Aizstājvārdus var izmantot parametrā *HostMetadataItem*, bet ne parametrā *HostnameItem*.

1. piemērs: lietotāja 'zabbix' ID iegūšana.

```
Alias=zabbix.userid:vfs.file.regexp[/etc/passwd,"^zabbix:.:([0-9]+)",,,,\1]
```

Tagad saīsināto atslēgu **zabbix.userid** var izmantot datu iegūšanai.

2. piemērs: CPU noslodzes iegūšana ar noklusējuma un pielāgotiem parametriem.

```
Alias=cpu.util:system.cpu.util
Alias=cpu.util[*]:system.cpu.util[*]
```

Tas ļauj izmantot atslēgu **cpu.util**, lai iegūtu CPU noslodzes procentuālo vērtību ar noklusējuma parametriem, kā arī izmantot **cpu.util[all, idle, avg15]**, lai iegūtu konkrētus datus par CPU noslodzi.

3. piemērs: vairāku zemā līmeņa atklāšanas noteikumu palaišana, kas apstrādā vienus un tos pašus atklāšanas vienumus.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Tagad ir iespējams iestatīt vairākus atklāšanas noteikumus, izmantojot **vfs.fs.discovery** ar dažādiem parametriem katram noteikumam, piemēram, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]** utt.

AllowKey

Atļaut izpildīt to vienumu atslēgas, kas atbilst šablonam. Atslēgas šablons ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Var definēt vairākus atslēgu atbilstības noteikumus kombinācijā ar DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

BufferSend

Laika intervāls sekundēs, kas nosaka, cik bieži vērtības tiek sūtītas no bufera uz Zabbix serveri. Ņemiet vērā, ka, ja buferis ir pilns, dati tiks nosūtīti agrāk.

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Maksimālais vērtību skaits atmiņas buferī. Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns. Šis parametrs jāizmanto tikai tad, ja pastāvīgais buferis ir atspējots (*EnablePersistentBuffer=0*).

Noklusējums: 1000
 Diapazons: 2-65535

ControlSocket

Vadības ligzda, ko izmanto izpildlaika komandu nosūtīšanai ar opciju '-R'.

Noklusējums: `/tmp/agent.sock`

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

DenyKey

Aizliegt to vienumu atslēgu izpildi, kas atbilst šablonam. Atslēgas šablons ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Vairāki atslēgu atbilstības noteikumi var tikt definēti kombinācijā ar AllowKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: aģenta pārbaužu ierobežošana.

EnablePersistentBuffer

Iespējo lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem. Ja pastāvīgā krātuve ir atspējota, tiks izmantots atmiņas buferis.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

ForceActiveChecksOnStart

Veikt aktīvās pārbaudes uzreiz pēc restartēšanas pirmajai saņemtajai konfigurācijai. Pieejams arī kā katra spraudņa konfigurācijas parametrs, piemēram: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

HeartbeatFrequency

Sirdspukstu ziņojumu biežums sekundēs. Izmanto aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Neobligāts parametrs, kas nosaka hosta saskarni (IP adresi vai DNS nosaukumu), ko izmanto hosta automātiskās reģistrācijas procesa laikā. Šī vērtība tiks izmantota, lai aizpildītu saskarni jaunizveidotajā hostā, un ļauj skaidri konfigurēt vai nu IP, vai DNS adresi. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Ja nav definēts, vērtība tiks iegūta no HostInterfaceItem.

Aģents izdos kļūdu un nestartēsies, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Neobligāts parametrs, kas definē vienumu, ko izmanto, lai noteiktu hosta saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas procesa laikā.
Šī vērtība tiek izmantota tikai tad, ja HostInterface nav definēts.
Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 255 rakstzīmju ierobežojumu.

Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

HostMetadata

Neobligāts parametrs, kas nosaka metadatus, ko izmanto, lai identificētu vai atšķirtu hostu hosta automātiskās reģistrācijas procesā. HostMetadata ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Ja tas nav definēts, vērtība tiks iegūta no HostMetadataItem.

Aģents ziņos par kļūdu un nestartēsies, ja norādītā vērtība pārsniedz 2034 baitu ierobežojumu vai ir virkne, kas nav UTF-8 formātā. Ja parametrs paredz IP adresi vai DNS nosaukumu, tiks noraidītas un atzīmētas kā nederīgas arī vērtības, kas ir derīgā UTF-8 formātā, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Diapazons: 0-2034 baiti

HostMetadataItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto, lai iegūtu hosta metadatus. Šī opcija tiek izmantota tikai tad, ja HostMetadata nav definēts.

HostMetadataItem vērtība tiek iegūta katrā automātiskās reģistrācijas mēģinājumā un tiek izmantota tikai hosta automātiskās reģistrācijas procesā. HostMetadataItem ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai jābūt UTF-8 virknei, pretējā gadījumā tā tiks ignorēta. Ja parametrs sagaida IP adresi vai DNS nosaukumu, vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi, arī tiks noraidītas un atzīmētas kā nederīgas.

Vairākrindu metadati netiek atbalstīti — izvade tiks saīsināta pie pirmās rindas pārtraukuma.

Hostname

Ar komatiem atdalītu, unikālu, reģistrjutīgu hostu nosaukumu saraksts. Nepieciešams aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie konfigurēti serverī. Ja vērtība nav definēta, tā tiek iegūta no HostnameItem.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: Iestata HostnameItem

HostnameItem

Neobligāts parametrs, kas nosaka vienumu, ko izmanto hosta nosaukuma iegūšanai. Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts. Lietotāja parametri vai aizstājvārdi netiek atbalstīti, bet vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey vērtībām.

Noklusējums: `system.hostname`

Iekļaut

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā. Instalēšanas laikā Zabbix izveidos iekļaušanas direktoriju mapē /usr/local/etc, ja vien tas nav mainīts kompilēšanas laikā. Ceļš var būt relatīvs attiecībā pret *zabbix_agent2.conf* faila atrašanās vietu.
Lai iekļautu tikai atbilstošos failus norādītajā direktorijā, tiek atbalstīta aizstājējzīme zvaigznīte rakstu atbilstības noteikšanai.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās aģents klausās. Pirmā IP adrese tiek nosūtīta Zabbix serveris, ja tiek veidots savienojums ar to, lai iegūtu aktīvo pārbaužu sarakstu.

Noklusējums: `0.0.0.0`

ListenPort

Aģents klausīsies šajā portā savienojumiem no servera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Žurnālfaila nosaukums.

Noklusējums: `/tmp/zabbix_agent2.log`
 Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija kāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar parametru LogFile;
system - rakstīt žurnālu syslog;
console - rakstīt žurnālu standarta izvadē

Noklusējums: `file`

PersistentBufferFile

Fails, kur Zabbix aģents 2 glabā SQLite datubāzi. Jānorāda pilns faila nosaukums. Šis parametrs tiek izmantots tikai tad, ja ir iespējota noturīgā buferatmiņa (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Laika periods, cik ilgi dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri. Vecāki dati tiks zaudēti. Žurnāla dati tiks saglabāti. Šis parametrs tiek izmantots tikai tad, ja noturīgais buferis ir iespējots (*EnablePersistentBuffer=1*).

Noklusējums: 1h
 Diapazons: 1m-365d

PidFile

PID faila nosaukums.

Noklusējums: `/tmp/zabbix_agent2.pid`

Plugins.<PluginName>.System.Capacity

Vienlaikus izpildāmo pārbaužu limits katram `<PluginName>` spraudnim.

Noklusējums: 1000 Diapazons: 1-1000

Plugins.Log.MaxLinesPerSecond

Maksimālais jauno rindu skaits, ko aģents nosūtīs sekundē uz Zabbix serveri vai starpniekserveri, apstrādājot aktīvās pārbaudes 'log' un 'logrt'. Norādīto vērtību pārrakstīs parametrs 'maxlines', kas norādīts 'log' un 'logrt' vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu, nekā iestatīts *MaxLinesPerSecond*, lai log vienumos meklētu nepieciešamo virkni.

Noklusējums: 20
 Diapazons: 1-1000

Plugins.SystemRun.LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus. Komandas tiks reģistrētas tikai tad, ja tās tiks izpildītas attālināti. Žurnāla ieraksti netiks izveidoti, ja system.run[] tiek palaists lokāli ar parametriem HostMetadataItem, HostInterfaceItem vai HostnameItem.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

PluginSocket

Ceļš uz UNIX ligzdu ielādējamo spraudņu saziņai.

Noklusējums: `/tmp/agent.plugin.sock`

PluginTimeout

Savienojumu ar ielādējamiem spraudņiem noildze sekundēs.

Noklusējums: `Timeout`
 Diapazons: 1-30

RefreshActiveChecks

Cik bieži aktīvo pārbaužu saraksts tiek atsvaidzināts, sekundēs. Ņemiet vērā, ka pēc neveiksmīga mēģinājuma atsvaidzināt aktīvās pārbaudes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Server

Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot,

lai atļautu jebkuru IPv4 adresi. Atstarpes ir atļautas. Ja šis parametrs nav norādīts, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes. Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums un neobligāts ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairākas servera vai starpniekservera grupas dalībnieku adreses, kas atdalītas ar semikolu. Var norādīt vairākus Zabbix serverus/klasterus un Zabbix starpniekserverus, atdalot tos ar komatu. Ja netiek izmantotas starpniekserveru grupas, no katra Zabbix servera/klastera nevajadzētu norādīt vairāk par vienu Zabbix starpniekserveri. Ja ir norādīts Zabbix starpniekserveris, tad šī starpniekservera Zabbix serveri/klasteri nevajadzētu norādīt.
Lai paralēli izmantotu vairākus neatkarīgus Zabbix serverus, var norādīt vairākas ar komatiem atdalītas adreses. Atstarpes ir atļautas.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
IPv6 adresēm jābūt ietvertām kvadrātiekavās, ja šim hostam ir norādīts ports. Ja ports nav norādīts, kvadrātiekavas IPv6 adresēm nav obligātas.
Ja šis parametrs nav norādīts, aktīvās pārbaudes tiek atspējotas.

Piemērs Zabbix starpniekserverim:

```
ServerActive=127.0.0.1:10051
```

Piemērs Zabbix starpniekserveru grupai:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Piemērs vairākiem serveriem:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Augstas pieejamības piemērs:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Augstas pieejamības piemērs ar diviem klasteriem un vienu serveri:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Avota IP adrese:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri.
- savienojumu izveidei, izpildot dažus vienumus (web.page.get, net.tcp.port u.c.).

StatusPort

Ja iestatīts, aģents klausīsies šajā portā HTTP statusa pieprasījumiem (http://localhost:<port>/status).

Diapazons: 1024-32767

Noildze

Norāda, cik ilgi (sekundēs) gaidīt savienojuma izveidi un datu apmaiņu ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- atbildes gaidīšanu no Zabbix servera;
- pieprasījumu sūtīšanu uz Zabbix serveri, tostarp vienumu konfigurācijas pieprasījumus un vienumu datus aktīvajās pārbaudēs;
- žurnāla datu iegūšanu, izmantojot logfile;
- heartbeat ziņojumu sūtīšanu;
- maksimālo `vfs.*` pārbaužu ilgumu;
- izmantošanu kā rezerves vērtību gadījumos, kad serveris vai starpniekserveris, kas ir vecāks par versiju 7.0, nosūta pārbaudes bez noildzēm.

Šī noildze **netiks** izmantota tām aģenta pārbaudēm, kurām lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālā, starpniekservera vai atsevišķa vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Ienākošie savienojumi, kurus pieņemt. Izmanto pasīvajām pārbaudēm. Var norādīt vairākas vērtības, atdalot tās ar komatu:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
psk - pieņemt savienojumus ar TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieņemt savienojumus ar TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai ar Zabbix komponentiem.

TLSCipherAll

OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus sertifikātu un uz PSK balstītai šifrēšanai.

Piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128:kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherAll13

OpenSSL šifru virkne (TLS 1.3) priekš OpenSSL 1.1.1 vai jaunākas versijas. Pārraksta noklusējuma šifru kopas atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert

OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus uz sertifikātu balstītai šifrēšanai.

Piemērs:

```
EECDH+aRSA+AES128:RSA+aRSA+AES128
```

Ņemiet vērā, ka šo parametru nevar izmantot kopā ar `TLSAccept=cert,psk`; sertifikāta savienojumam (`TLSConnect=cert`) tā vietā izmantojiet TLSCipherAll.

TLSCipherCert13

OpenSSL šifru virkne (TLS 1.3) priekš OpenSSL 1.1.1 vai jaunākas versijas. Pārraksta noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem balstītai šifrēšanai.

Ņemiet vērā, ka šo parametru nevar izmantot kopā ar `TLSAccept=cert,psk`; sertifikāta savienojumam (`TLSConnect=cert`) tā vietā izmantojiet TLSCipherAll13.

TLSCipherPSK

OpenSSL šifru virkne (TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
kECDHEPSK+AES128:kPSK+AES128
```

TLSCipherPSK13

OpenSSL šifru virkne (TLS 1.3) OpenSSL 1.1.1 vai jaunākai versijai. Pārraksta noklusējuma šifru kopas atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Kā aģentam vajadzētu izveidot savienojumu ar Zabbix serveri vai starpniekserveri. Tiek izmantots aktīvajām pārbaudēm. Var norādīt tikai vienu vērtību:
unencrypted - izveidot savienojumu bez šifrēšanas (noklusējums)
psk - izveidot savienojumu, izmantojot TLS un iepriekš koplietotu atslēgu (PSK)
cert - izveidot savienojumu, izmantojot TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCRLFile

Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur aģents privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Atļautā servera (starpniekservera) sertifikāta izdevējs.

TLSServerCertSubject

Atļautā servera (starpniekservera) sertifikāta tēma.

UnsafeUserParameters

Atļaut visas rakstzīmes nodot argumentos lietotāja definētiem parametriem. Šādas rakstzīmes nav atļautas: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Papildus tam nav atļautas arī jaunās rindas rakstzīmes.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Lietotāja definēts parametrs uzraudzībai. Var būt vairāki lietotāja definēti parametri.
Formāts: UserParameter=<key>,<shell command>
Ņemiet vērā, ka čaulas komanda nedrīkst atgriezt tukšu virkni vai tikai EOL. Čaulas komandām var būt relatīvi ceļi, ja ir norādīts parametrs UserParameterDir.

Piemērs:

```
UserParameter=system.test,who|wc -l
UserParameter=check_cpu,./custom_script.sh
```

UserParameterDir

Noklusējuma meklēšanas ceļš UserParameter komandām. Ja tiek izmantots, aģents pirms komandas izpildes nomainīs savu darba direktoriju uz šeit norādīto. Tādējādi UserParameter komandām var izmantot relatīvu prefiksu `./` pilna ceļa vietā.
Ir atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=/opt/myscripts
```

5 Zabbix aģents (Windows)

Pārskats

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Windows Zabbix aģents konfigurācijas fails (zabbix_agentd.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parameter	Description
Alias	Iestata aizstājvārdu vienuma atslēgai.
AllowKey	Atļaut to vienumu atslēgu izpildi, kas atbilst šablonam.
BufferSend	Neglabāt datus buferī ilgāk par N sekundēm.
BufferSize	Maksimālais vērtību skaits atmiņas buferī.
DebugLevel	Atkļūdošanas līmenis.
DenyKey	Aizliegt to vienumu atslēgu izpildi, kas atbilst šablonam.
EnableRemoteCommands	Vai ir atļautas attālās komandas no Zabbix servera.
HeartbeatFrequency	Sirdspukstu ziņojumu biežums sekundēs.
HostInterface	Neobligāts parametrs, kas definē hosta saskarni.
HostInterfaceItem	Neobligāts parametrs, kas definē vienumu, ko izmanto hosta saskarnes iegūšanai.
HostMetadata	Neobligāts parametrs, kas definē hosta metadatus.
HostMetadataItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta metadatu iegūšanai.
Hostname	Neobligāts parametrs, kas definē hosta nosaukumu.
HostnameItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai.

Parameter	Description
Include	Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
ListenBacklog	Maksimālais gaidošo savienojumu skaits TCP rindā.
ListenIP	Ar komatiem atdalīts IP adrešu saraksts, kurās aģentam jāgaida savienojumi.
ListenPort	Aģents gaidīs savienojumus no servera šajā portā.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais izmērs.
LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
LogType	Žurnāla izvades tips.
MaxLinesPerSecond	Maksimālais jauno rindu skaits, ko aģents nosūtīs sekundē uz Zabbix serveri vai starpniekserveri, apstrādājot aktīvās pārbaudes 'log' un 'logrt'.
PerfCounter	Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
PerfCounterEn	Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Salīdzinājumā ar PerfCounter, perfcounter ceļiem jābūt angļu valodā.
RefreshActiveChecks	Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts.
Server	Ar komatiem atdalīts Zabbix serveru un Zabbix starpniekserveru IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi.
ServerActive	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras iegūt aktīvās pārbaudes.
SourceIP	Avota IP adrese.
StartAgents	Iepriekš atzaroto zabbix_agentd instanču skaits, kas apstrādā pasīvās pārbaudes.
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSConnect	Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim.
TLSCRLFile	Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kas satur aģenta privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSPSKFile	Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu un tiek izmantots šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Iepriekš koplietotās atslēgas identitātes virkne, kas tiek izmantota šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Atļautais servera (starpniekservera) sertifikāta izdevējs.
TLSServerCertSubject	Atļautais servera (starpniekservera) sertifikāta subjekts.
UnsafeUserParameters	Atļaut visām rakstzīmēm tikt nodotām lietotāja definēto parametru argumentos.
UserParameter	Lietotāja definēts parametrs uzraudzībai.
UserParameterDir	Noklusējuma meklēšanas ceļš UserParameter komandām.

Visi parametri ir neobligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo dēmona noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta tikai konfigurācijas failus UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

Alias

Iestata aizstājvārdu vienuma atslēgai. To var izmantot, lai aizstātu garu un sarežģītu vienuma atslēgu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki *Alias* parametri. Vairāki parametri ar vienu un to pašu *Alias* atslēgu nav atļauti.
 Dažādas *Alias* atslēgas var atsaukties uz vienu un to pašu vienuma atslēgu.
 Aizstājvārdus var izmantot parametrā *HostMetadataItem*, bet ne *HostnameItem* vai *PerfCounter* parametrā.

1. piemērs: peidžinga faila izmantojuma iegūšana procentos no servera.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Tagad saīsināto atslēgu **pg_usage** var izmantot datu iegūšanai.

2. piemērs: CPU slodzes iegūšana ar noklusējuma un pielāgotiem parametriem.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Tas ļauj izmantot atslēgu **cpu.load**, lai iegūtu CPU slodzi ar noklusējuma parametriem, kā arī izmantot **cpu.load[percpu,avg15]**, lai iegūtu konkrētus datus par CPU slodzi.

3. piemērs: vairāku zemā līmeņa atklāšanas noteikumu palaišana, kas apstrādā vienus un tos pašus atklāšanas vienumus.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Tagad ir iespējams iestatīt vairākus atklāšanas noteikumus, izmantojot **vfs.fs.discovery** ar dažādiem parametriem katram noteikumam, piemēram, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]** utt.

AllowKey

Atļaut izpildīt tos vienumu atslēgu modeļus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Var definēt vairākus atslēgu atbilstības noteikumus kombinācijā ar DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

BufferSend

Neglabāt datus buferī ilgāk par N sekundēm.

Noklusējums: 5
 Diapazons: 1-3600

BufferSize

Maksimālais vērtību skaits atmiņas buferī. Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns.

Noklusējums: 100
 Diapazons: 2-65535

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

DenyKey

Aizliegt to vienumu atslēgu izpildi, kas atbilst šablonam. Atslēgas šablons ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Var definēt vairākus atslēgu atbilstības noteikumus kombinācijā ar AllowKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: aģenta pārbaužu ierobežošana.

EnableRemoteCommands

Vai ir atļautas attālās komandas no Zabbix serveris. Šis parametrs ir **novecojis**, tā vietā izmantojiet AllowKey=system.run[*] vai DenyKey=system.run[*].
Tas ir iekšējs aizstājvārds parametriem AllowKey/DenyKey atkarībā no vērtības:
0 - DenyKey=system.run[*]
1 - AllowKey=system.run[*]

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

HeartbeatFrequency

Sirdspukstu ziņojumu biežums sekundēs. Izmanto aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Neobligāts parametrs, kas nosaka hosta saskarni (IP adresi vai DNS nosaukumu), ko izmanto hosta automātiskās reģistrācijas procesā. Šī vērtība tiks izmantota, lai aizpildītu saskarni jaunizveidotajā hostā, un ļauj skaidri konfigurēt vai nu IP, vai DNS adresi. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Ja nav definēts, vērtība tiks iegūta no HostInterfaceItem.

Aģents izdos kļūdu un nestartēsies, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Neobligāts parametrs, kas definē vienumu, ko izmanto, lai noteiktu hosta saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas procesa laikā.
Šī vērtība tiek izmantota tikai tad, ja HostInterface nav definēts.
Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 255 rakstzīmju ierobežojumu.

Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

HostMetadata

Neobligāts parametrs, kas definē metadatus, ko izmanto, lai identificētu vai atšķirtu hostu hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadata ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Ja tas nav definēts, vērtība tiks iegūta no HostMetadataItem.

Aģents ziņos par kļūdu un nestartēsies, ja norādītā vērtība pārsniedz 2034 baitu ierobežojumu vai nav UTF-8 virkne. Ja parametrs sagaida IP adresi vai DNS nosaukumu, tiks noraidītas un atzīmētas kā nederīgas arī vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Diapazons: 0-2034 baiti

HostMetadataItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto, lai iegūtu hosta metadatus. Šī opcija tiek izmantota tikai tad, ja HostMetadata nav definēts.

HostMetadataItem vērtība tiek iegūta katrā automātiskās reģistrācijas mēģinājumā un tiek izmantota tikai hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadataItem ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Tiek atbalstīti lietotāja parametri, veiktspējas skaitītāji un aizstājvārdi. Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents žurnālā ierakstīs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai jābūt UTF-8 virknei, pretējā gadījumā tā tiks ignorēta. Ja parametrs paredz IP adresi vai DNS nosaukumu, vērtības, kas ir derīgas UTF-8 virknes, bet nav derīgas IP adreses vai DNS nosaukumi, arī tiks noraidītas un atzītas par nederīgām.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Hostname

Ar komatiem atdalītu, unikālu, reģistrjutīgu hostu nosaukumu saraksts. Nepieciešams aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie konfigurēti serverī. Ja vērtība nav definēta, tā tiek iegūta no HostnameItem.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: iestata HostnameItem

HostnameItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai. Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts. Lietotāja parametri, veiktspējas skaitītāji vai aizstājvārdi netiek atbalstīti, bet system.run[] vienums tiek atbalstīts neatkarīgi no AllowKey/DenyKey vērtībām.
Skatiet arī detalizētāku aprakstu.

Noklusējums: `system.hostname`

Iekļaut

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā (pēc noklusējuma tas atrodas `C:\Program Files\Zabbix Agent`, ja Zabbix aģents ir instalēts, izmantojot Windows MSI instalēšanas pakotnes; atrodas instalēšanas laikā norādītajā mapē, ja Zabbix aģents ir instalēts kā zip arhīvs). Visiem iekļautajiem failiem jābūt ar pareizu sintaksi, pretējā gadījumā aģents netiks palaists.
Lai norādītajā direktorijā iekļautu tikai atbilstošos failus, šablonu atbilstībai tiek atbalstīta aizstājējzīme zvaigznīte.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=C:\Program Files\Zabbix Agent\zabbix_agentd.d\*.conf
```

ListenBacklog

Maksimālais gaidošo savienojumu skaits TCP rindā.
Noklusējuma vērtība ir stingri kodēta konstante, kas ir atkarīga no sistēmas.
Arī maksimālā atbalstītā vērtība ir atkarīga no sistēmas; pārāk lielas vērtības var tikt klusi saīsinātas līdz "implementācijas noteiktajam maksimumam".

Noklusējums: `SOMAXCONN`
 Diapazons: 0 - INT_MAX

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās aģents klausīsies.

Noklusējums: `0.0.0.0`

ListenPort

Aģents klausīsies šajā portā savienojumiem no servera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Aģenta žurnālfaila nosaukums.

Noklusējums: `c:\zabbix_agentd.log`
 Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija jebkāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus. Komandas tiks reģistrētas tikai tad, ja tās tiks izpildītas attālināti. Žurnāla ieraksti netiks izveidoti, ja system.run[] tiek palaists lokāli ar parametriem HostMetadataItem, HostInterfaceItem vai HostnameItem.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar parametru LogFile;
system - rakstīt žurnālu Windows notikumu žurnālā;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

MaxLinesPerSecond

Maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs Zabbix serverim vai starpniekserverim, apstrādājot aktīvās pārbaudes `log`, `logrt` un `eventlog`. Norādīto vērtību pārrakstīs parametrs `maxlines`, kas norādīts `log`, `logrt` vai `eventlog` vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jaunu rindu, nekā iestatīts *MaxLinesPerSecond*, lai žurnāla vienumos atrastu nepieciešamo virkni.

Noklusējums: 20
 Diapazons: 1-1000

PerfCounter

Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
Sintakse: <parameter_name>,"<perf_counter_path>",<period>

Piemēram, ja vēlaties saņemt procesora pārtraukumu vidējo skaitu sekundē par pēdējo minūti, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Varat palaist "typeperf -qx", lai iegūtu visu Windows pieejamo veiktspējas skaitītāju sarakstu.

PerfCounterEn

Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Salīdzinot ar PerfCounter, perfcounter ceļiem jābūt angļu valodā. Atbalstīts tikai **Windows Server 2008/Vista** un jaunākās versijās.
Sintakse: <parameter_name>,"<perf_counter_path>",<period>

Piemēram, ja vēlaties saņemt procesora pārtraukumu vidējo skaitu sekundē par pēdējo minūti, varat definēt jaunu parametru ”interrupts” šādi:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultās pēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Angļu virkņu sarakstu var atrast, apskatot šādu reģistra atslēgu: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009`.

RefreshActiveChecks

Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts, sekundēs. Ņemiet vērā, ka pēc neveiksmīga mēģinājuma atsvaidzināt aktīvās pārbaudes nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Server

Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad ’127.0.0.1’, ’::127.0.0.1’, ’::ffff:127.0.0.1’ tiek uzskatītas par vienādām, un ’::/0’ atļaus jebkuru IPv4 vai IPv6 adresi. ’0.0.0.0/0’ var izmantot, lai atļautu jebkuru IPv4 adresi. Ņemiet vērā, ka ”ar IPv4 saderīgas IPv6 adreses” (prefikss 0000::/96) tiek atbalstītas, bet RFC4291 tās ir atzītas par novecojušām. Atstarpes ir atļautas.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligāts: jā, ja StartAgents nav tieši iestatīts uz 0

ServerActive

Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes. Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums un neobligāts ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairākas servera vai starpniekserveru grupas dalībnieku adreses, kas atdalītas ar semikolu. Var norādīt vairākus Zabbix serverus/klasterus un Zabbix starpniekserverus, atdalot tos ar komatu. Ja netiek izmantotas starpniekserveru grupas, no katra Zabbix servera/klastera nevajadzētu norādīt vairāk par vienu Zabbix starpniekserveri. Ja ir norādīts Zabbix starpniekserveris, tad šī starpniekservera Zabbix serveri/klasteri nevajadzētu norādīt.
Lai paralēli izmantotu vairākus neatkarīgus Zabbix serverus, var norādīt vairākas ar komatiem atdalītas adreses. Atstarpes ir atļautas.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
IPv6 adresēm jābūt ietvertām kvadrātiekavās, ja šim hostam ir norādīts ports. Ja ports nav norādīts, kvadrātiekavas IPv6 adresēm nav obligātas.
Ja šis parametrs nav norādīts, aktīvās pārbaudes ir atspējotas.

Piemērs Zabbix starpniekserverim:

```
ServerActive=127.0.0.1:10051
```

Piemērs Zabbix starpniekserveru grupai:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Piemērs vairākiem serveriem:

```
ServerActive=127.0.0.1:20051,zabbix.domain,[::1]:30051,::1,[12fc::1]
```

Piemērs augstai pieejamībai:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Piemērs augstai pieejamībai ar diviem klasteriem un vienu serveri:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

Diapazons: (*)

SourceIP

Avota IP adrese:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (web.page.get, net.tcp.port u.c.).

StartAgents

Iepriekš atzaroto zabbix_agentd instanču skaits, kas apstrādā pasīvās pārbaudes. Ja iestatīts uz 0, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.

Noklusējums: 10
 Diapazons: 0-100 (*)

Timeout

Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- atbildes gaidīšanu no Zabbix servera;
- pieprasījumu sūtīšanu uz Zabbix serveri, tostarp vienumu konfigurācijas pieprasījumus un vienumu datus aktīvajās pārbaudēs;
- žurnāla datu iegūšanu, izmantojot logfile vai Windows notikumu žurnāla uzraudzību;
- heartbeat ziņojumu sūtīšanu;
- maksimālo ilgumu `vfs.*` pārbaudēm;
- izmantošanu Zabbix aģenta moduļos;
- izmantošanu kā rezerves vērtību gadījumos, kad serveris vai starpniekserveris, kas ir vecāks par versiju 7.0, sūta pārbaudes bez noildzēm.

Šī noildze **netiks** izmantota tām aģenta pārbaudēm, kurām lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālajā, starpniekservera vai katra vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Pieņemamie ienākošie savienojumi. Izmanto pasīvajām pārbaudēm. Var norādīt vairākas vērtības, atdalot tās ar komatu:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
psk - pieņemt savienojumus ar TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieņemt savienojumus ar TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus(-u) partnera sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai ar Zabbix komponentiem.

TLSConnect

Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim. Izmanto aktīvajām pārbaudēm. Var norādīt tikai vienu vērtību:
unencrypted - pieslēgties bez šifrēšanas (noklusējums)
psk - pieslēgties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieslēgties, izmantojot TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCRLFile

Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur aģents privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Atļautā servera (starpniekservera) sertifikāta izdevējs.

TLSServerCertSubject

Atļautā servera (starpniekservera) sertifikāta subjekts.

UnsafeUserParameters

Atļaut visas rakstzīmes nodot argumentos lietotāja definētiem parametriem. Šādas rakstzīmes nav atļautas: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Papildus tam nav atļautas arī jaunās rindas rakstzīmes.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Lietotāja definēts parametrs uzraudzībai. Var būt vairāki lietotāja definēti parametri.
Formāts: UserParameter=<key>,<shell command>
Ņemiet vērā, ka čaulas komanda nedrīkst atgriezt tukšu virkni vai tikai EOL. Čaulas komandām var būt relatīvi ceļi, ja ir norādīts parametrs UserParameterDir.

Piemērs:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Noklusējuma meklēšanas ceļš UserParameter komandām. Ja tiek izmantots, aģents pirms komandas izpildes nomainīs savu darba direktoriju uz šeit norādīto. Tādējādi UserParameter komandām pilna ceļa vietā var izmantot relatīvo prefiksu `./`. Ir atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=C:\opt\myscripts
```

Note:
(*) Aktīvo serveru skaitam, kas norādīts ServerActive, pieskaitot pasīvo pārbaužu iepriekš atzaroto instanču skaitu, kas norādīts StartAgents, jābūt mazākam par 64.

6 Zabbix aģents 2 (Windows)

Pārskats

Zabbix aģents 2 ir jaunas paaudzes Zabbix aģents, un to var izmantot Zabbix aģenta vietā.

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Windows Zabbix aģenta 2 konfigurācijas fails (zabbix_agent2.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parameter	Description
Alias	Iestata vienuma atslēgai aizstājvārdu.
AllowKey	Atļauj izpildīt tās vienumu atslēgas, kas atbilst paraugam.
BufferSend	Neglabāt datus buferī ilgāk par N sekundēm.
BufferSize	Maksimālais vērtību skaits atmiņas buferī.
ControlSocket	Vadības ligzda, ko izmanto izpildlaika komandu nosūtīšanai ar opciju '-R'.
DebugLevel	Atkļūdošanas līmenis.
DenyKey	Aizliedz izpildīt tās vienumu atslēgas, kas atbilst paraugam.
EnablePersistentBuffer	Iespējo lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem.
ForceActiveChecksOnStart	Veikt aktīvās pārbaudes uzreiz pēc restartēšanas pirmajai saņemtajai konfigurācijai.
HeartbeatFrequency	Sirdspuksta ziņojumu biežums sekundēs.
HostInterface	Neobligāts parametrs, kas definē hosta saskarni.
HostInterfaceItem	Neobligāts parametrs, kas definē vienumu, ko izmanto hosta saskarnes iegūšanai.
HostMetadata	Neobligāts parametrs, kas definē hosta metadatus.
HostMetadataItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta metadatu iegūšanai.
Hostname	Neobligāts parametrs, kas definē hosta nosaukumu.
HostnameItem	Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto hosta nosaukuma iegūšanai.
Include	Konfigurācijas failā var iekļaut atsevišķus failus vai visus failus direktorijā.
ListenIP	Ar komatiem atdalīts IP adrešu saraksts, uz kurām aģentam jāgaida savienojumi.
ListenPort	Aģents gaidīs savienojumus no servera šajā portā.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais izmērs.
LogType	Žurnāla izvades tips.
PersistentBufferFile	Fails, kurā Zabbix aģentam 2 jāglabā SQLite datubāze.
PersistentBufferPeriod	Laika periods, cik ilgi dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri.
Plugins.<PluginName>.System.Capacity	Pārbaužu limits katram spraudnim, ko var izpildīt vienlaikus.
Plugins.Log.MaxLinesPerSecond	Maksimālais jauno rindu skaits sekundē, ko aģents nosūtīs uz Zabbix serveri vai starpniekserveri, apstrādājot 'log' un 'logrt' aktīvās pārbaudes.
Plugins.SystemRun.LogRemoteCommands	Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus.
PluginSocket	Ceļš uz UNIX ligzdu ielādējamo spraudņu saziņai.

Parameter	Description
PluginTimeout	Savienojumu ar ielādējamiem spraudņiem noildze sekundēs.
PerfCounter	Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
PerfCounterEn	Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Salīdzinot ar PerfCounter, perfcounter ceļiem jābūt angļu valodā.
RefreshActiveChecks	Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts.
Server	Ar komatiem atdalīts Zabbix serveru un Zabbix starpniekserveru IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi.
ServerActive	Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes.
SourceIP	Avota IP adrese.
StatusPort	Ja iestatīts, aģents gaidīs HTTP statusa pieprasījumus šajā portā (http://localhost:<port>/status).
Timeout	Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāti partnera sertifikāta pārbaudei; izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kurā ir aģenta sertifikāts vai sertifikātu ķēde; izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSCipherAll13	OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru komplekta atlases kritērijus sertifikātu un PSK šifrēšanai.
TLSCipherCert13	OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru komplekta atlases kritērijus uz sertifikātiem balstītai šifrēšanai.
TLSCipherPSK13	OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru komplekta atlases kritērijus uz PSK balstītai šifrēšanai.
TLSConnect	Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim.
TLSCRLFile	Pilns ceļš uz failu, kurā ir atsauktie sertifikāti. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kurā ir aģenta privātā atslēga; izmanto šifrētai saziņai starp Zabbix komponentiem.
TLSPSKFile	Pilns ceļš uz failu, kurā ir aģenta iepriekš koplietotā atslēga; izmanto šifrētai saziņai ar Zabbix serveri.
TLSPSKIdentity	Iepriekš koplietotās atslēgas identitātes virkne; izmanto šifrētai saziņai ar Zabbix serveri.
TLSServerCertIssuer	Atļautais servera (starpniekservera) sertifikāta izdevējs.
TLSServerCertSubject	Atļautais servera (starpniekservera) sertifikāta subjekts.
UnsafeUserParameters	Atļauj visām rakstzīmēm tikt nodotām lietotāja definēto parametru argumentos.
UserParameter	Lietotāja definēts parametrs uzraudzībai.
UserParameterDir	Noklusējuma meklēšanas ceļš UserParameter komandām.

Visi parametri ir neobligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta tikai konfigurācijas failus UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametru informācija

Alias

Iestata aizstājvārdu vienuma atslēgai. To var izmantot, lai aizstātu garu un sarežģītu vienuma atslēgu ar īsāku un vienkāršāku.
 Var būt norādīti vairāki *Alias* parametri. Vairāki parametri ar vienu un to pašu *Alias* atslēgu nav atļauti.
 Dažādas *Alias* atslēgas var atsaukties uz vienu un to pašu vienuma atslēgu.
 Aizstājvārdus var izmantot parametrā *HostMetadataItem*, bet ne parametrā *HostnameItem*.

1. piemērs: Lapošanas faila izmantojuma procentos iegūšana no servera.

```
Alias=pg_usage:perf_counter[\Paging File(_Total)\% Usage]
```

Tagad saīsināto atslēgu **pg_usage** var izmantot datu iegūšanai.

2. piemērs: CPU slodzes iegūšana ar noklusējuma un pielāgotiem parametriem.

```
Alias=cpu.load:system.cpu.load
Alias=cpu.load[*]:system.cpu.load[*]
```

Tas ļauj izmantot atslēgu **cpu.load**, lai iegūtu CPU slodzi ar noklusējuma parametriem, kā arī izmantot **cpu.load[percpu,avg15]**, lai iegūtu konkrētus datus par CPU slodzi.

3. piemērs: Vairāku zemā līmeņa atklāšanas noteikumu palaišana, kas apstrādā vienus un tos pašus atklāšanas vienumus.

```
Alias=vfs.fs.discovery[*]:vfs.fs.discovery
```

Tagad ir iespējams iestatīt vairākus atklāšanas noteikumus, izmantojot **vfs.fs.discovery** ar dažādiem parametriem katram noteikumam, piemēram, **vfs.fs.discovery[foo]**, **vfs.fs.discovery[bar]** utt.

AllowKey

Atļaut izpildīt tos vienumu atslēgu modeļus, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Vairākus atslēgu atbilstības noteikumus var definēt kopā ar DenyKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

BufferSend

The time interval in seconds which determines how often values are sent from the buffer to Zabbix server.
Note, that if the buffer is full, the data will be sent sooner.

Default: 5
 Range: 1-3600

BufferSize

Maksimālais vērtību skaits atmiņas buferī. Aģents nosūtīs visus savāktos datus uz Zabbix serveri vai starpniekserveri, ja buferis būs pilns.
Šis parametrs jāizmanto tikai tad, ja pastāvīgais buferis ir atspējots (*EnablePersistentBuffer=0*).

Noklusējums: 1000
 Diapazons: 2-65535

ControlSocket

Vadības ligzda, ko izmanto izpildlaika komandu nosūtīšanai ar opciju '-R'.

Noklusējums: `\\.\pipe\agent.sock`

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

DenyKey

Aizliegt to vienumu atslēgu izpildi, kas atbilst paraugam. Atslēgas paraugs ir aizstājējzīmju izteiksme, kas atbalsta rakstzīmi "*", lai atbilstu jebkuram jebkādu rakstzīmju skaitam.
Vairākus atslēgu atbilstības noteikumus var definēt kombinācijā ar AllowKey. Parametri tiek apstrādāti pa vienam atbilstoši to parādīšanās secībai. Skatiet arī: Aģenta pārbaužu ierobežošana.

EnablePersistentBuffer

Iespējo lokālās pastāvīgās krātuves izmantošanu aktīvajiem vienumiem. Ja pastāvīgā krātuve ir atspējota, tiks izmantots atmiņas buferis.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

ForceActiveChecksOnStart

Veikt aktīvās pārbaudes uzreiz pēc restartēšanas pirmajai saņemtajai konfigurācijai. Pieejams arī kā katra spraudņa konfigurācijas parametrs, piemēram: `Plugins.Uptime.System.ForceActiveChecksOnStart=1`

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

HeartbeatFrequency

Sirdspukstu ziņojumu biežums sekundēs. Izmanto aktīvo pārbaužu pieejamības uzraudzībai.
0 - sirdspukstu ziņojumi ir atspējoti.

Noklusējums: 60
 Diapazons: 0-3600

HostInterface

Neobligāts parametrs, kas nosaka hosta saskarni (IP adresi vai DNS nosaukumu), ko izmanto hosta automātiskās reģistrācijas procesa laikā. Šī vērtība tiks izmantota, lai aizpildītu saskarni jaunizveidotajā hostā, un ļauj skaidri konfigurēt vai nu IP, vai DNS adresi. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Ja nav definēts, vērtība tiks iegūta no HostInterfaceItem.

Aģents izdos kļūdu un nestartēsies, ja vērtība pārsniegs 255 rakstzīmju ierobežojumu.

Diapazons: 0-255 rakstzīmes

HostInterfaceItem

Neobligāts parametrs, kas definē vienumu, ko izmanto, lai noteiktu hosta saskarni (IP adresi vai DNS nosaukumu) hosta automātiskās reģistrācijas procesa laikā. Šī vērtība tiek izmantota tikai tad, ja HostInterface nav definēts. Plašāku informāciju skatiet sadaļā DNS izmantošana kā noklusējuma saskarne.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 255 rakstzīmju ierobežojumu.

Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

HostMetadata

Neobligāts parametrs, kas definē metadatus, ko izmanto, lai identificētu vai atšķirtu hostu hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadata ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Ja tas nav definēts, vērtība tiks iegūta no HostMetadataItem.

Aģents ziņos par kļūdu un nestartēsies, ja norādītā vērtība pārsniegs 2034 baitu ierobežojumu vai būs ne UTF-8 virkne. Ja parametrs paredz IP adresi vai DNS nosaukumu, vērtības, kas ir derīgas UTF-8, bet nav derīgas IP adreses vai DNS nosaukumi, arī tiks noraidītas un atzīmētas kā nederīgas.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Diapazons: 0-2034 baiti

HostMetadataItem

Neobligāts parametrs, kas definē Zabbix aģenta vienumu, ko izmanto, lai iegūtu hosta metadatus. Šī opcija tiek izmantota tikai tad, ja HostMetadata nav definēts.

HostMetadataItem vērtība tiek iegūta katrā automātiskās reģistrācijas mēģinājumā un tiek izmantota tikai hosta automātiskās reģistrācijas procesā (aktīvais aģents). HostMetadataItem ļauj atšķirt hostus ne tikai pēc hosta nosaukuma.

Tiek atbalstīti lietotāja parametri un aizstājvārdi. Vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey iestatījumiem.

Automātiskās reģistrācijas pieprasījuma laikā aģents reģistrēs brīdinājuma ziņojumu, ja norādītā vienuma atgrieztā vērtība pārsniedz 65535 UTF-8 koda punktu ierobežojumu. Vienuma atgrieztajai vērtībai jābūt UTF-8 virknei, pretējā gadījumā tā tiks ignorēta. Ja parametrs paredz IP adresi vai DNS nosaukumu, vērtības, kas ir derīgas UTF-8, bet nav derīgas IP adreses vai DNS nosaukumi, arī tiks noraidītas un atzīmētas kā nederīgas.

Vairākrindu metadati netiek atbalstīti — izvade tiks apgriezta pie pirmās rindas pārtraukuma.

Hostname

Ar komatiem atdalītu, unikālu, reģistrjutīgu hostu nosaukumu saraksts. Nepieciešams aktīvajām pārbaudēm, un tam jāsakrīt ar hostu nosaukumiem, kā tie konfigurēti serverī. Ja vērtība nav definēta, tā tiek iegūta no HostnameItem.
Atļautās rakstzīmes: burtciparu rakstzīmes, '.', ' ', '_' un '-'. Maksimālais garums: 128 rakstzīmes katram hosta nosaukumam, 2048 rakstzīmes visai rindai.

Noklusējums: Iestata HostnameItem

HostnameItem

Neobligāts parametrs, kas definē vienumu, ko izmanto hosta nosaukuma iegūšanai. Šī opcija tiek izmantota tikai tad, ja Hostname nav definēts. Lietotāja parametri vai aizstājvārdi netiek atbalstīti, bet vienums system.run[] tiek atbalstīts neatkarīgi no AllowKey/DenyKey vērtībām.

Noklusējums: `system.hostname`

Iekļaut

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā (pēc noklusējuma tas atrodas `C:\Program Files\Zabbix Agent 2`, ja Zabbix aģents ir instalēts, izmantojot Windows MSI instalētāja pakotnes; tas atrodas instalēšanas laikā norādītajā mapē, ja Zabbix aģents ir instalēts kā zip arhīvs). Visiem iekļautajiem failiem jābūt ar pareizu sintaksi, pretējā

gadījumā aģents netiks palaists. Ceļš var būt relatīvs attiecībā pret *zabbix_agent2.conf* faila atrašanās vietu (piemēram, `Include=.\zabbix_agent2.d\plugins.d\*.conf`).
Lai iekļautu tikai atbilstošos failus norādītajā direktorijā, tiek atbalstīta zvaigznītes aizstājējzīme šablonu atbilstības noteikšanai.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=C:\Program Files\Zabbix Agent2\zabbix_agent2.d\*.conf
```

ListenIP

Ar komatiem atdalītu IP adrešu saraksts, kurās aģents klausās. Pirmā IP adrese tiek nosūtīta Zabbix serverim, ja tiek veidots savienojums ar to, lai iegūtu aktīvo pārbaužu sarakstu.

Noklusējums: `0.0.0.0`

ListenPort

Aģents klausīsies šajā portā savienojumiem no servera.

Noklusējums: 10050
 Diapazons: 1024-32767

LogFile

Aģenta žurnālfaila nosaukums.

Noklusējums: `c:\zabbix_agent2.log`
 Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija kāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts parametrā LogFile;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

PersistentBufferFile

Fails, kur Zabbix aģents 2 glabā SQLite datubāzi. Jānorāda pilns faila nosaukums. Šis parametrs tiek izmantots tikai tad, ja ir iespējota pastāvīgā buferēšana (*EnablePersistentBuffer=1*).

PersistentBufferPeriod

Laika periods, cik ilgi dati jāglabā, ja nav savienojuma ar serveri vai starpniekserveri. Vecāki dati tiks zaudēti. Žurnāla dati tiks saglabāti. Šis parametrs tiek izmantots tikai tad, ja noturīgais buferis ir iespējots (*EnablePersistentBuffer=1*).

Noklusējums: 1h
 Diapazons: 1m-365d

Plugins.<PluginName>.System.Capacity

Pārbaužu limits katram `<PluginName>` spraudnim, ko var izpildīt vienlaikus.

Noklusējums: 1000 Diapazons: 1-1000

Plugins.Log.MaxLinesPerSecond

Maksimālais jauno rindu skaits, ko aģents nosūtīs sekundē uz Zabbix serveri vai starpniekserveri, apstrādājot aktīvās pārbaudes `log`, `logrt` un `eventlog`. Norādīto vērtību pārrakstīs parametrs `maxlines`, kas norādīts `log`, `logrt` vai `eventlog` vienuma atslēgā.
Piezīme: Zabbix apstrādās 10 reizes vairāk jauno rindu, nekā iestatīts *MaxLinesPerSecond*, lai log vienumos meklētu nepieciešamo virkni.

Noklusējums: 20
 Diapazons: 1-1000

Plugins.SystemRun.LogRemoteCommands

Iespējot izpildīto čaulas komandu reģistrēšanu kā brīdinājumus. Komandas tiks reģistrētas tikai tad, ja tās tiks izpildītas attālināti. Žurnāla ieraksti netiks izveidoti, ja system.run[] tiks palaists lokāli ar parametriem HostMetadataItem, HostInterfaceItem vai HostnameItem.

Noklusējums: 0
 Vērtības: 0 - atspējots, 1 - iespējots

PluginSocket

Ceļš uz UNIX ligzdu ielādējamo spraudņu saziņai.

Noklusējums: `\\.\pipe\agent.plugin.sock`

PluginTimeout

Savienojumu ar ielādējamiem spraudņiem noildze sekundēs.

Noklusējums: Timeout
 Diapazons: 1-30

PerfCounter

Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs).
Sintakse: <parameter_name>,"<perf_counter_path>",<period>

Piemēram, ja vēlaties saņemt procesora pārtraukumu vidējo skaitu sekundē par pēdējo minūti, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounter = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultpēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Varat palaist "typeperf -qx", lai iegūtu visu Windows pieejamo veiktspējas skaitītāju sarakstu.

PerfCounterEn

Definē jaunu parametru <parameter_name>, kas ir sistēmas veiktspējas skaitītāja <perf_counter_path> vidējā vērtība norādītajā laika periodā <period> (sekundēs). Salīdzinot ar PerfCounter, perfcounter ceļiem jābūt angļu valodā. Atbalstīts tikai **Windows Server 2008/Vista** un jaunākās versijās.
Sintakse: <parameter_name>,"<perf_counter_path>",<period>

Piemēram, ja vēlaties saņemt procesora pārtraukumu vidējo skaitu sekundē par pēdējo minūti, varat definēt jaunu parametru "interrupts" šādi:

```
PerfCounterEn = interrupts,"\Processor(0)\Interrupts/sec",60
```

Lūdzu, ņemiet vērā dubultās pēdiņas ap veiktspējas skaitītāja ceļu. Parametra nosaukums (interrupts) jāizmanto kā vienuma atslēga, veidojot vienumu. Paraugi vidējās vērtības aprēķināšanai tiks ņemti katru sekundi.
Angļu virkņu sarakstu var atrast, apskatot šādu reģistra atslēgu: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Perflib\009.

RefreshActiveChecks

Cik bieži tiek atsvaidzināts aktīvo pārbaužu saraksts, sekundēs. Ņemiet vērā, ka pēc neveiksmīga aktīvo pārbaužu saraksta atsvaidzināšanas nākamais atsvaidzināšanas mēģinājums tiks veikts pēc 60 sekundēm.

Noklusējums: 5
 Diapazons: 1-86400

Server

Ar komatiem atdalītu IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai Zabbix serveru vai Zabbix starpniekserveru DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem. Ja ir iespējots IPv6 atbalsts, tad '127.0.0.1', '::127.0.0.1', '::ffff:127.0.0.1' tiek uzskatītas par vienādām, un '::/0' atļaus jebkuru IPv4 vai IPv6 adresi. '0.0.0.0/0' var izmantot, lai atļautu jebkuru IPv4 adresi. Atstarpes ir atļautas. Ja šis parametrs nav norādīts, pasīvās pārbaudes tiek atspējotas un aģents neklausīsies nevienā TCP portā.

Piemērs:

```
Server=127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

ServerActive

Zabbix servera/starpniekservera adrese vai klastera konfigurācija, no kuras saņemt aktīvās pārbaudes. Servera/starpniekservera adrese ir IP adrese vai DNS nosaukums un neobligāts ports, kas atdalīts ar kolu.
Klastera konfigurācija ir viena vai vairākas servera vai starpniekservera grupas dalībnieku adreses, kas atdalītas ar semikolu. Var norādīt vairākus Zabbix serverus/klasterus un Zabbix starpniekserverus, atdalot tos ar komatu. Ja netiek izmantotas starpniekserveru grupas, no katra Zabbix servera/klastera nevajadzētu norādīt vairāk nekā vienu Zabbix starpniekserveri. Ja ir norādīts Zabbix starpniekserveris, tad šī starpniekservera Zabbix serveri/klasteri nevajadzētu norādīt.
Lai paralēli izmantotu vairākus neatkarīgus Zabbix serverus, var norādīt vairākas ar komatiem atdalītas adreses. Atstarpes ir atļautas.
Ja ports nav norādīts, tiek izmantots noklusējuma ports.
IPv6 adresēm jābūt ietvertām kvadrātiekavās, ja šim hostam ir norādīts ports. Ja ports nav norādīts, IPv6 adrešu kvadrātiekavas nav obligātas.
Ja šis parametrs nav norādīts, aktīvās pārbaudes ir atspējotas.

Piemērs Zabbix starpniekserverim:

```
ServerActive=127.0.0.1:10051
```

Piemērs Zabbix starpniekserveru grupai:

```
ServerActive=proxy1.example.com;proxy2.example.com;proxy3.example.com;proxy4.example.com;proxy5.example.co
```

Piemērs vairākiem serveriem:

```
ServerActive=127.0.0.1:20051,zabbix.domain,\[::1\]:30051,::1,\[12fc::1\]
```

Augstas pieejamības piemērs:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051;zabbix.cluster.node3
```

Augstas pieejamības piemērs ar diviem klasteriem un vienu serveri:

```
ServerActive=zabbix.cluster.node1;zabbix.cluster.node2:20051,zabbix.cluster2.node1;zabbix.cluster2.node2,z
```

SourceIP

Avota IP adrese:

- izejošajiem savienojumiem ar Zabbix serveri vai Zabbix starpniekserveri;
- savienojumu izveidei, izpildot dažus vienumus (web.page.get, net.tcp.port u.c.).

StatusPort

Ja iestatīts, aģents klausīsies šajā portā HTTP statusa pieprasījumiem (http://localhost:<port>/status).

Diapazons: 1024-32767

Noildze

Norāda, cik ilgi gaidīt (sekundēs), lai izveidotu savienojumu un apmainītos ar datiem ar Zabbix starpniekserveri vai serveri.

Šis parametrs nosaka dažādu saziņas darbību ilgumu, tostarp:

- atbildes gaidīšanu no Zabbix servera;
- pieprasījumu sūtīšanu uz Zabbix serveri, tostarp vienumu konfigurācijas pieprasījumus un vienumu datus aktīvajās pārbaudēs;
- žurnāla datu iegūšanu, izmantojot logfile vai Windows notikumu žurnāla uzraudzību;
- heartbeat ziņojumu sūtīšanu;
- maksimālo ilgumu `vfs.*` pārbaudēm;
- izmantošanu kā rezerves vērtību scenārijos, kad serveris vai starpniekserveris, kas ir vecāks par versiju 7.0, sūta pārbaudes bez noildzēm.

Šī noildze **netiks** izmantota tām aģenta pārbaudēm, kurām lietotāja saskarnē ir konfigurējami noildzes iestatījumi (globālā, starpniekservera vai katra vienuma līmenī).

Noklusējums: 3
 Diapazons: 1-30

TLSAccept

Pieņemtie ienākošie savienojumi. Izmanto pasīvajām pārbaudēm. Var norādīt vairākas vērtības, atdalot tās ar komatu:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
psk - pieņemt savienojumus ar TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieņemt savienojumus ar TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCAFile

Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus(-u) partnera sertifikāta verifikācijai un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai ar Zabbix komponentiem.

TLSCipherAll13

OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru komplektu atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem un PSK.

Piemērs:

```
TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSCipherCert13

OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru kopas atlases kritērijus šifrēšanai, kas balstīta uz sertifikātiem.

Ņemiet vērā, ka šo parametru nevar izmantot kopā ar `TLSAccept=cert,psk`; savienojumam ar sertifikātu (`TLSConnect=cert`) tā vietā izmantojiet TLSCipherAll13.

TLSCipherPSK13

OpenSSL šifru virkne (TLS 1.3). Pārraksta noklusējuma šifru komplekta atlases kritērijus uz PSK balstītai šifrēšanai.

Piemērs:

```
TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128_GCM_SHA256
```

TLSConnect

Kā aģentam jāpieslēdzas Zabbix serverim vai starpniekserverim. Izmanto aktīvajām pārbaudēm. Var norādīt tikai vienu vērtību:
unencrypted - pieslēgties bez šifrēšanas (noklusējums)
psk - pieslēgties, izmantojot TLS un iepriekš koplietotu atslēgu (PSK)
cert - pieslēgties, izmantojot TLS un sertifikātu

Obligāts: jā, ja ir definēti TLS sertifikāta vai PSK parametri (pat *unencrypted* savienojumam); pretējā gadījumā nē

TLSCRLFile

Pilns ceļš uz failu, kas satur atsauktos sertifikātus. Šis parametrs tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur aģents privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSPSKFile

Pilns ceļš uz failu, kas satur aģenta iepriekš koplietoto atslēgu un tiek izmantots šifrētai saziņai ar Zabbix serveri.

TLSPSKIdentity

Iepriekš koplietotās atslēgas identitātes virkne, ko izmanto šifrētai saziņai ar Zabbix serveri.

TLSServerCertIssuer

Atļautā servera (starpniekservera) sertifikāta izdevējs.

TLSServerCertSubject

Atļautā servera (starpniekservera) sertifikāta subjekts.

UnsafeUserParameters

Atļaut visas rakstzīmes nodot argumentos lietotāja definētiem parametriem. Šādas rakstzīmes nav atļautas: \ ' " ` * ? [ ] { } ~ $! & ; ( ) < > | # @ %
Papildus tam nav atļautas arī jaunās rindas rakstzīmes.

Noklusējums: 0
 Vērtības: 0 - neatļaut, 1 - atļaut

UserParameter

Lietotāja definēts parametrs uzraudzībai. Var būt vairāki lietotāja definēti parametri.
Formāts: UserParameter=<key>,<shell command>
Ņemiet vērā, ka čaulas komanda nedrīkst atgriezt tukšu virkni vai tikai EOL. Čaulas komandām var būt relatīvi ceļi, ja ir norādīts parametrs UserParameterDir.

Piemērs:

```
UserParameter=system.test,dir /b | find /c /v ""
UserParameter=check_cpu,.\custom\_script.bat
```

UserParameterDir

Noklusējuma meklēšanas ceļš UserParameter komandām. Ja tiek izmantots, aģents pirms komandas izpildes nomainīs savu darba direktoriju uz šeit norādīto. Tādējādi UserParameter komandām pilna ceļa vietā var izmantot relatīvo prefiksu `.\`.
Ir atļauts tikai viens ieraksts.

Piemērs:

```
UserParameterDir=C:\opt\myscripts
```

7 Spraudņu konfigurācijas parametri

Skatiet Zabbix aģents 2 spraudņu konfigurācijas parametrus:

- Ceph spraudnis
- Docker spraudnis
- Ember+ spraudnis
- Memcached spraudnis
- Modbus spraudnis
- MongoDB spraudnis

- MQTT spraudnis
- MSSQL spraudnis
- MySQL spraudnis
- NVIDIA GPU spraudnis
- Oracle spraudnis
- PostgreSQL spraudnis
- Redis spraudnis
- S.M.A.R.T. spraudnis

Spraudņu konfigurācijas principi

Šajā lapā ir aprakstīti spraudņu konfigurācijas principi un labākā prakse.

Visi spraudņi tiek konfigurēti, izmantojot parametru *Plugins.**, kas var būt vai nu daļa no Zabbix aģenta 2 konfigurācijas faila, vai arī paša spraudņa konfigurācijas faila. Ja spraudnis izmanto atsevišķu konfigurācijas failu, ceļš uz šo failu jānorāda Zabbix aģenta 2 konfigurācijas faila parametrā Include.

Tipiska spraudņa parametra struktūra ir šāda:

Plugins.<PluginName>.<Parameter>=<Value>

Papildus tam ir divas specifiskas parametru grupas:

- *Plugins.<PluginName>.Default.<Parameter>=<Value>* tiek izmantots, lai definētu noklusējuma parametru vērtības.
- *Plugins.<PluginName>.<SessionName>.<Parameter>=<Value>* tiek izmantots, lai definētu atsevišķas parametru kopas dažādiem uzraudzības mērķiem, izmantojot nosauktās sesijas.

Visiem parametru nosaukumiem jāatbilst šādām prasībām:

- ieteicams jūsu spraudņu nosaukumus rakstīt ar lielo sākumburtu
- parametra nosaukums jāraksta ar lielo sākumburtu
- speciālās rakstzīmes nav atļautas
- ligzdošanas līmenis nav ierobežots
- parametru skaits nav ierobežots

Piemēram, lai veiktu aktīvās pārbaudes, kurām nav *Scheduling* atjaunināšanas intervāla, uzreiz pēc aģenta restartēšanas tikai Uptime spraudnim, konfigurācijas failā iestatiet `Plugins.Uptime.System.ForceActiveChecksOnStart=1`. Līdzīgi, lai CPU spraudnim iestatītu pielāgotu vienlaicīgo pārbaužu limitu, konfigurācijas failā iestatiet `Plugins.CPU.System.Capacity=N`.

Noklusējuma vērtības

Savienojuma parametriem (URI, lietotājvārds, parole u.c.) varat iestatīt noklusējuma vērtības konfigurācijas failā šādā formātā:

```
Plugins.<PluginName>.Default.<Parameter>=<Value>
```

Piemēram, `Plugins.Mysql.Default.Username=zabbix`, `Plugins.MongoDB.Default.Uri=tcp://127.0.0.1:27017` u.c.

Ja šāda parametra vērtība nav norādīta vienuma atslēgā vai nosauktās sesijas parametros, spraudnis izmantos noklusējuma vērtību. Ja arī noklusējuma parametrs nav definēts, tiks izmantotas stingri kodētas noklusējuma vērtības.

> **Note:**
> Ja vienuma atslēgai nav neviena parametra, Zabbix aģents 2 mēģinās ievākt metriku, izmantojot noklusējuma parametru sadaļā definētās vērtības.

Nosauktās sesijas

Nosauktās sesijas ir papildu spraudņa parametru līmenis, un tās var izmantot, lai norādītu atsevišķas autentifikācijas parametru kopas katrai no uzraudzītajām instancēm. Katram nosauktās sesijas parametram jābūt šādā struktūrā:

```
Plugins.<PluginName>.Sessions.<SessionName>.<Parameter>=<Value>
```

Sesijas nosaukumu var izmantot kā vienuma atslēgas parametru connString, nevis atsevišķi norādīt URI, lietotājvārdu un/vai paroli.

Vienumu atslēgās pirmais parametrs var būt vai nu connString, vai URI. Ja pirmais atslēgas parametrs neatbilst nevienam sesijas nosaukumam, tas tiks apstrādāts kā URI. Ņemiet vērā, ka vienuma atslēgā iegultu URI akreditācijas datu nodošana netiek atbalstīta; tā vietā izmantojiet nosauktās sesijas parametrus.

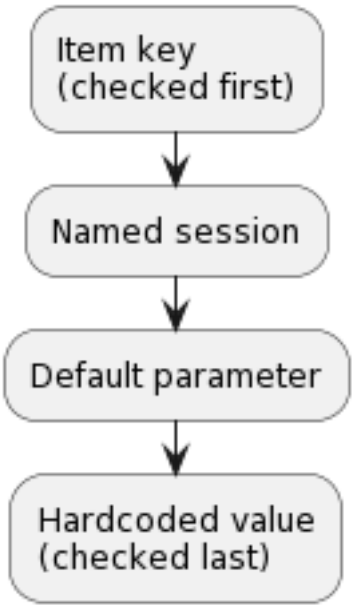
Pieejamo nosaukto sesiju parametru saraksts ir atkarīgs no spraudņa.

Sesijas parametrus ir iespējams pārrakstīt, vienuma atslēgas parametros norādot jaunas vērtības (skatiet piemēru).

Ja parametrs nosauktajai sesijai nav definēts, Zabbix aģents 2 izmantos vērtību, kas definēta noklusējuma spraudņa parametrā.

Parametru prioritāte

Zabbix aģents 2 spraudņi meklē ar savienojumu saistīto parametru vērtības šādā secībā:



1. Pirmais vienuma atslēgas parametrs tiek salīdzināts ar sesiju nosaukumiem. Ja atbilstība netiek atrasta, tas tiek uzskatīts par faktisko vērtību; šajā gadījumā 3. solis tiks izlaists. Ja atbilstība tiek atrasta, parametra vērtībai (parasti URI) jābūt definētai nosauktajā sesijā.
2. Pārējie parametri tiks ņemti no vienuma atslēgas, ja tie ir definēti.
3. Ja vienuma atslēgas parametrs (piemēram, parole) ir tukšs, spraudnis meklēs atbilstošo nosauktās sesijas parametru.
4. Ja arī sesijas parametrs nav norādīts, tiks izmantota atbilstošajā noklusējuma parametrā definētā vērtība.
5. Ja nekas no iepriekš minētā neizdodas, spraudnis izmantos stingri ieprogrammēto noklusējuma vērtību.

Piemērs 1

Divu instanču "MySQL1" un "MySQL2" uzraudzība.

Konfigurācijas parametri:

```
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
```

Šīs konfigurācijas rezultātā katru sesijas nosaukumu var izmantot kā connString vienuma atslēgā, piemēram, `mysql.ping[MySQL1]` vai `mysql.ping[MySQL2]`.

Piemērs 2

Dažu parametru norādīšana vienuma atslēgā.

Konfigurācijas parametri:

```
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
```

Vienuma atslēga: `pgsql.ping[session1,new_username,,postgres]`

Šīs konfigurācijas rezultātā aģents izveidos savienojumu ar PostgreSQL, izmantojot šādus parametrus:

- URI no sesijas parametra: `192.0.2.234:5432`
- Lietotājvārds no vienuma atslēgas: `new_username`
- Parole no sesijas parametra (jo tā nav norādīta vienuma atslēgā): `session_password`
- Datubāzes nosaukums no vienuma atslēgas: `postgres`

Piemērs 3

Metrikas vākšana, izmantojot noklusējuma konfigurācijas parametrus.

Konfigurācijas parametri:

```
Plugins.PostgreSQL.Default.Uri=tcp://192.0.2.234:5432
Plugins.PostgreSQL.Default.User=zabbix
Plugins.PostgreSQL.Default.Password=password
```

Vienuma atslēga: `pgsql.ping[,,,postgres]`

Šīs konfigurācijas rezultātā aģents izveidos savienojumu ar PostgreSQL, izmantojot šādus parametrus:

- Noklusējuma URI: `192.0.2.234:5432`
- Noklusējuma lietotājvārds: `zabbix`
- Noklusējuma parole: `password`
- Datubāzes nosaukums no vienuma atslēgas: `postgres`

Savienojumi

Daži spraudņi atbalsta metriku vākšanu no vairākām instancēm vienlaikus. Var uzraudzīt gan lokālās, gan attālinātās instances. Tiek atbalstīti TCP un Unix-socket savienojumi.

Ieteicams konfigurēt spraudņus tā, lai savienojumi ar instancēm tiktu uzturēti atvērtā stāvoklī. Ieguvumi ir mazāka tīkla noslodze, zemāka latentce, kā arī mazāks CPU un atmiņas patēriņš, jo ir mazāks savienojumu skaits. Par to rūpējas klienta bibliotēka.

> **Note:**
> Laika periodu, cik ilgi neizmantotajiem savienojumiem jāpaliek atvērtiem, var noteikt ar parametru *Plugins.<PluginName>.KeepAlive*. Piemērs: *Plugins.Memcached.KeepAlive*

1 Ceph spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti Ceph spraudņa darbībai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `ceph.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Ceph.Default.ApiKey				Noklusējuma API atslēga savienojumam ar Ceph; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Ceph.Default.Mode		native / restful	restful (novecojis)	Noklusējuma režīms savienojumam ar Ceph. Native režīms tiek atbalstīts tikai Linux vidē un начиная ar Ceph 16. Restful režīms nedarbosies ar Ceph versiju 20 vai jaunāku. Ņemiet vērā, ka lietotāja akreditācijas datu kopa katram režīmam atšķiras, un tie nav savstarpēji saderīgi.
Plugins.Ceph.Default.User				Noklusējuma lietotājvārds savienojumam ar Ceph; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Ceph.Default.Uri	nē		https://localhost:8003	Savienojuma URI savienojumam ar Ceph; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība. Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti). Jāatbilst URI formātam. Tiek atbalstīta tikai `https` shēma; shēmu var nenorādīt. Portu var nenorādīt (noklusējums=8003). Piemēri: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.InsecureSkipVerify	nē	false / true	false	Nosaka, vai http klientam jāpārbauda servera sertifikātu ķēde un hosta nosaukums. Ja *true*, TLS pieņem jebkuru servera uzrādītu sertifikātu un jebkuru hosta nosaukumu šajā sertifikātā. Šajā režīmā TLS ir pakļauts "man-in-the-middle" uzbrukumiem (izmantojams tikai testēšanai).
Plugins.Ceph.KeepAlive	nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pirms neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.Ceph.Sessions.<SessionName>.ApiKey	nē			Nosauktās sesijas API atslēga. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Ceph.Sessions.<SessionName>.Mode	nē	native / restful	restful (novecojis)	Nosauktās sesijas režīms savienojumam ar Ceph. Native režīms tiek atbalstīts tikai Linux vidē un начиная ar Ceph 16. Restful režīms nedarbosies ar Ceph versiju 20 vai jaunāku. Ņemiet vērā, ka lietotāja akreditācijas datu kopa katram režīmam atšķiras, un tie nav savstarpēji saderīgi.
Plugins.Ceph.Sessions.<SessionName>.User	nē			Nosauktās sesijas lietotājvārds. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Ceph.Sessions.<SessionName>.Uri	nē			Nosauktās sesijas savienojuma virkne. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās. Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti). Jāatbilst URI formātam. Tiek atbalstīta tikai `https` shēma; shēmu var nenorādīt. Portu var nenorādīt (noklusējums=8003). Piemēri: `https://127.0.0.1:8003` `localhost`
Plugins.Ceph.Timeout	nē	1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

2 Docker spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti Docker spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `docker.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;

- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Docker.Endpoint	Nē		unix:///var/run/docker.sock	Docker programmona unix-socket atrašanās vieta. Jāsatur shēma (tiek atbalstīts tikai `unix://`).
Plugins.Docker.Timeout	Nē	1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

3 Ember+ spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots sprauduņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti Ember+ sprauduņa darbībai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `emberplus.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ember+ spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts Ember+ sprauduņa repozitorijā.

Šis spraudnis pašlaik ir pieejams tikai būvēšanai no pirmkoda (gan Unix, gan Windows vidē).

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Opcijas

Parameter	Description
-V --version	Izdrukāt sprauduņa versiju un licences informāciju.
-h --help	Izdrukāt palīdzības informāciju (saīsināti).
-t, --test <item key>	Palaist spraudni testēšanai (sprauduņa konfigurācija tiek ignorēta).

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.EmberPlus.Default.Uri	Nē		tcp://localhost:9998	Noklusējuma URI savienojuma izveidei. Vienīgā atbalstītā shēma ir `tcp://`. Shēmu var nenorādīt. Iegultie akreditācijas dati tiks ignorēti.
Plugins.EmberPlus.KeepAlive	Nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pēc kura neizmantotie sprauduņa savienojumi tiek aizvērti.
Plugins.EmberPlus.Sessions.<SessionName>.Uri	Nē		tcp://localhost:9998	URI savienojuma izveidei nosauktajai sesijai. Vienīgā atbalstītā shēma ir `tcp://`. Shēmu var nenorādīt. Iegultie akreditācijas dati tiks ignorēti. **<SessionName>** — definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.EmberPlus.System.Path	Nē			Ceļš uz Ember+ sprauduņa izpildāmo failu. Piemērs: `Plugins.EmberPlus.System.Path=/usr/libexec/zabbix/zabbix`
Plugins.EmberPlus.Timeout	Nē	1-30	globālais taimauts	Laiks sekundēs, cik ilgi gaidīt servera atbildi, pirmoreiz izveidojot savienojumu un veicot turpmākās darbības sesijā.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

4 Memcached spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots sprauduņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti Memcached sprauduņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `memcached.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parameter	Mandatory	Range	Default	Description
Plugins.Memcachēd.Default.Password				Noklusējuma parole savienojumam ar Memcached; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.

Plugins.Memcachēd.Default.Uri			tcp://localhost:11211	Noklusējuma URI savienojumam ar Memcached; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
				
				Tam nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=11211).
				Piemēri: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcachēd.Default.User				Noklusējuma lietotājvārds savienojumam ar Memcached; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Memcachēd.KeepAlive		60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pirms tiek aizvērti neizmantotie sprauduņa savienojumi.
Plugins.Memcachēd.Sessions.<SessionName>.Password				Nosauktās sesijas parole. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.Memcachēd.Sessions.<SessionName>.Uri				Nosauktās sesijas savienojuma virkne.
				<SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Tam nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=11211).
				Piemēri: `tcp://localhost:11211`
				`localhost`
				`unix:/var/run/memcached.sock`

Plugins.Memcachēd.Sessions.<SessionName>.User				Nosauktās sesijas lietotājvārds. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Memcachēd.Timeout		1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

5 Modbus spraudnis

Pārskats

Zabbix aģenta 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģenta 2 konfigurācijas parametri tiek atbalstīti Modbus spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `modbus.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģenta 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma iestatījumus, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Modbus.Sessions.<SessionName>.Endpoint				Endpoint ir savienojuma virkne, kas sastāv no protokola shēmas, hosta adreses un porta vai seriālā porta nosaukuma un atribūtiem. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Modbus.Sessions.<SessionName>.SlaveID				Nosauktās sesijas Slave ID. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās. Piemērs: `Plugins.Modbus.Sessions.MB1.SlaveID=20` *Ņemiet vērā*, ka šis nosauktās sesijas parametrs tiek pārbaudīts tikai tad, ja vienuma atslēgas slave ID parametra vērtība ir tukša.
Plugins.Modbus.Sessions.<SessionName>.Timeout				Nosauktās sesijas noildze sekundēs. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās. Piemērs: `Plugins.Modbus.Sessions.MB1.Timeout=2`

Ja nepieciešams iestatīt pieprasījuma izpildes noildzi (cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas), izmantojiet vienuma konfigurācijas formu.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru aprakstu: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumus spraudņu konfigurēšanai

6 MongoDB spraudnis

Pārskats

Zabbix aģenta 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģenta 2 konfigurācijas parametri tiek atbalstīti MongoDB spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `mongo.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģenta 2 konfigurācijai.

MongoDB spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts MongoDB spraudņa repozitorijā.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.

- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Opcijas

Parameter	Description
-V --version	Izdrukāt spraudņa versiju un licences informāciju.
-h --help	Izdrukāt palīdzības informāciju (saīsināti).
-t, --test <item key>	Palaist spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Parametri

Pieslēdzoties MongoDB, spraudnis pārbauda savu konfigurāciju noteiktā secībā, kas nosaka, kuru vērtību izmantot. Secība no augstākās līdz zemākajai prioritātei ir šāda:

1. Parametri, kas norādīti savienojuma URI (piemēram, vaicājuma parametri, kas definēti aiz ? parametrā `Plugins.MongoDB.Sessions.` ir ar visaugstāko prioritāti).
2. Parametri Zabbix aģenta 2 vienuma atslēgā (piemēram, mongodb.collection.stats).
3. Parametri, kas definēti nosauktajā sesijā (piemēram, `Plugins.MongoDB.Sessions.<SessionName>.User`, `Plugins.MongoDB.Se`
4. Noklusējuma vērtības (piemēram, `Plugins.MongoDB.Default.User`, `Plugins.MongoDB.Default.Password`).

Piemēram:

- Ja savienojuma autentificēšanai ar MongoDB serveri norādāt datubāzi parametrā `*.<SessionName>.Uri` (piemēram, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://user:password@127.0.0.1:27017/?authSource=admin` tad spraudnis uzskatīs, ka parametrs `*.<SessionName>.AuthSource` ir iestatīts uz `admin`, pat ja parametrs `*.<SessionName>.AuthSource` ir iestatīts uz citu vērtību.
- Ja norādāt URI parametrā `*.<SessionName>.Uri` (piemēram, `Plugins.MongoDB.Sessions.<session_name>.Uri=mongodb://` bet lietotājvārdu un paroli norādāt Zabbix aģenta 2 vienuma atslēgā, spraudnis izmantos lietotājvārdu un paroli no vienuma atslēgas, pat ja `Plugins.MongoDB.Sessions.<SessionName>.User` un `Plugins.MongoDB.Sessions.<SessionName>.Passw` ir konfigurēti.

Attention:
Ja tiek izmantota shēma `mongodb+srv://`, URI var iegūt parametrus no DNS TXT ieraksta. Parametri, kas iegūti no šī TXT ieraksta, ignorē visas pārējās konfigurācijas, tostarp URI parametrus, Zabbix aģenta 2 vienuma atslēgas parametrus un sesijas parametrus.
 Savienojuma virknē izmantotajam resursdatora nosaukumam drīkst pastāvēt tikai viens TXT ieraksts.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.System.Path	No			Ceļš uz MongoDB spraudņa izpildāmo failu. Piemērs: `Plugins.MongoDB.System.Path=/usr/libexec/zabbix/zabbix-a`
Plugins.MongoDB.Timeout	No	1-30	global timeout	Pieprasījuma izpildes noildze (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).
Plugins.MongoDB.KeepAlive	No	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pēc kura neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.MongoDB.Sessions.<SessionName>.Uri	No		`tcp://127.0.0.1:27017`	Nosauktās sesijas savienojuma virkne. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Jāatbilst URI formātam. Atbalstītās shēmas: `mongodb://` un `mongodb+srv://`; shēma `tcp://` ir novecojusi un tiek saglabāta atpakaļsaderībai ar esošajām konfigurācijām. Portu var nenorādīt (noklusējums=27017). Piemēri: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Sessions.<SessionName>.User	No			Nosauktās sesijas lietotājvārds. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MongoDB.Sessions.<SessionName>.Password	No			Nosauktās sesijas parole. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.Direct	No		false	Savienojuma metode, ko izmanto, lai noteiktu, vai automātiskā MongoDB repliku kopas mezglu atklāšana ir jāiespējo vai jāatspējo. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: `true` - spraudnis pieslēgsies tikai norādītajam serverim, ignorējot citus repliku kopas dalībniekus; `false` - spraudnis automātiski atklās visus repliku kopas mezglus. Ja tiek izmantota shēma `tcp://` (novecojusi), parametrs pēc noklusējuma tiek iestatīts uz `true`.
Plugins.MongoDB.Sessions.<SessionName>.AuthMechanism	No			Autentifikācijas mehānisms pieslēgšanai MongoDB serverim. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: `SCRAM-SHA-1`, `SCRAM-SHA-256` un `MONGODB-X509`.
Plugins.MongoDB.Sessions.<SessionName>.AuthSource	No			Datubāze savienojuma autentificēšanai ar MongoDB serveri. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Ja tiek izmantots autentifikācijas mehānisms `MONGODB-X509`, šis parametrs tiek automātiski iestatīts uz `$external`; tā mainīšana uz jebkuru citu vērtību izraisīs kļūdu palaišanas laikā.
Plugins.MongoDB.Sessions.<SessionName>.ReplicaSet	No			Repliku kopas nosaukums, ko izmanto MongoDB repliku kopas mezglu automātiskai atklāšanai. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Ja šis parametrs tiek izmantots ar shēmu `mongodb+srv://`, tas darbosies tā, it kā būtu norādīti vairāki URL; DNS serverim var būt TXT ieraksts, kas var ignorēt šo (vai jebkuru citu) parametru. Nav pieejams, ja parametrs `Plugins.MongoDB.Sessions.<SessionName>.Direct` ir iestatīts uz `true`; aģentu neizdosies palaist.
Plugins.MongoDB.Sessions.<SessionName>.ReadPreference	No			Secība pieslēgšanai MongoDB repliku kopas mezgliem vai atkāpšanās variants. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` un `NearestMode`. Nav pieejams, ja parametrs `Plugins.MongoDB.Sessions.<SessionName>.Direct` ir iestatīts uz `true`; aģentu neizdosies palaist.
Plugins.MongoDB.Sessions.<SessionName>.TLSConnect	No			Šifrēšanas veids saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: `required` - pieprasa TLS savienojumu; `verify\_ca` - pārbauda sertifikātus; `verify\_full` - pārbauda sertifikātus un IP adresi. Atbalstīts kopš spraudņa versijas 1.2.1.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Sessions.<SessionName>.TLSCAFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSConnect is set to one of: verify_ca, verify_full			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātu(-us) otras puses sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile is specified			Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MongoDB.Sessions.<SessionName>.TLSKeyFile	Yes, if Plugins.MongoDB.Sessions.<SessionName>.TLSCertFile is specified			Pilns ceļš uz failu, kas satur datubāzes privāto atslēgu, ko izmanto šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MongoDB.Default.Uri			`tcp://127.0.0.1:27017`	Noklusējuma URI pieslēgšanai MongoDB. Jāatbilst URI formātam. Atbalstītās shēmas: `mongodb://` un `mongodb+srv://`. Novecojusi shēma: `tcp://` (izmantota kā noklusējuma vērtība atpakaļsaderībai ar esošajām konfigurācijām). Portu var nenorādīt (noklusējums=27017). Piemēri: `mongodb://127.0.0.1:27017`, `mongodb+srv://example.com`, `localhost`.
Plugins.MongoDB.Default.User				Noklusējuma lietotājvārds pieslēgšanai MongoDB.
Plugins.MongoDB.Default.Password				Noklusējuma parole pieslēgšanai MongoDB.
Plugins.MongoDB.Default.Direct			`false`	Noklusējuma savienojuma metode, ko izmanto, lai noteiktu, vai automātiskā MongoDB repliku kopas mezglu atklāšana ir jāiespējo vai jāatspējo. Atbalstītās vērtības: `true` - spraudnis pieslēgsies tikai norādītajam serverim, ignorējot citus repliku kopas dalībniekus; `false` - spraudnis automātiski atklās visus repliku kopas mezglus. Ja tiek izmantota shēma `tcp://` (novecojusi), parametrs pēc noklusējuma tiek iestatīts uz `true`.
Plugins.MongoDB.Default.AuthMechanism				Noklusējuma autentifikācijas mehānisms pieslēgšanai MongoDB serverim. Atbalstītās vērtības: `SCRAM-SHA-1`, `SCRAM-SHA-256` un `MONGODB-X509`. Ja vērtība nav norādīta ne šeit, ne vienuma atslēgā, ne nosauktajā sesijā, ne URI, tiks izmantots uz SCRAM balstīts mehānisms (ja ir norādīts lietotājvārds; pretējā gadījumā autentifikācija netiks izmantota).
Plugins.MongoDB.Default.AuthSource			`admin`	Noklusējuma datubāze savienojuma autentificēšanai ar MongoDB serveri. Ja tiek izmantots autentifikācijas mehānisms `MONGODB-X509`, šis parametrs tiek automātiski iestatīts uz `$external`; tā mainīšana uz jebkuru citu vērtību izraisīs kļūdu palaišanas laikā.

Parameter	Mandatory	Range	Default	Description
Plugins.MongoDB.Default.ReplicaSet				Noklusējuma repliku kopas nosaukums; tiek izmantots MongoDB repliku kopas mezglu automātiskai atklāšanai. Ja šis parametrs tiek izmantots ar shēmu `mongodb+srv://`, tas darbosies tā, it kā būtu norādīti vairāki URL; DNS serverim var būt TXT ieraksts, kas var ignorēt šo (vai jebkuru citu) parametru. Nav pieejams, ja parametrs `Plugins.MongoDB.Default.Direct` ir iestatīts uz `true`; aģentu neizdosies palaist.
Plugins.MongoDB.Default.ReadPreference				Noklusējuma secība pieslēgšanai MongoDB repliku kopas mezgliem vai atkāpšanās variants. Atbalstītās vērtības: `PrimaryMode`, `PrimaryPreferredMode`, `SecondaryMode`, `SecondaryPreferredMode` un `NearestMode`. Nav pieejams, ja parametrs `Plugins.MongoDB.Default.Direct` ir iestatīts uz `true`; aģentu neizdosies palaist.
Plugins.MongoDB.Default.TLSConnect				Noklusējuma šifrēšanas veids saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm. Atbalstītās vērtības: `required` - pieprasa TLS savienojumu. `verify\_ca` - pārbauda sertifikātus. `verify\_full` - pārbauda sertifikātus un IP adresi. Ja vērtība nav norādīta ne šeit, ne vienuma atslēgā, ne nosauktajā sesijā, ne URI, TLS savienojums netiks izveidots.
Plugins.MongoDB.Default.TLSCAFile				Noklusējuma ceļš uz failu, kas satur augstākā līmeņa CA sertifikātu(-us) otras puses sertifikāta pārbaudei. Tiek izmantots šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
Plugins.MongoDB.Default.TLSCertFile				Noklusējuma ceļš uz failu, kas satur MongoDB klienta sertifikātu vai sertifikātu ķēdi. Tiek izmantots šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.
Plugins.MongoDB.Default.TLSKeyFile				Noklusējuma ceļš uz failu, kas satur MongoDB klienta privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix aģentu 2 un uzraudzītajām datubāzēm.

Skatiet arī:

- Vispārīgo Zabbix aģenta 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

7 MQTT spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots sprauduņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti MQTT sprauduņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `mqtt.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.MQTT.Default.Password	nē			Noklusējuma parole savienojumam ar MQTT; tiek izmantota, ja vērtība nav norādīta vienuma atslēgā vai nosauktajā sesijā.
Plugins.MQTT.Default.TLSCAFile	nē			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri; tiek izmantots, ja vērtība nav norādīta nosauktajā sesijā.
Plugins.MQTT.Default.TLSCertFile	nē			Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri; tiek izmantots, ja vērtība nav norādīta nosauktajā sesijā.
Plugins.MQTT.Default.TLSKeyFile	nē			Pilns ceļš uz failu, kas satur MQTT privāto atslēgu šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri; tiek izmantots, ja vērtība nav norādīta nosauktajā sesijā.

Plugins.MQTT.Default.Topic	nē			Noklusējuma tēma MQTT abonēšanai; tiek izmantota, ja vērtība nav norādīta vienuma atslēgā vai nosauktajā sesijā.
				
				Tēma var saturēt aizstājējzīmes ("+","#")
				Piemēri: `path/to/file`
				`path/to/#`
				`path/+/topic`

Plugins.MQTT.Default.Url	nē		tcp://localhost:1883	Noklusējuma MQTT brokera savienojuma virkne; tiek izmantota, ja vērtība nav norādīta vienuma atslēgā vai nosauktajā sesijā.
				
				Nedrīkst ietvert vaicājuma parametrus.
				Jāatbilst URL formātam.
				Atbalstītās shēmas: `tcp` (noklusējuma), `ws`, `tls`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=1883).
				Piemēri: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Default.User	nē			Noklusējuma lietotājvārds savienojumam ar MQTT; tiek izmantots, ja vērtība nav norādīta vienuma atslēgā vai nosauktajā sesijā.
Plugins.MQTT.Sessions.<SessionName>.Password	nē			Nosauktās sesijas parole. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MQTT.Sessions.<SessionName>.TLSCAFile	nē			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MQTT.Sessions.<SessionName>.TLSCertFile	nē			Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.MQTT.Sessions.<SessionName>.TLSKeyFile	nē			Pilns ceļš uz failu, kas satur MQTT privāto atslēgu, ko izmanto šifrētai saziņai starp Zabbix aģents 2 un MQTT brokeri. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.MQTT.Sessions.<SessionName>.Topic	nē			Nosauktās sesijas tēma MQTT abonēšanai.
				<SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Tēma var saturēt aizstājējzīmes ("+","#")
				Piemēri: `path/to/file`
				`path/to/#`
				`path/+/topic`

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts

Plugins.MQTT.Sessions.<SessionName>.Url				Nosauktās sesijas savienojuma virkne.
				<SessionName> - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Nedrīkst ietvert vaicājuma parametrus.
				Jāatbilst URL formātam.
				Atbalstītās shēmas: `tcp` (noklusējuma), `ws`, `tls`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=1883).
				Piemēri: `tcp://host:1883`
				`localhost`
				`ws://host:8080`

Plugins.MQTT.Sessions.<SessionName>.User				Nosauktās sesijas lietotājvārds. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienumu atslēgās.

Ja nepieciešams iestatīt pieprasījuma izpildes noildzi (cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā apturēšanas), izmantojiet vienuma konfigurācijas formu.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

8 MSSQL spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti MSSQL spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `mssql.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

MSSQL spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts MSSQL plugin repository.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Opcijas

Parameter	Description
-V --version	Izdrukāt spraudņa versiju un licences informāciju.
-h --help	Izdrukāt palīdzības informāciju (saīsinātā forma).
-t, --test <item key>	Palaist spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.MSSQL.CustomQueriesDir	Nē		`/usr/local/share/zabbix/custom_queries/mssql` Unix sistēmām `*:\Program Files\Zabbix Agent 2\Custom Queries\MSSQL` Windows sistēmām, kur `*` ir diska nosaukums no `ProgramFiles` vides mainīgā	Norāda ceļu uz direktoriju, kurā atrodas lietotāja definēti .sql faili ar pielāgotiem vaicājumiem, kurus spraudnis var izpildīt. Spraudnis startēšanas laikā ielādē visus pieejamos .sql failus konfigurētajā direktorijā. Tas nozīmē, ka jebkādas izmaiņas pielāgoto vaicājumu failos netiks atspoguļotas, kamēr spraudnis netiks restartēts. Spraudnis tiek palaists un apturēts kopā ar Zabbix aģents 2.
Plugins.MSSQL.CustomQueriesEnabled	Nē		false	Ja iestatīts, iespējo `mssql.custom.query` vienums atslēgas izpildi. Ja atspējots, no pielāgoto vaicājumu direktorijas ceļa netiks ielādēts neviens vaicājums.
Plugins.MSSQL.Default.CACertPath	Nē			Noklusējuma faila ceļš uz sertifikātu iestādes (CA) publiskās atslēgas sertifikātu, kas izdevusi MSSQL serveris sertifikātu. Sertifikātam jābūt PEM formātā.
Plugins.MSSQL.Default.Database	Nē			Noklusējuma datubāzes nosaukums, kurai pieslēgties.
Plugins.MSSQL.Default.Encrypt	Nē			Norāda noklusējuma savienojuma šifrēšanas tipu. Iespējamās vērtības: *true* - datu sūtīšana starp spraudni un serveri ir šifrēta; *false* - datu sūtīšana starp spraudni un serveri nav šifrēta ārpus pieteikšanās paketes; *strict* - datu sūtīšana starp spraudni un serveri ir E2E šifrēta, izmantojot TDS8; *disable* - datu sūtīšana starp spraudni un serveri nav šifrēta.
Plugins.MSSQL.Default.HostNameInCertificate	Nē			MSSQL serveris sertifikāta vispārpieņemtais nosaukums (CN) pēc noklusējuma.
Plugins.MSSQL.Default.Password	Nē			Parole, kas pēc noklusējuma tiks nosūtīta aizsargātam MSSQL serveris.
Plugins.MSSQL.Default.TLSMinVersion	Nē			Minimālā TLS versija, ko izmantot pēc noklusējuma. Iespējamās vērtības: `1.0`, `1.1`, `1.2`, `1.3`.
Plugins.MSSQL.Default.TrustServerCertificate	Nē			Vai spraudnim pēc noklusējuma uzticēties servera sertifikātam bez tā validēšanas. Iespējamās vērtības: `true`, `false`.
Plugins.MSSQL.Default.Uri	Nē		sqlserver://localhost:1433	Noklusējuma URI savienojumam. Vienīgā atbalstītā shēma ir `sqlserver://`. Shēmu var izlaist. Iegultie akreditācijas dati tiks ignorēti.
Plugins.MSSQL.Default.User	Nē			Noklusējuma lietotājvārds, kas tiks nosūtīts aizsargātam MSSQL serveris.
Plugins.MSSQL.KeepAlive	Nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pirms neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.MSSQL.Sessions.<SessionName>.CACertPath	Nē			Faila ceļš uz sertifikātu iestādes (CA) publiskās atslēgas sertifikātu, kas izdevusi MSSQL serveris sertifikātu nosauktajai sesijai. Sertifikātam jābūt PEM formātā. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.Database	Nē			Datubāzes nosaukums, kurai pieslēgties nosauktajai sesijai. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.MSSQL.Sessions.<SessionName>.Encrypt	Nē			Norāda savienojuma šifrēšanas tipu nosauktajai sesijai. Iespējamās vērtības: *true* - datu sūtīšana starp spraudni un serveri ir šifrēta; *false* - datu sūtīšana starp spraudni un serveri nav šifrēta ārpus pieteikšanās paketes; *strict* - datu sūtīšana starp spraudni un serveri ir E2E šifrēta, izmantojot TDS8; *disable* - datu sūtīšana starp spraudni un serveri nav šifrēta. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.HostNameInCertificate	Nē			MSSQL serveris sertifikāta vispārpieņemtais nosaukums (CN) nosauktajai sesijai. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.Password	Nē			Parole, kas tiks nosūtīta aizsargātam MSSQL serveris nosauktajai sesijai. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.TLSMinVersion	Nē			Minimālā TLS versija, ko izmantot nosauktajai sesijai. Iespējamās vērtības: `1.0`, `1.1`, `1.2`, `1.3`. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.TrustServerCertificate	Nē			Vai spraudnim uzticēties servera sertifikātam bez tā validēšanas nosauktajai sesijai. Iespējamās vērtības: `true`, `false`. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.Uri	Nē		sqlserver://localhost:1433	URI savienojumam nosauktajai sesijai. Vienīgā atbalstītā shēma ir `sqlserver://`. Shēmu var izlaist. Iegultie akreditācijas dati tiks ignorēti. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.Sessions.<SessionName>.User	Nē			Lietotājvārds, kas tiks nosūtīts aizsargātam MSSQL serveris nosauktajai sesijai. **<SessionName>** - definējiet sesijas nosaukumu izmantošanai vienums atslēgās.
Plugins.MSSQL.System.Path	Nē			Ceļš uz MSSQL spraudņa izpildāmo failu. Globāls iestatījums MSSQL spraudnim. Tiek piemērots visiem savienojumiem. Piemērs: `Plugins.MSSQL.System.Path=/usr/libexec/zabbix/zabbix-agen`
Plugins.MSSQL.Timeout	Nē	1-30	globālais taimauts	Ilgums sekundēs, cik ilgi gaidīt servera atbildi, pirmoreiz pieslēdzoties un veicot turpmākās darbības sesijā.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

9 MySQL spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti MySQL spraudņa darbībai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `mysql.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;

- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parameter	Mandatory	Range	Default	Description
Plugins.Mysql.CallTimeout	nē	1-30	globālais taimauts	Maksimālais laiks sekundēs, cik ilgi gaidīt, līdz pieprasījums tiek izpildīts.
Plugins.Mysql.CustomQueriesPath	nē		`/usr/local/share/zabbix/external_scripts/mysql` Unix sistēmām `*:\Program Files\Zabbix Agent 2\Custom Queries\Mysql` Windows sistēmām, kur * ir diska nosaukums no `ProgramFiles` vides mainīgā	Pilns ceļš uz direktoriju, kurā atrodas **mysql** faili ar pielāgotiem vaicājumiem.
Plugins.Mysql.CustomQueriesEnabled	nē		false	Ja iestatīts, iespējo `mysql.custom.query` vienuma atslēgas izpildi. Ja atspējots, no pielāgoto vaicājumu direktorijas ceļa netiks ielādēts neviens vaicājums.
Plugins.Mysql.Default.Password	nē			Noklusējuma parole savienojumam ar MySQL; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Mysql.Default.TLSCAFile	nē (jā, ja Plugins.Mysql.Default.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.
Plugins.Mysql.Default.TLSCertFile	nē (jā, ja Plugins.Mysql.Default.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.
Plugins.Mysql.Default.TLSConnect	nē			Šifrēšanas veids saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība. Atbalstītās vērtības: *required* - pieprasīt TLS savienojumu; *verify_ca* - pārbaudīt sertifikātus; *verify_full* - pārbaudīt sertifikātus un IP adresi.
Plugins.Mysql.Default.TLSKeyFile	nē (jā, ja Plugins.Mysql.Default.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns ceļš uz failu, kas satur datubāzes privāto atslēgu šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Parameter	Mandatory	Range	Default	Description

Plugins.Mysql.Default.Uri	nē		tcp://localhost:3306	Noklusējuma URI savienojumam ar MySQL; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
				
				Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=3306).
				Piemēri: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Default.User	nē			Noklusējuma lietotājvārds savienojumam ar MySQL; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Mysql.KeepAlive	nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pirms neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.Mysql.Sessions.<SessionName>.Password	nē			Nosauktās sesijas parole. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Mysql.Sessions.<SessionName>.TLSCAFile	nē (jā, ja Plugins.Mysql.Sessions.<SessionName>.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Mysql.Sessions.<SessionName>.TLSCertFile	jā, ja norādīts Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile			Pilns ceļš uz failu, kas satur aģenta sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Mysql.Sessions.<SessionName>.TLSConnect	nē			Šifrēšanas veids saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: *required* - pieprasīt TLS savienojumu; *verify_ca* - pārbaudīt sertifikātus; *verify_full* - pārbaudīt sertifikātus un IP adresi.
Plugins.Mysql.Sessions.<SessionName>.TLSKeyFile	jā, ja norādīts Plugins.Mysql.Sessions.<SessionName>.TLSCertFile			Pilns ceļš uz failu, kas satur datubāzes privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.Mysql.Sessions.<SessionName>.Uri	nē			Nosauktās sesijas savienojuma virkne.
				<SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=3306).
				Piemēri: `tcp://localhost:3306`
				`localhost`
				`unix:/var/run/mysql.sock`

Plugins.Mysql.Sessions.<SessionName>.User	nē			Nosauktās sesijas lietotājvārds. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Mysql.Timeout	nē	1-30	globālais taimauts	Maksimālais gaidīšanas laiks sekundēs, kad ir jāizveido savienojums.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

10 NVIDIA GPU spraudnis

Pārskats

Zabbix aģenta 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģenta 2 konfigurācijas parametri tiek atbalstīti NVIDIA GPU spraudņa darbībai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `nvidiagpu.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģenta 2 konfigurācijai.

NVIDIA GPU spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts NVIDIA GPU plugin repository.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.NVIDIA.System.Path	Jā			Ceļš uz NVIDIA GPU spraudņa izpildāmo failu. Lietošanas piemērs: `Plugins.NVIDIA.System.Path=/path/to/executable/nvidia`
Plugins.NVIDIA.Timeout	Nē	1-30	globālais taimauts	Norāda maksimālo laiku (sekundēs), cik ilgi gaidīt serveris atbildi savienojuma mēģinājumu laikā un turpmāko darbību laikā sesijā. Globālais vienuma tipa taimauts vai individuālais vienuma taimauts aizstās šo vērtību, ja tas būs lielāks.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

11 Oracle spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti Oracle spraudņa darbībai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `oracle.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Oracle.CallTimeout	Nē	1-30	globālais taimauts	Maksimālais gaidīšanas laiks sekundēs, līdz pieprasījums tiek pabeigts.
Plugins.Oracle.ConnectTimeout	Nē	1-30	globālais taimauts	Maksimālais gaidīšanas laiks sekundēs, līdz savienojums tiek izveidots.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Oracle.CustomQueriesPath	Nē		/usr/local/share/zabbix/custom_queries/oracle Unix sistēmām *:\Program Files\Zabbix Agent 2\Custom Queries\Oracle Windows sistēmām, kur * ir diska nosaukums no vides mainīgā ProgramFiles	Pilns ceļš uz direktoriju, kurā ir .sql faili ar pielāgotiem vaicājumiem. Piemērs: /etc/zabbix/oracle/sql
Plugins.Oracle.CustomQueriesEnabled	Nē		false	Ja iestatīts, iespējo oracle.custom.query vienuma atslēgas izpildi. Ja atspējots, no pielāgoto vaicājumu direktorijas ceļa netiks ielādēts neviens vaicājums.
Plugins.Oracle.Default.Password	Nē			Noklusējuma parole savienojumam ar Oracle; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Oracle.Default.Service	Nē			Noklusējuma servisa nosaukums savienojumam ar Oracle (SID netiek atbalstīts); tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Oracle.Default.Uri	Nē			Noklusējuma URI savienojumam ar Oracle; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība. Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti). Jāatbilst URI formātam. Tiek atbalstītas tikai tcp un tcps shēmas; shēmu var nenorādīt. Portu var nenorādīt (noklusējums=1521). Kā savienojuma virkni iespējams norādīt arī TNS atslēgu vai TNS vērtību. TNS vērtībai jābūt izveidotai bez atstarpēm. Piemēri: tcp://127.0.0.1:1521 localhost zbx_tns_example (TNS atslēga) (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR (TNS vērtība)
Plugins.Oracle.Default.User	Nē			Noklusējuma lietotājvārds savienojumam ar Oracle; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Oracle.KeepAlive	Nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pēc kura neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.Oracle.ResolveTNS	Nē	true / false	true	Opcija nosaka, kā interpretēt savienojuma virkni (ConnString), lai izveidotu savienojumu ar Oracle serveri. Ja tā ir iestatīta uz false, ConnString tiks pievienota noklusējuma shēma un ports (ja to nav). Ja opcija ir iestatīta uz true, noklusējuma shēma un ports tiks izlaisti (ja vien tie nav skaidri norādīti ConnString), un ConnString tiks nodota Oracle klientam tāda, kāda tā ir. Ja Oracle klients atrod ConnString failā tnsnames.ora, savienojumam ar Oracle serveri tiks izmantots atrastais savienojuma apraksts.
Plugins.Oracle.Sessions.<SessionName>.Password	Nē			Nosauktās sesijas parole. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.Oracle.Sessions.<SessionName>.Service	Nē			Nosauktās sesijas servisa nosaukums, kas tiks izmantots savienojumam (SID netiek atbalstīts). **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts

Plugins.Oracle.Sessions.<SessionName>.Uri	Nē			Nosauktās sesijas Oracle savienojuma virkne.
				<SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Nedrīkst ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Tiek atbalstītas tikai `tcp` un `tcps` shēmas; shēmu var nenorādīt.
				Portu var nenorādīt (noklusējums=1521).
				Kā savienojuma virkni iespējams norādīt arī TNS atslēgu vai TNS vērtību. TNS vērtībai jābūt izveidotai bez atstarpēm.
				Piemēri: `tcp://127.0.0.1:1521`
				`localhost`
				`zbx_tns_example` (TNS atslēga)
				`(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(POR`
				(TNS vērtība)

Plugins.Oracle.Sessions.<SessionName>.User	Nē			Nosauktās sesijas lietotājvārds. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

12 PostgreSQL spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti PostgreSQL spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `postgresql.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

PostgreSQL spraudnis ir ielādējams spraudnis, un tas ir pieejams un pilnībā aprakstīts PostgreSQL plugin repository.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Opcijas

Parametrs	Apraksts
-V --version	Izdrukā spraudņa versiju un licences informāciju.
-h --help	Izdrukā palīdzības informāciju (saīsinātā forma).
-t, --test <item key>	Palaiž spraudni testēšanai (spraudņa konfigurācija tiek ignorēta).

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.PostgreSQL.Default.CacheMode			prepare	Kešatmiņas režīms PostgreSQL savienojumam. Atbalstītās vērtības: *prepare* (noklusējums) - izveidos sagatavotos priekšrakstus PostgreSQL serverī; *describe* - izmantos anonīmo sagatavoto priekšrakstu, lai aprakstītu priekšrakstu, neveidojot priekšrakstu serverī. Ņemiet vērā, ka "describe" galvenokārt ir noderīgs, ja vide neatļauj sagatavotos priekšrakstus, piemēram, izmantojot savienojumu pūleri, piemēram, PgBouncer.
Plugins.PostgreSQL.CallTimeout		1-30	globālais taimauts	Maksimālais gaidīšanas laiks (sekundēs), līdz pieprasījums tiek pabeigts.
Plugins.PostgreSQL.CustomQueriesPath			`/usr/local/share/zabbix/custom_queries/postgresql` Unix sistēmām `*:\Program Files\Zabbix Agent 2\Custom Queries\PostgreSQL` Windows sistēmām, kur * ir diska nosaukums no `ProgramFiles` vides mainīgā	Pilns direktorija ceļš, kur atrodas pielāgotiem vaicājumiem.
Plugins.PostgreSQL.CustomQueriesEnabled			false	Ja iestatīts, iespējo `postgresql.custom.query` vienuma atslēgas izpildi. Ja atspējots, no pielāgoto vaicājumu direktorija ceļa netiks ielādēts neviens vaicājums.
Plugins.PostgreSQL.Default.Database				Noklusējuma datubāze savienojumam ar PostgreSQL; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.PostgreSQL.Default.Password				Noklusējuma parole savienojumam ar PostgreSQL; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.PostgreSQL.Default.TLSCAFile				Pilns faila ceļš uz failu, kas satur augstākā līmeņa CA sertifikātu(-us) partnera sertifikāta pārbaudei šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.
Plugins.PostgreSQL.Default.TLSCertFile				Pilns faila ceļš uz failu, kas satur PostgreSQL sertifikātu vai sertifikātu ķēdi šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.
Plugins.PostgreSQL.Default.TLSConnect				Šifrēšanas veids saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība. Atbalstītās vērtības: *required* - izveidot savienojumu, izmantojot TLS kā transporta režīmu bez identitātes pārbaudēm; *verify_ca* - izveidot savienojumu, izmantojot TLS, un pārbaudīt sertifikātu; *verify_full* - izveidot savienojumu, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, ka DBHost norādītā datubāzes identitāte (CN) atbilst tās sertifikātam. Nenorādīts šifrēšanas veids nozīmē nešifrētu savienojumu.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.PostgreSQL.Default.TLSKeyFile	Nē (jā, ja Plugins.PostgreSQL.Default.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns faila ceļš uz failu, kas satur PostgreSQL privāto atslēgu šifrētai saziņai starp Zabbix aģents 2 un uzraudzītajām datubāzēm; tiek izmantots, ja nosauktajā sesijā nav norādīta vērtība.

Plugins.PostgreSQL.Default.Uri	Nē			Noklusējuma URI savienojumam ar PostgreSQL; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
				
				Tam nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`.
				Piemēri: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Default.User	Nē			Noklusējuma lietotājvārds savienojumam ar PostgreSQL; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.PostgreSQL.KeepAlive	Nē	60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pēc kura neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.PostgreSQL.Sessions.<SessionName>.CachePrepMode	Nē		*prepare*	Kešatmiņas režīms PostgreSQL savienojumam. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: *prepare* (noklusējums) - izveidos sagatavotos priekšrakstus PostgreSQL serverī; *describe* - izmantos anonīmo sagatavoto priekšrakstu, lai aprakstītu priekšrakstu, neveidojot priekšrakstu serverī. Ņemiet vērā, ka "describe" galvenokārt ir noderīgs, ja vide neatļauj sagatavotos priekšrakstus, piemēram, izmantojot savienojumu pūleri, piemēram, PgBouncer.
Plugins.PostgreSQL.Sessions.<SessionName>.Database	Nē			Datubāze sesijas savienojumam. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.PostgreSQL.Sessions.<SessionName>.Password	Nē	Jāatbilst paroles formātam.		Parole sesijas savienojumam. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCAFile	Nē (jā, ja Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect ir iestatīts uz *verify_ca* vai *verify_full*)			Pilns faila ceļš uz failu, kas satur augstākā līmeņa CA sertifikātu(-us) partnera sertifikāta pārbaudei. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile	Jā, ja norādīts Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile			Pilns faila ceļš uz failu, kas satur PostgreSQL sertifikātu vai sertifikātu ķēdi. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.PostgreSQL.Sessions.<SessionName>.TLSConnect	Nē			Šifrēšanas veids PostgreSQL savienojumam. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās. Atbalstītās vērtības: *required* - izveidot savienojumu, izmantojot TLS kā transporta režīmu bez identitātes pārbaudēm; *verify_ca* - izveidot savienojumu, izmantojot TLS, un pārbaudīt sertifikātu; *verify_full* - izveidot savienojumu, izmantojot TLS, pārbaudīt sertifikātu un pārbaudīt, ka DBHost norādītā datubāzes identitāte (CN) atbilst tās sertifikātam. Nenorādīts šifrēšanas veids nozīmē nešifrētu savienojumu.

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.PostgreSQL.Sessions.<SessionName>.TLSKeyFile	Ja, ja nav norādīts Plugins.PostgreSQL.Sessions.<SessionName>.TLSCertFile			Pilns faila ceļš uz failu, kas satur PostgreSQL privāto atslēgu. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.

Plugins.PostgreSQL.Sessions.<SessionName>.Uri	Nē			Nosauktās sesijas savienojuma virkne.
				<SessionName> - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
				
				Tam nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`.
				Piemēri: `tcp://127.0.0.1:5432`
				`tcp://localhost`
				`unix:/var/run/postgresql/.s.PGSQL.5432`

Plugins.PostgreSQL.Sessions.<SessionName>.User	Nē			Nosauktās sesijas lietotājvārds. **<SessionName>** - definē sesijas nosaukumu izmantošanai vienumu atslēgās.
Plugins.PostgreSQL.System.Path	Jā			Ceļš uz PostgreSQL spraudņa izpildāmo failu. Piemērs: `Plugins.PostgreSQL.System.Path=/usr/libexec/zabbix/zabbi`
Plugins.PostgreSQL.Timeout	Nē	1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi spraudņu konfigurēšanai

13 Redis spraudnis

Pārskats

Zabbix aģenta 2 konfigurācijas fails tiek izmantots spraudņu konfigurēšanai. Šie Zabbix aģenta 2 konfigurācijas parametri tiek atbalstīti Redis spraudņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `redis.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģenta 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības.
- Vērtības atbalsta vides mainīgos.
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM.
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.Password	Nē			Noklusējuma parole savienojumam ar Redis; tiek izmantota, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Redis.Default.TLSConnect	Nē			Klienta (Zabbix aģents 2) TLS verifikācijas prasība. Atbalstītās vērtības: `required` - šifrēts, neverificēts (tikai testēšanai) `verify_ca` - šifrēts, servera sertifikāts `verify_full` - šifrēts, servera sertifikāts un servera nosaukums verificēts ar SAN Tukša vai neiestatīta vērtība tiek uzskatīta par bez tls.
Plugins.Redis.Default.TLSCAFile	Nē			Noklusējuma pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus.
Plugins.Redis.Default.TLSCertFile	Nē			Noklusējuma pilns ceļš uz failu, kas satur aģenta sertifikātu.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Default.TLSKeyFile				Noklusējuma pilns ceļš uz failu, kas satur aģenta privāto atslēgu.

Plugins.Redis.Default.Uri			tcp://localhost:6379	Noklusējuma URI savienojumam ar Redis; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
				
				Tam nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var izlaist.
				Portu var izlaist (noklusējums=6379).
				Piemēri: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Default.User			default	Noklusējuma lietotājs, ko nosūtīt aizsargātajam Redis serverim; tiek izmantots, ja vienuma atslēgā vai nosauktajā sesijā nav norādīta vērtība.
Plugins.Redis.KeepAlive		60-900	300	Maksimālais gaidīšanas laiks (sekundēs), pirms neizmantotie spraudņa savienojumi tiek aizvērti.
Plugins.Redis.Sessions.<SessionName>.Password				Nosauktās sesijas parole. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.
Plugins.Redis.Sessions.<SessionName>.TLSConnect				Klienta (Zabbix aģents 2) TLS verifikācijas prasība. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās. Atbalstītās vērtības: `required` - šifrēts, neverificēts (tikai testēšanai) `verify_ca` - šifrēts, servera sertifikāts `verify_full` - šifrēts, servera sertifikāts un servera nosaukums verificēts ar SAN Tukša vai neiestatīta vērtība tiek uzskatīta par bez tls.
Plugins.Redis.Sessions.<SessionName>.TLSCAFile	(jā, ja TLSConnect ir `verify_ca` vai `verify_full`)			Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus Redis servera sertifikāta verifikācijai. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.
Plugins.Redis.Sessions.<SessionName>.TLSCertFile				Pilns ceļš uz failu, kas satur aģenta sertifikātu klienta autentifikācijai. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.
Plugins.Redis.Sessions.<SessionName>.TLSKeyFile				Pilns ceļš uz failu, kas satur aģenta privāto atslēgu klienta autentifikācijai. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.

Plugins.Redis.Sessions.<SessionName>.Uri			localhost:6379	Nosauktās sesijas savienojuma virkne.
				<SessionName> - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.
				
				Tai nevajadzētu ietvert iegultus akreditācijas datus (tie tiks ignorēti).
				Jāatbilst URI formātam.
				Atbalstītās shēmas: `tcp`, `unix`; shēmu var izlaist.
				Portu var izlaist (noklusējums=6379).
				Piemēri: `tcp://localhost:6379`
				`localhost`
				`unix:/var/run/redis.sock`

Plugins.Redis.Sessions.<SessionName>.User			default	Lietotājs, ko nosūtīt aizsargātajam Redis serverim. **<SessionName>** - definējiet sesijas nosaukumu, kas tiks izmantots vienumu atslēgās.

Parameter	Mandatory	Range	Default	Description
Plugins.Redis.Timeout	nē	1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā pārtraukšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix agent 2 (UNIX) / Zabbix agent 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

14 SMART spraudnis

Pārskats

Zabbix aģents 2 konfigurācijas fails tiek izmantots sprauduņu konfigurēšanai. Šie Zabbix aģents 2 konfigurācijas parametri tiek atbalstīti SMART sprauduņa darbināšanai.

Ieteicams tos norādīt atsevišķā konfigurācijas failā (piemēram, `smart.conf`) un pēc tam izmantot Include direktīvu, lai pievienotu šo failu Zabbix aģents 2 konfigurācijai.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Ceļš uz `smartctl` izpildāmo failu ir jānorāda, vai nu pievienojot tā direktoriju sistēmas `PATH` vides mainīgajam, vai konfigurējot `Plugins.Smart.Path`; tas attiecas gan uz Linux, gan Windows;
- Zabbix atbalsta konfigurācijas failus tikai UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar "#", tiek atbalstīti tikai rindas sākumā.

Parametri

Parametrs	Obligāts	Diapazons	Noklusējums	Apraksts
Plugins.Smart.Path	nē		smartctl	Ceļš uz smartctl izpildāmo failu.
Plugins.Smart.Timeout	nē	1-30	globālais taimauts	Pieprasījuma izpildes taimauts (ilgums sekundēs, cik ilgi gaidīt pieprasījuma pabeigšanu pirms tā apturēšanas).

Skatiet arī:

- Vispārīgo Zabbix aģents 2 konfigurācijas parametru apraksts: Zabbix aģents 2 (UNIX) / Zabbix aģents 2 (Windows)
- Norādījumi sprauduņu konfigurēšanai

8 Zabbix Java gateway

Ja izmantojat `startup.sh` un `shutdown.sh` skriptus, lai palaistu Zabbix Java gateway, tad nepieciešamos konfigurācijas parametrus varat norādīt failā `settings.sh`. Palaišanas un apturēšanas skripti ielādē iestatījumu failu un nodrošina čaulas mainīgo (uzskaitīti pirmajā kolonnā) pārveidošanu par Java īpašībām (uzskaitītas otrajā kolonnā).

Ja palaižat Zabbix Java gateway manuāli, tieši izpildot `java`, tad atbilstošās Java īpašības norādāt komandrindā.

Variable	Property	Mandatory	Range	Default	Description
LISTEN_IP	zabbix.listenIP	no		0.0.0.0	IP adrese, kurā klausīties.
LISTEN_PORT	zabbix.listenPort	no	1024-32767	10052	Ports, kurā klausīties.
PID_FILE	zabbix.pidFile	no		/tmp/zabbix_java.pid	PID faila nosaukums. Ja tas nav norādīts, Zabbix Java Gateway tiek palaists kā konsoles lietojumprogramma.

Variable	Property	Mandatory	Range	Default	Description
PROPERTIES_FILE	zabbix.propertiesFile	no			Īpašību faila nosaukums. Var izmantot, lai iestatītu papildu īpašības, izmantojot atslēgas-vērtības formātu, tādējādi tās nav redzamas komandrindā, vai lai pārrakstītu esošās. Piemēram: ”javax.net.ssl.trustStorePassword=<p
START_POLLERS	zabbix.startPollers	no	1-1000	5	Palaižamo darba pavedienu skaits.
TIMEOUT	zabbix.timeout	no	1-30	3	Cik ilgi gaidīt tīkla operācijas (sekundēs).

Warning:
Ports 10052 nav IANA reģistrēts.

9 Zabbix tīmekļa pakalpojums

Pārskats

Zabbix tīmekļa pakalpojums ir process, ko izmanto saziņai ar ārējiem tīmekļa pakalpojumiem.

Šajā sadaļā ir uzskaitīti parametri, ko atbalsta Zabbix tīmekļa pakalpojuma konfigurācijas fails (zabbix_web_service.conf).

Parametri ir uzskaitīti bez papildu informācijas. Noklikšķiniet uz parametra, lai skatītu pilnu informāciju.

Parametrs	Apraksts
AllowedIP	Ar komatiem atdalīts Zabbix serveru un Zabbix starpniekserveru IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi.
DebugLevel	Atkļūdošanas līmenis.
IgnoreURLCertErrors	Norāda TLS sertifikāta validācijas kļūdu apstrādi, piekļūstot lietotāja saskarnes URL.
Include	Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā.
ListenPort	Pakalpojums šajā portā gaidīs savienojumus no servera.
LogFile	Žurnālfaila nosaukums.
LogFileSize	Žurnālfaila maksimālais izmērs.
LogType	Žurnāla izvades veids.
Timeout	Maksimālais laiks (sekundēs), kas tiek patērēts paneļa PDF atskaites formatēšanai.
TLSAccept	Kādus ienākošos savienojumus pieņemt.
TLSCAFile	Pilns ceļš uz failu, kurā ir augstākā līmeņa CA sertifikāti vienādranga sertifikāta pārbaudei un kas tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSCertFile	Pilns ceļš uz failu, kurā ir pakalpojuma sertifikāts vai sertifikātu ķēde un kas tiek izmantots šifrētai saziņai starp Zabbix komponentiem.
TLSKeyFile	Pilns ceļš uz failu, kurā ir pakalpojuma privātā atslēga un kas tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

Visi parametri nav obligāti, ja vien nav skaidri norādīts, ka parametrs ir obligāts.

Ņemiet vērā:

- Noklusējuma vērtības atspoguļo procesa noklusējuma vērtības, nevis piegādātajos konfigurācijas failos norādītās vērtības;
- Vērtības atbalsta vides mainīgos;
- Zabbix atbalsta tikai konfigurācijas failus UTF-8 kodējumā bez BOM;
- Komentāri, kas sākas ar ”#”, tiek atbalstīti tikai rindas sākumā.

Parametru informācija

AllowedIP

Ar komatiem atdalīts Zabbix serveru un Zabbix starpniekserveru IP adrešu saraksts, pēc izvēles CIDR pierakstā, vai DNS nosaukumi. Ienākošie savienojumi tiks pieņemti tikai no šeit norādītajiem hostiem.
Ja IPv6 atbalsts ir iespējots, tad `127.0.0.1`, `::127.0.0.1`, `::ffff:127.0.0.1` tiek uzskatītas par vienādām, un `::/0` atļaus jebkuru IPv4 vai IPv6 adresi. `0.0.0.0/0` var izmantot, lai atļautu jebkuru IPv4 adresi.

Piemērs:

```
127.0.0.1,192.168.1.0/24,::1,2001:db8::/32,zabbix.example.com
```

Obligāts: jā

DebugLevel

Norādiet atkļūdošanas līmeni:
0 - pamatinformācija par Zabbix procesu palaišanu un apturēšanu
1 - kritiska informācija;
2 - kļūdu informācija;
3 - brīdinājumi;
4 - atkļūdošanai (rada daudz informācijas);
5 - paplašināta atkļūdošana (rada vēl vairāk informācijas).

Noklusējums: 3
 Diapazons: 0-5

IgnoreURLCertErrors

Norāda TLS sertifikāta validācijas kļūdu apstrādi, piekļūstot lietotāja saskarne URL:
0 - neignorēt sertifikāta kļūdas;
1 - ignorēt sertifikāta kļūdas.

Noklusējums: 0
 Diapazons: 0-1

Iekļaut

Konfigurācijas failā varat iekļaut atsevišķus failus vai visus failus direktorijā. Instalēšanas laikā Zabbix izveidos iekļaušanas direktoriju mapē /usr/local/etc, ja vien tas nav mainīts kompilēšanas laikā. Ceļš var būt relatīvs attiecībā pret *zabbix_web_service.conf* faila atrašanās vietu.
Lai iekļautu tikai atbilstošos failus norādītajā direktorijā, tiek atbalstīta zvaigznītes aizstājējzīme modeļu atbilstības noteikšanai.
Skatiet īpašās piezīmes par ierobežojumiem.

Piemērs:

```
Include=/absolute/path/to/config/files/*.conf
```

ListenPort

Pakalpojums klausīsies šajā portā savienojumiem no servera.

Noklusējums: 10053
 Diapazons: 1024-32767

LogFile

Žurnālfaila nosaukums.

Piemērs:

```
/tmp/zabbix_web_service.log
```

Obligāts: Jā, ja LogType ir iestatīts uz *file*; pretējā gadījumā nē

LogFileSize

Maksimālais žurnālfaila izmērs MB.
0 - atspējot automātisko žurnālfaila rotāciju.
Piezīme: Ja tiek sasniegts žurnālfaila izmēra ierobežojums un faila rotācija jebkāda iemesla dēļ neizdodas, esošais žurnālfails tiek saīsināts un sākts no jauna.

Noklusējums: 1
 Diapazons: 0-1024

LogType

Žurnāla izvades tips:
file - rakstīt žurnālu failā, kas norādīts ar parametru LogFile;
system - rakstīt žurnālu syslog;
console - rakstīt žurnālu standarta izvadē.

Noklusējums: `file`

Noildze

Maksimālais laiks (sekundēs), kas tiek patērēts paneļa PDF atskaites formatēšanai.

Noklusējums: 10
 Diapazons: 1-30

TLSAccept

Kādus ienākošos savienojumus pieņemt:
unencrypted - pieņemt savienojumus bez šifrēšanas (noklusējums)
cert - pieņemt savienojumus ar TLS un sertifikātu

Noklusējums: `unencrypted`

TLSCAFile

Pilns faila ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

TLSCertFile

Pilns ceļš uz failu, kas satur servisa sertifikātu vai sertifikātu ķēdi un tiek izmantots šifrētai saziņai ar Zabbix komponentiem.

TLSKeyFile

Pilns ceļš uz failu, kas satur servisa privāto atslēgu un tiek izmantots šifrētai saziņai starp Zabbix komponentiem.

10 Vides mainīgie

Pārskats

Vides mainīgie ļauj konfigurēt Zabbix komponentus, neiekodējot vērtības konfigurācijas failos. Tas atvieglo konfigurāciju pārvaldību dinamiskās vidēs, piemēram, Docker, kur mainīgos var nodot izpildes laikā, lai pielāgotos dažādām konfigurācijām.

Vienkāršākajā gadījumā, iestatot Zabbix servera DebugLevel konfigurācijas parametra vērtību uz vides mainīgo, to pēc tam var izmantot servera konfigurēšanai palaišanas laikā:

```
#### Zabbix servera konfigurācijas fails:
DebugLevel=${NEW_DEBUG_LEVEL}

#### Zabbix servera palaišana:
NEW_DEBUG_LEVEL=5 /usr/sbin/zabbix_server
```

Vides mainīgos atbalsta šādi Zabbix komponenti:

- Serveris
- Starpniekserveris
- Aģents (UNIX vai Windows)
- Aģents 2 (UNIX vai Windows), ieskaitot spraudņus
- Tīmekļa serviss
- Zabbix sender (izmantojot -c, --config option)

Svarīgas piezīmes

- Ja konfigurācijas parametrs ir iestatīts uz vides mainīgo, kas nav norādīts, palaižot komponentu, tiek izmantota parametra noklusējuma vērtība.
- Izmantojot izpildlaika komandas (piemēram, lai palielinātu aģents žurnāla līmeni), ir jānorāda visi iepriekš izmantotie vides mainīgie. Tas ir tāpēc, ka Zabbix komponenti izmanto savu konfigurācijas failu, lai izpildītu izpildlaika komandas; ja vides mainīgie nav norādīti, tiks izmantotas konfigurācijas parametru noklusējuma vērtības. Skatiet Piemērus.
- `userparameter_reload` izpildlaika komanda neatbalsta vides mainīgo pārlādēšanu. Pārlādes laikā mainīgie tiek ignorēti, un tiek pārlādēti tikai parametri ar parastām vērtībām.
- Procesa pašreizējie vides mainīgie, kas tika izmantoti konfigurācijas failos, pēc Zabbix komponenta palaišanas tiek notīrīti. Tas nodrošina, ka bērnprocesi (piemēram, Zabbix izpildītie attālinātie skripti) nevar piekļūt šiem mainīgajiem. Tomēr ņemiet vērā, ka procesa sākotnējos mainīgos joprojām var iegūt (piemēram, izmantojot failu `/proc/<PID>/environ`).

Sintakse

Vides mainīgajiem jāizmanto šāda sintakse: `${alphanumerics/underscores}`.

Mainīgā nosaukums drīkst ietvert tikai burtus (a-z, A-Z), pasvītrojuma zīmes (_), un ciparus (0-9), un tas nedrīkst sākties ar ciparu.

Mainīgie, kas neatbilst prasītajai sintaksei vai ir apvienoti ar parastu vērtību, tiks apstrādāti kā parastas vērtības, kas var izraisīt kļūdas.

Pareiza mainīgo sintakse:

```
DebugLevel=${NEW_DEBUG_LEVEL}
Hostname=${ZBX_HOSTNAME}
LogFile=${LogFile_001}
```

Nepareiza mainīgo sintakse:

```
DebugLevel=${5_DebugLevel}
Hostname=${ZBX.HOSTNAME 1}
LogFile=/${HOME}/zabbix/zabbix_server.log
```

Note:
Sistēmā Windows vides mainīgo nosaukumos netiek ņemts vērā burtu reģistrs.

Piemēri

Tālāk sniegtie piemēri parāda, kā konfigurēt un izmantot vides mainīgos ar Zabbix komponentiem.

Piemērs 1: Zabbix aģenta konfigurēšana un testēšana

1. Iestatiet vides mainīgos aģenta konfigurācijas failā:

```
Hostname=${ZBX_HOSTNAME}
ServerActive=${ServerActive}
```

2. Pārbaudiet konfigurācijas failu:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

3. Palaidiet aģentu ar vides mainīgajiem:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Izmantojot izpildlaika komandas (piemēram, lai palielinātu aģenta žurnāla līmeni), ir jānorāda visi iepriekš izmantotie vides mainīgie:

```
ZBX_HOSTNAME="New Zabbix agent" ServerActive=127.0.0.1 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agent
```

Tas ir tāpēc, ka aģents izmanto savu konfigurācijas failu, lai izpildītu izpildlaika komandas; ja vides mainīgie netiek norādīti, tiks izmantotas konfigurācijas parametru noklusējuma vērtības.

Alternatīvi, pēc vides mainīgo iestatīšanas aģenta konfigurācijas failā, varat padarīt tos pieejamus procesiem (piemēram, izmantojot komandu `export`). Tas samazina neparedzētas darbības risku trūkstošu vai nepareizi iestatītu mainīgo dēļ.

```
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
```

Piemērs 2: Zabbix aģenta konfigurēšana konteineram

Ja veidojat un konfigurējat savu pielāgoto attēlu Zabbix komponentiem (piemēram, Zabbix aģentam), varat definēt konfigurācijas parametrus, izmantojot vides mainīgos, un pēc tam palaist konteineru ar šiem mainīgajiem.

1. Sagatavojot attēlu, iestatiet vides mainīgos aģenta konfigurācijas failā:

```
Hostname=${ZBX_HOSTNAME}
BufferSize=${BUFSZ}
ListenPort=${LISTENPORT}
UserParameter=${_UsrPar01}
UserParameter=${_UsrPar02}
```

2. Pēc konteinera attēla izveides palaidiet aģenta konteineru (piemēram, Docker) ar vides mainīgajiem:

```
docker run --name my-zabbix-agent -e ZBX_HOSTNAME="new-hostname" -e BUFSZ=1000 -e LISTENPORT=20050 -e _Usr
```

3. Izmantojot izpildlaika komandas (piemēram, lai palielinātu aģenta žurnāla līmeni), piekļūstiet konteinera čaulai un izpildiet izpildlaika komandu:

```
docker exec -it <containerid> sh
/usr/sbin/zabbix_agentd -R log_level_increase
```

Attention:
Izpildlaika komanda `userparameter_reload` neatbalsta vides mainīgo pārlādēšanu. Pārlādes laikā mainīgie tiek ignorēti, un tiek pārlādēti tikai parametri ar parastām vērtībām.

11 Iekļaušana

Pārskats

Papildu failus vai direktorijus var iekļaut servera, starpniekservera, aģenta un tīmekļa pakalpojuma konfigurācijā, izmantojot parametru `Include`.

Piezīmes par iekļaušanu

Ja parametrs `Include` tiek izmantots faila iekļaušanai, failam jābūt lasāmam.

Ja parametrs `Include` tiek izmantots direktorija iekļaušanai:

- Visiem failiem direktorijā jābūt lasāmiem.
- Nevajadzētu pieņemt, ka iekļaušana notiek kādā noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā). Tāpēc nedefinējiet vienu parametru vairākos "Include" failos (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku).
- Visi faili direktorijā tiek iekļauti konfigurācijā.
- Ņemiet vērā failu rezerves kopijas, ko automātiski izveido daži teksta redaktori. Piemēram, ja, rediģējot failu "include/my_specific.conf", tiek izveidota rezerves kopija "include/my_specific_conf.BAK", tad konfigurācijā tiks iekļauti abi faili. Pārvietojiet "include/my_specific.conf.BAK" ārpus "Include" direktorija. Linux vidē "Include" direktorija saturu var pārbaudīt ar komandu "ls -al", lai atrastu nevajadzīgus failus.

Ja parametrs `Include` tiek izmantots failu iekļaušanai, izmantojot šablonu:

- Visiem failiem, kas atbilst šablonam, jābūt lasāmiem.
- Nevajadzētu pieņemt, ka iekļaušana notiek kādā noteiktā secībā (piemēram, faili netiek iekļauti alfabētiskā secībā). Tāpēc nedefinējiet vienu parametru vairākos "Include" failos (piemēram, lai pārrakstītu vispārīgu iestatījumu ar specifisku).

3 Protokoli

Lūdzu, izmantojiet sānjoslu, lai piekļūtu šīs sadaļas saturam.

1 Servera un starpniekservera datu apmaiņas protokols

Pārskats

Serveris - starpniekserveris datu apmaiņa ir balstīta uz JSON formātu.

Pieprasījuma un atbildes ziņojumiem jāsākas ar galveni un datu garumu.

Pasīvais starpniekserveris

Konfigurācijas pieprasījums

Serveris vispirms nosūtīs tukšu `proxy config` pieprasījumu. Šis pieprasījums tiek nosūtīts ik pēc `ProxyConfigFrequency` (servera konfigurācijas parametra) sekundēm.

Starpniekserveris atbild ar pašreizējo starpniekservera versiju, sesijas marķieri un konfigurācijas revīziju. Serveris atbild ar konfigurācijas datiem, kas ir jāatjaunina.

name	value type	description
server→proxy:		
request	*string*	'proxy config'
proxy→server:		
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
session	*string*	Starpniekservera konfigurācijas sesijas marķieris.
config_revision	*number*	Starpniekservera konfigurācijas revīzija.
server→proxy:		
full_sync	*number*	1 - ja tiek nosūtīti pilni konfigurācijas dati; pretējā gadījumā nav norādīts (neobligāti).
data	*array*	Tabulas datu objekts. Nav norādīts, ja konfigurācija nav mainīta (neobligāti).
<table>	*object*	Viens vai vairāki objekti ar <table> datiem (neobligāti, atkarībā no izmaiņām).
fields	*array*	Lauku nosaukumu masīvs.
-	*string*	Lauka nosaukums.
data	*array*	Rindu masīvs.
-	*array*	Kolonnu masīvs.

name	value type	description
-	*string*,*number*	Kolonnas vērtība, kuras tips ir atkarīgs no kolonnas tipa datubāzes shēmā.
macro.secrets	*object*	Slepeno makrosu informācija; nav norādīta, ja vault makrosos nav izmaiņu (neobligāti).
config_revision	*number*	Konfigurācijas kešatmiņas revīzija - tiek nosūtīta kopā ar konfigurācijas datiem (neobligāti).
del_hostids	*array*	Noņemto hostu ID masīvs (neobligāti).
-	*number*	Hosta identifikators.
del_macro_hostids	*array*	Hostu ID masīvs, kuriem ir noņemti visi makrosi (neobligāti).
-	*number*	Hosta identifikators.
proxy→server:		
response	*string*	Informācija par pieprasījuma izpildes rezultātu ('success' vai 'failed').
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).

Piemērs:

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"
}
```

Datu pieprasījums

Pieprasījums `proxy data` tiek izmantots, lai no starpniekservera iegūtu hosta saskarnes pieejamības, vēsturiskos, atklāšanas un automātiskās reģistrācijas datus. Šis pieprasījums tiek nosūtīts ik pēc `ProxyDataFrequency` (servera konfigurācijas parametrs) sekundēm.

name	value type	description
server→proxy:		
request	*string*	'proxy data'
proxy→server:		
session	*string*	Datu sesijas marķieris.
interface availability	*array*	*(optional)* Saskarnes pieejamības datu objektu masīvs.
interfaceid	*number*	Saskarnes identifikators.
available	*number*	Saskarnes pieejamība: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - nezināma **1**, *INTERFACE_AVAILABLE_TRUE* - pieejama **2**, *INTERFACE_AVAILABLE_FALSE* - nav pieejama
error	*string*	Saskarnes kļūdas ziņojums vai tukša virkne.
history data	*array*	*(optional)* Vēsturisko datu objektu masīvs.
itemid	*number*	Vienuma identifikators.
clock	*number*	Vienuma vērtības laikspiedols (sekundes).
ns	*number*	Vienuma vērtības laikspiedols (nanosekundes).
value	*string*	*(optional)* Vienuma vērtība.
id	*number*	Vērtības identifikators (augošs skaitītājs, unikāls vienas datu sesijas ietvaros).
timestamp	*number*	*(optional)* Žurnāla tipa vienumu laikspiedols.
source	*string*	*(optional)* Eventlog vienuma avota vērtība.
severity	*number*	*(optional)* Eventlog vienuma svarīguma vērtība.
eventid	*number*	*(optional)* Eventlog vienuma eventid vērtība.
state	*string*	*(optional)* Vienuma stāvoklis: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Pēdējais žurnāla izmērs žurnāla tipa vienumiem.
mtime	*number*	*(optional)* Žurnāla tipa vienumu modificēšanas laiks.
discovery data	*array*	*(optional)* Atklāšanas datu objektu masīvs.
clock	*number*	Atklāšanas datu laikspiedols.
druleid	*number*	Atklāšanas kārtulas identifikators.
dcheckid	*number*	Atklāšanas pārbaudes identifikators vai null atklāšanas kārtulas datiem.

name	value type	description
type	*number*	Atklāšanas pārbaudes tips: **-1** atklāšanas kārtulas dati **0**, *SVC_SSH* - SSH servisa pārbaude **1**, *SVC_LDAP* - LDAP servisa pārbaude **2**, *SVC_SMTP* - SMTP servisa pārbaude **3**, *SVC_FTP* - FTP servisa pārbaude **4**, *SVC_HTTP* - HTTP servisa pārbaude **5**, *SVC_POP* - POP servisa pārbaude **6**, *SVC_NNTP* - NNTP servisa pārbaude **7**, *SVC_IMAP* - IMAP servisa pārbaude **8**, *SVC_TCP* - TCP porta pieejamības pārbaude **9**, *SVC_AGENT* - Zabbix aģents **10**, *SVC_SNMPv1* - SNMPv1 aģents **11**, *SVC_SNMPv2* - SNMPv2 aģents **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 aģents **14**, *SVC_HTTPS* - HTTPS servisa pārbaude **15**, *SVC_TELNET* - Telnet pieejamības pārbaude
ip	*string*	Hosta IP adrese.
dns	*string*	Hosta DNS nosaukums.
port	*number*	*(optional)* Servisa porta numurs.
key_	*string*	*(optional)* Vienuma atslēga atklāšanas pārbaudei ar tipu **9** *SVC_AGENT*
value	*string*	*(optional)* No servisa saņemtā vērtība; lielākajai daļai servisu tā var būt tukša.
status	*number*	*(optional)* Servisa statuss: **0**, *DOBJECT_STATUS_UP* - Serviss ir UP **1**, *DOBJECT_STATUS_DOWN* - Serviss ir DOWN
auto reg- is- tra- tion	*array*	*(optional)* Automātiskās reģistrācijas datu objektu masīvs.
clock	*number*	Automātiskās reģistrācijas datu laikspiedols.
host	*string*	Hosta nosaukums.
ip	*string*	*(optional)* Hosta IP adrese.
dns	*string*	*(optional)* No IP adreses atrisinātais DNS nosaukums.
port	*string*	*(optional)* Hosta ports.
host_metadata	*string*	*(optional)* Hosta metadati, ko nosūtījis aģents (pamatojoties uz aģenta konfigurācijas parametru HostMetadata vai HostMetadataItem).
tasks	*array*	*(optional)* Uzdevumu masīvs.
type	*number*	Uzdevuma tips: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - attālinātās komandas rezultāts
status	*number*	Attālinātās komandas izpildes statuss: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - attālinātā komanda veiksmīgi pabeigta **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - attālinātās komandas izpilde neizdevās
error	*string*	*(optional)* Kļūdas ziņojums.
parent_taskid	*number*	Vecākuzdevuma ID.
more	*number*	*(optional)* 1 - ir vēl vēsturiskie dati, ko nosūtīt.
clock	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (sekundes).
ns	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (nanosekundes).
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Pieprasījuma izdošanās informācija (’success’ vai ’failed’).
tasks	*array*	*(optional)* Uzdevumu masīvs.

name	value type	description
type	*number*	Uzdevuma tips: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - attālinātā komanda
clock	*number*	Uzdevuma izveides laiks.
ttl	*number*	Laiks sekundēs, pēc kura uzdevuma derīgums beidzas.
commandtype	*number*	Attālinātās komandas tips: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - izmantot pielāgotu skriptu **1**, *ZBX_SCRIPT_TYPE_IPMI* - izmantot IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - izmantot SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - izmantot Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - izmantot globālu skriptu (pašlaik funkcionāli ekvivalents pielāgotam skriptam)
command	*string*	Izpildāmā attālinātā komanda.
execute_on	*number*	Izpildes mērķis pielāgotiem skriptiem: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - izpildīt skriptu uz aģenta **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - izpildīt skriptu uz servera **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - izpildīt skriptu uz starpniekservera
port	*number*	*(optional)* Ports Telnet un SSH komandām.
authtype	*number*	*(optional)* Autentifikācijas tips SSH komandām.
username	*string*	*(optional)* Lietotājvārds Telnet un SSH komandām.
password	*string*	*(optional)* Parole Telnet un SSH komandām.
publickey	*string*	*(optional)* Publiskā atslēga SSH komandām.
privatekey	*string*	*(optional)* Privātā atslēga SSH komandām.
parent_taskid	*number*	Vecākuzdevuma ID.
hostid	*number*	Mērķa hosta ID.

Piemērs:

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
      }
  ]
}
```

Uzdevumu pieprasījums

`proxy tasks` apmaiņa nodrošina attālināto komandu uzdevumu piegādi un apstiprināšanu starp serveri un pasīvu starpniekserveri. Kad serveris aptaujā starpniekserveri, lai saņemtu uzdevumu rezultātus, tas nosūta tukšu `proxy tasks` pieprasījumu ar intervālu, kas noteikts `ProxyDataFrequency`. Starpniekserveris atbild ar savu pašreizējo versiju un visiem gaidošajiem uzdevumu rezultātiem (ieskaitot laikspiedolus). Pēc tam serveris apstiprina saņemšanu, atgriežot `response`, un var iekļaut jaunus uzdevumus, kas starpniekserverim jāizpilda.

name	value type	description
serveris→starpniekserveris:		
request	*string*	'proxy tasks'
starpniekserveris→serveris:		
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
clock	*number*	*(neobligāts)* Datu pārsūtīšanas laikspiedols (sekundes).
ns	*number*	*(neobligāts)* Datu pārsūtīšanas laikspiedols (nanosekundes).
serveris→starpniekserveris:		
response	*string*	Pieprasījuma izdošanās informācija ('success' vai 'failed').

Piemērs:

serveris→starpniekserveris:

```
{
  "request":"proxy tasks"
}
```

starpniekserveris→serveris:

```
{
  "version":"7.0.0",
  "clock":1721059872,
  "ns":848141522
}
```

serveris→starpniekserveris:

```
{
  "response":"success"
}
```

Aktīvais starpniekserveris

Konfigurācijas pieprasījums

`proxy config` pieprasījumu aktīvais starpniekserveris nosūta, lai iegūtu starpniekservera konfigurācijas datus. Šis pieprasījums tiek nosūtīts ik pēc `ProxyConfigFrequency` (starpniekservera konfigurācijas parametra) sekundēm.

name	value type	description
proxy→server:		
request	*string*	'proxy config'
host	*string *	Starpniekservera nosaukums.
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
session	*string*	Starpniekservera konfigurācijas sesijas marķieris.
config_revision	*number*	Starpniekservera konfigurācijas revīzija.
hostmap_revision	*number*	Pašreizējā hostu un starpniekservera kartējuma revīzija.
server→proxy:		
full_sync	*number*	1 - ja tiek nosūtīti pilni konfigurācijas dati, pretējā gadījumā nav norādīts (neobligāti).
data	*array*	Tabulas datu objekts. Nav norādīts, ja konfigurācija nav mainīta (neobligāti).
<table>	*object*	Viens vai vairāki objekti ar <table> datiem (neobligāti, atkarībā no izmaiņām).
fields	*array*	Lauku nosaukumu masīvs.
-	*string*	Lauka nosaukums.
data	*array*	Rindu masīvs.
-	*array*	Kolonnu masīvs.
-	*string,number*	Kolonnas vērtība, kuras tips ir atkarīgs no kolonnas tipa datubāzes shēmā.
macro.secrets	*object*	Slepeno makrosu informācija; nav norādīta, ja vault makrosos nav izmaiņu (neobligāti).
proxy_group	*string*	Tā starpniekserveru grupas nosaukums, kurai pieder starpniekserveris.
config_revision	*number*	Konfigurācijas kešatmiņas revīzija - tiek nosūtīta kopā ar konfigurācijas datiem (neobligāti).
del_hostids	*array*	Noņemto hostid masīvs (neobligāti).
-	*number*	Hosta identifikators.
del_macro_hostids	*array*	Hostid masīvs, kuriem ir noņemti visi makrosi (neobligāti).
-	*number*	Hosta identifikators.

Piemērs:

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
}
```


Datu pieprasījums

`proxy data` pieprasījumu nosūta starpniekserveris, lai nodrošinātu hosta saskarnes pieejamības, vēstures, atklāšanas un automātiskās reģistrācijas datus. Šis pieprasījums tiek nosūtīts ik pēc `DataSenderFrequency` (starpniekservera konfigurācijas parametrs) sekundēm. Ņemiet vērā, ka aktīvais starpniekserveris joprojām aptaujās Zabbix serveri katru sekundi attālināto komandu uzdevumu saņemšanai (ar tukšu `proxy data` pieprasījumu).

name	value type	description
proxy→server:		
request	*string*	'proxy data'
host	*string*	Starpniekservera nosaukums.
session	*string*	Datu sesijas marķieris.
interface availability	*array*	*(optional)* Saskarnes pieejamības datu objektu masīvs.
interfaceid	*number*	Saskarnes identifikators.

name	value type	description
available	*number*	Saskarnes pieejamība: **0**, *INTERFACE_AVAILABLE_UNKNOWN* - nezināma **1**, *INTERFACE_AVAILABLE_TRUE* - pieejama **2**, *INTERFACE_AVAILABLE_FALSE* - nav pieejama
error	*string*	Saskarnes kļūdas ziņojums vai tukša virkne.
history	*array*	*(optional)* Vēstures datu objektu masīvs.
data		
itemid	*number*	Vienuma identifikators.
clock	*number*	Vienuma vērtības laikspiedols (sekundes).
ns	*number*	Vienuma vērtības laikspiedols (nanosekundes).
value	*string*	*(optional)* Vienuma vērtība.
id	*number*	Vērtības identifikators (augošs skaitītājs, unikāls vienas datu sesijas ietvaros).
timestamp	*number*	*(optional)* Žurnāla tipa vienumu laikspiedols.
source	*string*	*(optional)* Eventlog vienuma avota vērtība.
severity	*number*	*(optional)* Eventlog vienuma nopietnības vērtība.
eventid	*number*	*(optional)* Eventlog vienuma eventid vērtība.
state	*string*	*(optional)* Vienuma stāvoklis: **0**, *ITEM_STATE_NORMAL* **1**, *ITEM_STATE_NOTSUPPORTED*
lastlogsize	*number*	*(optional)* Pēdējais žurnāla izmērs žurnāla tipa vienumiem.
mtime	*number*	*(optional)* Modifikācijas laiks žurnāla tipa vienumiem.
discovery	*array*	*(optional)* Atklāšanas datu objektu masīvs.
data		
clock	*number*	Atklāšanas datu laikspiedols.
druleid	*number*	Atklāšanas kārtulas identifikators.
dcheckid	*number*	Atklāšanas pārbaudes identifikators vai null atklāšanas kārtulas datiem.
type	*number*	Atklāšanas pārbaudes tips: **-1** atklāšanas kārtulas dati **0**, *SVC_SSH* - SSH servisa pārbaude **1**, *SVC_LDAP* - LDAP servisa pārbaude **2**, *SVC_SMTP* - SMTP servisa pārbaude **3**, *SVC_FTP* - FTP servisa pārbaude **4**, *SVC_HTTP* - HTTP servisa pārbaude **5**, *SVC_POP* - POP servisa pārbaude **6**, *SVC_NNTP* - NNTP servisa pārbaude **7**, *SVC_IMAP* - IMAP servisa pārbaude **8**, *SVC_TCP* - TCP porta pieejamības pārbaude **9**, *SVC_AGENT* - Zabbix aģents **10**, *SVC_SNMPv1* - SNMPv1 aģents **11**, *SVC_SNMPv2* - SNMPv2 aģents **12**, *SVC_ICMPPING* - ICMP ping **13**, *SVC_SNMPv3* - SNMPv3 aģents **14**, *SVC_HTTPS* - HTTPS servisa pārbaude **15**, *SVC_TELNET* - Telnet pieejamības pārbaude
ip	*string*	Hosta IP adrese.
dns	*string*	Hosta DNS nosaukums.
port	*number*	*(optional)* Servisa porta numurs.
key_	*string*	*(optional)* Vienuma atslēga atklāšanas pārbaudei ar tipu **9** *SVC_AGENT*
value	*string*	*(optional)* No servisa saņemtā vērtība; lielākajai daļai servisu tā var būt tukša.
status	*number*	*(optional)* Servisa statuss: **0**, *DOBJECT_STATUS_UP* - Serviss darbojas **1**, *DOBJECT_STATUS_DOWN* - Serviss nedarbojas
autoregistration	*array*	*(optional)* Automātiskās reģistrācijas datu objektu masīvs.
clock	*number*	Automātiskās reģistrācijas datu laikspiedols.
host	*string*	Hosta nosaukums.
ip	*string*	*(optional)* Hosta IP adrese.

name	value type	description
dns	*string*	*(optional)* No IP adreses atrisinātais DNS nosaukums.
port	*string*	*(optional)* Hosta ports.
host_metadata	*string*	*(optional)* Hosta metadati, ko nosūtījis aģents (pamatojoties uz HostMetadata vai HostMetadataItem aģenta konfigurācijas parametru).
tasks	*array*	*(optional)* Uzdevumu masīvs.
type	*number*	Uzdevuma tips: **0**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND_RESULT* - attālinātās komandas rezultāts
status	*number*	Attālinātās komandas izpildes statuss: **0**, *ZBX_TM_REMOTE_COMMAND_COMPLETED* - attālinātā komanda veiksmīgi pabeigta **1**, *ZBX_TM_REMOTE_COMMAND_FAILED* - attālinātā komanda neizdevās
error	*string*	*(optional)* Kļūdas ziņojums.
parent_taskid	*number*	Vecākuzdevuma ID.
more	*number*	*(optional)* 1 - ir vēl vēstures dati, ko nosūtīt
clock	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (sekundes).
ns	*number*	*(optional)* Datu pārsūtīšanas laikspiedols (nanosekundes).
version	*string*	Starpniekservera versija (<major>.<minor>.<build>).
server→proxy:		
response	*string*	Pieprasījuma veiksmes informācija (’success’ vai ’failed’).
upload	*string*	Vēsturisko datu augšupielādes kontrole (vēsture, automātiskā reģistrācija, hosta pieejamība, tīkla atklāšana). Iespējamās vērtības: **enabled** - normāla darbība **disabled** - serveris nepieņem datus (iespējams, iekšējās kešatmiņas limita pārsniegšanas dēļ)
tasks	*array*	*(optional)* Uzdevumu masīvs.
type	*number*	Uzdevuma tips: **1**, *ZBX_TM_TASK_PROCESS_REMOTE_COMMAND* - attālinātā komanda
clock	*number*	Uzdevuma izveides laiks.
ttl	*number*	Laiks sekundēs, pēc kura uzdevuma derīgums beidzas.
commandtype	*number*	Attālinātās komandas tips: **0**, *ZBX_SCRIPT_TYPE_CUSTOM_SCRIPT* - izmantot pielāgotu skriptu **1**, *ZBX_SCRIPT_TYPE_IPMI* - izmantot IPMI **2**, *ZBX_SCRIPT_TYPE_SSH* - izmantot SSH **3**, *ZBX_SCRIPT_TYPE_TELNET* - izmantot Telnet **4**, *ZBX_SCRIPT_TYPE_GLOBAL_SCRIPT* - izmantot globālu skriptu (pašlaik funkcionāli ekvivalents pielāgotam skriptam)
command	*string*	Izpildāmā attālinātā komanda.
execute_on	*number*	Izpildes mērķis pielāgotiem skriptiem: **0**, *ZBX_SCRIPT_EXECUTE_ON_AGENT* - izpildīt skriptu uz aģenta **1**, *ZBX_SCRIPT_EXECUTE_ON_SERVER* - izpildīt skriptu uz servera **2**, *ZBX_SCRIPT_EXECUTE_ON_PROXY* - izpildīt skriptu uz starpniekservera
port	*number*	*(optional)* Ports Telnet un SSH komandām.
authtype	*number*	*(optional)* Autentifikācijas tips SSH komandām.
username	*string*	*(optional)* Lietotājvārds Telnet un SSH komandām.
password	*string*	*(optional)* Parole Telnet un SSH komandām.
publickey	*string*	*(optional)* Publiskā atslēga SSH komandām.
privatekey	*string*	*(optional)* Privātā atslēga SSH komandām.
parent_taskid	*number*	Vecākuzdevuma ID.
hostid	*number*	Mērķa hosta ID.

Piemērs:

proxy→server:

```
{
"request": "proxy data",
"host": "Zabbix proxy",
"session": "818cdd1b537bdc5e50c09ed4969235b6",
"interface availability": [{
"interfaceid": 1,
"available": 1,
"error": ""
}],
"history data": [{
"id": 1114,
"itemid": 44162,
"clock": 1665730632,
"ns": 798953105,
"value": "1"
}, {
"id": 1115,
"itemid": 44161,
"clock": 1665730633,
"ns": 811684663,
"value": "58"
}],
"auto registration": [{
"clock": 1665730633,
"host": "Zabbix server",
"ip": "127.0.0.1",
"dns": "localhost",
"port": "10053",
"host_metadata": "58",
"tls_accepted": 1
}],
"discovery data": [{
"clock": 1665732232,
"drule": 2,
"dcheck": 2,
"ip": "127.0.0.1",
"dns": "localhost",
"port": 10052,
"status": 1
}, {
"clock": 1665732232,
"drule": 2,
"dcheck": null,
"ip": "127.0.0.1",
"dns": "localhost",
"status": 1
}],
"host data": [{
"hostid": 10084,
"active_status": 1
}],
"tasks": [{
"type": 3,
"clock": 1665730985,
"ttl": 0,
"status": -1,
"info": "Remote commands are not enabled",
"parent_taskid": 3
}],
"version": "8.0.0",
"clock": 1665730643,
"ns": 65389964
```

```
}
```

server→proxy:

```
{
"upload": "enabled",
"response": "success",
"tasks": [{
"type": 2,
"clock": 1665730986,
"ttl": 600,
"commandtype": 0,
"command": "ping -c 3 127.0.0.1; case $? in [01]) true;; *) false;; esac",
"execute_on": 2,
"port": 0,
"authtype": 0,
"username": "",
"password": "",
"publickey": "",
"privatekey": "",
"alertid": 0,
"parent_taskid": 4,
"hostid": 10084
}]
}
```

2 Zabbix aģenta/agent2 protokols

Lūdzu, skatiet lapu Passive and active agent checks, lai iegūtu vairāk informācijas par Zabbix aģenta un Zabbix agent 2 protokoliem.

4 Zabbix aģents 2 spraudņa protokols

Zabbix aģents 2 protokols ir balstīts uz koda, izmēra un datu modeli.

Kods

Type	Size	Comments
Baits	4	Derīgās slodzes tips, pašlaik tiek atbalstīts tikai JSON.

Izmērs

Tips	Izmērs	Komentāri
Baits	4	Pašreizējās lietderīgās slodzes izmērs baitos.

Slodzes dati

Tips	Izmērs	Komentāri
Baits	Definēts ar lauku *Izmērs*	JSON formāta dati.

Noslodzes datu definīcija

Kopīgie dati

Šie parametri ir pieejami visos pieprasījumos/atbildēs:

Name	Type	Comments
id	uint32	Pieprasījumiem — pieaugošs identifikators, ko izmanto, lai sasaistītu pieprasījumus ar atbildēm. Unikāls viena pieprasījuma virziena ietvaros (t.i., no aģents uz spraudni vai no spraudņa uz aģents). Atbildēm — atbilstošā pieprasījuma ID.
type	uint32	Pieprasījuma tips.

Žurnāla pieprasījums

Pieprasījums, ko spraudnis nosūta, lai ierakstītu žurnāla ziņojumu aģenta žurnāla failā.

virziens	spraudnis → aģents
atbilde	nē

Parametri, kas raksturīgi žurnāla pieprasījumiem:

Nosaukums	Type	Komentāri
severity	uint32	Ziņojuma nozīmīgums (žurnāla līmenis).
message	string	Žurnālā ierakstāmais ziņojums.

Piemērs:

```
{"id":0,"type":1,"severity":3,"message":"message"}
```

Reģistrācijas pieprasījums

Pieprasījums, ko aģents nosūta aģenta palaišanas fāzē, lai iegūtu nodrošinātos metriku datus spraudņa reģistrēšanai.

direction	aģents → spraudnis
response	jā

Reģistrācijas pieprasījumiem specifiski parametri:

Name	Type	Comments
version	string	Protokola versija <major>.<minor>

Piemērs:

```
{"id":1,"type":2,"version":"1.0"}
```

Reģistrēšanas atbilde

Spraudņa atbilde uz reģistrēšanas pieprasījumu.

virziens	spraudnis → aģents
atbilde	n/a

Reģistrēšanas atbildēm raksturīgie parametri:

Nosaukums	Tips	Komentāri
name	string	Spraudņa nosaukums.
metrics	string masīvs (neobligāts)	Metrikas ar aprakstiem, kā tās tiek izmantotas spraudnī. Atgriež RegisterMetrics(). Nav norādīts, ja tiek atgriezta kļūda.
interfaces	uint32 (neobligāts)	Spraudņa atbalstīto saskarņu bitu maska. Nav norādīts, ja tiek atgriezta kļūda.

Nosaukums	Tips	Komentāri
error	string (neobligāts)	Kļūdas ziņojums, kas tiek atgriezts, ja spraudni nevar palaist. Nav norādīts, ja tiek atgrieztas metrikas.

Piemēri:

```
{"id":2,"type":3,"metrics":["external.test", "External exporter Test."], "interfaces": 4}
```

vai

```
{"id":2,"type":3,"error":"error message"}
```

Start pieprasījums

Pieprasījums izpildīt Runner saskarnes Start funkciju.

virziens	aģents → spraudnis
atbilde	nē

Pieprasījumam nav specifisku parametru, tas satur tikai kopīgo datu parametrus.

Piemērs:

```
{"id":3,"type":4}
```

Pārtraukt pieprasījumu

Pieprasījums, ko aģents nosūta, lai izslēgtu spraudni.

direction	aģents → spraudnis
response	nē

Pieprasījumam nav specifisku parametru, tas satur tikai kopīgo datu parametrus.

Piemērs:

```
{"id":3,"type":5}
```

Eksportēšanas pieprasījums

Pieprasījums izpildīt Exporter saskarnes funkciju Export.

virziens	aģents → spraudnis
atbilde	nē

Eksportēšanas pieprasījumiem specifiski parametri:

Nosaukums	Type	Komentāri
key	string	Spraudņa atslēga.
parameters	array of strings (optional)	Funkcijas Export parametri.

Piemērs:

```
{"id":4,"type":6,"key":"test.key","parameters":["foo","bar"]}
```

Eksportēšanas atbilde

Atbilde no Exporter interfeisa funkcijas Export.

virziens	spraudnis → aģents
atbilde	nav piemērojams

Parametri, kas raksturīgi eksportēšanas atbildēm:

Nosaukums	Tips	Komentāri
value	string (optional)	Atbildes vērtība no funkcijas Export. Nav norādīta, ja tiek atgriezta kļūda.
error	string (optional)	Kļūdas ziņojums, ja funkcija Export nav izpildīta veiksmīgi. Nav norādīts, ja tiek atgriezta vērtība.

Piemēri:

```
{"id":5,"type":7,"value":"response"}
```

vai

```
{"id":5,"type":7,"error":"error message"}
```

Konfigurēt pieprasījumu

Pieprasījums izpildīt *Configurator* saskarnes funkciju *Configure*.

virziens	aģents → spraudnis
atbilde	n/a

Parametri, kas raksturīgi *Configure* pieprasījumiem:

Nosaukums	Tips	Komentāri
global_options	JSON object	JSON objekts, kas satur globālās aģenta konfigurācijas opcijas.
private_options	JSON object (optional)	JSON objekts, kas satur privātās spraudņa konfigurācijas opcijas, ja tādas ir norādītas.

Piemērs:

```
{"id":6,"type":8,"global_options":{...},"private_options":{...}}
```

Validēt pieprasījumu

Pieprasījums izpildīt *Configurator* saskarnes *Validate* funkciju.

direction	aģents → spraudnis
response	jā

Parametri, kas raksturīgi *Validate* pieprasījumiem:

Name	Type	Comments
private_options	JSON object (optional)	JSON objekts, kas satur privātās spraudņa konfigurācijas opcijas, ja tādas ir norādītas.

Piemērs:

```
{"id":7,"type":9,"private_options":{...}}
```

Validēt atbildi

Atbilde no *Configurator* saskarnes *Validate* funkcijas.

virziens	spraudnis → aģents
atbilde	n/a

Validate atbildēm specifiski parametri:

Nosaukums	Tips	Komentāri
error	string (optional)	Kļūdas ziņojums, kas tiek atgriezts, ja funkcija Validate netiek izpildīta veiksmīgi. Ja izpilde ir veiksmīga, šī lauka nav.

Piemērs:

```
{"id":8,"type":10}
```

vai

```
{"id":8,"type":10,"error":"error message"}
```

5 Zabbix sender protokols

Pārskats

Zabbix serveris un Zabbix starpniekserveris izmanto uz JSON balstītu saziņas protokolu, lai saņemtu datus no Zabbix sender. Datus var saņemt, izmantojot trapper vienumu vai HTTP aģenta vienumu ar iespējotu uztveršanu.

Pieprasījuma un atbildes ziņojumiem jāsākas ar galveni un datu garumu.

Zabbix sender pieprasījums

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value"
        }
    ]
}
```

Zabbix serveris atbilde

```
{
    "response": "success",
    "info": "processed: 1; failed: 0; total: 1; seconds spent: 0.060753"
}
```

Zabbix sender pieprasījums ar laikspiedolu

Alternatīvi, Zabbix sender var nosūtīt pieprasījumu ar laikspiedolu un nanosekundēm.

```
{
    "request": "sender data",
    "data": [
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710794,
            "ns": 592397170
        },
        {
            "host": "<hostname>",
            "key": "trap",
            "value": "test value",
            "clock": 1516710795,
            "ns": 192399456
        }
    ],
    "clock": 1516712029,
    "ns": 873386094
}
```

Zabbix serveris atbilde

```
{
    "response": "success",
    "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.060904"
}
```

6 Galvene

Pārskats

Galvene ir iekļauta visos pieprasījumu un atbilžu ziņojumos starp Zabbix komponentiem. Tā ir nepieciešama, lai noteiktu ziņojuma garumu, vai tas ir saspiests vai nē, vai tā ir liela pakete vai nē.

Zabbix sakaru protokolam ir 1 GB paketes izmēra ierobežojums vienam savienojumam. 1 GB ierobežojums attiecas gan uz saņemtās paketes datu garumu, gan uz nesaspiesto datu garumu.

Sūtot konfigurāciju uz Zabbix starpniekserveris, paketes izmēra ierobežojums tiek palielināts līdz 4 GB, lai ļautu sinhronizēt lielas konfigurācijas. Ja datu garums pirms saspiešanas pārsniedz 4 GB, Zabbix serveris automātiski sāk izmantot lielās paketes formātu (0x04 karodziņš), kas palielina paketes izmēra ierobežojumu līdz 16 GB.

Ņemiet vērā, ka, lai gan lielās paketes formātu var izmantot jebkādu datu sūtīšanai, pašlaik tikai Zabbix starpniekserveris konfigurācijas sinhronizētājs var apstrādāt paketes, kas ir lielākas par 1 GB.

Structure

The header consists of four fields. All numbers in the header are formatted as little-endian.

Field	Size	Size (large packet)	Description
<PROTOCOL>	4	4	"ZBXD" or 5A 42 58 44
<FLAGS>	1	1	Protocol flags: 0x01 - Zabbix communications protocol 0x02 - compression 0x04 - large packet
<DATALEN>	4	8	Data length.
<RESERVED>	4	8	When compression is used (0x02 flag) - the length of uncompressed data When compression is not used - 00 00 00 00

Piemēri

Šeit ir daži koda fragmenti, kas parāda, kā pievienot Zabbix protokola galveni datiem, kurus vēlaties nosūtīt, lai iegūtu paketi, kas jānosūta uz Zabbix, lai tā tiktu pareizi interpretēta. Šie koda fragmenti pieņem, ka dati nav lielāki par 1GB, tādēļ lielo pakešu formāts netiek izmantots.

Python

```
packet = b"ZBXD\1" + struct.pack("<II", len(data), 0) + data
```

vai

```
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;
```

vai

```
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;
```

vai

```
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 Ar jaunu rindu atdalīts JSON eksporta protokols

Šajā sadaļā ir sniegta informācija par eksporta protokolu ar jaunu rindu atdalītā JSON formātā, kas tiek izmantots:

- datu eksportēšanai failos
- straumēšanai uz ārējām sistēmām

Var eksportēt tālāk norādīto:

- trigeru notikumus
- vienumu vērtības
- tendences (tikai eksportēšanai failos)

Visiem failiem ir .ndjson paplašinājums. Katra eksporta faila rinda ir JSON objekts.

Trigeru notikumi

Tālāk norādītā informācija tiek eksportēta problēmas notikumam:

Lauks		Type	Apraksts
clock		number	Sekunžu skaits kopš Epoch līdz brīdim, kad tika konstatēta problēma (veselā daļa).
ns		number	Nanosekunžu skaits, kas jāpieskaita `clock`, lai iegūtu precīzu problēmas konstatēšanas laiku.
value		number	1 (vienmēr).
eventid		number	Problēmas notikuma ID.
name		string	Problēmas notikuma nosaukums.
severity		number	Problēmas notikuma nopietnība (0 - Nav klasificēts, 1 - Informācija, 2 - Brīdinājums, 3 - Vidējs, 4 - Augsts, 5 - Katastrofa).
hosts		array	To hostu saraksts, kas iesaistīti trigera izteiksmē; masīvā jābūt vismaz vienam elementam.
-		object	
	host	string	Hosta nosaukums.
	name	string	Redzamais hosta nosaukums.
groups		array	Visu to hostu grupu saraksts, kas iesaistīti trigera izteiksmē; masīvā jābūt vismaz vienam elementam.
-		string	Hostu grupas nosaukums.
tags		array	Problēmas birku saraksts (var būt tukšs).
-		object	
	tag	string	Birkas nosaukums.
	value	string	Birkas vērtība (var būt tukša).

Tālāk norādītā informācija tiek eksportēta atkopšanas notikumam:

Lauks	Type	Apraksts
clock	number	Sekunžu skaits kopš Epoch līdz brīdim, kad problēma tika novērsta (veselā daļa).
ns	number	Nanosekunžu skaits, kas jāpieskaita `clock`, lai iegūtu precīzu problēmas novēršanas laiku.
value	number	0 (vienmēr).
eventid	number	Atkopšanas notikuma ID.
p_eventid	number	Problēmas notikuma ID.

Piemēri

Problēma:

```
{"clock":1519304285,"ns":123456789,"value":1,"name":"Vai nu Zabbix aģents nav sasniedzams hostā Host B, va
```

Atjaunošana:

```
{"clock":1519304345,"ns":987654321,"value":0,"eventid":43,"p_eventid":42}
```

Problēma (vairāku problēmas notikumu ģenerēšana):

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Vai nu Zabbix aģents nav sasniedzams hos

{"clock":1519304286,"ns":123456789,"value":1,"eventid":43,"name":"Vai nu Zabbix aģents nav sasniedzams hos

Atjaunošana:

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":43}

{"clock":1519304346,"ns":987654321,"value":0,"eventid":44,"p_eventid":42}

Vienumu vērtības

Tālāk norādītā informācija tiek eksportēta savāktai vienuma vērtībai:

Lauks	Type	Apraksts
host	object	Vienuma hosta hosta nosaukums.
host	string	Hosta nosaukums.
name	string	Redzamais hosta nosaukums.
groups	array	Vienuma hosta hostu grupu saraksts; masīvā jābūt vismaz vienam elementam.
-	string	Hostu grupas nosaukums.
item_tags	array	Vienuma birku saraksts (var būt tukšs).
-	object	
tag	string	Birkas nosaukums.
value	string	Birkas vērtība (var būt tukša).
itemid	number	Vienuma ID.
name	string	Redzamais vienuma nosaukums.
clock	number	Sekunžu skaits kopš Epoch līdz brīdim, kad vērtība tika savākta (veselā daļa).
ns	number	Nanosekunžu skaits, kas jāpieskaita `clock`, lai iegūtu precīzu vērtības savākšanas laiku.
timestamp (*Log* only)	number	0, ja nav pieejams.
source (*Log* only)	string	Tukša virkne, ja nav pieejams.
severity (*Log* only)	number	0, ja nav pieejams.
eventid (*Log* only)	number	0, ja nav pieejams.
value	number (for numeric items) or string (for text items)	Savāktā vienuma vērtība.
type	number	Savāktās vērtības tips: 0 - skaitlis ar peldošo komatu, 1 - rakstzīme, 2 - žurnāls, 3 - bez zīmes vesels skaitlis, 4 - teksts, 5 - binārs, 6 - JSON

Piemēri

Skaitliska (bez zīmes) vērtība:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

Skaitliska (peldošā komata) vērtība:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

Rakstzīmju, teksta vērtība:

{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

Žurnāla vērtība:

{"host":{"host":"Host A","name":"Host A visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"

Tendences

Aprēķinātai tendences vērtībai tiek eksportēta šāda informācija:

Lauks	Type	Apraksts
host	object	Vienuma hosta hosta nosaukums.
host	string	Hosta nosaukums.
name	string	Redzamais hosta nosaukums.
groups	array	Vienuma hosta hostu grupu saraksts; masīvā jābūt vismaz vienam elementam.
-	string	Hostu grupas nosaukums.
item_tags	array	Vienuma birku saraksts (var būt tukšs).
-	object	
tag	string	Birkas nosaukums.
value	string	Birkas vērtība (var būt tukša).
itemid	number	Vienuma ID.
name	string	Redzamais vienuma nosaukums.
clock	number	Sekunžu skaits kopš Epoch līdz brīdim, kad vērtība tika savākta (veselā daļa).
count	number	Noteiktai stundai savākto vērtību skaits.
min	number	Minimālā vienuma vērtība noteiktai stundai.
avg	number	Vidējā vienuma vērtība noteiktai stundai.
max	number	Maksimālā vienuma vērtība noteiktai stundai.
type	number	Vērtības tips: 0 - peldošā komata skaitlis, 3 - nenegatīvs vesels skaitlis

Piemēri

Skaitliska (bez zīmes) vērtība:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

Skaitliska (peldošā komata) vērtība:

```
{"host":{"host":"Host B","name":"Host B visible"},"groups":["Group X","Group Y","Group Z"],"item_tags":[{"
```

4 Vienumi

Lūdzu, izmantojiet sānjoslu, lai piekļūtu saturam šajā sadaļā.

1 vm.memory.size parametri

Pārskats

Šajā sadaļā ir sniegta informācija par dažiem aģenta vienuma vm.memory.size[<mode>] parametriem.

Parametri

Šim vienumam ir pieejami šādi parametri:

- **active** - atmiņa, kas pašlaik tiek izmantota vai ir izmantota pavisam nesen, tādēļ tā atrodas RAM
- **anon** - atmiņa, kas nav saistīta ar failu (to nevar no tā atkārtoti nolasīt)
- **available** - pieejamā atmiņa, kas tiek aprēķināta atšķirīgi atkarībā no platformas (skatiet tabulu zemāk)
- **buffers** - kešatmiņa tādām lietām kā failu sistēmas metadati
- **cached** - kešatmiņa dažādām lietām
- **exec** - izpildāmais kods, parasti no (programmas) faila
- **file** - kešatmiņa nesen piekļūtu failu saturam
- **free** - atmiņa, kas ir uzreiz pieejama jebkurai vienībai, kura pieprasa atmiņu
- **inactive** - atmiņa, kas ir atzīmēta kā neizmantota
- **pavailable** - 'available' atmiņa procentos no 'total' (aprēķināts kā `available/total*100`)
- **pinned** - tas pats, kas 'wired'
- **pused** - 'used' atmiņa procentos no 'total' (aprēķināts kā `used/total*100`)
- **shared** - atmiņa, kurai vienlaikus var piekļūt vairāki procesi

- **slab** - kopējais atmiņas apjoms, ko kodols izmanto datu struktūru kešošanai savām vajadzībām
- **total** - kopējā pieejamā fiziskā atmiņa
- **used** - izmantotā atmiņa, kas tiek aprēķināta atšķirīgi atkarībā no platformas (skatiet tabulu zemāk)
- **wired** - atmiņa, kas ir atzīmēta tā, lai vienmēr paliktu RAM. Tā nekad netiek pārvietota uz disku.

Warning:
Daži no šiem parametriem ir specifiski platformai un var nebūt pieejami jūsu platformā. Sīkāku informāciju skatiet Zabbix aģenta vienumi.

Platformai specifisks **available** un **used** aprēķins:

Platforma	**"available"**	**"used"**
AIX	free + cached	izmantotā reālā atmiņa
FreeBSD	inactive + cached + free	active + wired + cached
HP UX	free	total - free
Linux<3.14	free + buffers + cached	total - free
Linux 3.14+ (arī atpakaļportēts uz 3.10 RHEL 7)	/proc/meminfo, sīkāku informāciju skatiet Linux kodola dokumentācijā par "MemAvailable". Ņemiet vērā, ka free + buffers + cached vairs nav vienāds ar 'available', jo ne visu lapu kešatmiņu var atbrīvot un aprēķinā tiek izmantota zemā ūdenszīme.	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:
vm.memory.size[used] un *vm.memory.size[available]* summa ne vienmēr ir vienāda ar kopējo apjomu. Piemēram, FreeBSD gadījumā:
 * Active, inactive, wired, cached atmiņa tiek uzskatīta par izmantotu, jo tā glabā noderīgu informāciju.
 * Vienlaikus inactive, cached, free atmiņa tiek uzskatīta par pieejamu, jo šāda veida atmiņu var nekavējoties piešķirt procesiem, kas pieprasa vairāk atmiņas.
Tātad inactive atmiņa vienlaikus ir gan izmantota, gan pieejama. Šī iemesla dēļ vienums *vm.memory.size[used]* ir paredzēts tikai informatīviem nolūkiem, savukārt *vm.memory.size[available]* ir paredzēts izmantošanai trigeros.

Skatiet arī

1. Papildu informācija par atmiņas aprēķināšanu dažādās operētājsistēmās

2 Pasīvās un aktīvās aģenta pārbaudes

Pārskats

Šajā sadaļā ir sniegta detalizēta informācija par pasīvajām un aktīvajām pārbaudēm, ko veic Zabbix aģents un Zabbix aģents 2.

Zabbix saziņai ar aģentiem izmanto uz JSON balstītu komunikācijas protokolu.

Zabbix aģenta un Zabbix aģenta 2 protokoli ir apvienoti kopš Zabbix 7.0. Atšķirība starp Zabbix aģenta un Zabbix aģenta 2 pieprasījumiem/atbildēm tiek izteikta ar "variant" taga vērtību.

Pasīvās pārbaudes

Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa noteiktus datus (piemēram, CPU slodzi), un Zabbix aģents nosūta rezultātu atpakaļ serverim.

Pasīvās pārbaudes tiek izpildītas asinhroni — nav nepieciešams saņemt atbildi uz vienu pieprasījumu, pirms tiek sāktas citas pārbaudes. Arī DNS atrisināšana notiek asinhroni.

Aģenta aptaujātājs mēģinās izveidot savienojumu ar visām adresēm, kas atgrieztas DNS vaicājumā. Tas nodrošina, ka, ja viena IP adrese nav sasniedzama, aptaujātājs mēģinās nākamo pieejamo adresi, tādējādi palielinot veiksmīga savienojuma iespējamību. Šis uzlabojums attiecas gan uz Zabbix serveri, gan starpniekserveri.

Asinhrono pārbaužu maksimālais vienlaicīgums ir 1000 (noteikts ar MaxConcurrentChecksPerPoller).

Asinhrono aģenta aptaujātāju skaitu nosaka parametrs StartAgentPollers.

Servera pieprasījums

Galvenes un datu garuma definīciju skatiet protokola detalizācijā.

```
{
  "request": "passive checks",
  "data": [
    {
      "key": "agent.version",
      "timeout": 3
    }
  ]
}
```

Lauks		Tips	Obligāts	Vērtība
request		*string*	jā	`"passive checks"`
data		*array of object*	jā	Pasīvās pārbaudes vienums.
	key	*string*	jā	Vienuma atslēga ar izvērstiem makrosiem.
	timeout	*number*	jā	Saziņas noildze.

Aģenta atbilde

```
{
  "version": "8.0.0",
  "variant": 2,
  "data": [
    {
      "value": "8.0.0"
    }
  ]
}
```

Lauks		Tips	Obligāts	Vērtība
version		*string*	jā	Aģenta versijas numurs.
variant		*number*	jā	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix aģents 2).
data		*array of object*	jā	Satur pārbaudes rezultātu.
	value	*string*	nē	Vienuma vērtība, ja pārbaude bija veiksmīga.
	error	*string*	nē	Kļūdas ziņojums, ja pārbaude nebija veiksmīga.

Piemēram, atbalstītiem vienumiem:

1. Serveris atver TCP savienojumu
2. Serveris nosūta **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”agent.ping”,”timeout”:3}]}**
3. Aģents nolasa pieprasījumu un atbild ar **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”value”:1}]}**
4. Serveris apstrādā datus, lai iegūtu vērtību, mūsu gadījumā — ’1’
5. TCP savienojums tiek aizvērts

Piemēram, neatbalstītiem vienumiem:

1. Serveris atver TCP savienojumu
2. Serveris nosūta **<HEADER><DATALEN>{”request”:”passive checks”,”data”:[{”key”:”vfs.fs.size[/nono]”,”timeout”:3}]}**
3. Aģents nolasa pieprasījumu un atbild ar **<HEADER><DATALEN>{”version”:”8.0.0”,”variant”:2,”data”:[{”error”:”Unsupport item key.”}]}**
4. Serveris apstrādā datus, maina vienuma stāvokli uz neatbalstīts ar norādīto kļūdas ziņojumu
5. TCP savienojums tiek aizvērts

Pārslēgšanās uz veco protokolu

Lai nodrošinātu, ka Zabbix serveris vai starpniekserveris var strādāt ar aģentiem no versijām pirms 7.2, kurās tiek izmantots vienkāršā teksta protokols, ir ieviesta pārslēgšanās uz veco protokolu.

Pasīvās pārbaudes tiek veiktas, izmantojot JSON protokolu (7.0 un jaunākās versijās), pēc restartēšanas vai kad tiek mainīta saskarnes konfigurācija. Ja atbildē netiek saņemts derīgs JSON (aģents nosūtīja ”ZBX_NOTSUPPORTED”), Zabbix saglabās saskarni kešatmiņā kā veco protokolu un **atkārtos** pārbaudi, nosūtot tikai vienuma atslēgu.

Ņemiet vērā, ka katru stundu Zabbix serveris/starpniekserveris atkal mēģinās strādāt ar jauno protokolu visās saskarnēs, vajadzības gadījumā pārslēdzoties atpakaļ uz veco protokolu.

Aktīvās pārbaudes

Aktīvajām pārbaudēm ir nepieciešama sarežģītāka apstrāde. Aģentam vispirms no servera/starpniekservera jāiegūst vienumu un/vai attālināto komandu saraksts neatkarīgai apstrādei.

Serveri/starpniekserveri, no kuriem iegūt aktīvās pārbaudes, ir norādīti aģenta konfigurācijas faila parametrā 'ServerActive'. Šo pārbaužu pieprasīšanas biežumu nosaka parametrs 'RefreshActiveChecks' tajā pašā konfigurācijas failā. Tomēr, ja aktīvo pārbaužu atsvaidzināšana neizdodas, atkārtots mēģinājums tiek veikts pēc stingri noteiktām 60 sekundēm.

Note:
Sākot ar Zabbix 6.4, aģents (aktīvajā režīmā) vairs nesaņem no servera/starpniekservera pilnu konfigurācijas kopiju reizi divās minūtēs (pēc noklusējuma). Tā vietā, lai samazinātu tīkla datplūsmu un resursu izmantošanu, ik pēc 5 sekundēm (pēc noklusējuma) tiek veikta inkrementāla konfigurācijas sinhronizācija, kuras laikā serveris/starpniekserveris nodrošina pilnu konfigurācijas kopiju **tikai** tad, ja aģents to vēl nav saņēmis vai ir notikušas izmaiņas hosta konfigurācijā, globālajos makrosos vai globālajās regulārajās izteiksmēs.

Pēc tam aģents periodiski nosūta jaunās vērtības serverim(-iem). Ja aģents ir saņēmis izpildei kādas attālās komandas, tiks nosūtīts arī izpildes rezultāts. Ņemiet vērā, ka attālo komandu izpilde aktīvā aģentā tiek atbalstīta kopš Zabbix aģenta 7.0.

Note:
Ja aģents atrodas aiz ugunsmūra, varat apsvērt izmantot tikai aktīvās pārbaudes, jo šajā gadījumā nebūs nepieciešams mainīt ugunsmūra iestatījumus, lai atļautu sākotnējos ienākošos savienojumus.

Vienumu saraksta iegūšana

Aģenta pieprasījums

Aktīvo pārbaužu pieprasījums tiek izmantots, lai iegūtu aktīvās pārbaudes, kas jāapstrādā aģentam. Šo pieprasījumu aģents nosūta startēšanas laikā un pēc tam ar RefreshActiveChecks intervāliem.

```
{
  "request": "active checks",
  "host": "Zabbix server",
  "host_metadata": "mysql,nginx",
  "interface": "zabbix.server.lan",
  "ip": "159.168.1.1",
  "port": 12050,
  "version": "8.0.0",
  "variant": 2,
  "config_revision": 1,
  "session": "e3dcbd9ace2c9694e1d7bbd030eeef6e"
}
```

Field	Type	Mandatory	Value
request	*string*	yes	`active checks`
host	*string*	yes	Hosta nosaukums.
host_metadata	*string*	no	Konfigurācijas parametra HostMetadata vai HostMetadataItem metrikas vērtība.
interface	*string*	no	Konfigurācijas parametra HostInterface vai HostInterfaceItem metrikas vērtība.
ip	*string*	no	Konfigurācijas parametra ListenIP pirmā IP adrese, ja iestatīta.
port	*number*	no	Konfigurācijas parametra ListenPort vērtība, ja iestatīta un nav noklusējuma aģenta klausīšanās ports.
version	*string*	yes	Aģenta versijas numurs.
variant	*number*	yes	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix aģents 2).
config_revision	*number*	no	Konfigurācijas identifikators inkrementālajai konfigurācijas sinhronizācijai.
session	*string*	no	Sesijas identifikators inkrementālajai konfigurācijas sinhronizācijai.

Servera atbilde

Aktīvo pārbaužu atbildi serveris nosūta atpakaļ aģentam pēc aktīvo pārbaužu pieprasījuma apstrādes.

```
{
  "response": "success",
```

```
  "config_revision": 2,
  "data": [
    {
      "key": "system.uptime",
      "itemid": 1234,
      "delay": "10s",
      "lastlogsize": 0,
      "mtime": 0
    },
    {
      "key": "agent.version",
      "itemid": 5678,
      "delay": "10m",
      "lastlogsize": 0,
      "mtime": 0,
      "timeout": "30s"
    }
  ],
  "commands": [
    {
      "command": "df -h --output=source,size / | awk 'NR>1 {print $2}'",
      "id": 1324,
      "wait": 1
    }
  ]
}
```

Field	Type	Mandatory	Value
response	*string*	yes	`success` | `failed`
info	*string*	no	Kļūdas informācija neveiksmes gadījumā.
data	*array of objects*	no	Aktīvo pārbaužu vienumi. Izlaists, ja hosta konfigurācija nav mainījusies.
key	*string*	no	Vienuma atslēga ar izvērstiem makrosiem.
itemid	*number*	no	Vienuma identifikators.
delay	*string*	no	Vienuma atjaunināšanas intervāls. Elastīgos/plānošanas intervālus atbalsta gan Zabbix aģents, gan Zabbix aģents 2 начиная no Zabbix 7.0.
lastlogsize	*number*	no	Vienuma lastlogsize.
mtime	*number*	no	Vienuma mtime.
timeout	*string*	no	Vienuma noildze.
refresh_unsupported	*number*	no	Neatbalstīta vienuma atsvaidzināšanas intervāls.
regexp	*array of objects*	no	Globālās regulārās izteiksmes.
name	*string*	no	Globālās regulārās izteiksmes nosaukums.
expression	*string*	no	Globālā regulārā izteiksme.
expression_type	*number*	no	Globālās regulārās izteiksmes tips.
exp_delimiter	*string*	no	Globālās regulārās izteiksmes atdalītājs.
case_sensitive	*number*	no	Globālās regulārās izteiksmes reģistrjutības iestatījums.
commands	*array of objects*	no	Attālinātās komandas izpildei. Iekļautas, ja attālināto komandu izpildi ir aktivizējusi darbības operācija vai manuāla skripta izpilde. Ņemiet vērā, ka attālināto komandu izpilde aktīvā aģentā tiek atbalstīta kopš Zabbix aģenta 7.0. Vecāki aktīvie aģenti ignorēs visas attālinātās komandas, kas iekļautas servera aktīvo pārbaužu atbildē.
command	*string*	no	Attālinātā komanda.
id	*number*	no	Attālinātās komandas identifikators.
wait	*number*	no	Attālinātās komandas izpildes režīms (”0” (nowait) komandām no darbības operācijām; ”1” (wait) komandām no manuālas skripta izpildes).
timeout	*number*	no	Attālinātās komandas izpildes noildze serveris/starpniekserveris konfigurācijā.

Field	Type	Mandatory	Value
config_revision	*number*	no	Konfigurācijas identifikators inkrementālajai konfigurācijas sinhronizācijai. Izlaists, ja hosta konfigurācija nav mainījusies. Palielināts, ja hosta konfigurācija ir mainījusies.

Serverim jāatbild ar success.

Piemēram:

1. Aģents atver TCP savienojumu
2. Aģents pieprasa pārbaužu sarakstu
3. Serveris atbild ar vienumu sarakstu un izpildāmajām attālinātajām komandām
4. Aģents parsē atbildi
5. TCP savienojums tiek aizvērts
6. Aģents sāk periodisku datu vākšanu un izpilda attālinātās komandas (atbalstīts kopš Zabbix aģenta 7.0)

Attention:
Ņemiet vērā, ka, izmantojot aktīvo pārbaudi, pusēm, kurām ir piekļuve Zabbix servera trapper portam, var kļūt pieejami (sensitīvi) konfigurācijas dati. Tas ir iespējams, jo jebkurš var izlikties par aktīvu aģentu un pieprasīt vienumu konfigurācijas datus; autentifikācija nenotiek, ja vien neizmantojat šifrēšanas opcijas.

Savākto datu nosūtīšana

Aģents nosūta

Aģenta datu pieprasījums satur savāktās vienumu vērtības un izpildīto attālināto komandu vērtības (ja tādas ir).

```
{
  "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
}
```

Lauks	Tips	Obligāts	Vērtība
request	*string*	jā	`agent data`
data	*array of objects*	jā	Vienumu vērtības.

Lauks	Tips	Obligāts	Vērtība
id	*number*	jā	Vērtības identifikators (pieaugošs skaitītājs, ko izmanto dublētu vērtību pārbaudei tīkla problēmu gadījumā).
itemid	*number*	jā	Vienuma identifikators.
value	*string*	nē	Vienuma vērtība.
lastlogsize	*number*	nē	Vienuma lastlogsize.
mtime	*number*	nē	Vienuma mtime.
state	*number*	nē	Vienuma stāvoklis.
source	*string*	nē	Vērtības notikumu žurnāla avots.
eventid	*number*	nē	Vērtības notikumu žurnāla eventid.
severity	*number*	nē	Vērtības notikumu žurnāla smaguma pakāpe.
timestamp	*number*	nē	Vērtības notikumu žurnāla laikspiedols.
clock	*number*	jā	Vērtības laikspiedols (sekundes kopš Epoch).
ns	*number*	jā	Vērtības laikspiedola nanosekundes.
commands	*array of objects*	nē	Attālināto komandu izpildes rezultāts. Ņemiet vērā, ka attālināto komandu izpilde aktīvā aģentā tiek atbalstīta kopš Zabbix aģenta 7.0. Vecāki aktīvie aģenti ignorēs visas attālinātās komandas, kas iekļautas aktīvo pārbaužu servera atbildē.
id	*number*	nē	Attālinātās komandas identifikators.
value	*string*	nē	Attālinātās komandas izpildes rezultāts, ja izpilde bija veiksmīga.
error	*string*	nē	Attālinātās komandas izpildes kļūdas ziņojums, ja izpilde neizdevās.
session	*string*	jā	Unikāls sesijas identifikators, kas tiek ģenerēts katru reizi, kad aģents tiek palaists.
host	*string*	jā	Hosta nosaukums.
version	*string*	jā	Aģenta versijas numurs.
variant	*number*	jā	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix aģents 2).

Katram vērtības ierakstam tiek piešķirts virtuāls ID. Vērtības ID ir vienkāršs augošs skaitītājs, unikāls vienas datu sesijas ietvaros (ko identificē sesijas marķieris). Šis ID tiek izmantots, lai atmestu dublētas vērtības, kas var tikt nosūtītas sliktas savienojamības apstākļos.

Servera atbilde

Aģenta datu atbildi serveris nosūta atpakaļ aģentam pēc aģenta datu pieprasījuma apstrādes.

```
{
  "response": "success",
  "info": "processed: 2; failed: 0; total: 2; seconds spent: 0.003534"
}
```

Lauks	Tips	Obligāts	Vērtība
response	*string*	jā	`success` | `failed`
info	*string*	jā	Vienumu apstrādes rezultāti.

Attention:
Ja dažu vērtību nosūtīšana serverī neizdodas (piemēram, tāpēc, ka hosts vai vienums ir atspējots vai dzēsts), aģents nemēģinās šīs vērtības nosūtīt atkārtoti.

Piemēram:

1. Aģents atver TCP savienojumu
2. Aģents nosūta vērtību sarakstu
3. Serveris apstrādā datus un nosūta statusu atpakaļ
4. TCP savienojums tiek aizvērts

Attention:
Kļūdas ziņojums servera pusē tiks saīsināts līdz 2048 simboliem.

Heartbeat ziņojums

Aģents nosūta

Heartbeat ziņojumu aktīvais aģents nosūta uz Zabbix serveri/starpniekserveri ik pēc HeartbeatFrequency sekundēm (konfigurēts Zabbix aģenta/ aģenta 2 konfigurācijas failā).

To izmanto, lai uzraudzītu aktīvo pārbaužu pieejamību.

```
{
  "request": "active check heartbeat",
  "host": "Zabbix server",
  "heartbeat_freq": 60,
  "version": "8.0.0",
  "variant": 2
}
```

Lauks	Tips	Obligāts	Vērtība
request	*string*	jā	`active check heartbeat`
host	*string*	jā	Hosta nosaukums.
heartbeat_freq	*number*	jā	Aģenta heartbeat biežums (konfigurācijas parametrs HeartbeatFrequency).
version	*string*	jā	Aģenta versijas numurs.
variant	*number*	jā	Aģenta variants (*1* - Zabbix aģents, *2* - Zabbix aģents 2).

Pāradresācijas atbilde

Kad hosts ir pārdalīts, serveris var norādīt aģentam pāradresēt tā heartbeat (un turpmākās aktīvās pārbaudes) uz citu starpniekservera vai servera instanci.

```
  {
    "response": "failed",
    "redirect": {
      "revision": 2,
      "address": "192.0.2.0:10055"
    }
  }
```

Lauks		Tips	Obligāts	Vērtība
response		*string*	jā	`success` | `failed`
redirect		*object*	jā	Pāradresācijas norādījumi.
	revision	*number*	jā	Konfigurācijas revīzijas identifikators.
	address	*string*	jā	Mērķa servera/starpniekservera adrese.

Vecāks XML protokols

Note:
Zabbix pieņems līdz 16 MB XML Base64 kodētu datu, taču viena atkodēta vērtība nedrīkst būt lielāka par 64 KB, pretējā gadījumā tā atkodēšanas laikā tiks saīsināta līdz 64 KB.

3 Minimālais atļauju līmenis Windows aģenta vienumiem

Pārskats

Uzraugot sistēmas, izmantojot aģentu, laba prakse ir iegūt metriku no hosta, kurā aģents ir instalēts. Lai izmantotu mazāko privilēģiju principu, ir nepieciešams noteikt, kāda metrika tiek iegūta no aģenta.

Šajā dokumentā esošā tabula ļauj izvēlēties minimālās tiesības, kas nepieciešamas garantētai korektai Zabbix aģenta darbībai.

Ja aģenta darbībai tiek izvēlēts cits lietotājs, nevis 'LocalSystem', tad, lai aģents darbotos kā Windows serviss, jaunajam lietotājam jābūt piešķirtām tiesībām "Log on as a service" no "Local Policy→User Rights Assignment", kā arī tiesībām izveidot, rakstīt un dzēst Zabbix aģenta žurnālfailu. Active Directory lietotājs ir jāpievieno grupai *Performance Monitor Users*.

Note:
Strādājot ar aģenta tiesībām, kas balstītas uz "minimum technically acceptable" grupu, ir nepieciešama iepriekšēja tiesību piešķiršana uzraugāmajiem objektiem.

Biežāk izmantotie Windows atbalstītie aģenta vienumi

Vienuma atslēga	Lietotāju grupa	
	Ieteicamā	Tehniski minimāli pieļaujamā (funkcionalitāte ir ierobežota)
agent.hostname	Guests	Guests
agent.ping	Guests	Guests
agent.variant	Guests	Guests
agent.version	Guests	Guests
log	Administrators	Guests
log.count	Administrators	Guests
logrt	Administrators	Guests
logrt.count	Administrators	Guests
net.dns	Guests	Guests
net.dns.perf	Guests	Guests
net.dns.record	Guests	Guests
net.if.discovery	Guests	Guests
net.if.in	Guests	Guests
net.if.out	Guests	Guests
net.if.total	Guests	Guests
net.tcp.listen	Guests	Guests
net.tcp.port	Guests	Guests
net.tcp.service	Guests	Guests
net.tcp.service.perf	Guests	Guests
net.udp.service	Guests	Guests
net.udp.service.perf	Guests	Guests
proc.num	Administrators	Guests
system.cpu.discovery	Performance Monitor Users	Performance Monitor Users
system.cpu.load	Performance Monitor Users	Performance Monitor Users
system.cpu.num	Guests	Guests
system.cpu.util	Performance Monitor Users	Performance Monitor Users
system.hostname	Guests	Guests
system.localtime	Guests	Guests
system.run	Administrators	Guests
system.sw.arch	Guests	Guests
system.swap.size	Guests	Guests
system.uname	Guests	Guests
system.uptime	Performance Monitor Users	Performance Monitor Users
vfs.dir.count	Administrators	Guests
vfs.dir.get	Administrators	Guests
vfs.dir.size	Administrators	Guests
vfs.file.cksum	Administrators	Guests
vfs.file.contents	Administrators	Guests
vfs.file.exists	Administrators	Guests
vfs.file.md5sum	Administrators	Guests
vfs.file.regexp	Administrators	Guests
vfs.file.regmatch	Administrators	Guests
vfs.file.size	Administrators	Guests
vfs.file.time	Administrators	Guests
vfs.fs.discovery	Administrators	Guests
vfs.fs.size	Administrators	Guests
vfs.fs.get	Administrators	Guests
vm.memory.size	Guests	Guests
web.page.get	Guests	Guests
web.page.perf	Guests	Guests
web.page.regexp	Guests	Guests
zabbix.stats	Guests	Guests

Windows specifiskās vienumu atslēgas

Vienuma atslēga	Lietotāju grupa	
	Ieteicamā	Minimāli tehniski pieļaujamā (funkcionalitāte ir ierobežota)

Vienuma atslēga	Lietotāju grupa	
eventlog	Event Log Readers	Guests
net.if.list	Guests	Guests
perf_counter	Performance Monitor Users	Performance Monitor Users
proc_info	Administrators	Guests
service.discovery	Guests	Guests
service.info	Guests	Guests
services	Guests	Guests
wmi.get	Administrators	Guests
vm.vmemory.size	Guests	Guests

4 Atgriezto vērtību kodējums

Zabbix serveris sagaida, ka katra atgrieztā teksta vērtība būs UTF8 kodējumā. Tas attiecas uz jebkura veida pārbaudēm: Zabbix aģents, SSH, Telnet utt.

Dažādas uzraudzītās sistēmas/ierīces un pārbaudes vērtībā var atgriezt rakstzīmes, kas nav ASCII. Šādos gadījumos gandrīz visas iespējamās **zabbix** atslēgas satur papildu vienuma atslēgas parametru — **<encoding>**. Šis atslēgas parametrs nav obligāts, taču tas jānorāda, ja atgrieztā vērtība nav UTF8 kodējumā un tajā ir rakstzīmes, kas nav ASCII. Pretējā gadījumā rezultāts var būt negaidīts un neprognozējams.

Tālāk ir sniegts apraksts par darbību šādos gadījumos ar dažādām datu bāzu aizmugursistēmām.

MySQL

Ja vērtība satur ne-ASCII rakstzīmi kodējumā, kas nav UTF8, šī rakstzīme un tai sekojošās tiks atmestas, kad datubāze saglabās šo vērtību. Brīdinājuma ziņojumi netiks ierakstīti *zabbix_server.log*.
 Attiecas vismaz uz MySQL versiju 5.1.61.

PostgreSQL

Ja vērtība satur ne-ASCII rakstzīmi kodējumā, kas nav UTF8, tas izraisīs neveiksmīgu SQL vaicājumu (PGRES_FATAL_ERROR:ERROR invalid byte sequence for encoding), un dati netiks saglabāti. Atbilstošs brīdinājuma ziņojums tiks ierakstīts *zabbix_server.log*.
 Attiecas vismaz uz PostgreSQL versiju 9.1.3.

5 Lielu failu atbalsts

Lielu failu atbalsts, ko bieži saīsina kā LFS, ir termins, ko lieto, lai aprakstītu spēju strādāt ar failiem, kas ir lielāki par 2 GB 32 bitu operētājsistēmās. Lielu failu atbalsts ietekmē vismaz žurnālfailu uzraudzību un visus vfs.file.* vienumus. Lielu failu atbalsts ir atkarīgs no sistēmas iespējām Zabbix kompilēšanas laikā, taču 32 bitu Solaris sistēmās tas ir pilnībā atspējots, jo nav saderīgs ar procfs un swapctl.

6 Sensors

Pārskats

Katram sensora čipam sysfs `/sys/devices` kokā ir sava direktorija. Lai atrastu visus sensora čipus, ir ērtāk sekot ierīču simboliskajām saitēm no `/sys/class/hwmon/hwmon*`, kur `*` ir reāls skaitlis (0,1,2,...).

Sensoru rādījumi atrodas vai nu `/sys/class/hwmon/hwmon*/` direktorijā virtuālajām ierīcēm, vai `/sys/class/hwmon/hwmon*/device` direktorijā nevirtuālajām ierīcēm. Fails ar nosaukumu `name`, kas atrodas `hwmon*` vai `hwmon*/device` direktorijās, satur čipa nosaukumu, kas atbilst sensora čipa izmantotā kodola draivera nosaukumam.

Katrā failā ir tikai viena sensora rādījuma vērtība. Kopējā failu nosaukumu shēma failiem, kas satur sensoru rādījumus jebkurā no iepriekš minētajām direktorijām, ir: `<type><number>_<item>`, kur

- **type** - sensora čipiem tas ir ”in” (spriegums), ”temp” (temperatūra), ”fan” (ventilators) utt.
- **item** - ”input” (izmērītā vērtība), ”max” (augšējais slieksnis), ”min” (apakšējais slieksnis) utt.
- **number** - vienmēr tiek izmantots elementiem, kas var būt sastopami vairāk nekā vienu reizi (parasti sākas ar 1, izņemot spriegumus, kas sākas ar 0). Ja faili neattiecas uz konkrētu elementu, tiem ir vienkāršs nosaukums bez numura.

Informāciju par sensoru pieejamību hostā var iegūt, izmantojot rīkus **sensors-detect** un **sensors** (lm-sensors package). **Sensors-detect** palīdz noteikt, kuri moduļi ir nepieciešami pieejamajiem sensoriem. Kad moduļi ir ielādēti, programmu **sensors** var izmantot, lai parādītu visu sensora čipu rādījumus. Šīs programmas izmantotais sensoru rādījumu marķējums var atšķirties no kopējās nosaukumu shēmas (`<type><number>_<item>`):

- ja ir fails ar nosaukumu <type><number>_label, tad šajā failā esošais marķējums tiks izmantots <type><number><item> nosaukuma vietā;
- ja faila <type><number>_label nav, tad programma meklē nosaukuma aizstāšanu failā /etc/sensors.conf (tas var būt arī /etc/sensors3.conf vai cits).

Šis marķējums ļauj lietotājam noteikt, kāda veida aparatūra tiek izmantota. Ja nav ne faila <type><number>_label, ne marķējuma konfigurācijas failā, aparatūras tipu var noteikt pēc nosaukuma atribūta (hwmon*/device/name). Faktiskos sensoru nosaukumus, kurus pieņem zabbix_agent, var iegūt, palaižot programmu **sensors** ar parametru -u (sensors -u).

Programmā **sensors** pieejamie sensori ir atdalīti pēc kopnes tipa (ISA adapteris, PCI adapteris, SPI adapteris, virtuālā ierīce, ACPI saskarne, HID adapteris).

Sensora identifikatoru iegūšana

Sensora etiķetes, kā tās izdrukā komanda sensors, ne vienmēr var izmantot tieši, jo marķējums atšķiras starp dažādiem mikroshēmu ražotājiem. Piemēram, sensors izvade var saturēt šādas rindas:

```
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)
```

No tiem tikai viens attēlojamais nosaukums tieši atbilst iekšējam identifikatoram:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in0]
2.240000
```

Mēģinājums izmantot citas attēlojamās etiķetes (piemēram, Vcore vai +12V) nedarbosies:

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,Vcore]
ZBX_NOTSUPPORTED
```

Lai atrastu iekšējo sensora identifikatoru, ko Zabbix pieņem, palaidiet sensors -u. Izvadē var redzēt sekojošo:

```
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
...
```

Tātad etiķete Vcore atbilst identifikatoram in1, bet +12V atbilst in4. Saskaņā ar kodola hwmon sysfs specifikāciju, tie ir spriegumi uz mikroshēmas izvadiem, un vispārīgi runājot tiem var būt nepieciešama mērogošana.

```
zabbix_get -s 127.0.0.1 -k sensor[lm85-i2c-0-2e,in1]
1.301000
```

Zabbix var iegūt ne tikai sprieguma (in), bet arī strāvas (curr), temperatūras (temp) un ventilatora ātruma (fan) rādījumus.

Saderība

Mantotās distribūcijas

Sensora rādījumi tiek iegūti no direktorija /proc/sys/dev/sensors:

- **device** - ierīces nosaukums (ja tiek izmantots <mode>, tā ir regulārā izteiksme)
- **sensor** - sensora nosaukums (ja tiek izmantots <mode>, tā ir regulārā izteiksme)
- **mode** - iespējamās vērtības: avg, max, min (ja šis parametrs ir izlaists, device un sensor tiek interpretēti burtiski)

Atslēgas piemērs: sensor[w83781d-i2c-0-2d,temp1].

Pašreizējās distribūcijas

Sensora rādījumi tiek iegūti no direktorija /sys/class/hwmon:

- **device** - ierīces nosaukums (neregulārā izteiksme). Ierīces nosaukums var būt faktiskais ierīces nosaukums (piemēram, 0000:00:18.3) vai nosaukums, kas iegūts, izmantojot programmu sensors (piemēram, k8temp-pci-00c3). Lietotājs pats izvēlas, kuru nosaukumu izmantot.
- **sensor** - sensora nosaukums (neregulārā izteiksme).
- **mode** - iespējamās vērtības: avg, max, min (ja šis parametrs ir izlaists, device un sensor tiek apstrādāti burtiski).

Atslēgu piemēri:

sensor[k8temp-pci-00c3,temp,max] vai sensor[0000:00:18.3,temp1].

sensor[smsc47b397-isa-0880,in,avg] vai sensor[smsc47b397.2176,in1].

7 Piezīmes par memtype parametru proc.mem vienumos

Pārskats

Parametrs **memtype** tiek atbalstīts Linux, AIX, FreeBSD un Solaris platformās.

Visās šajās platformās tiek atbalstītas trīs izplatītas 'memtype' vērtības: pmem, rss un vsize. Papildus tam dažās platformās tiek atbalstītas arī platformai specifiskas 'memtype' vērtības.

AIX

Skatiet AIX atbalstītās parametra memtype vērtības tabulā.

Atbalstītā vērtība	Apraksts	Avots struktūrā procentry64	Mēģina būt saderīgs ar
vsize [1]	Virtuālās atmiņas izmērs	pi_size	
pmem	Reālās atmiņas procentuālā daļa	pi_prm	ps -o pmem
rss	Rezidentās kopas izmērs	pi_trss + pi_drss	ps -o rssize
size	Procesa izmērs (kods + dati)	pi_dvm	"ps gvw" SIZE kolonna
dsize	Datu izmērs	pi_dsize	
tsize	Teksta (koda) izmērs	pi_tsize	"ps gvw" TSIZ kolonna
sdsize	Datu izmērs no koplietojamās bibliotēkas	pi_sdsize	
drss	Datu rezidentās kopas izmērs	pi_drss	
trss	Teksta rezidentās kopas izmērs	pi_trss	

Piezīmes par AIX:

1. Izvēloties parametrus AIX sistēmas proc.mem[] vienuma atslēgai, mēģiniet norādīt šaurus procesu atlases kritērijus. Pretējā gadījumā pastāv risks, ka proc.mem[] rezultātā tiks iekļauti nevēlami procesi.

Piemērs:

```
$ zabbix_agentd -t proc.mem[,,,NonExistingProcess,rss]
proc.mem[,,,NonExistingProcess,rss]           [u|2879488]
```

Šis piemērs parāda, ka, norādot tikai komandrindas parametru (atbilstības regulāro izteiksmju parametrs), Zabbix aģents uzskaita pats sevi — visticamāk, tas nav tas, ko vēlaties.

2. Neizmantojiet "ps -ef" procesu pārlūkošanai — tas parāda tikai ne-kodola procesus. Izmantojiet "ps -Af", lai redzētu visus procesus, kurus redzēs Zabbix aģents.
3. Apskatīsim 'topasrec' piemēru, kā Zabbix aģents proc.mem[] atlasa procesus.

```
$ 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[] ir argumenti:

proc.mem[<name>,<user>,<mode>,<cmdline>,<memtype>]

1. kritērijs ir procesa nosaukums (arguments <name>). Mūsu piemērā Zabbix aģents to redzēs kā 'topasrec'. Lai tas atbilstu, jums ir vai nu jānorāda 'topasrec', vai arī jāatstāj tas tukšs.
2. kritērijs ir lietotāja vārds (arguments <user>). Lai tas atbilstu, jums ir vai nu jānorāda 'root', vai arī jāatstāj tas tukšs.

3. kritērijs, ko izmanto procesa atlasei, ir arguments <cmdline>. Zabbix aģents redzēs tā vērtību kā '/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'. Lai tas atbilstu, jums ir vai nu jānorāda regulārā izteiksme, kas atbilst šai virknei, vai arī jāatstāj tas tukšs.

Argumenti <mode> un <memtype> tiek piemēroti pēc iepriekš minēto trīs kritēriju izmantošanas.

FreeBSD

Skatiet FreeBSD atbalstītās parametra 'memtype' vērtības tabulā.

Atbalstītā vērtība	Apraksts	Avots struktūrā kinfo_proc	Mēģina būt saderīgs ar
vsize	Virtuālās atmiņas izmērs	kp_eproc.e_vm.vm_map_size vai ki_size	ps -o vsz
pmem	Reālās atmiņas procentuālā daļa	aprēķināts no rss	ps -o pmem
rss	Rezidentās kopas izmērs	kp_eproc.e_vm.vm_rssize vai ki_rssize	ps -o rss
size [1]	Procesa izmērs (kods + dati + steks)	tsize + dsize + ssize	
tsize	Teksta (koda) izmērs	kp_eproc.e_vm.vm_tsize vai ki_tsize	ps -o tsiz
dsize	Datu izmērs	kp_eproc.e_vm.vm_dsize vai ki_dsize	ps -o dsiz
ssize	Steka izmērs	kp_eproc.e_vm.vm_ssize vai ki_ssize	ps -o ssiz

Linux

Tabulā skatiet Linux atbalstītās parametra 'memtype' vērtības.

Atbalstītā vērtība	Apraksts	Avots failā /proc/<pid>/status
vsize [1]	Virtuālās atmiņas lielums	VmSize
pmem	Reālās atmiņas procentuālā daļa	(VmRSS/total_memory) * 100
rss	Rezidentās kopas lielums	VmRSS
data	Datu segmenta lielums	VmData
exe	Koda segmenta lielums	VmExe
hwm	Maksimālais rezidentās kopas lielums	VmHWM
lck	Bloķētās atmiņas lielums	VmLck
lib	Koplietojamo bibliotēku lielums	VmLib
peak	Maksimālais virtuālās atmiņas lielums	VmPeak
pin	Piesprausto lapu lielums	VmPin
pte	Lapu tabulas ierakstu lielums	VmPTE
size	Procesa koda + datu + steka segmentu lielums	VmExe + VmData + VmStk
stk	Steka segmenta lielums	VmStk
swap	Izmantotās maiņvietas lielums	VmSwap

Piezīmes par Linux:

1. Ne visas 'memtype' vērtības atbalsta vecākas Linux kodola versijas. Piemēram, Linux 2.4 kodoli neatbalsta `hwm`, `pin`, `peak`, `pte` un `swap` vērtības.
2. Esam novērojuši, ka Zabbix aģents aktīvās pārbaudes procesa pašuzraudzība ar `proc.mem[...,...,...,...,data]` parāda vērtību, kas ir par 4 kB lielāka nekā tā, par kuru ziņo `VmData` rinda aģents failā /proc/<pid>/status. Pašmērījuma laikā aģents datu segments palielinās par 4 kB un pēc tam atgriežas iepriekšējā lielumā.

Solaris

Tabulā skatiet Solaris sistēmā parametram 'memtype' atbalstītās vērtības.

Atbalstītā vērtība	Apraksts	Avots psinfo struktūrā	Mēģina būt saderīgs ar
vsize [1]	Procesa attēla izmērs	pr_size	ps -o vsz
pmem	Reālās atmiņas procentuālā daļa	pr_pctmem	ps -o pmem

Atbalstītā vērtība	Apraksts	Avots psinfo struktūrā	Mēģina būt saderīgs ar
rss	Rezidentās kopas izmērs Tas var būt novērtēts par zemu — skatiet rss aprakstu sadaļā ”man ps”.	pr_rssize	ps -o rss

Piezīmes

1 Noklusējuma vērtība.

8 Piezīmes par procesu atlasi proc.mem un proc.num vienumos

Procesi, kas modificē savu komandrindu

Dažas programmas izmanto savas komandrindas modificēšanu kā metodi, lai attēlotu savu pašreizējo aktivitāti. Lietotājs var redzēt aktivitāti, palaižot ps un top komandas. Šādu programmu piemēri ir *PostgreSQL*, *Sendmail*, *Zabbix*.

Apskatīsim piemēru no Linux. Pieņemsim, ka vēlamies uzraudzīt Zabbix aģenta procesu skaitu.

Komanda ps parāda interesējošos procesus šādi:

```
$ 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]
...
```

Procesu atlase pēc nosaukuma un lietotāja darbojas:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd,zabbix]'
6
```

Tagad pārdēvēsim izpildāmo failu zabbix_agentd par zabbix_agentd_30 un restartēsim to.

Tagad ps parāda:

```
$ 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]
...
```

Tagad procesu atlase pēc nosaukuma un lietotāja dod nepareizu rezultātu:

```
$ zabbix_get -s localhost -k 'proc.num[zabbix_agentd_30,zabbix]'
1
```

Kāpēc vienkārša izpildāmā faila pārdēvēšana uz garāku nosaukumu rada tik atšķirīgu rezultātu?

Zabbix aģents sāk ar procesa nosaukuma pārbaudi. Tiek atvērts fails /proc/<pid>/status un pārbaudīta rinda Name. Mūsu gadījumā Name rindas ir šādas:

```
$ 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
```

Procesa nosaukums failā `status` tiek saīsināts līdz 15 rakstzīmēm.

Līdzīgu rezultātu var redzēt ar komandu `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
 ...
```

Acīmredzot tas nav vienāds ar mūsu `proc.num[]` parametra `name` vērtību `zabbix_agentd_30`. Tā kā neizdevās saskaņot procesa nosaukumu no faila `status`, Zabbix aģents pāriet pie faila `/proc/<pid>/cmdline`.

To, kā aģents redz failu "cmdline", var ilustrēt, palaižot komandu:

```
$ 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
```

Faili `/proc/<pid>/cmdline` mūsu gadījumā satur neredzamus, nedrukājamus nulles baitus, kas *C* valodā tiek izmantoti virkņu terminēšanai. Šajā piemērā nulles baiti ir parādīti kā "<NUL>".

Zabbix aģents pārbauda "cmdline" galvenajam procesam un iegūst `zabbix_agentd_30`, kas atbilst mūsu parametra `name` vērtībai `zabbix_agentd_30`. Tāpēc galvenais process tiek ieskaitīts ar vienumu `proc.num[zabbix_agentd_30,zabbix]`.

Pārbaudot nākamo procesu, aģents no faila `cmdline` iegūst `zabbix_agentd_30: collector [idle 1 sec]`, un tas neatbilst mūsu parametram `name` `zabbix_agentd_30`. Tāpēc tiek ieskaitīts tikai galvenais process, kas nemodificē savu komandrindu. Citi aģenta procesi modificē savu komandrindu un tiek ignorēti.

Šis piemērs parāda, ka parametru `name` šajā gadījumā nevar izmantot `proc.mem[]` un `proc.num[]` procesu atlasei.

Note:
Vienumam `proc.get[]`, kad Zabbix aģents pārbauda procesa nosaukumu laukā "cmdline", tas izmantos tikai to nosaukuma daļu, kas sākas no pēdējās slīpsvītras un turpinās līdz pirmajai atstarpei vai kolam. Procesa nosaukums, kas iegūts no cmdline faila, tiks izmantots tikai tad, ja tā sākums pilnībā atbilst saīsinātajam procesa nosaukumam failā `status`. Algoritms ir vienāds gan procesa nosaukumam filtrā, gan JSON izvadē.

Izmantojot parametru `cmdline` ar atbilstošu regulāro izteiksmi, tiek iegūts pareizs rezultāts:

```
$ zabbix_get -s localhost -k 'proc.num[,zabbix,,zabbix_agentd_30[ :]]'
6
```

Esiet uzmanīgi, izmantojot `proc.get[]`, `proc.mem[]` un `proc.num[]` vienumus, lai uzraudzītu programmas, kas modificē savas komandrindas.

Pirms ievietot parametrus `name` un `cmdline` vienumos `proc.get[]`, `proc.mem[]` un `proc.num[]`, iespējams, vēlēsieties pārbaudīt šos parametrus, izmantojot vienumu `proc.num[]` un komandu `ps`.

Linux kodola pavedieni

Pavedienus nevar atlasīt ar `cmdline` parametru `proc.get[]`, `proc.mem[]` un `proc.num[]` vienumos

Apskatīsim kā piemēru vienu no kodola pavedieniem:

```
$ ps -ef| grep kthreadd
root         2     0  0 09:33 ?        00:00:00 [kthreadd]
```

To var atlasīt ar procesa `name` parametru:

```
$ zabbix_get -s localhost -k 'proc.num[kthreadd,root]'
1
```

Taču atlase pēc procesa `cmdline` parametra nedarbojas:

```
$ zabbix_get -s localhost -k 'proc.num[,root,,kthreadd]'
0
```

Iemesls ir tāds, ka Zabbix aģents ņem regulāro izteiksmi, kas norādīta `cmdline` parametrā, un piemēro to procesa `/proc/<pid>/cmdline` saturam. Kodola pavedieniem to `/proc/<pid>/cmdline` faili ir tukši. Tāpēc `cmdline` parametrs nekad neatbilst.

Pavedienu skaitīšana `proc.mem[]` un `proc.num[]` vienumos

Linux kodola pavedieni tiek skaitīti ar `proc.num[]` vienumu, taču tie neziņo par atmiņu `proc.mem[]` vienumā. Piemēram:

```
$ 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.
```

Bet kas notiek, ja ir lietotāja process ar tādu pašu nosaukumu kā kodola pavedienam? Tad tas varētu izskatīties šādi:

```
$ 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[]` saskaitīja gan kodola pavedienu, gan lietotāja procesu. `proc.mem[]` ziņo par atmiņu tikai lietotāja procesam un uzskata kodola pavediena atmiņu par 0. Tas atšķiras no iepriekšējā gadījuma, kad tika ziņots ZBX_NOTSUPPORTED.

Esiet piesardzīgi, izmantojot `proc.mem[]` un `proc.num[]` vienumus, ja programmas nosaukums nejauši sakrīt ar kāda pavediena nosaukumu.

Pirms parametru ievietošanas `proc.mem[]` un `proc.num[]` vienumos, iespējams, vēlēsieties pārbaudīt parametrus, izmantojot `proc.num[]` vienumu un komandu `ps`.

9 net.tcp.service un net.udp.service pārbaužu ieviešanas detaļas

net.tcp.service un net.udp.service pārbaužu ieviešana ir detalizēti aprakstīta šajā lapā dažādiem pakalpojumiem, kas norādīti service parametrā.

Vienuma net.tcp.service parametri

ftp

Izveido TCP savienojumu un sagaida, ka atbildes pirmās 4 rakstzīmes būs "220 ", pēc tam nosūta "QUIT\r\n". Ja ports nav norādīts, pēc noklusējuma tiek izmantots 21. ports.

http

Izveido TCP savienojumu, neko nesagaidot un neko nenosūtot. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 80. ports.

https

Izmanto libcurl (un darbojas tikai ar to), nepārbauda sertifikāta autentiskumu, nepārbauda hosta nosaukumu SSL sertifikātā, iegūst tikai atbildes galveni (HEAD pieprasījums). Ja ports nav norādīts, pēc noklusējuma tiek izmantots 443. ports.

imap

Izveido TCP savienojumu un sagaida, ka atbildes pirmās 4 rakstzīmes būs "* OK", pēc tam nosūta "a1 LOGOUT\r\n". Ja ports nav norādīts, pēc noklusējuma tiek izmantots 143. ports.

ldap

Atver savienojumu ar LDAP serveri un izpilda LDAP meklēšanas operāciju ar filtru (objectClass=*). Sagaida veiksmīgu pirmā ieraksta pirmā atribūta izgūšanu. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 389. ports.

nntp

Izveido TCP savienojumu un sagaida, ka atbildes pirmās 3 rakstzīmes būs ”200” vai ”201”, pēc tam nosūta ”QUIT\r\n”. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 119. ports.

pop

Izveido TCP savienojumu un sagaida, ka atbildes pirmās 3 rakstzīmes būs ”+OK”, pēc tam nosūta ”QUIT\r\n”. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 110. ports.

smtp

Izveido TCP savienojumu un sagaida, ka atbildes pirmās 3 rakstzīmes būs ”220”, kam seko atstarpe, rindas beigu simbols vai defise. Rindas, kurās ir defise, pieder daudzrindu atbildei, un atbilde tiks nolasīta atkārtoti, līdz tiks saņemta rinda bez defises. Pēc tam nosūta ”QUIT\r\n”. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 25. ports.

ssh

Izveido TCP savienojumu. Ja savienojums ir izveidots, abas puses apmainās ar identifikācijas virkni (SSH-major.minor-XXXX), kur major un minor ir protokola versijas, bet XXXX ir virkne. Zabbix pārbauda, vai ir atrasta specifikācijai atbilstoša virkne, un pēc tam nosūta atpakaļ virkni ”SSH-major.minor-zabbix_agent\r\n” vai ”0\n”, ja tā neatbilst. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 22. ports.

tcp

Izveido TCP savienojumu, neko nesagaidot un neko nenosūtot. Atšķirībā no citām pārbaudēm šai pārbaudei ir jānorāda porta parametrs.

telnet

Izveido TCP savienojumu un sagaida pieteikšanās uzvedni (beidzas ar ’:’). Ja ports nav norādīts, pēc noklusējuma tiek izmantots 23. ports.

Vienuma net.udp.service parametri

ntp

Nosūta SNTP paketi, izmantojot UDP, un pārbauda atbildi saskaņā ar RFC 4330, 5. sadaļu. Ja ports nav norādīts, pēc noklusējuma tiek izmantots 123.

10 proc.get parametri

Pārskats

Vienums **proc.get**[<name>,<user>,<cmdline>,<mode>] tiek atbalstīts Linux, Windows, FreeBSD, OpenBSD un NetBSD.

Ar vienumu atgriezto procesa parametru saraksts atšķiras atkarībā no operētājsistēmas un argumenta ’mode’ vērtības.

Linux

Tālāk norādītie procesa parametri tiek atgriezti Linux vidē katram režīmam:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: procesa nosaukums
ppid: vecākprocesa PID	ppid: vecākprocesa PID	processes: procesu skaits
name: procesa nosaukums	name: procesa nosaukums	vsize: virtuālās atmiņas izmērs
cmdline: komanda ar argumentiem	user: lietotājs (reālais), kura vārdā process darbojas	pmem: reālās atmiņas procentuālā daļa
user: lietotājs (reālais), kura vārdā process darbojas	group: grupa (reālā), kurai process pieder	rss: rezidentās kopas izmērs
group: grupa (reālā), kurai process pieder	uid: lietotāja ID	data: datu segmenta izmērs
uid: lietotāja ID	gid: grupas ID, kurai process pieder	exe: koda segmenta izmērs
gid: grupas ID, kurai process pieder	tid: pavediena ID	lib: koplietojamo bibliotēku izmērs
vsize: virtuālās atmiņas izmērs	tname: pavediena nosaukums	lck: bloķētās atmiņas izmērs
pmem: reālās atmiņas procentuālā daļa	cputime_user: kopējais CPU laiks lietotāja režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)	pin: piesaistīto lapu izmērs

mode=process	mode=thread	mode=summary
rss: rezidentās kopas izmērs	cputime_system: kopējais CPU laiks sistēmas režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)	pte: lapu tabulas ierakstu izmērs
data: datu segmenta izmērs	state: pavediena stāvoklis	size: procesa koda + datu + steka segmentu izmērs
exe: koda segmenta izmērs	ctx_switches: konteksta pārslēgšanu skaits	stk: steka segmenta izmērs
hwm: maksimālais rezidentās kopas izmērs	page_faults: lapu kļūdu skaits	swap: izmantotās mijmaiņas vietas izmērs
lck: bloķētās atmiņas izmērs		cputime_user: kopējais CPU laiks lietotāja režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)
lib: koplietojamo bibliotēku izmērs		cputime_system: kopējais CPU laiks sistēmas režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)
peak: maksimālais virtuālās atmiņas izmērs		ctx_switches: konteksta pārslēgšanu skaits
pin: piesaistīto lapu izmērs		threads: pavedienu skaits
pte: lapu tabulas ierakstu izmērs		page_faults: lapu kļūdu skaits
size: procesa koda + datu + steka segmentu izmērs		pss: proporcionālās kopas izmēra atmiņa
stk: steka segmenta izmērs		
swap: izmantotās mijmaiņas vietas izmērs		
cputime_user: kopējais CPU laiks lietotāja režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)		
cputime_system: kopējais CPU laiks sistēmas režīmā (vērtība izteikta takts impulsos — daliet ar `getconf CLK_TCK` vai `sysconf(_SC_CLK_TCK)`, lai iegūtu sekundes)		
state: procesa stāvoklis (caurspīdīgi iegūts no procfs, pilnā formā)		
ctx_switches: konteksta pārslēgšanu skaits		
threads: pavedienu skaits		
page_faults: lapu kļūdu skaits		
pss: proporcionālās kopas izmēra atmiņa		

BSD bāzēta OS

Tālāk norādītie procesa parametri tiek atgriezti sistēmās FreeBSD, OpenBSD un NetBSD katram režīmam:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: procesa nosaukums

mode=process	mode=thread	mode=summary
ppid: vecākprocesa PID	ppid: vecākprocesa PID	processes: procesu skaits
jid: cietuma ID (tikai FreeBSD)	jid: cietuma ID (tikai FreeBSD)	vsize: virtuālās atmiņas izmērs
jname: cietuma nosaukums (tikai FreeBSD)	jname: cietuma nosaukums (tikai FreeBSD)	pmem: reālās atmiņas procentuālā daļa (tikai FreeBSD)
name: procesa nosaukums	name: procesa nosaukums	rss: rezidentās atmiņas izmērs
cmdline: komanda ar argumentiem	user: lietotājs (reālais), kura vārdā process darbojas	size: procesa izmērs (kods + dati + steks)
user: lietotājs (reālais), kura vārdā process darbojas	group: grupa (reālā), kuras vārdā process darbojas	tsize: teksta (koda) izmērs
group: grupa (reālā), kuras vārdā process darbojas	uid: lietotāja ID	dsize: datu izmērs
uid: lietotāja ID	gid: grupas ID, kuras vārdā process darbojas	ssize: steka izmērs
gid: grupas ID, kuras vārdā process darbojas	tid: pavediena ID	cputime_user: kopējais CPU laiks lietotāja režīmā (norādīts sekundēs)
vsize: virtuālās atmiņas izmērs	tname: pavediena nosaukums	cputime_system: kopējais CPU laiks sistēmas režīmā (norādīts sekundēs)
pmem: reālās atmiņas procentuālā daļa (tikai FreeBSD)	cputime_user: kopējais CPU laiks lietotāja režīmā (norādīts sekundēs)	ctx_switches: konteksta pārslēgšanu skaits
rss: rezidentās atmiņas izmērs	cputime_system: kopējais CPU laiks sistēmas režīmā (norādīts sekundēs)	threads: pavedienu skaits (NetBSD netiek atbalstīts)
size: procesa izmērs (kods + dati + steks)	state: pavediena stāvoklis	stk: steka segmenta izmērs
tsize: teksta (koda) izmērs	ctx_switches: konteksta pārslēgšanu skaits	page_faults: lapu kļūdu skaits
dsize: datu izmērs	io_read_op: reižu skaits, kad sistēmai bija jāveic ievade	fds: failu deskriptoru skaits (tikai OpenBSD)
ssize: steka izmērs	io_write_op: reižu skaits, kad sistēmai bija jāveic izvade	swap: izmantotās maiņvietas izmērs
cputime_user: kopējais CPU laiks lietotāja režīmā (norādīts sekundēs)		io_read_op: reižu skaits, kad sistēmai bija jāveic ievade
cputime_system: kopējais CPU laiks sistēmas režīmā (norādīts sekundēs)		io_write_op: reižu skaits, kad sistēmai bija jāveic izvade
state: procesa stāvoklis (gaidīšana uz disku/darbojas/guļ/izsekošanas apturēšana/zombijs/cits)		
ctx_switches: konteksta pārslēgšanu skaits		
threads: pavedienu skaits (NetBSD netiek atbalstīts)		
page_faults: lapu kļūdu skaits		
fds: failu deskriptoru skaits (tikai OpenBSD)		
swap: izmantotās maiņvietas izmērs		
io_read_op: reižu skaits, kad sistēmai bija jāveic ievade		
io_write_op: reižu skaits, kad sistēmai bija jāveic izvade		

Windows

Tālāk norādītie procesa parametri tiek atgriezti sistēmā Windows katram režīmam:

mode=process	mode=thread	mode=summary
pid: PID	pid: PID	name: procesa nosaukums
ppid: vecākprocesa PID	ppid: vecākprocesa PID	processes: procesu skaits
name: procesa nosaukums	name: procesa nosaukums	vmsize: virtuālās atmiņas lielums
user: lietotājs, kura vārdā process darbojas	user: lietotājs, kura vārdā process darbojas	wkset: procesa darba kopas lielums

mode=process	mode=thread	mode=summary
sid: lietotāja SID	sid: lietotāja SID	cputime_user: kopējais CPU laiks, kas pavadīts lietotāja režīmā (norādīts milisekundēs)
vmsize: virtuālās atmiņas lielums	tid: pavediena ID	cputime_system: kopējais CPU laiks, kas pavadīts sistēmas režīmā (norādīts milisekundēs)
wkset: procesa darba kopas lielums		threads: pavedienu skaits
cputime_user: kopējais CPU laiks, kas pavadīts lietotāja režīmā (norādīts milisekundēs)		page_faults: lapu kļūdu skaits
cputime_system: kopējais CPU laiks, kas pavadīts sistēmas režīmā (norādīts milisekundēs)		handles: turiņu skaits
threads: pavedienu skaits		io_read_b: nolasīto IO baitu skaits
page_faults: lapu kļūdu skaits		io_write_b: ierakstīto IO baitu skaits
handles: turiņu skaits		io_read_op: IO nolasīšanas operācijas
io_read_b: nolasīto IO baitu skaits		io_write_op: IO rakstīšanas operācijas
io_write_b: ierakstīto IO baitu skaits		io_other_b: pārsūtīto IO baitu skaits, izņemot nolasīšanas un rakstīšanas operācijas
io_read_op: IO nolasīšanas operācijas		io_other_op: IO operācijas, izņemot nolasīšanas un rakstīšanas operācijas
io_write_op: IO rakstīšanas operācijas		
io_other_b: pārsūtīto IO baitu skaits, izņemot nolasīšanas un rakstīšanas operācijas		
io_other_op: IO operācijas, izņemot nolasīšanas un rakstīšanas operācijas		

11 Nesasniedzama/nepieejama hosta saskarnes iestatījumi

Pārskats

Vairāki konfigurācijas parametri nosaka, kā Zabbix serverim jārīkojas, ja aģenta pārbaude (Zabbix, SNMP, IPMI, JMX) neizdodas un hosta saskarne kļūst nesasniedzama.

Nesasniedzama saskarne

Hosta saskarne tiek uzskatīta par nesasniedzamu pēc neveiksmīgas pārbaudes (tīkla kļūda, noildze), ko veic Zabbix, SNMP, IPMI vai JMX aģenti. Kopš Zabbix 6.2.0 arī aktīvās Zabbix aģenta pārbaudes ietekmē saskarnes pieejamību. Kad aktīvās pārbaudes kļūst nepieejamas, tās ietekmē aģenta saskarnes kopējo pieejamības statusu.

No brīža, kad saskarne kļūst nesasniedzama, **UnreachableDelay** nosaka, cik bieži tā tiek pārbaudīta atkārtoti, izmantojot vienu no vienumiem (ieskaitot LLD kārtulas). Šīs atkārtotās pārbaudes veic nesasniedzamības aptaujātāji (vai IPMI aptaujātāji IPMI pārbaudēm). Pēc noklusējuma intervāls starp secīgām sasniedzamības pārbaudēm ir 15 sekundes.

Attention:
Pārbaudes, ko veic asinhronie aptaujātāji, netiek pārvietotas uz nesasniedzamības aptaujātājiem.

Zabbix servera žurnālā nesasniedzamība tiek norādīta ar šādiem ziņojumiem:

```
Zabbix agent item "system.cpu.load[percpu,avg1]" on host "New host" failed: first network error, wait for
Zabbix agent item "system.cpu.load[percpu,avg15]" on host "New host" failed: another network error, wait f
```

Žurnāla ziņojumos ir norādīts precīzs neveiksmīgais vienums un tā tips (Zabbix aģents).

Note:
Parametrs *Timeout* arī ietekmē, cik ātri saskarne tiek pārbaudīta atkārtoti nesasniedzamības laikā. Ja Timeout ir iestatīts uz 20 sekundēm un UnreachableDelay uz 30 sekundēm, nākamā pārbaude notiks pēc 50 sekundēm no pirmā mēģinājuma.

Parametrs **UnreachablePeriod** nosaka kopējo nesasniedzamības perioda ilgumu. Pēc noklusējuma UnreachablePeriod ir 45 sekundes. Šai vērtībai jābūt vairākas reizes lielākai par UnreachableDelay, lai nodrošinātu, ka saskarne tiek pārbaudīta atkārtoti vairākas reizes, pirms tā tiek atzīmēta kā nepieejama.

Iekšējais vienums zabbix[host,active_agent,available] ļauj uzraudzīt aktīvo pārbaužu pieejamību nesasniedzamības scenārijos.

Saskarnes pārslēgšana atpakaļ uz pieejamu

Kad nesasniedzamības periods ir beidzies, saskarne tiek aptaujāta vēlreiz, samazinot prioritāti vienumam, kas padarīja saskarni nesasniedzamu. Ja nesasniedzamā saskarne atkal kļūst pieejama, uzraudzība automātiski atgriežas normālā režīmā:

```
atsāk Zabbix aģenta pārbaudes hostam "New host": savienojums atjaunots
```

Note:
Kad saskarne kļūst pieejama, hosts neaptaujā visus savus vienumus nekavējoties divu iemeslu dēļ:

- Tas var pārslogot hostu.
- Saskarnes atjaunošanas laiks ne vienmēr sakrīt ar plānoto vienumu aptaujas grafika laiku.

Tāpēc pēc tam, kad saskarne kļūst pieejama, vienumi netiek aptaujāti nekavējoties, bet tiek pārplānoti uz nākamo aptaujas ciklu.

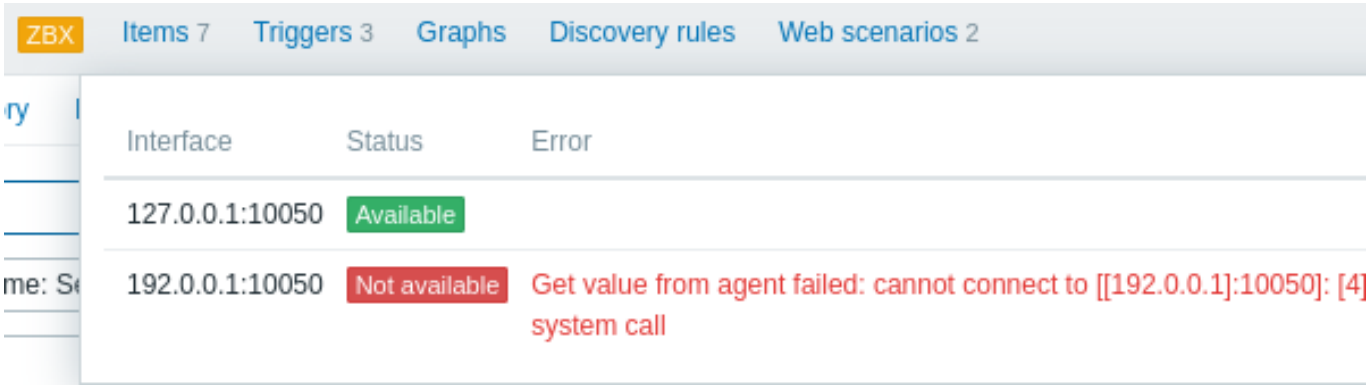
Nepieejama saskarne

Pēc tam, kad beidzas UnreachablePeriod un saskarne nav atkal kļuvusi pieejama, saskarne tiek uzskatīta par nepieejamu.

Servera žurnālā tas tiek norādīts ar šādiem ziņojumiem:

```
temporarily disabling Zabbix agent checks on host "New host": interface unavailable
```

un lietotāja saskarnē hosta pieejamības ikona mainās no zaļas/pelēkas uz dzeltenu/sarkanu (informāciju par nepieejamo saskarni var redzēt uznirstošajā lodziņā, kas tiek parādīts, novietojot peles kursoru uz hosta pieejamības ikonas):



Parametrs **UnavailableDelay** nosaka, cik bieži saskarne tiek pārbaudīta tās nepieejamības laikā.

Pēc noklusējuma tas ir 60 sekundes (tātad šajā gadījumā "temporarily disabling", kā norādīts iepriekš redzamajā žurnāla ziņojumā, nozīmē pārbaužu atspējošanu uz vienu minūti).

Kad savienojums ar saskarni tiek atjaunots, uzraudzība arī automātiski atgriežas normālā režīmā:

```
enabling Zabbix agent checks on host "New host": interface became available
```

12 Attālināta Zabbix statistikas uzraudzība

Pārskats

Ir iespējams padarīt dažus Zabbix servera un starpniekservera iekšējos metriku datus attālināti pieejamus no citas Zabbix instances vai trešās puses rīka. Tas var būt noderīgi, lai atbalsta sniedzēji/pakalpojumu sniedzēji varētu attālināti uzraudzīt savu klientu Zabbix serverus/starpniekserverus vai organizācijās, kur Zabbix nav galvenais uzraudzības rīks, lai Zabbix iekšējās metrikas varētu uzraudzīt trešās puses sistēma centralizētas uzraudzības konfigurācijā.

Zabbix iekšējā statistika tiek atklāta konfigurējamam adrešu kopumam, kas norādīts jaunajā 'StatsAllowedIP' servera/starpniekservera parametrā. Pieprasījumi tiks pieņemti tikai no šīm adresēm.

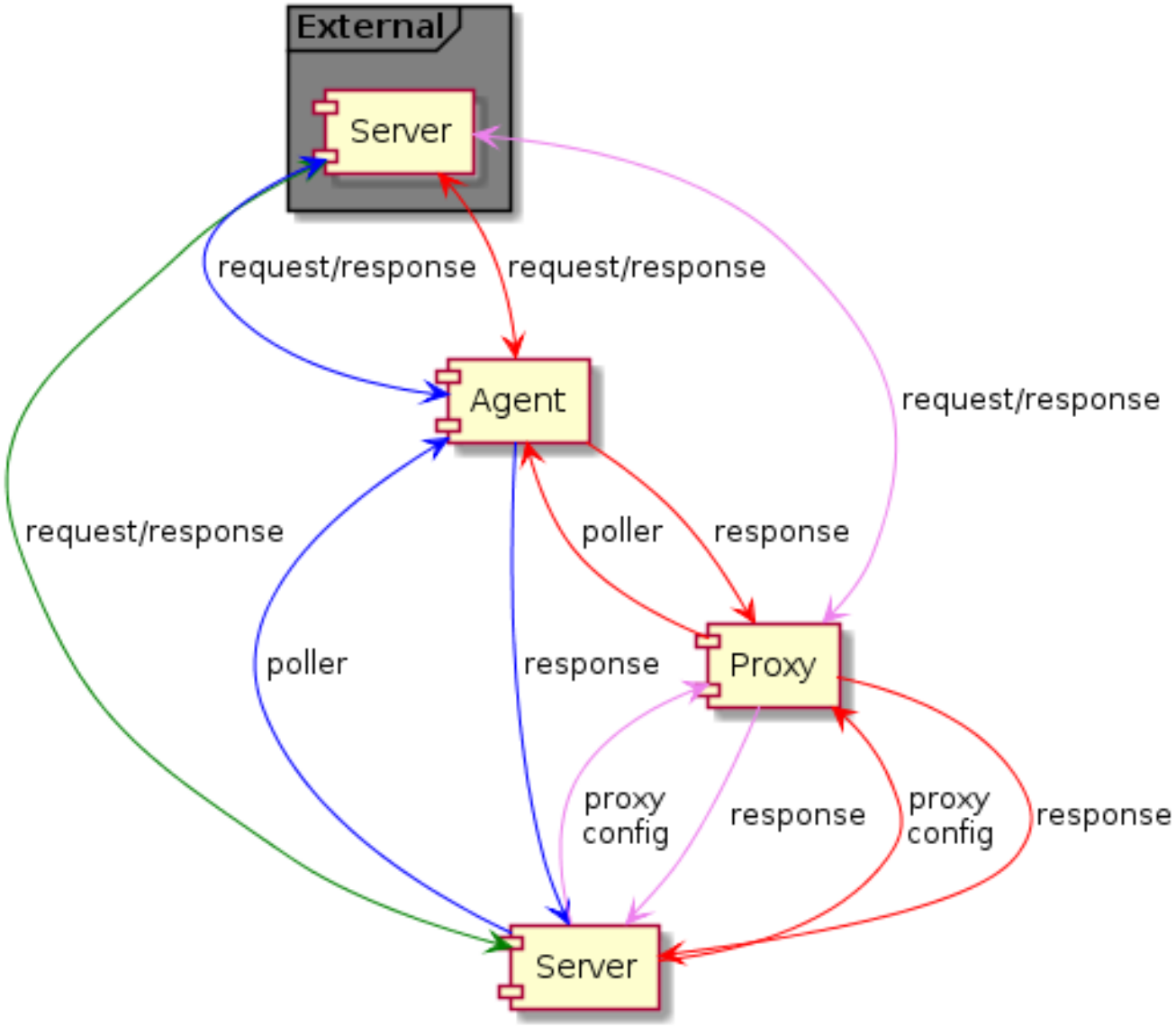
Vienumi

Lai konfigurētu iekšējās statistikas vaicāšanu citā Zabbix instancē, jūs varat izmantot divus vienumus:

- iekšējais vienums `zabbix[stats,<ip>,<port>]` — tiešiem attālinātiem Zabbix servera/starpniekservera vaicājumiem. <ip> un <port> tiek izmantoti, lai identificētu mērķa instanci.
- aģenta vienums `zabbix.stats[<ip>,<port>]` — uz aģentu balstītiem attālinātiem Zabbix servera/starpniekservera vaicājumiem. <ip> un <port> tiek izmantoti, lai identificētu mērķa instanci.

Skatiet arī: Iekšējie vienumi, Zabbix aģenta vienumi

Tālāk redzamā diagramma ilustrē jebkura no šiem vienumiem izmantošanu atkarībā no konteksta.



- — Serveris → ārēja Zabbix instance (`zabbix[stats,<ip>,<port>]`)
- — Serveris → starpniekserveris → ārēja Zabbix instance (`zabbix[stats,<ip>,<port>]`)
- — Serveris → aģents → ārēja Zabbix instance (`zabbix.stats[<ip>,<port>]`)
- — Serveris → starpniekserveris → aģents → ārēja Zabbix instance (`zabbix.stats[<ip>,<port>]`)

Lai pārliecinātos, ka mērķa instance atļauj tās vaicāšanu no ārējās instances, norādiet ārējās instances adresi mērķa instances parametrā 'StatsAllowedIP'.

Pieejamie metriķi

Statistikas vienumi apkopo statistiku vairumā un atgriež JSON, kas ir pamats atkarīgajiem vienumiem datu iegūšanai. Tālāk norādītie iekšējie metriķi tiek atgriezti ar jebkuru no šiem diviem vienumiem:

- `zabbix[boottime]`
- `zabbix[hosts]`
- `zabbix[items]`
- `zabbix[items_unsupported]`
- `zabbix[preprocessing]` (tikai serverim)
- `zabbix[preprocessing_queue]` (tikai serverim)
- `zabbix[process,<type>,<mode>,<state>]` (tikai uz procesa tipa balstīta statistika)
- `zabbix[rcache,<cache>,<mode>]`
- `zabbix[requiredperformance]`

- zabbix[triggers] (tikai serverim)
- zabbix[uptime]
- zabbix[vcache,buffer,<mode>] (tikai serverim)
- zabbix[vcache,cache,<parameter>]
- zabbix[version]
- zabbix[vmware,buffer,<mode>]
- zabbix[wcache,<cache>,<mode>] (trends kešatmiņas tips tikai serverim)

Veidnes

Veidnes ir pieejamas Zabbix servera vai starpniekservera iekšējo metriku attālinātai uzraudzībai no ārējas instances:

- Attālināta Zabbix servera veselība
- Attālināta Zabbix starpniekservera veselība

Ņemiet vērā, ka, lai izmantotu veidni vairāku ārējo instanču attālinātai uzraudzībai, katras ārējās instances uzraudzībai ir nepieciešams atsevišķs hosts.

Trapper process

Iekšējo metriku pieprasījumu saņemšanu no ārējas Zabbix instances apstrādā trapper process, kas validē pieprasījumu, apkopo metrikas, izveido JSON datu buferi un nosūta sagatavoto JSON atpakaļ, piemēram, no serveris:

```
{
  "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
      }
    }
  }
}
```

Iekšējās rindas vienumi

Ir arī vēl divi vienumi, kas īpaši ļauj attālināti vaicāt iekšējās rindas statistiku citā Zabbix instancē:

- `zabbix[stats,<ip>,<port>,queue,<from>,<to>]` iekšējais vienums — tiešiem iekšējās rindas vaicājumiem attālinātā Zabbix serverī/starpniekserverī
- `zabbix.stats[<ip>,<port>,queue,<from>,<to>]` aģenta vienums — uz aģentu balstītiem iekšējās rindas vaicājumiem attālinātā Zabbix serverī/starpniekserverī

Skatiet arī: Iekšējie vienumi, Zabbix aģenta vienumi

13 Kerberos konfigurēšana ar Zabbix

Pārskats

Kerberos autentifikāciju var izmantot tīmekļa uzraudzībā un HTTP vienumos Zabbix.

Šajā lapā ir aprakstīts piemērs, kā konfigurēt Kerberos Zabbix serverim, lai Debian/Ubuntu vidē veiktu `www.example.com` tīmekļa uzraudzību, izmantojot Kerberos principālu Zabbix procesam.

Konfigurācija

1. Instalējiet KDC un klienta utilītas:

```
sudo apt update
sudo apt install krb5-kdc krb5-admin-server krb5-user
```

Pakotņu iestatīšanas laikā atbildiet uz uzvednēm, piemēram:

```
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. Piesaistiet saprotamu resursdatora nosaukumu (neobligāti, lokālai testēšanai).

Rediģējiet /etc/hosts un pievienojiet ierakstu savam DC un tīmekļa serverim, ja jums nav DNS:

```
sudo vi /etc/hosts
```

Piemēra rinda, ko varat pievienot:

```
192.168.1.100  dc01.example.com dc01
```

3. Konfigurējiet Kerberos klientu un KDC realm:

```
sudo vi /etc/krb5.conf
```

Piemēra iestatījumi:

```
[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
```

Ja plānojat izmantot `.localdomain` vai citus nepubliskus nosaukumus, pievienojiet skaidras domain→realm atbilstības, lai hostname→realm kartēšana darbotos. Neatbilstības šeit izraisa `Server not found in Kerberos database` kļūdas.

4. Inicializējiet Kerberos datubāzi (vienreiz, KDC hostā). Kad tiek prasīts, iestatiet drošu galveno paroli:

```
sudo krb5_newrealm
```

5. Izveidojiet `HTTP/host.fqdn@REALM` principal, izmantojot precīzu resursdatora nosaukumu, ko klienti izmantos; ieteicams izmantot mazos burtus (piemēram, `HTTP/web.example.com@EXAMPLE.COM`). Lielo/mazo burtu vai nosaukuma neatbilstība izraisa `Server not found in Kerberos database`.

```
sudo kadmin.local
```

`kadmin.local` iekšpusē:

```
addprinc kerb-admin@EXAMPLE.COM      # administratīvais principal
addprinc -randkey HTTP/dc01.example.com@EXAMPLE.COM
ktadd -k /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
quit
```

Pārvietojiet keytab uz tīmekļa hostu (vai atstājiet lokāli, ja tā ir viena un tā pati mašīna) un iestatiet Apache izmantojamas atļaujas:

```
chown www-data:www-data /etc/apache2/http.keytab
chmod 600 /etc/apache2/http.keytab
#### verify
sudo -u www-data -k /etc/apache2/http.keytab
```

6. Instalējiet un iespējojiet Apache GSSAPI moduli:

```
sudo apt install libapache2-mod-auth-gssapi
sudo a2enmod auth_gssapi
sudo a2enmod headers
sudo systemctl restart apache2
```

> **Attention:**
> Ne visas mod_auth_gssapi versijas atbalsta katru `Gssapi*` direktīvu. Ja Apache neizdodas palaist ar `Invalid command 'GssapiCredStore'`, noņemiet neatbalstīto direktīvu vai jauniniet moduli.

7. Konfigurējiet VirtualHost (pielāgojiet `DocumentRoot` / ceļu uz savu Zabbix UI):

```
sudo vi /etc/apache2/sites-available/zabbix.conf
```

`zabbix.conf` iekšpusē:

```
<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>
```

Restartējiet Apache:

```
sudo systemctl restart apache2
```

8. Iespējojiet/palaidiet KDC servisus un pārbaudiet klausīšanās portus (KDC hostā):

```
sudo systemctl enable --now krb5-kdc krb5-admin-server
ss -tnlp | grep :80    # or: sudo netstat -tnlp | grep :80
```

9. Iegūstiet TGT testēšanai (palaidiet kā lietotājs, kurš izmantos biļeti).

Biļešu sarakstā jums vajadzētu redzēt `krbtgt/EXAMPLE.COM@EXAMPLE.COM`. Palaidiet `kinit` kā tas pats OS lietotājs, kuram nepieciešama biļete (piemēram, `zabbix` tīmekļa pārbaudēm vai `www-data`/Apache interaktīviem pārlūka SSO testiem). Citādam

OS lietotājam izsniegtās biļetes nebūs redzamas, ja vien netiks pielāgoti KRB5CCNAME un atļaujas.

```
kinit kerb-admin@EXAMPLE.COM
klist
```

10. Pārbaudiet SPNEGO apmaiņu ar curl (no klienta ar derīgu TGT). 200 OK (vai pāradresācija uz lietotni) norāda, ka SPNEGO izdevās:

```
curl -v --negotiate -u : http://dc01.example.com/
```

11. Neobligāti, ja Zabbix lietotāja saskarnei jāpieņem ar HTTP autentificētas pieteikšanās, iespējojiet HTTP autentifikāciju Zabbix lietotāja saskarnē (ui/conf/zabbix.conf.php):

```
$ALLOW_HTTP_AUTH = true;
```

Tīmekļa UI atveriet *Users* > *Authentication* un pārejiet uz cilni *HTTP settings*. Atzīmējiet izvēles rūtiņu *Enable HTTP authentication* un uznirstošajā logā noklikšķiniet uz *Ok*. Nolaižamajā sarakstā *Default login form* atlasiet "HTTP login form". Izlemiet, vai *Case-sensitive login* atbilst jūsu direktorija politikai. Lai pabeigtu, noklikšķiniet uz pogas *Update*.

12. Pārlūka konfigurācija (kā piemērs izmantots Firefox): iestatiet network.negotiate-auth.trusted-uris uz hostiem, kas veic Negotiate (dc01.example.com), lai pārlūks automātiski sūtītu Kerberos marķierus.

about:config iekšpusē:

```
network.negotiate-auth.trusted-uris = dc01.example.com
```

Tagad, apmeklējot http://dc01.example.com, jums būtu jāielogojas Zabbix bez formas.

13. Uzturiet atslēgas/biļetes aktuālas. Noklusējuma Kerberos biļetes derīguma laiks ir aptuveni 10 stundas. Pievienojiet cron/systemd taimeri, lai izvairītos no derīguma termiņa beigām:

```
####for the web service
kinit -kt /etc/apache2/http.keytab HTTP/dc01.example.com@EXAMPLE.COM
####for the monitoring user
kinit -kt /var/lib/zabbix/kerb.keytab kerb-admin@EXAMPLE.COM
```

14. Uzturēšanas pārbaudes:

- klist -k /etc/apache2/http.keytab — pārbaudiet, vai keytab satur servisa principal.
- sudo tail -f /var/log/apache2/error.log — sekojiet GSSAPI kļūdām (gss_acquire_cred[_from]() failed to get server creds nozīmē keytab/atļauju problēmu vai trūkstošu principal).
- curl --negotiate, kas atgriež 401/403, bieži nozīmē nepareizu principal, nav biļetes, host galvenes neatbilstību vai failu sistēmas atļauju problēmu; pārbaudiet žurnālus un /etc/krb5.conf domēna kartējumus.

Drošības un failu atļauju piezīmes

Keytab failiem jābūt lasāmiem tikai tam kontam, kuram tie ir nepieciešami. Atļauju piemēri: 0400, īpašnieks zabbix:zabbix — zabbix lietotāja keytab failam, vai 0440 un root:www-data — Apache keytab failam.

Izvairieties no ilgtermiņa atklāta teksta paroļu glabāšanas hostā. Ja iespējams, izmantojiet keytab failus vai domēnam pievienotas datora galvenās identitātes.

Palaižot testus vai skriptus, kas iestata KRB5CCNAME vai kopē keytab failus, pēc darbības vēlreiz pārbaudiet īpašnieku un atļaujas — tīmekļa servera atteikums pieņemt akreditācijas datus bieži ir saistīts ar failu atļauju problēmu.

14 modbus.get parametri

Pārskats

Tālāk esošajā tabulā ir sniegta informācija par modbus.get vienums parametriem.

Parametri

Parametrs	Apraksts	Noklusējuma vērtības	Piemērs
endpoint	Galapunkta protokols un adrese, definēta kā `protocol://connection_string` Iespējamās protokola vērtības: *rtu*, *ascii* (tikai Agent 2), *tcp* Savienojuma virknes formāts: ar *tcp* - `address:port` ar seriālo līniju: *rtu*, *ascii* - `port_name:speed:params` kur 'speed' - 1200, 9600 utt. 'params' - datu biti (5,6,7 vai 8), paritāte (n,e vai o nozīmē none/even/odd), stopbiti (1 vai 2)	protocol: none *rtu/ascii* protokols: port_name: none speed: 115200 params: 8n1 *tcp* protokols: 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	Ierīces Modbus adrese, kurai tas ir paredzēts (no 1 līdz 247), skatiet MODBUS Messaging Implementation Guide (23. lappuse) tcp ierīce (nevis GW) ignorēs šo lauku	serial: 1 tcp: 255 (0xFF)	2
function	Tukšs vai atbalstītas funkcijas vērtība: 1 - Read Coil, 2 - Read Discrete Input, 3 - Read Holding Registers, 4 - Read Input Registers	tukšs	3
address	Pirmā reģistra, spoles vai ievades adrese. Ja 'function' ir tukšs, tad 'address' jābūt diapazonā priekš: Coil - 00001 - 09999 Discrete input - 10001 - 19999 Input register - 30001 - 39999 Holding register - 40001 - 49999 Ja 'function' nav tukšs, lauks 'address' būs no 0 līdz 65535 un tiks izmantots bez izmaiņām (PDU)	tukša function: 00001 netukša function: 0	9999
count	Secīgu 'type' elementu skaits, kas tiks nolasīti no ierīces, kur: Coil vai Discrete input gadījumā 'type' = 1 bits citos gadījumos: (count*sizeof(type))/2 = faktiskais nolasāmo reģistru skaits Ja 'offset' nav 0, vērtība tiks pieskaitīta pie 'faktiskā skaita' Pieļaujamais 'faktiskā skaita' diapazons ir 1:65535	1	2
type	Datu tips: Read Coil un Read Discrete Input gadījumā - *bit* Read Holding Registers un Read Input Registers gadījumā: *int8* - 8 bitu *uint8* - 8 bitu (bez zīmes) *int16* - 16 bitu *uint16* - 16 bitu (bez zīmes) *int32* - 32 bitu *uint32* - 32 bitu (bez zīmes) *float* - 32 bitu *uint64* - 64 bitu (bez zīmes) *double* - 64 bitu	bit uint16	uint64

Parametrs	Apraksts	Noklusējuma vērtības	Piemērs
endianness	Baitu secības tips: *be* - Big Endian *le* - Little Endian *mbe* - Mid-Big Endian *mle* - Mid-Little Endian Ierobežojumi: 1 bitam - be 8 bitiem - be,le 16 bitiem - be,le	be	le
offset	Reģistru skaits, sākot no 'address', kuru rezultāts tiks atmests. Katra reģistra izmērs ir 16 bitu (nepieciešams, lai atbalstītu iekārtas, kas neatbalsta nejaušas piekļuves nolasīšanu).	0	4

15 Pielāgotu VMware veiktspējas skaitītāju nosaukumu izveide

Pārskats

VMware veiktspējas skaitītāja ceļam ir `group/counter[rollup]` formāts, kur:

- `group` - veiktspējas skaitītāja grupa, piemēram, *cpu*
- `counter` - veiktspējas skaitītāja nosaukums, piemēram, *usagemhz*
- `rollup` - veiktspējas skaitītāja apkopojuma tips, piemēram, *average*

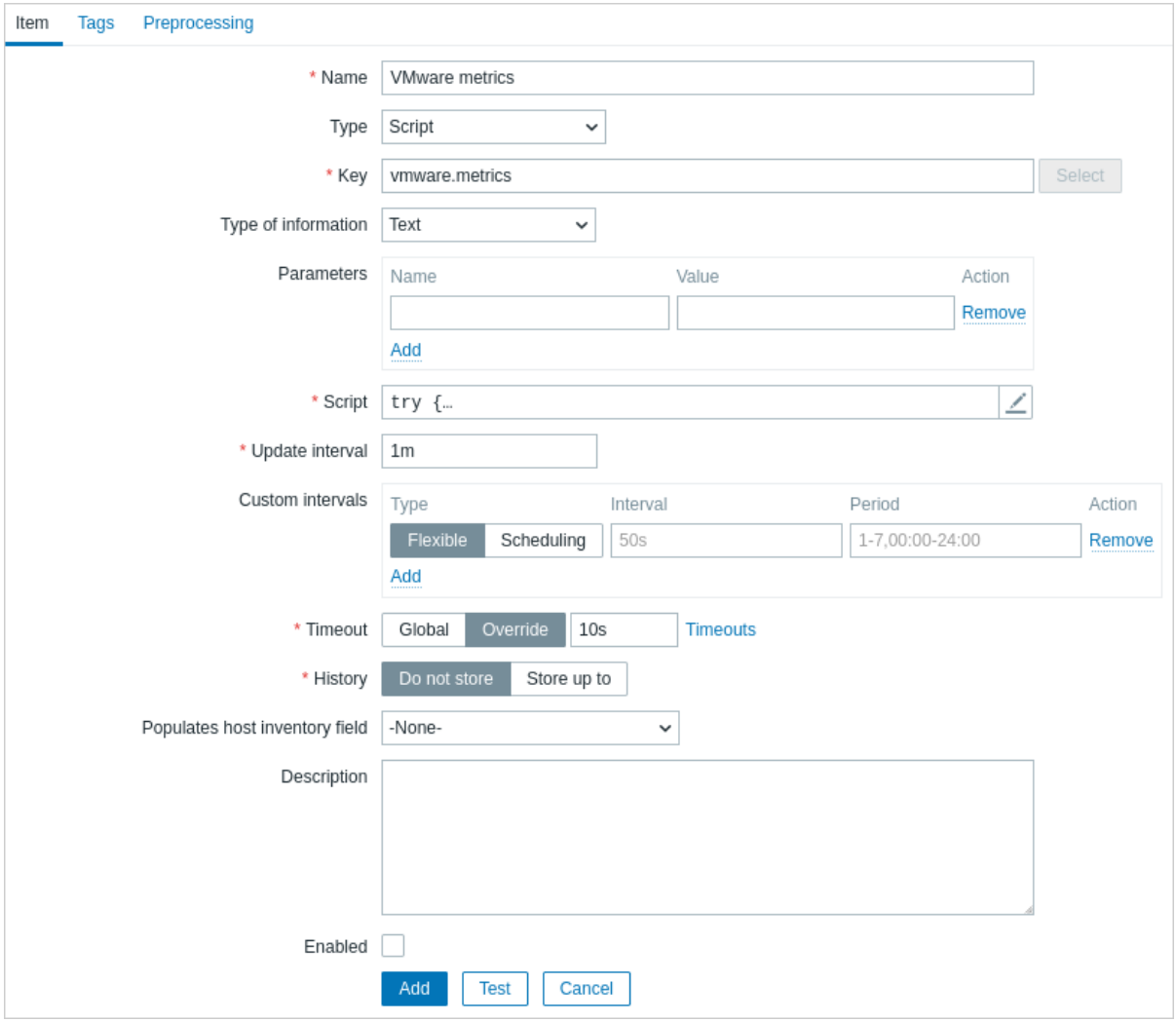
Tātad iepriekš minētais piemērs dotu šādu skaitītāja ceļu: `cpu/usagemhz[average]`

Veiktspējas skaitītāju grupu apraksti, skaitītāju nosaukumi un apkopojuma tipi ir atrodami VMware dokumentācijā.

Ir iespējams iegūt iekšējos nosaukumus un izveidot pielāgotus veiktspējas skaitītāju nosaukumus, izmantojot skripta vienums Zabbix.

Konfigurācija

1. Izveidojiet atspējotu Script vienumu galvenajā VMware hostā (kur atrodas **eventlog[]** vienums) ar šādiem parametriem:



- *Name*: VMware metrics
- *Type*: Script
- *Key*: vmware.metrics
- *Type of information*: Text
- *Script*: kopējiet un ielīmējiet zemāk norādīto skriptu
- *Timeout*: 10
- *History*: Neglabāt
- *Enabled*: nav atzīmēts

Skripts

```
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;
```

Kad vienums ir konfigurēts, nospiediet pogu *Test*, pēc tam nospiediet *Get value*.

Kopējiet saņemto XML jebkurā XML formatētājā un atrodiet vajadzīgo metriku.

Viena metrikas XML piemērs:

```
<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>
```

Izmantojiet XPath, lai no saņemtā XML izvilktu skaitītāja ceļu. Iepriekš minētajam piemēram XPath būs šāds:

field	xPath	value
group	//groupInfo[../key=6]/key	cpu
counter	//nameInfo[../key=6]/key	usagemhz
rollup	//rollupType[../key=6]	average

Rezultējošais veiktspējas skaitītāja ceļš šajā gadījumā ir: `cpu/usagemhz[average]`

16 Atgrieztās vērtības priekš system.sw.packages.get

Pārskats

Šajā sadaļā ir sniegta informācija par atgrieztajām vērtībām Zabbix aģents vienumam `system.sw.packages.get`.

Detalizēta informācija

Šī vienuma izvade ir objektu masīvs, kur katrs objekts satur šādas atslēgas:

- **name** — pakotnes nosaukums
- **manager** — pakotņu pārvaldnieks, kas ziņoja šos datus (`rpm`, `dpkg`, `pacman`, `pkgtool` vai `portage`)
- **version** — pakotnes versija
- **size** — nesaspiestas pakotnes lielums baitos (ja nav pieejams, iestatīts uz 0)
- **arch** — pakotnes arhitektūra
- **buildtime** — objekts ar 2 ierakstiem:
  - **timestamp** — UNIX laika zīmogs, kad pakotne tika izveidota (ja nav pieejams, iestatīts uz 0)
  - **value** — cilvēkam lasāms datums un laiks, kad pakotne tika izveidota (ja nav pieejams, iestatīts uz tukšu virkni)
- **installtime** — objekts ar 2 ierakstiem:
  - **timestamp** — UNIX laika zīmogs, kad pakotne tika instalēta (ja nav pieejams, iestatīts uz 0)
  - **value** — cilvēkam lasāms datums un laiks, kad pakotne tika instalēta (ja nav pieejams, iestatīts uz tukšu virkni)

Piemērs:

```
[
    {
        "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 Atgrieztās vērtības priekš net.dns.get

Pārskats

Šajā sadaļā ir sniegta informācija par `net.dns.get` Zabbix aģents 2 vienuma atgrieztajām vērtībām.

Detalizēta informācija

Šī vienuma izvade ir objekts, kas satur DNS ieraksta informāciju, pamatojoties uz vienuma atslēgā norādītajiem parametriem.

Piemēram, vienums `net.dns.get[,example.com]` var atgriezt šādu atteikta vaicājuma JSON:

```
{
    "flags": [
        "RA"
    ],
    "query_time": "0.019030",
    "question_section": [
        {
            "qclass": "IN",
            "qname": "example.com.",
            "qtype": "SOA"
        }
    ],
    "response_code": "REFUSED",
    "zbx_error_code": 0
}
```

Norādot DNS servera IP adresi, vienums `net.dns.get[192.0.2.0,example.com]` var atgriezt šādu 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
}
```


Ja rodas savienojuma problēma, vienums `net.dns.get[192.0.2.0,example.com]` var atgriezt kļūdu:


```
{
    "zbx_error_code": -1,
    "zbx_error_msg": "Communication error: read udp 192.0.2.0:12345->192.0.2.0:53: i/o timeout"
}
```


Ir iespējami šādi kļūdu kodu tipi:

Scenārijs	"zbx_error_code"	"zbx_error_msg"
Nav kļūdu, un DNS atbilde tika saņemta un parsēta.	0	
DNS nav pieejams.	-1	"Communication error"
JSON parsēšanas laikā rodas kļūda	-2	"Received unexpected response"

Ar papildu parametriem vienums `net.dns.get[192.0.2.0,example.com,ANY,5,5,tcp,"cdflag,rdflag,dnssec,nsid,edns0,aa` var atgriezt šādu 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
}
```


Skatiet arī

Lai iegūtu vairāk informācijas par DNS ierakstiem, skatiet:

- Domēna vārdi — ieviešana un specifikācija
- Domēna vārdu sistēmas (DNS) parametri

18 Piezīmes par system.cpu.util vienumiem operētājsistēmā Windows

vienums system.cpu.util nodrošina CPU noslodzes procentuālo vērtību.

Kad Zabbix aģents operētājsistēmai Windows palaiž savācēja procesu, tiek inicializēts buferis N vienumiem N loģiskajiem procesoriem (pavedieniem). Vērtības buferī tiek periodiski atjauninātas, izmantojot Windows veiktspējas skaitītājus. Šīs vērtības tiek ziņotas, kad tiek pieprasīts `system.cpu.util[n]`, kur 0 <= n < N.

Vēl viens vienums buferī tiek izmantots `system.cpu.util[total]`.

Zabbix automātiski izvēlas dažādus veiktspējas skaitītājus, lai atbalstītu gan NUMA sistēmas, gan vecākas Windows versijas bez procesoru grupu atbalsta.

Ja loģisko procesoru (pavedienu) skaits ir mazāks vai vienāds ar 64 un procesoru grupu skaits ir vienāds ar 1, tad:

```
N ir loģisko procesoru (pavedienu) skaits.
```

```
\Processor(_Total)\% Processor Time
\Processor(0)\% Processor Time
\Processor(1)\% Processor Time
\Processor(2)\% Processor Time
...
\Processor(N-1)\% Processor Time
```

Pretējā gadījumā:

```
G ir procesoru grupu skaits, un N ir loģisko procesoru (pavedienu) skaits grupā.

\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 Lielu JSON vērtību atbalsts

Lai nodrošinātu uzticamu lielu vērtību apstrādi ar JSON datu tipu (1MiB vai vairāk), pārskatiet un pēc nepieciešamības piemērojiet šādas konfigurācijas izmaiņas:

1. Ja izmantojat MySQL vai MariaDB, palieliniet maksimālo atļauto paketes izmēru failā `/etc/mysql/my.cnf` (dažās sistēmās noklusējuma vērtība var būt 1MB):

```
[mysqld]
max_allowed_packet=128M

[mysqldump]
max_allowed_packet=1G
```

2. Pēc nepieciešamības pielāgojiet papildu MySQL/MariaDB servera sistēmas mainīgos un InnoDB sistēmas mainīgos. Piemēram:

```
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. Palieliniet `HistoryCacheSize` līdz 2G Zabbix servera konfigurācijas failā.

4. Ja izmantojat Zabbix starpniekserveri, palieliniet `HistoryCacheSize` līdz 2G un `ProxyMemoryBufferSize` līdz 2G starpniekservera konfigurācijas failā.

5. Ja izmantojat Zabbix aģentu, palieliniet komunikācijas noildzes starp Zabbix serveri/starpniekserveri un aģentu (piemēram, vienuma noildzes vai parametra `Timeout` vērtību Zabbix servera/starpniekservera un aģenta konfigurācijas failos). Pretējā gadījumā ziņojumu apmaiņa starp tiem var netikt pabeigta, un var tikt parādītas kļūdas, piemēram, *"message length does not match expected length"*. Pēc nepieciešamības pārskatiet arī citas noildzes, piemēram, skripta noildzi vai vienuma testa noildzi.

5 Atbalstītās funkcijas

Skatiet Trigera izteiksme.

6 Makrosi

Ir iespējams izmantot jau gatavus atbalstītos makrosus un lietotāja makrosus, ko atbalsta pēc atrašanās vietas.

1 Atrašanās vietas atbalstītie makrosi

Pārskats

Tabulā makrosi, ko atbalsta atrašanās vieta ir ietverta atsauce uz visiem makrosiem (iebūvētajiem, izteiksmju un lietotāja definējamiem), kas tiek atbalstīti konkrētajā atrašanās vietā.

Ja tiek atbalstīti visi makrosi no saistītās grupas, tad tiek norādīts tikai grupas nosaukums. Noklikšķiniet uz katra makrosa nosaukuma vai grupas nosaukuma, lai skatītu pilnu informāciju par makrosiem.

Sarakstā makrosu detalizēta informācija ir ietverts detalizēts visu **iebūvēto** makrosu saraksts, kas sagrupēti pēc lietojumprogrammas jomas. Ņemiet vērā, ka, lai pielāgotu makrosu vērtības (piemēram, saīsinātu vai izvilktu noteiktas apakšvirknes), varat izmantot makrosu funkcijas.

Makrosi, ko atbalsta atrašanās vieta

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Darbība		
Laika perioda nosacījums, Noklusējuma darbības soļa ilgums, Soļa ilgums	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts
Paziņojumi un komandas		
Balstīts uz trigeri	Darbības makrosi Datuma/laika makrosi {ESC.HISTORY} Notikuma makrosi, Cēloņa/simptoma notikuma makrosi Notikuma atjaunināšanas makrosi: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} Izteiksmes makross: {?EXPRESSION} {FUNCTION.VALUE} Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Mērķa hosta makrosi (tikai komandām) Hosta inventāra makrosi Vienuma makrosi (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) Starpniekservera makrosi Trigera makrosi: {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} Lietotāja definēti makrosi	Skatiet arī: Makrosu izmantošana ziņojumos

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Problēmas atjaunināj ums	Darbības makrosi Datuma/laika makrosi {ESC.HISTORY} Notikuma makrosi, Cēloņa/simptoma notikuma makrosi, Atkopšanas notikuma makrosi Notikuma atjaunināšanas makrosi: visi (izņemot {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}) Izteiksmes makross: {?EXPRESSION} Funkcijas makrosi Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Mērķa hosta makrosi (tikai komandām) Hosta inventāra makrosi Vienuma makrosi (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) Starpniekservera makrosi Trigera makrosi: {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} Lietotāja definēti makrosi	
Problēmas atkopšana	Atkopšanas notikuma makrosi {FUNCTION.RECOVERY.VALUE} {TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}	
Atklāšana	Darbības makrosi Datuma/laika makrosi Atklāšanas makrosi Notikuma makrosi: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Mērķa hosta makrosi (tikai komandām) Starpniekservera makrosi	
Automātiskā reģistrācija	Darbības makrosi Datuma/laika makrosi Notikuma makrosi: {EVENT.AGE}, {EVENT.DATE}, {EVENT.ID}, {EVENT.OBJECT}, {EVENT.SOURCE}, {EVENT.TIME}, {EVENT.TIMESTAMP} Hosta makrosi: {HOST.HOST}, {HOST.IP}, {HOST.PORT}, {HOST.METADATA} Mērķa hosta makrosi (tikai komandām) Starpniekservera makrosi	
Balstīts uz pakalpojumu	Darbības makrosi Datuma/laika makrosi {ESC.HISTORY} Notikuma makrosi: visi, izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA} Pakalpojuma makrosi Lietotāja definēti makrosi	
Pakalpojuma atkopšana	Notikuma makrosi (izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA}), Atkopšanas notikuma makrosi	

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Pakalpojuma atjaunināj ums	Darbības makrosi Datuma/laika makrosi {ESC.HISTORY} Notikuma makrosi (izņemot {EVENT.ACK.STATUS}, {EVENT.OPDATA}) Notikuma atjaunināšanas makrosi: {EVENT.UPDATE.DATE}, {EVENT.UPDATE.NSEVERITY}, {EVENT.UPDATE.SEVERITY}, {EVENT.UPDATE.STATUS}, {EVENT.UPDATE.TIME}, {EVENT.UPDATE.TIMESTAMP} Pakalpojuma makrosi Lietotāja definēti makrosi	
Iekšējie paziņojumi	Darbības makrosi Datuma/laika makrosi {ESC.HISTORY} Notikuma makrosi (izņemot {EVENT.NSEVERITY}, {EVENT.SEVERITY}), Atkopšanas notikuma makrosi Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Vienuma makrosi: {ITEM.DESCRIPTION}, {ITEM.DESCRIPTION.ORIG}, {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG}, {ITEM.NAME}, {ITEM.NAME.ORIG}, {ITEM.VALUETYPE} Starpniekservera makrosi	
Balstīts uz vienumu	Vienuma makrosi: {ITEM.STATE}, {ITEM.STATE.ERROR}	
Balstīts uz LLD kārtulu	Zema līmeņa atklāšanas paziņojuma makrosi	
Balstīts uz trigeri	Trigera makrosi: {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} Lietotāja definēti makrosi	
Brīdinājuma skripta parametri	Brīdinājuma makrosi Lietotāja definēti makrosi (tikai globālie)	
Savienotājs		
URL, Lietotājvārds, Parole, Bearer token, HTTP starpniekserveris, SSL sertifikāts, SSL atslēgas fails, SSL atslēgas parole lauks	Lietotāja definēti makrosi (tikai globālie)	
Timeout lauks	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Grafika nosaukums	Izteiksmes makross: {?EXPRESSION}	Šajā atrašanās vietā šajā makrosā tiek atbalstīta tikai viena **avg**, **last**, **max**, **min** funkcija ar sekundēm kā parametru (var izmantot laika sufiksus). Makrosu {HOST.HOST<1-9>} var izmantot kā hostu makrosā (skatiet arī indeksētos makrosus). Lietošanas piemērs: {?avg(/{HOST.HOST}/item.key,1h)} Neatbalstīta lietošana: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Hosts, hosta prototips		
Saskarnes IP/DNS	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME} Lietotāja definēti makrosi [1]	[1] IP adresei vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts
Saskarnes ports	Lietotāja definēti makrosi	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts
SNMP kopiena *SNMPv3 konteksta nosaukums, Drošības nosaukums, Autentifikācijas parole, Privātuma parole* *IPMI lietotājvārds, parole*	Lietotāja definēti makrosi	
Vienums, vienuma prototips, LLD kārtula		
Nosaukums	Lietotāja definēti makrosi	Netiek atbalstīts LLD kārtulu nosaukumos.
Apraksts	Lietotāja definēti makrosi	

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Atslēgas parametri	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	Atbalstītie {HOST.*} makrosi tiks atrisināti uz saskarni, kas ir atlasīta vienumam. Ja tie tiek izmantoti vienumos bez saskarnēm, tie tiks atrisināti uz hosta Zabbix aģenta, SNMP, JMX vai IPMI saskarni šādā prioritātes secībā vai uz 'UNKNOWN', ja hostam nav nevienas saskarnes.
Atjaunināšanas intervāls, Pielāgotie intervāli, Timeout (pieejams atbalstītajiem vienumu tipiem), Glabāt līdz (vēsturei/tendencēm) lauki	Lietotāja definēti makrosi	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Dzēst zaudētos resursus, Atspējot zaudētos resursus lauki	Lietotāja definēti makrosi	Šie lauki tiek atbalstīti tikai LLD kārtulām. Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Priekšapstrādes soļa parametri (ieskaitot pielāgotus skriptus), *pielāgotās kļūdu apstrādes parametri* (*Iestatīt vērtību uz* un *Iestatīt kļūdu uz* lauki)	Lietotāja definēti makrosi	
Filtra regulārās izteiksmes	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	Šie lauki tiek atbalstīti tikai LLD kārtulām.
Aizstāšanas filtra regulārās izteiksmes	Lietotāja definēti makrosi	Šie lauki tiek atbalstīti tikai LLD kārtulām.
Aizstāšanas darbības: *Atjaunināšanas intervāls*, *Vēstures glabāšanas periods*, *Tendenču glabāšanas periods* (vienumu prototipiem)	Lietotāja definēti makrosi	Šie lauki tiek atbalstīti tikai LLD kārtulām. Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Vienumu tipi		
Pārlūka vienums *parametru nosaukumi un vērtības*	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Vienuma makrosi: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definēti makrosi	

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Skripta vienums *parametru nosaukumi un vērtības*		
Aprēķinātais/agregētais vienums formula (*izteiksmes konstantes*, *funkciju parametri*, *vienuma atslēgas parametri*, (tikai agregētam vienumam) *filtra nosacījumi* (hostu grupas nosaukums un birkas nosaukums))	Lietotāja definēti makrosi	
Datubāzes monitors vienums *Lietotājvārds, Parole, SQL vaicājums* lauki	Lietotāja definēti makrosi	
HTTP aģenta vienums *URL, Vaicājuma lauki, Pieprasījuma pamatteksts, Galvenes (nosaukumi un vērtības), SSL sertifikāta fails, SSL atslēgas fails* lauki	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definēti makrosi	URL, kas satur slepeno makrosu, nedarbosies, jo tajos esošais makross tiks atrisināts kā ”******”
HTTP aģenta vienums *HTTP starpniekservera* lauks	Vienuma makrosi: {ITEM.ID}, {ITEM.KEY}, {ITEM.KEY.ORIG} Lietotāja definēti makrosi	
HTTP aģenta vienums *Obligātie statusa kodi, HTTP autentifikācijas lietotājvārds/parole, SSL atslēgas parole* lauki	Lietotāja definēti makrosi	
HTTP aģenta vienums *Atļautie hosti* lauks	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	
JMX aģenta vienums *Galapunkta* lauks	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.PORT} Lietotāja definēti makrosi	
Trapper vienums *Atļautie hosti* lauks		
SNMP aģenta vienums *SNMP OID* lauks	Lietotāja definēti makrosi	
SSH un Telnet vienums *skripts*	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	
SSH aģenta vienums *Lietotājvārds, Publiskās atslēgas fails, Privātās atslēgas fails, Parole* lauki	Lietotāja definēti makrosi	

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Telnet aģenta vienums *Lietotājvārds, Parole* lauki	Lietotāja definēti makrosi	
Karte		
URL, URL nosaukums	Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Hosta inventāra makrosi Kartes makrosi {TRIGGER.ID}	
Elementa etiķete	Izteiksmes makross: {?EXPRESSION} [1] Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} {HOSTGROUP.ID} Hosta inventāra makrosi Kartes makrosi Trigera makrosi: {TRIGGER.EVENTS.*}, {TRIGGER.ID}, {TRIGGER.PROBLEM.EVENTS.*}, {TRIGGERS.ACK}, {TRIGGERS.PROBLEM.ACK}, {TRIGGERS.PROBLEM.UNACK}, {TRIGGERS.UNACK}	[1] Šajā atrašanās vietā šajā makrosā tiek atbalstīta tikai viena **avg**, **last**, **max**, **min** funkcija ar sekundēm kā parametru (var izmantot laika sufiksus). Makrosu {HOST.HOST<1-9>} var izmantot kā hostu makrosā. Lietošanas piemērs: {?avg(/{HOST.HOST}/item.key,1h)} Neatbalstīta lietošana: {?last(/host/item1)/last(/host/item2)} {?last(/host/item1)*10} {?count(/host/item1,5m)}
Saites etiķete	Izteiksmes makross: {?EXPRESSION} [1]	
Formas etiķete	Izteiksmes makross: {?EXPRESSION} [1]	
Teksta lauks kartes formās	{MAP.NAME}	
Tīkla atkļūšana		
Atjaunināšanas intervāla lauks	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
SNMP kopienas, SNMP OID lauki *SNMPv3 konteksta nosaukuma, Drošības nosaukuma, Autentifikācijas paroles, Privātuma paroles* lauki	Lietotāja definēti makrosi (tikai globālie)	
Starpniekserveris		

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Adrese aktīvajiem aģentiem > *Ports* (ja starpniekserveris pieder grupai) Pasīvā starpniekservera režīmam: saskarnes *Adrese* un *Ports* lauki *Timeouts for item types* lauki aizstāšanas režīmā	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Starpniekserveru grupa		
Pārslēgšanās periods, Minimālais starpniekserveru skaits lauki	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Skripti		
Komandas, apstiprinājuma teksts (Tips: skripts, manuāla **hosta darbība**)	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID} [1], {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi [2] {MANUALINPUT} Lietotājvārda makrosi Lietotāja definēti makrosi	[1] Tikai apstiprinājuma tekstam [2] Atbalstīts Zabbix serverim un Zabbix starpniekserverim
Komandas (Tips: skripts, manuāla **notikuma darbība**)	Datuma/laika makrosi Notikuma makrosi, Cēloņa/simptoma notikuma makrosi, Atkopšanas notikuma makrosi (ja atkopšana ir notikusi) Notikuma atjauninašanas makrosi: {EVENT.UPDATE.HISTORY}, {EVENT.UPDATE.STATUS} {FUNCTION.VALUE} Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Vienuma makrosi (izņemot {ITEM.STATE}, {ITEM.STATE.ERROR}) {MANUALINPUT} Starpniekservera makrosi Trigera makrosi: {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} Lietotājvārda makrosi Lietotāja definēti makrosi	
Apstiprinājuma teksts (Tips: skripts, manuāla **notikuma darbība**)	Notikuma makrosi: {EVENT.ID}, {EVENT.NAME}, {EVENT.NSEVERITY}, {EVENT.SEVERITY}, {EVENT.STATUS}, {EVENT.VALUE} Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi {MANUALINPUT} Lietotājvārda makrosi Lietotāja definēti makrosi	

Atrašanās vieta	Atbalstītie makrosi	Komentāri
URL, apstiprinājuma teksts (Tips: URL, manuāla **hosta/notikuma darbība**)		
Birkas nosaukums un vērtība	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Trigera makrosi: {TRIGGER.ID} (tikai trigera birkas vērtībai) Lietotāja definēti makrosi	Birkas nosaukuma un vērtības makrosi tiek atrisināti tikai notikuma ģenerēšanas procesa laikā.
Trigeris		
Nosaukums	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Pozicionālie makrosi/atsauces: $1...$9 Lietotāja definēti makrosi	
Notikuma nosaukums	Datuma/laika makrosi: {TIME}, {TIMESTAMP} Izteiksmes makross: {?EXPRESSION} Funkcijas makrosi Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Trigera makrosi: {TRIGGER.EXPRESSION.EXPLAIN}	
Darbības dati	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Lietotāja definēti makrosi	
Izteiksme	{TRIGGER.VALUE} Lietotāja definēti makrosi [1]	[1] Tikai konstantēs un funkciju parametros; slepenie makrosi netiek atbalstīti
Birka atbilstībai, Izvēlnes ieraksta nosaukums, Izvēlnes ieraksta URL lauki	Lietotāja definēti makrosi	URL, kas satur slepeno makrosu, nedarbosies, jo tajos esošais makross tiks atrisināts kā ”******”.
Apraksts	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} Lietotāja definēti makrosi	
URL	{EVENT.ID} Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Vienuma makrosi: {ITEM.LASTVALUE*}, {ITEM.LOG.*}, {ITEM.VALUE*} {TRIGGER.ID}	
Lietotāja parametru komandas	Pozicionālie makrosi/atsauces: $1...$9	
Tīmekļa scenārijs		

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Nosaukums, Aģents, HTTP starpniekserveris lauki	Lietotāja definēti makrosi	
Atjaunināšanas intervāla lauks	Lietotāja definēti makrosi	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Mainīgo, Galveņu, SSL sertifikāta, SSL atslēgas faila lauki	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	
Lietotāja, Paroles, SSL atslēgas paroles lauki	Lietotāja definēti makrosi	
Tīmekļa scenārija solis *Nosaukuma, Mainīgo (tikai vērtības)* lauki	Lietotāja definēti makrosi	
Tīmekļa scenārija solis *Timeout, Obligātie statusa kodi* lauki	Lietotāja definēti makrosi	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Tīmekļa scenārija solis *URL, Galvenes (nosaukumi un vērtības), Obligātā virkne* lauki	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	
Tīmekļa scenārija solis *Post* lauks	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.IP}, {HOST.NAME}, {HOST.PORT}	
Webhook medija tips		
Parametru vērtības	Brīdinājuma makrosi Visi makrosi, kas tiek atbalstīti uz trigeri balstītos problēmu paziņojumos (skatīt iepriekš)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
JavaScript skripts, *JavaScript* skripta parametra nosaukums	Lietotāja definēti makrosi (tikai globālie)	
Izvēlnes ieraksta nosaukums, Izvēlnes ieraksta URL	{EVENT.TAGS.<tag name>}	
Logdaļas		
Mērinstrumenta logdaļa *Apraksta* lauks	Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Vienuma makrosi (izņemot {ITEM.STATE.ERROR}) Lietotāja definēti makrosi (tikai globālie)	
Honeycomb logdaļa *Primārās/sekundārās etiķetes* lauks		
Vienuma vērtības logdaļa *Apraksta* lauks		

Atrašanās vieta	Atbalstītie makrosi	Komentāri
Top hosts logdaļa *Teksta datu* kolonna	Hosta makrosi: {HOST.CONN}, {HOST.DESCRIPTION}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Hosta inventāra makrosi Lietotāja definēti makrosi (tikai globālie)	
URL (dinamiskā) logdaļa *URL* lauks	Hosta makrosi: {HOST.CONN}, {HOST.DNS}, {HOST.HOST}, {HOST.ID}, {HOST.IP}, {HOST.NAME}, {HOST.PORT} Lietotāja definēti makrosi	
Citas atrašanās vietas		
Darba laika lauks	Lietotāja definēti makrosi (tikai globālie)	Vienam makrosam ir jāaizpilda viss lauks; vairāki makrosi/maisījums ar tekstu netiek atbalstīts.
Timeouts for item types lauks (globālais iestatījums)		
Lietotāja medijs *Kad aktīvs* lauks	Lietotāja definēti makrosi (tikai globālie)	
E-pasta medija tips *Lietotājvārda, Paroles* lauki	Lietotāja definēti makrosi (tikai globālie)	Ieteicams izmantot slepenos makrosus.
Plānotie pārskati *Tēmas, Ziņojuma* lauki	Datuma/laika makrosi: {TIME}, {TIMESTAMP}	

Makrosu detalizēta informācija

Šajā sarakstā ir ietverta detalizēta informācija par katru iebūvēto makrosu, kas sagrupēti pēc lietojuma jomas:

- Darbību makrosi
- Brīdinājumu makrosi
- Datuma/laika makrosi
- Atklāšanas makrosi
- Notikumu makrosi
  - Notikumu atjaunināšanas makrosi
  - Cēloņa/simptoma notikumu makrosi
  - Atkopšanas notikumu makrosi
- Eskalācijas makrosi
- Funkciju makrosi
- Hostu makrosi
  - Mērķa hosta makrosi
- Hostu grupu makrosi
- Hostu inventāra makrosi
- Vienumu makrosi
- Zema līmeņa atklāšanas paziņojumu makrosi
- Karšu makrosi
- Starpniekservera makrosi
- Skriptu makrosi
- Pakalpojumu makrosi
- Trigeru makrosi
- Lietotājvārdu makrosi

Darbību makrosi

{ACTION.ID}

Iedarbinātās darbības skaitliskais ID.

{ACTION.NAME}

Iedarbinātās darbības nosaukums.

Brīdinājumu makrosi

{ALERT.MESSAGE}

Default message vērtība no darbības konfigurācijas.

{ALERT.SENDTO}

Send to vērtība no lietotāja medija konfigurācijas.

{ALERT.SUBJECT}

Default subject vērtība no darbības konfigurācijas.

Datuma un laika makrosi

{DATE}

Pašreizējais datums `yyyy.mm.dd` formātā.

{TIME}

Pašreizējais laiks `hh:mm:ss` formātā.

{TIMESTAMP}

Pašreizējais laiks UNIX laika zīmoga formātā.

Atklāšanas makrosi

{DISCOVERY.DEVICE.IPADDRESS}

Atklātās ierīces IP adrese.
 Vienmēr pieejams, nav atkarīgs no tā, vai hosts ir pievienots.

{DISCOVERY.DEVICE.DNS}

Atklātās ierīces DNS nosaukums.
 Vienmēr pieejams, nav atkarīgs no tā, vai hosts ir pievienots.

{DISCOVERY.DEVICE.STATUS}

Atklātās ierīces statuss (UP/DOWN).

{DISCOVERY.DEVICE.UPTIME}

Laiks kopš pēdējās atklāšanas statusa maiņas konkrētai ierīcei.
 Piemēram: 1h 29m 01s.
 Ierīcēm ar statusu DOWN tas ir to dīkstāves periods.

{DISCOVERY.RULE.NAME}

Atklāšanas noteikuma nosaukums, kas atklāja ierīces/pakalpojuma esamību vai neesamību.

{DISCOVERY.SERVICE.NAME}

Atklātā servisa nosaukums.
 Piemēram: HTTP.

{DISCOVERY.SERVICE.PORT}

Atklātā servisa ports.
 Piemēram: 80.

{DISCOVERY.SERVICE.STATUS}

Atklātā servisa statuss (UP/DOWN).

{DISCOVERY.SERVICE.UPTIME}

Laiks kopš pēdējās atklāšanas statusa maiņas konkrētam servisam.
 Piemēram: 1h 29m 01s.
 Servisiem ar statusu DOWN tas ir to dīkstāves periods.

Eskalācijas makrosi

{ESC.HISTORY}

Iepriekš nosūtīto paziņojumu žurnāls, to eskalācijas solis un statuss (*nosūtīts*, *procesā* vai *neizdevās*).

Notikumu makrosi

{EVENT.ACK.STATUS}

Notikuma apstiprināšanas statuss (Jā/Nē).

{EVENT.AGE}

Notikuma, kas izraisīja darbību, vecums ar precizitāti līdz sekundei.
 Noderīgi eskalētos ziņojumos. Piemēram: 9m 13s.

{EVENT.DATE}

Datums, kad notika notikums, kas izraisīja darbību, `yyyy.mm.dd` formātā.
 Piemēram: 2025.04.14.

{EVENT.DURATION}

Notikuma ilgums (laika starpība starp problēmas un atkopšanas notikumiem) ar precizitāti līdz sekundei.
 Noder problēmas atkopšanas ziņojumos.

{EVENT.ID}

Notikuma skaitliskais ID, kas izraisīja darbību.

{EVENT.NAME}

Notikuma nosaukums, kas izraisīja darbību.

{EVENT.NSEVERITY}

Notikuma smaguma pakāpes skaitliskā vērtība.
 Iespējamās vērtības: *0* - Neklasificēts, *1* - Informācija, *2* - Brīdinājums, *3* - Vidējs, *4* - Augsts, *5* - Katastrofa.

{EVENT.OBJECT}

Notikuma objekta skaitliskā vērtība.
 Iespējamās vērtības: *0* - Trigeris, *1* - Atklātais hosts, *2* - Atklātais pakalpojums, *3* - Automātiskā reģistrācija, *4* - Vienums, *5* - Zema līmeņa atklāšanas kārtula.

{EVENT.OPDATA}

Problēmas pamatā esošā trigera darbības dati.

{EVENT.SEVERITY}

Notikuma smaguma pakāpes nosaukums.

{EVENT.SOURCE}

Notikuma avota skaitliskā vērtība.
 Iespējamās vērtības: *0* - trigeris, *1* - atklāšana, *2* - automātiskā reģistrācija, *3* - iekšējs, *4* - pakalpojums.

{EVENT.STATUS}

Darbības izraisījušā notikuma verbālā vērtība.

{EVENT.TAGS}

Ar komatiem atdalīts notikuma tagu saraksts.
Ja tagu nav, tiek izvērsts par tukšu virkni.

{EVENT.TAGSJSON}

JSON masīvs, kas satur notikuma birku objektus.
Ja birkas nepastāv, tiek izvērsts par tukšu masīvu.

{EVENT.TAGS.<tag name>}

Notikuma birkas vērtība, uz kuru atsaucas birkas nosaukums.
 Birkas nosaukumam, kas satur ne burtciparu rakstzīmes (tostarp neangļu daudzbaitu UTF rakstzīmes), jābūt ietvertam dubultpēdiņās. Pēdiņas un atpakaļvērstās slīpsvītras citētā birkas nosaukumā jāatveido ar atsoļa rakstzīmi — atpakaļvērsto slīpsvītru.

{EVENT.TIME}

Notikuma laiks, kas izraisīja darbību, `hh:mm:ss` formātā.
 Piemēram: 12:57:53.

{EVENT.TIMESTAMP}

Darbību izraisījušā notikuma UNIX laikspiedols.

{EVENT.VALUE}

Darbību izraisījušā notikuma skaitliskā vērtība.
 Iespējamās vērtības: *1* - problēma, *0* - atkopšanās.

Cēloņa/simptoma notikuma makrosi

{EVENT.CAUSE.ACK.STATUS}

Cēloņa notikuma apstiprināšanas statuss (Jā/Nē).
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.AGE}

Cēloņa notikuma vecums ar precizitāti līdz sekundei.
 Noderīgs eskalētos ziņojumos.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.DATE}

Cēloņa notikuma datums `yyyy.mm.dd` formātā.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.DURATION}

Cēloņa notikuma ilgums (laika starpība starp problēmas un atkopšanas notikumiem) ar precizitāti līdz sekundei.
 Noder problēmas atkopšanas ziņojumos.
 Tiek izmantots simptoma notikuma kontekstā.

{EVENT.CAUSE.ID}

Cēloņa notikuma skaitliskais ID.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.NAME}

Cēloņa notikuma nosaukums.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.NSEVERITY}

Cēloņa notikuma svarīguma skaitliskā vērtība.
 Iespējamās vērtības: *0* - Nav klasificēts, *1* - Informācija, *2* - Brīdinājums, *3* - Vidējs, *4* - Augsts, *5* - Katastrofa.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.OBJECT}

Cēloņa notikuma objekta skaitliskā vērtība.
 Iespējamās vērtības: *0* - Trigeris, *1* - Atklātais hosts, *2* - Atklātais pakalpojums, *3* - Automātiskā reģistrācija, *4* - Vienums, *5* - Zema līmeņa atklāšanas kārtula.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.OPDATA}

Cēloņa problēmas pamatā esošā trigera darbības dati.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.SEVERITY}

Cēloņa notikuma svarīguma līmeņa nosaukums.
 Iespējamās vērtības: *Neklasificēts*, *Informācija*, *Brīdinājums*, *Vidējs*, *Augsts*, *Katastrofa*.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.SOURCE}

Cēloņa notikuma avota skaitliskā vērtība.
 Iespējamās vērtības: *0* - Trigeris, *1* - Atklāšana, *2* - Automātiskā reģistrācija, *3* - Iekšējs.
 Tiek izmantots simptomu notikuma kontekstā.

{EVENT.CAUSE.STATUS}

Cēloņa notikuma verbālā vērtība.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.TAGS}

Ar komatiem atdalīts cēloņa notikuma tagu saraksts.
Ja tagi nepastāv, tiek izvērsts par tukšu virkni.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.TAGSJSON}

JSON masīvs, kas satur cēloņa notikuma birku objektus.
Ja birkas nepastāv, tiek izvērsts par tukšu masīvu.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.TAGS.<tag name>}

Cēloņa notikuma birkas vērtība, uz kuru atsaucas birkas nosaukums.
 Birkas nosaukumam, kas satur nealfanumēriskas rakstzīmes (tostarp neangļu daudzbaitu UTF rakstzīmes), jābūt ietvertam dubultpēdiņās. Pēdiņas un atpakaļvērstās slīpsvītras citētā birkas nosaukumā jāekranē ar atpakaļvērsto slīpsvītru.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.TIME}

Cēloņa notikuma laiks `hh:mm:ss` formātā.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.TIMESTAMP}

Cēloņa notikuma UNIX laikspiedols.
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.UPDATE.HISTORY}

Cēloņa problēmas atjauninājumu žurnāls (apstiprinājumi u.c.).
 Izmanto simptomu notikuma kontekstā.

{EVENT.CAUSE.VALUE}

Cēloņa notikuma skaitliskā vērtība.
 Iespējamās vērtības: *1* - Problēma, *0* - Atkopjas.
 Izmanto simptomu notikuma kontekstā.

{EVENT.SYMPTOMS}

Simptomu notikumu saraksts. Ietver šādu informāciju: hosta nosaukums, notikuma nosaukums, nopietnība, vecums, pakalpojumu birkas un vērtības.
 Šī makrokomanda tiek izmantota cēloņa notikuma kontekstā un atgriež informāciju par simptomu notikumiem.

Atkopšanas notikuma makrosi

{EVENT.RECOVERY.DATE}

Atkopšanas notikuma datums `yyyy.mm.dd` formātā.

{EVENT.RECOVERY.ID}

Atkopšanas notikuma skaitliskais ID.

{EVENT.RECOVERY.NAME}

Atkopšanas notikuma nosaukums.

{EVENT.RECOVERY.STATUS}

Atkopšanas notikuma verbālā vērtība.

{EVENT.RECOVERY.TAGS}

Ar komatiem atdalīts atkopšanas notikuma tagu saraksts. Tiek izvērsts par tukšu virkni, ja tagi nepastāv.

{EVENT.RECOVERY.TAGSJSON}

JSON masīvs, kas satur atkopšanas notikuma birku objektus. Tiek izvērsts par tukšu masīvu, ja birkas nepastāv.

{EVENT.RECOVERY.TIME}

Atkopšanas notikuma laiks `hh:mm:ss` formātā.

{EVENT.RECOVERY.TIMESTAMP}

Atkopšanas notikuma UNIX laikspiedols.

{EVENT.RECOVERY.VALUE}

Atkopšanas notikuma skaitliskā vērtība.

Notikuma atjaunināšanas makrosi

{EVENT.UPDATE.ACTION}

Cilvēkam lasāms darbības(-u) nosaukums, kas veiktas problēmas atjaunināšanas laikā.
 Tiek atrisināts uz šādām vērtībām: *apstiprināts*, *neapstiprināts*, *komentēts*, *mainīta nopietnība no (sākotnējā nopietnība) uz (atjauninātā nopietnība)* un *aizvērts* (atkarībā no tā, cik darbību tiek veiktas vienā atjaunināšanā).

{EVENT.UPDATE.ACTIONJSON}

JSON masīvs, kas satur detalizētu informāciju par darbību(-ām), kas veiktas problēmas atjaunināšanas laikā.
 Iespējamās JSON īpašību vērtības:
- true (`acknowledge`, `unacknowledge`, `close`, `unsuppress`, `cause` un `symptom` īpašībām);
- <ziņojuma virkne> (`message` īpašībai);
- laikspiedols (`suppress_until` un `timestamp` īpašībām) vai 0 (`suppress_until` īpašībai, ja nomākšana ir bez termiņa);
- 0, 1, 2, 3, 4, 5 (`old` un `new` nopietnības īpašībām).

 Piemēram: `{"acknowledge":true,"message":"Monthly maintenance.","severity":{"old":2,"new":1},"suppress_until":1730851`

{EVENT.UPDATE.DATE}

Notikuma atjaunināšanas datums (apstiprinājums u.c.) `yyyy.mm.dd` formātā.

{EVENT.UPDATE.HISTORY}

Problēmas atjauninājumu žurnāls (apstiprinājumi u.c.).

{EVENT.UPDATE.MESSAGE}

Problēmas atjaunināšanas ziņojums.

{EVENT.UPDATE.NSEVERITY}

Jaunā notikuma nopietnības līmeņa skaitliskā vērtība, kas iestatīta problēmas atjaunināšanas darbības laikā.

{EVENT.UPDATE.SEVERITY}

Jaunā notikuma svarīguma pakāpes nosaukums, kas iestatīts problēmas atjaunināšanas darbības laikā.

{EVENT.UPDATE.STATUS}

Problēmas atjaunināšanas statusa skaitliskā vērtība.
 Iespējamās vērtības: *0* - webhook tika izsaukts problēmas/atkopšanas notikuma dēļ, *1* - atjaunināšanas operācija.

{EVENT.UPDATE.TIME}

Notikuma atjaunināšanas laiks (`acknowledgment` u.c.) `hh:mm:ss` formātā.

{EVENT.UPDATE.TIMESTAMP}

Notikuma atjaunināšanas UNIX laikspiedols (apstiprinājums u.c.).

Funkciju makrosi

{FUNCTION.VALUE}

N-tās uz vienumu balstītās funkcijas vērtība trigera izteiksmē notikuma brīdī.
 Tiek skaitītas tikai funkcijas, kurām kā pirmais parametrs ir `/host/key`.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {FUNCTION.VALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. funkciju trigera izteiksmē. Skatiet indeksētos makrosus.

{FUNCTION.RECOVERY.VALUE}

N-tās uz vienumu balstītās funkcijas vērtība atkopšanas izteiksmē notikuma brīdī.*
 Tiek skaitītas tikai funkcijas, kurām kā pirmais parametrs ir `/hosts/atslēga`.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {FUNCTION.RECOVERY.VALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. funkciju trigera izteiksmē. Skatiet indeksētos makro.

Hosta makrosi

{HOST.CONN}

Hosta IP adrese vai DNS nosaukums atkarībā no hosta iestatījumiem.
 Globālajos skriptos, saskarņu IP/DNS laukos un tīmekļa scenārijos makross tiks atrisināts uz galveno aģenta saskarni. Ja aģenta saskarne nav definēta, tiks izmantota galvenā SNMP saskarne. Ja arī SNMP saskarne nav definēta, tiks izmantota galvenā JMX saskarne. Ja arī JMX saskarne nav definēta, tiks izmantota galvenā IPMI saskarne. Ja hostam nav nevienas saskarnes, makross tiek atrisināts kā 'UNKNOWN'.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.CONN<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētie makrosi.

{HOST.DESCRIPTION}

Hosta apraksts.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{HOST.DNS}

Hosta DNS nosaukums.
 Globālajos skriptos, saskarnes IP/DNS laukos un tīmekļa scenārijos makross tiks atrisināts uz galveno aģenta saskarni. Ja aģenta saskarne nav definēta, tiks izmantota galvenā SNMP saskarne. Ja arī SNMP saskarne nav definēta, tiks izmantota galvenā JMX saskarne. Ja arī JMX saskarne nav definēta, tiks izmantota galvenā IPMI saskarne. Ja hostam nav nevienas saskarnes, makross tiek atrisināts kā 'UNKNOWN'.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.DNS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{HOST.HOST}

Tehniskais hosta nosaukums.
 Makro funkcijas šim makro netiek atbalstītas, ja tas tiek izmantots kā vietturis vēstures funkcijas pirmajā parametrā, piemēram, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Šo makro var izmantot ar skaitlisku indeksu kā {HOST.HOST<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{HOST.ID}

Hosta ID.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.ID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{HOST.IP}

Hosta IP adrese.
 Globālajos skriptos, saskarņu IP/DNS laukos un tīmekļa scenārijos makross tiks atrisināts uz galveno aģenta saskarni. Ja aģenta saskarne nav definēta, tiks izmantota galvenā SNMP saskarne. Ja arī SNMP saskarne nav definēta, tiks izmantota galvenā JMX saskarne. Ja arī JMX saskarne nav definēta, tiks izmantota galvenā IPMI saskarne. Ja hostam nav nevienas saskarnes, makross tiek atrisināts kā 'UNKNOWN'.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.IP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{HOST.METADATA}

Hosta metadati.

 Izmanto tikai aktīvā aģenta automātiskajai reģistrācijai.

{HOST.NAME}

Redzamais hosta nosaukums.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{HOST.PORT}

Hosta (aģenta) ports.
 Globālajos skriptos, saskarnes IP/DNS laukos un tīmekļa scenārijos makross tiks atrisināts uz galveno aģenta saskarni. Ja aģenta saskarne nav definēta, tiks izmantota galvenā SNMP saskarne. Ja arī SNMP saskarne nav definēta, tiks izmantota galvenā JMX saskarne. Ja arī JMX saskarne nav definēta, tiks izmantota galvenā IPMI saskarne. Ja hostam nav nevienas saskarnes, makross tiek atrisināts kā 'UNKNOWN'.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {HOST.PORT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

Mērķa hosta makrosi

{HOST.TARGET.CONN}

Mērķa hosta IP adrese vai DNS nosaukums.

{HOST.TARGET.DNS}

Mērķa hosta DNS nosaukums.

{HOST.TARGET.HOST}

Mērķa hosta tehniskais nosaukums.

{HOST.TARGET.IP}

Mērķa hosta IP adrese.

{HOST.TARGET.NAME}

Mērķa hosta redzamais nosaukums.

Hostu grupu makrosi

{HOSTGROUP.ID}

Hostu grupas ID.

Hosta inventarizācijas makrosi

{INVENTORY.ALIAS}

Lauks *Alias* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.ALIAS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.ASSET.TAG}

Lauks *Asset tag* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.ASSET.TAG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.CHASSIS}

Lauks *Chassis* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CHASSIS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.CONTACT}

Lauks *Contact* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CONTACT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.CONTRACT.NUMBER}

Līguma numura lauks hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.CONTRACT.NUMBER<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.DEPLOYMENT.STATUS}

Hosta inventāra lauks *Izvietošanas statuss*.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.DEPLOYMENT.STATUS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HARDWARE}

Lauks *Aparatūra* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HARDWARE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HARDWARE.FULL}

Lauks *Aparatūra (Pilna informācija)* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HARDWARE.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HOST.NETMASK}

Lauks *Hosta apakštīkla maska* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.NETMASK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HOST.NETWORKS}

Lauks *Host networks* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.NETWORKS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HOST.ROUTER}

Lauks *Host router* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HOST.ROUTER<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HW.ARCH}

Lauks *Aparatūras arhitektūra* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.ARCH<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HW.DATE.DECOMM}

Lauks *Aparatūras ekspluatācijas pārtraukšanas datums* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.DECOMM<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.HW.DATE.EXPIRY}

Lauks *Datums, kad beidzas aparatūras uzturēšana* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.EXPIRY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HW.DATE.INSTALL}

Lauks *Datums, kad aparatūra instalēta* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.INSTALL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.HW.DATE.PURCHASE}

Lauks *Aparatūras iegādes datums* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.HW.DATE.PURCHASE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.INSTALLER.NAME}

Lauks *Instalētāja nosaukums* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.INSTALLER.NAME<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.LOCATION}

Location lauks hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.LOCATION.LAT}

Lauks *Atrašanās vietas platums* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION.LAT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.LOCATION.LON}

Lauks *Atrašanās vietas garums* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.LOCATION.LON<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.MACADDRESS.A}

Lauks *MAC address A* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MACADDRESS.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.MACADDRESS.B}

Lauks *MAC address B* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MACADDRESS.B<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.MODEL}

Lauks *Model* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.MODEL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.NAME}

Lauks *Name* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.NOTES}

Piezīmes lauks hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.OOB.IP}

Lauks *OOB IP address* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.IP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.OOB.NETMASK}

Lauks *OOB apakštīkla maska* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.NETMASK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.OOB.ROUTER}

Lauks *OOB maršrutētājs* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OOB.ROUTER<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.OS}

OS lauks hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.OS.FULL}

Lauks *OS (pilna informācija)* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.OS.SHORT}

Lauks *OS (Short)* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.OS.SHORT<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.CELL}

Lauks *Primary POC cell* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.CELL<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.EMAIL}

Lauks *Primary POC email* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.EMAIL<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.NAME}

Lauks *Primārās kontaktpersonas vārds* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.NOTES}

Lauks *Primārās kontaktpersonas piezīmes* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.PHONE.A}

Lauks *Primary POC phone A* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.PHONE.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.PHONE.B}

Lauks *Primary POC phone B* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.PHONE.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.PRIMARY.SCREEN}

Lauks *Primārās kontaktpersonas ekrāna nosaukums* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.PRIMARY.SCREEN<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.CELL}

Lauks *Secondary POC cell* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.CELL<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.EMAIL}

Lauks *Sekundārā POC e-pasts* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.EMAIL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.NAME}

Lauks *Sekundārās kontaktpersonas vārds* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.NOTES}

Lauks *Secondary POC notes* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piem., {INVENTORY.POC.SECONDARY.NOTES<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.PHONE.A}

Lauks *Secondary POC phone A* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.PHONE.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.PHONE.B}

Lauks *Secondary POC phone B* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.PHONE.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.POC.SECONDARY.SCREEN}

Lauks *Sekundārās POC ekrāna nosaukums* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.POC.SECONDARY.SCREEN<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.SERIALNO.A}

Lauks *Sērijas numurs A* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SERIALNO.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SERIALNO.B}

Lauks *Sērijas numurs B* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SERIALNO.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.ADDRESS.A}

Lauks *Vietnes adrese A* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.ADDRESS.B}

Lauks *Vietnes adrese B* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.ADDRESS.C}

Lauks *Vietnes adrese C* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ADDRESS.C<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.CITY}

Lauks *Vietnes pilsēta* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.CITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.COUNTRY}

Lauks *Vietnes valsts* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.COUNTRY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.NOTES}

Lauks *Piezīmes par vietu* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.NOTES<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.RACK}

Lauks *Vietnes skapja atrašanās vieta* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.RACK<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.STATE}

Lauks *Vietas štats/province* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SITE.ZIP}

Lauks *Vietas ZIP/pasta indekss* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SITE.ZIP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE}

Lauks *Programmatūra* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.APP.A}

Lauks *Programmatūras lietojumprogramma A* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.A<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.APP.B}

Lauks *Programmatūras lietojumprogramma B* hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.B<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.APP.C}

Lauks *Programmatūras lietojumprogramma C* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.C<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.APP.D}

Lauks *Programmatūras lietojumprogramma D* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.D<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.APP.E}

Lauks *Programmatūras lietojumprogramma E* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.APP.E<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.SOFTWARE.FULL}

Lauks *Programmatūra (pilna informācija)* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.SOFTWARE.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.TAG}

Lauks *Tag* hosta inventarizācijā.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TAG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makro.

{INVENTORY.TYPE}

Lauks *Type* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TYPE<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.TYPE.FULL}

Lauks *Type (Full details)* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.TYPE.FULL<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.URL.A}

URL A lauks hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.A<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.URL.B}

Lauks *URL B* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.B<1-9>}, lai trigera izteiksmē norādītu uz pirmo, otro, trešo utt. hostu. Skatiet indeksētos makrosus.

{INVENTORY.URL.C}

Lauks *URL C* hosta inventarizācijā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.URL.C<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{INVENTORY.VENDOR}

Vendor lauks hosta inventārā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {INVENTORY.VENDOR<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

Vienumu makrosi

{ITEM.DESCRIPTION}

N-tā vienuma apraksts trigera izteiksmē, kas izraisīja paziņojumu.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {ITEM.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.DESCRIPTION.ORIG}

N-tā vienuma sākotnējais apraksts (ar neatrisinātām makro vērtībām) trigera izteiksmē, kas izraisīja paziņojumu.

 Šo makro var izmantot ar skaitlisku indeksu kā {ITEM.DESCRIPTION.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētās makro.

{ITEM.ID}

Skaitliskais N-tā vienuma ID trigera izteiksmē, kas izraisīja paziņojumu.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {ITEM.ID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētie makrosi.

{ITEM.KEY}

N-tā vienuma atslēga trigera izteiksmē, kas izraisīja paziņojumu.
 Šim makrosam netiek atbalstītas makrosu funkcijas, ja tas tiek izmantots kā vietturis vēstures funkcijas pirmajā parametrā, piemēram, `last(/{HOST.HOST}/{ITEM.KEY})`.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {ITEM.KEY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.KEY.ORIG}

N-tā vienuma sākotnējā atslēga (ar neatrisinātām makrosiem) trigera izteiksmē, kas izraisīja paziņojumu.

 Šo makrosu var izmantot ar skaitlisku indeksu kā {ITEM.KEY.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. hostu trigera izteiksmē. Skatiet indeksētās makrosus.

{ITEM.LASTVALUE}

N-ā vienuma jaunākā vērtība trigera izteiksmē, kas izraisīja paziņojumu.
 Lietotāja saskarnē tā tiks atrisināta kā *UNKNOWN*, ja jaunākā vēstures vērtība ir savākta agrāk nekā pirms *Max history display period* laika (iestatīts izvēlnes sadaļā Administration → General).
 Ja to izmanto problēmas nosaukumā, makross netiks atrisināts uz jaunāko vienuma vērtību, skatot problēmu notikumus; tā vietā tas saglabās vienuma vērtību no brīža, kad problēma notika.
Ja to izmanto paziņojumos, dažos gadījumos makross var netikt atrisināts uz jaunāko vienuma vērtību brīdī, kad nostrādāja trigeris. Piemēram, ja vienums ātri saņem divas vērtības — ”A” un ”B” — un trigeris nostrādā pie ”A”, paziņojumos kā jaunākā vērtība nelielas apstrādes aizkaves dēļ var tikt parādīta ”B” — jaunākā vienuma vērtība mainījās starp brīdi, kad nostrādāja trigeris, un brīdi, kad tika izveidots paziņojums. Lai no tā izvairītos, varat izmantot makrosu {ITEM.VALUE}, kas tiek atrisināts uz vērtību brīdī, kad nostrādā trigeris, nodrošinot, ka paziņojumā tiek izmantota pareizā vērtība.
 Tas ir `last(/{HOST.HOST}/{ITEM.KEY})` aizstājvārds.
 Atrisinātā vērtība teksta/log vienumiem lietotāja saskarnē tiek saīsināta līdz 20 rakstzīmēm šādās vietās:
- Operatīvie dati;
- Trigera apraksts;
- Trigera URL;
- Trigera URL etiķetes;
- Vienuma vērtības logrīka apraksts.
Lai iegūtu pilnu vērtību, varat izmantot makrosu funkcijas, jo serveris nevienu vērtību nesaīsina. Piemēram: `{{ITEM.LASTVALUE}.regsub("(.*)", \1)}`.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LASTVALUE.AGE}

Laiks, kas pagājis starp jaunākās vienuma vērtības iegūšanu un makrosa novērtēšanu.
 Tiek attēlots cilvēkam lasāmā formātā (piemēram, 1m 45s).

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LASTVALUE.DATE}

Datums, kad tika savākta jaunākā vienuma vērtība.
 Tiek attēlots `YYYYMMDD` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (Monitoring -> Problems sarakstā), notikumu nosaukumos un birku nosaukumos un vērtībās. Visos pārējos gadījumos tiek izmantota lietotāja laika josla.

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.DATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.LASTVALUE.TIME}

Laiks, kad tika savākta jaunākā vienuma vērtība.
 Tiek attēlots HHMMSS formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (Monitoring -> Problems sarakstā), notikumu nosaukumos un birku nosaukumos un vērtībās. Visos pārējos gadījumos tiek izmantota lietotāja laika josla.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LASTVALUE.TIMESTAMP}

UNIX laikspiedols, kad tika savākta jaunākā vienuma vērtība.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LASTVALUE.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.AGE}

Žurnāla notikuma vecums.
Ar precizitāti līdz sekundei.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.DATE}

Datums, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.DATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.EVENTID}

Notikuma ID notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.EVENTID<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.NSEVERITY}

Notikuma skaitliskā nopietnība notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.NSEVERITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.SEVERITY}

Notikuma vārdiskā nopietnība notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.SEVERITY<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.SOURCE}

Notikuma avots notikumu žurnālā.
 Tikai Windows notikumu žurnāla uzraudzībai.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.SOURCE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.TIME}

Laiks, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.LOG.TIMESTAMP}

UNIX laikspiedols, kad žurnāla ieraksts tika ierakstīts žurnālā.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.LOG.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.NAME}

Vienuma nosaukums ar visiem atrisinātajiem makrosiem.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.NAME.ORIG}

Sākotnējais vienuma nosaukums (ar neatrisinātām makro vērtībām).

 Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.NAME.ORIG<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētie makro.

{ITEM.STATE}

N-ā vienuma pēdējais stāvoklis trigera izteiksmē, kas izraisīja paziņojumu.
 Iespējamās vērtības: *Not supported*, *Normal*.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.STATE.ERROR}

Kļūdas ziņojums ar detalizētu informāciju par to, kāpēc vienums kļuva neatbalstīts.
 Ja vienums nonāk neatbalstītā stāvoklī un pēc tam nekavējoties atkal kļūst atbalstīts, kļūdas lauks var būt tukšs.

{ITEM.VALUE}

Tiek atrisināts kā viens no šiem:

- N-tā vienuma vēsturiskā (notikuma brīdī esošā) vērtība trigera izteiksmē, ja tiek izmantots trigera stāvokļa maiņas kontekstā, piemēram, attēlojot notikumus vai sūtot paziņojumus;
- N-tā vienuma jaunākā vērtība trigera izteiksmē, ja tiek izmantots ārpus trigera stāvokļa maiņas konteksta, piemēram, attēlojot trigeru sarakstu uznirstošajā atlases logā. Šajā gadījumā tas darbojas tāpat kā {ITEM.LASTVALUE}.

Pirmajā gadījumā tas tiks atrisināts kā *UNKNOWN*, ja vēstures vērtība jau ir dzēsta vai nekad nav tikusi saglabāta.
Otrajā gadījumā un tikai lietotāja saskarnē tas tiks atrisināts kā *UNKNOWN*, ja jaunākā vēstures vērtība ir ievākta agrāk nekā pirms *Max history display period* laika (iestatīts izvēlnes sadaļā Administration→General).
Teksta/žurnāla vienumu atrisinātā vērtība lietotāja saskarnē tiek saīsināta līdz 20 rakstzīmēm šādās vietās:
- Operacionālie dati;
- Trigera apraksts;
- Trigera URL;
- Trigera URL etiķetes;
- Vienuma vērtības logrīka apraksts.
Lai atrisinātu uz pilnu vērtību, varat izmantot makro funkcijas, jo serveris nevienu vērtību nesaīsina. Piemēram: `{{ITEM.VALUE}.regsub("(.*)", \1)}`.

Šo makro var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makro.

{ITEM.VALUE.AGE}

Laiks, kas pagājis starp vienuma vērtības iegūšanu un makrosa novērtēšanu.
 Tiek attēlots cilvēkam lasāmā formātā (piemēram, 1m 45s).

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.AGE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.VALUE.DATE}

Datums, kad tika ievākta vienuma vērtība.
 Tiek attēlots `YYYYMMDD` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (Monitoring -> Problems sarakstā), notikumu nosaukumos un birku nosaukumos un vērtībās. Visos pārējos gadījumos tiek izmantota lietotāja laika josla.

Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.DATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.VALUE.TIME}

Laiks, kad tika ievākta vienuma vērtība.
 Tiek attēlots `HHMMSS` formātā.
 Servera laika josla tiek izmantota trigeru nosaukumos (Monitoring -> Problems sarakstā), notikumu nosaukumos un birku nosaukumos un vērtībās. Visos pārējos gadījumos tiek izmantota lietotāja laika josla.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.TIME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētie makrosi.

{ITEM.VALUE.TIMESTAMP}

UNIX laikspiedols, kad tika ievākta vienuma vērtība.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUE.TIMESTAMP<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{ITEM.VALUETYPE}

N-tā vienuma vērtības tips trigera izteiksmē, kas izraisīja paziņojumu.
 Iespējamās vērtības: *0* - peldošā komata skaitlis, *1* - rakstzīme, *2* - žurnāls, *3* - bezparaksta vesels skaitlis, *4* - teksts.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {ITEM.VALUETYPE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

Zema līmeņa atklāšanas paziņojumu makrosi

{LLDRULE.DESCRIPTION}

Zema līmeņa atklāšanas kārtulas apraksts, kas izraisīja paziņojumu.

{LLDRULE.DESCRIPTION.ORIG}

Sākotnējais zema līmeņa atklāšanas kārtulas apraksts (ar neatrisinātām makro vērtībām), kas izraisīja paziņojumu.

{LLDRULE.ID}

Zema līmeņa atklāšanas kārtulas skaitliskais ID, kas izraisīja paziņojumu.

{LLDRULE.KEY}

Zema līmeņa atklāšanas kārtulas atslēga, kas izraisīja paziņojumu.

{LLDRULE.KEY.ORIG}

Zema līmeņa atklāšanas kārtulas sākotnējā atslēga (ar neatrisinātiem makrosiem), kas izraisīja paziņojumu.

{LLDRULE.NAME}

Zema līmeņa atklāšanas kārtulas nosaukums, kas izraisīja paziņojumu.

{LLDRULE.NAME.ORIG}

Zema līmeņa atklāšanas kārtulas sākotnējais nosaukums (ar neatrisinātām makro vērtībām), kas izraisīja paziņojumu.

{LLDRULE.STATE}

Zema līmeņa atklāšanas kārtulas jaunākais stāvoklis.
 Iespējamās vērtības: *Nav atbalstīts*, *Normāls*.

 Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {LLDRULE.STATE<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētos makrosus.

{LLDRULE.STATE.ERROR}

Kļūdas ziņojums ar detalizētu informāciju par to, kāpēc zema līmeņa atklāšanas kārtula kļuva neatbalstīta.
 Ja zema līmeņa atklāšanas kārtula nonāk neatbalstītā stāvoklī un pēc tam nekavējoties atkal kļūst atbalstīta, kļūdas lauks var būt tukšs.

Kartes makrosi

{MAP.ID}

Tīkla kartes ID.

{MAP.NAME}

Tīkla kartes nosaukums.

Starpniekservera makrosi

{PROXY.DESCRIPTION}

Starpniekservera apraksts.
 Tiek aizvietots ar vienu no šīm vērtībām:

- N-tā vienuma starpniekserveris trigera izteiksmē (uz trigeriem balstītos paziņojumos). Šeit varat izmantot indeksētās makros;
- starpniekserveris, kas izpildīja atklāšanu (atklāšanas paziņojumos). Šeit izmantojiet {PROXY.DESCRIPTION} bez indeksēšanas;
- starpniekserveris, kurā reģistrējās aktīvs aģents (automātiskās reģistrācijas paziņojumos). Šeit izmantojiet {PROXY.DESCRIPTION} bez indeksēšanas.

Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {PROXY.DESCRIPTION<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētās makros.

{PROXY.NAME}

Starpniekservera nosaukums.
 Tiek aizvietots ar vienu no šīm vērtībām:

- N-tā vienuma starpniekserveris trigera izteiksmē (uz trigeri balstītos paziņojumos). Šeit varat izmantot indeksētās makrosu vērtības;
- starpniekserveris, kas izpildīja atklāšanu (atklāšanas paziņojumos). Šeit izmantojiet {PROXY.NAME} bez indeksēšanas;
- starpniekserveris, kurā reģistrējās aktīvais aģents (automātiskās reģistrācijas paziņojumos). Šeit izmantojiet {PROXY.NAME} bez indeksēšanas.

Šo makrosu var izmantot ar skaitlisku indeksu, piemēram, {PROXY.NAME<1-9>}, lai norādītu uz pirmo, otro, trešo utt. vienumu trigera izteiksmē. Skatiet indeksētie makrosi.

Skripta makrosi

{MANUALINPUT}

Manuālās ievades vērtība, ko lietotājs norādījis skripta izpildes laikā.

Servisa makrosi

{SERVICE.DESCRIPTION}

Pakalpojuma apraksts ar atrisinātiem makrosiem.

{SERVICE.ID}

Pakalpojuma skaitliskais ID, kas izraisīja darbību.

{SERVICE.NAME}

Pakalpojuma nosaukums ar atrisinātiem makrosiem.

{SERVICE.ROOTCAUSE}

Trigeru problēmu notikumu saraksts, kas izraisīja pakalpojuma kļūmi, sakārtots pēc svarīguma un hosta nosaukuma.
 Ietver šādu informāciju: hosta nosaukums, notikuma nosaukums, svarīgums, vecums, pakalpojuma birkas un vērtības.

{SERVICE.TAGS}

Ar komatiem atdalīts pakalpojuma notikuma tagu saraksts.
 Pakalpojuma notikuma tagus var definēt pakalpojuma konfigurācijas tagu sadaļā. Tiek izvērsts par tukšu virkni, ja tagi nepastāv.

{SERVICE.TAGSJSON}

JSON masīvs, kas satur servisa notikuma birku objektus.
 Servisa notikuma birkas var definēt servisa konfigurācijas birku sadaļā. Tiek izvērsts par tukšu masīvu, ja birkas nepastāv.

{SERVICE.TAGS.<tag name>}

Pakalpojuma notikuma birkas vērtība, uz kuru atsaucas birkas nosaukums.
 Pakalpojuma notikuma birkas var definēt pakalpojuma konfigurācijas birku sadaļā.
 Birkas nosaukumam, kas satur nealfanumēriskas rakstzīmes (tostarp neangļu daudzbaitu UTF rakstzīmes), jābūt ietvertam dubultpēdiņās. Pēdiņas un atpakaļvērstās slīpsvītras pēdiņās ietvertā birkas nosaukumā jāatveido ar atsoļa rakstzīmi — atpakaļvērsto slīpsvītru.

Trigeru makrosi

{TRIGGER.DESCRIPTION}

Trigera apraksts.
 Ja paziņojuma tekstā tiek izmantots `{TRIGGER.DESCRIPTION}`, tiks izvērstas visas makros, kas tiek atbalstītas trigera aprakstā.

{TRIGGER.EVENTS.ACK}

Apstiprināto notikumu skaits kartes elementam kartēs vai trigerim, kas ģenerēja pašreizējo notikumu paziņojumos.

{TRIGGER.EVENTS.PROBLEM.ACK}

Apstiprināto problēmu notikumu skaits visiem trigeriem, neņemot vērā to stāvokli.

{TRIGGER.EVENTS.PROBLEM.UNACK}

Neapstiprināto problēmu notikumu skaits visiem trigeriem, neņemot vērā to stāvokli.

{TRIGGER.EVENTS.UNACK}

Neapstiprināto notikumu skaits kartes elementam kartēs vai trigerim, kas ģenerēja pašreizējo notikumu paziņojumos.

{TRIGGER.EXPRESSION}

Trigera izteiksme.

{TRIGGER.EXPRESSION.EXPLAIN}

Daļēji novērtēta trigera izteiksme.
 Uz vienumiem balstītās funkcijas tiek novērtētas un aizstātas ar rezultātiem notikuma ģenerēšanas brīdī, savukārt visas pārējās funkcijas tiek attēlotas tā, kā tās ir rakstītas izteiksmē. Var izmantot trigera izteiksmju atkļūdošanai.

{TRIGGER.EXPRESSION.RECOVERY}

Trigera atkopšanas izteiksme, ja trigera konfigurācijā *OK notikuma ģenerēšana* ir iestatīta uz "Atkopšanas izteiksme"; pretējā gadījumā tiek atgriezta tukša virkne.

{TRIGGER.EXPRESSION.RECOVERY.EXPLAIN}

Daļēji novērtēta trigera atkopšanas izteiksme.
 Uz vienumiem balstītās funkcijas tiek novērtētas un aizstātas ar rezultātiem notikuma ģenerēšanas brīdī, savukārt visas pārējās funkcijas tiek attēlotas tā, kā tās ir uzrakstītas izteiksmē. Var izmantot trigera atkopšanas izteiksmju atkļūdošanai.

{TRIGGER.HOSTGROUP.NAME}

Sakārtots (pēc SQL vaicājuma), ar komatu un atstarpi atdalīts hostu grupu saraksts, kurās ir definēts trigeris.

{TRIGGER.ID}

Skaitliskais trigeris ID, kas izraisīja šo darbību.

{TRIGGER.NAME}

Trigera nosaukums ar atrisinātiem makro.
 Ņemiet vērā, ka `{EVENT.NAME}` var izmantot darbībās, lai parādītu aktivizētā notikuma/problēmas nosaukumu ar atrisinātiem makro.

{TRIGGER.NAME.ORIG}

Sākotnējais trigera nosaukums (ar neatrisinātām makrokomandām).

{TRIGGER.NSEVERITY}

Trigera skaitliskā nopietnība.
 Iespējamās vērtības: *0* - Nav klasificēts, *1* - Informācija, *2* - Brīdinājums, *3* - Vidējs, *4* - Augsts, *5* - Katastrofa.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.ACK}

Apstiprināto problēmu notikumu skaits trigeriem problēmas stāvoklī.

{TRIGGER.PROBLEM.EVENTS.PROBLEM.UNACK}

Neapstiprināto problēmu notikumu skaits trigeriem problēmas stāvoklī.

{TRIGGER.SEVERITY}

Trigera smaguma pakāpes nosaukums.
 Var definēt sadaļā *Administration* > *General* > *Trigger displaying options*.

{TRIGGER.STATE}

Trigera izteiksmes jaunākais stāvoklis.
 Iespējamās vērtības: *Nezināms*, *Normāls*.

{TRIGGER.STATE.ERROR}

Kļūdas ziņojums ar detalizētu informāciju par to, kāpēc trigeris kļuva neatbalstīts.
 Ja trigeris nonāk neatbalstītā stāvoklī un pēc tam nekavējoties atkal kļūst atbalstīts, kļūdas lauks var būt tukšs.

{TRIGGER.STATUS}

Trigera vērtība darbības soļa izpildes laikā.
 Iespējamās vērtības: *Problem*, *OK*.

{TRIGGER.TEMPLATE.NAME}

Sakārtots (pēc SQL vaicājuma), ar komatu un atstarpi atdalīts veidņu saraksts, kurās ir definēts trigeris, vai *UNKNOWN*, ja trigeris ir definēts hostā.

{TRIGGER.URL}

Trigera URL.

{TRIGGER.URL.NAME}

Trigera URL etiķete.

{TRIGGER.VALUE}

Pašreizējā skaitliskā trigera vērtība.
 Iespējamās vērtības: *0* - trigeris ir OK stāvoklī, *1* - trigeris ir problēmas stāvoklī.

{TRIGGERS.ACK}

Apstiprināto trigeru skaits kartes elementam, neņemot vērā trigeru stāvokli.
 Trigeris tiek uzskatīts par apstiprinātu, ja visi tā problēmu notikumi ir apstiprināti.

{TRIGGERS.PROBLEM.ACK}

Apstiprināto problēmu trigeru skaits kartes elementam.
 Trigeris tiek uzskatīts par apstiprinātu, ja visi tā problēmu notikumi ir apstiprināti.

{TRIGGERS.PROBLEM.UNACK}

Neapstiprināto problēmu trigeru skaits kartes elementam.
 Trigeris tiek uzskatīts par neapstiprinātu, ja vismaz viens no tā problēmas notikumiem nav apstiprināts.

{TRIGGERS.UNACK}

Neapstiprināto trigeru skaits kartes elementam, neņemot vērā trigeru stāvokli.
 Trigeris tiek uzskatīts par neapstiprinātu, ja vismaz viens no tā problēmas notikumiem nav apstiprināts.

Lietotājvārda makrosi

{USER.FULLNAME}

Lietotāja vārds, uzvārds un lietotājvārds, kurš pievienoja notikuma apstiprinājumu vai palaida skriptu.

{USER.NAME}

Lietotāja vārds, kurš palaida skriptu.

{USER.SURNAME}

Lietotāja, kurš palaida skriptu, uzvārds.

{USER.USERNAME}

Lietotāja lietotājvārds, kurš palaida skriptu.

Indeksētie makrosi

Indeksētā makrosa sintakse **{MACRO<1-9>}** var atsaukties tikai uz N-to vienumu, funkciju vai hostu trigera lauka *Expression* ietvaros:

- **{HOST.IP1}, {HOST.IP2}, {HOST.IP3}** tiek atrisināti kā pirmā, otrā un trešā hosta IP adreses trigera izteiksmē (ja tādi ir).
- **{ITEM.VALUE1}, {ITEM.VALUE2}, {ITEM.VALUE3}** tiek atrisināti kā pirmā, otrā un trešā vienuma vērtības trigera izteiksmē notikuma brīdī (ja tādi ir).
- **{FUNCTION.VALUE1}, {FUNCTION.VALUE2}, {FUNCTION.VALUE3}** tiek atrisināti kā pirmās, otrās un trešās uz vienumiem balstītās funkcijas vērtības notikuma brīdī (ja tādas ir).

Trigeru kontekstā indeksētie makrosi vienmēr attiecas uz trigera konfigurācijas lauku *Expression*, nevis uz *Recovery expression*. Piemēram, atkopšanas notikumā **{ITEM.VALUE2}** tiks atrisināts kā otrā vienuma vērtība no problēmas izteiksmes atkopšanas brīdī.

Makross **{HOST.HOST<1-9>}** tiek atbalstīts arī izteiksmes makrosā `{?func(/host/key,param)}` **grafiku nosaukumos**. Piemēram, `{?func(/{HOST.HOST2}/key,param)}` grafika nosaukumā tiks atrisināts kā otrā vienuma hosts grafikā.

Warning:
Indeksētie makrosi netiks atrisināti nevienā citā kontekstā ārpus šeit minētajiem gadījumiem. Citos kontekstos izmantojiet makrosus bez indeksa (**{HOST.HOST}, {HOST.IP}** u.c.).

2 Lietotāja makrosi, ko atbalsta atrašanās vieta

Pārskats

Šajā sadaļā ir ietverts to vietu saraksts, kur tiek atbalstīti lietotāja definējami makrosi.

Note:
Tikai globālā līmeņa lietotāja makrosi tiek atbalstīti sadaļām *Darbības*, *Tīkla atklāšana*, *Starpniekserveri* un visām vietām, kas šajā lapā norādītas sadaļā *Citas vietas*. Minētajās vietās hosta līmeņa un veidnes līmeņa makrosi netiks atrisināti.

Note:
Lai pielāgotu makrosu vērtības (piemēram, saīsinātu vai izvilktu noteiktas apakšvirknes), varat izmantot makrosu funkcijas.

Darbības

Aktivitātēs lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
Uz trigeriem balstīti paziņojumi un komandas		jā
Uz trigeriem balstīti iekšējie paziņojumi		jā
Problēmas atjaunināšanas paziņojumi		jā
Uz pakalpojumiem balstīti paziņojumi un komandas		jā
Pakalpojuma atjaunināšanas paziņojumi		jā
Laika perioda nosacījums		nē
Darbības		
	Noklusējuma darbības soļa ilgums	nē
	Soļa ilgums	nē

Hosti/hostu prototipi

hosta un hosta prototipa konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Saskarnes IP/DNS	tikai DNS

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
Saskarnes ports		nē
SNMP v1, v2		
	SNMP kopiena	jā
SNMP v3		
	Konteksta nosaukums	jā
	Drošības nosaukums	jā
	Autentifikācijas parole	jā
	Privātuma parole	jā
IPMI		
	Lietotājvārds	jā
	Parole	jā
Birkas[2]		
	Birku nosaukumi	jā
	Birku vērtības	jā

Vienumi / vienumu prototipi

vienuma vai vienuma prototipa konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Vienuma nosaukums	jā
Vienuma atslēgas parametri	jā
Atjaunināšanas intervāls	nē
Pielāgoti intervāli	nē
Noildze (piejama atbalstītajiem vienumu tipiem)	nē
Glabāt līdz (vēsturei un tendencēm)	nē
Apraksts	jā
Aprēķinātais/agregētais vienums	
Formula (izteiksmes konstantes un funkciju parametri; vienuma atslēgas parametri; (*tikai agregētam vienumam*) filtra nosacījumi (hostu grupas nosaukums un birkas nosaukums))	jā

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Datubāzes monitors	
Lietotājvārds	jā
Parole	jā
SQL vaicājums	jā
HTTP aģents	
URL[3]	jā
Vaicājuma lauki	jā
Pieprasījuma pamatteksts	jā
Galvenes (nosaukumi un vērtības)	jā
Nepieciešamie statusa kodi	jā
HTTP starpniekserveris	jā
HTTP autentifikācijas lietotājvārds	jā
HTTP autentifikācijas parole	jā
SSl sertifikāta fails	jā
SSl atslēgas fails	jā
SSl atslēgas parole	jā
Atļautie hosti	jā
JMX aģents	
JMX galapunkts	jā
Skripta vienums	
Parametru nosaukumi un vērtības	jā
Pārlūka vienums	
Parametru nosaukumi un vērtības	jā
SNMP aģents	
SNMP OID	jā
SSH aģents	
Lietotājvārds	jā
Publiskās atslēgas fails	jā
Privātās atslēgas fails	jā
Parole	jā
Skripts	jā
TELNET aģents	
Lietotājvārds	jā
Parole	jā
Skripts	jā
Zabbix trapper	
Atļautie hosti	jā
Birkas[2]	
Birku nosaukumi	jā
Birku vērtības	jā
Pirmsapstrādes soļi	
Parametri (ieskaitot pielāgotus skriptus)	jā
Pielāgotās kļūdu apstrādes parametri (*Iestatīt vērtību uz* un *Iestatīt kļūdu uz* lauki)	jā

Zema līmeņa atklāšana

Lietotāja makro var izmantot šādos zema līmeņa atklāšanas noteikuma laukos:

Atrašanās vieta		Vairāki makro/sajauku
Atslēgas parametri		jā
Atjaunināšanas intervāls		nē
Pielāgots intervāls		nē
Noildze (pieejama atbalstītajiem vienumu tipiem)		nē
Zaudēto resursu dzēšana		nē
Zaudēto resursu atspējošana		nē
Apraksts		jā
SNMP aģents		
	SNMP OID	jā
SSH aģents		
	Lietotājvārds	jā
	Publiskās atslēgas fails	jā
	Privātās atslēgas fails	jā
	Parole	jā
	Skripts	jā
TELNET aģents		
	Lietotājvārds	jā
	Parole	jā
	Skripts	jā
Zabbix trapper		
	Atļautie hosti	jā
Datubāzes monitors		
	Lietotājvārds	jā
	Parole	jā
	SQL vaicājums	jā
JMX aģents		
	JMX galapunkts	jā
HTTP aģents		
	URL[3]	jā
	Vaicājuma lauki	jā
	Pieprasījuma pamatteksts	jā
	Galvenes (nosaukumi un vērtības)	jā
	Nepieciešamie statusa kodi	jā
	HTTP autentifikācijas lietotājvārds	jā
	HTTP autentifikācijas parole	jā
Filtri		
	Regulārā izteiksme	jā
Pārrakstīšanas		
	Filtri: regulārā izteiksme	jā
	Darbības: atjaunināšanas intervāls (vienumu prototipiem)	nē
	Darbības: vēstures glabāšanas periods (vienumu prototipiem)	nē
	Darbības: tendenču glabāšanas periods (vienumu prototipiem)	nē

Tīkla atklāšana

Network discovery rule lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
Atjaunināšanas intervāls		nē
SNMP v1, v2		
	SNMP kopiena	jā
	SNMP OID	jā
SNMP v3		
	Konteksta nosaukums	jā
	Drošības nosaukums	jā
	Autentifikācijas parole	jā
	Privātuma parole	jā
	SNMP OID	jā

Starpniekserveri

Starpniekservera konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Adrese aktīvajiem aģentiem > *Ports* (ja starpniekserveris pieder grupai)	nē
Saskarnes adrese un ports (pasīvajam starpniekserverim)	nē
Noildzes vienumu tipiem	nē

Starpniekserveru grupas

starpniekserveru grupas konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Pārslēgšanās periods	nē
Minimālais starpniekserveru skaits	nē

Veidnes

veidnes konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
Birkas[2]		
	Birku nosaukumi	jā
	Birku vērtības	jā

Trigeri

trigera konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Nosaukums	jā
Darbības dati	jā
Izteiksme (tikai konstantēs un funkciju parametros; slepenie makrosi netiek atbalstīti)	jā
Tags atbilstības noteikšanai	jā
Izvēlnes ierakstā nosaukums	jā
Izvēlnes ierakstā URL[3]	jā
Apraksts	jā
Tagi[2]	
Tagu nosaukumi	jā
Tagu vērtības	jā

Tīmekļa scenārijs

Web scenario konfigurācijā lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
Nosaukums		jā
Atjaunināšanas intervāls		nē
Aģents		jā
HTTP starpniekserveris		jā
Mainīgie (tikai vērtības)		jā
Galvenes (nosaukumi un vērtības)		jā
Soļi		
	Nosaukums	jā
	URL[3]	jā
	Mainīgie (tikai vērtības)	jā
	Galvenes (nosaukumi un vērtības)	jā
	Noildze	nē
	Obligātā virkne	jā
	Obligātie statusa kodi	nē
Autentifikācija		

Atrašanās vieta		Vairāki makrosi/sajaukums ar tekstu[1]
	Lietotājs	jā
	Parole	jā
	SSL sertifikāts	jā
	SSL atslēgas fails	jā
	SSL atslēgas parole	jā
Birkas[2]		
	Birku nosaukumi	jā
	Birku vērtības	jā

Citas atrašanās vietas

Papildus šeit uzskaitītajām atrašanās vietām lietotāja makrosus var izmantot šādos laukos:

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Globālie skripti (URL, skripts, SSH, Telnet, IPMI), ieskaitot apstiprinājuma tekstu	jā
Webhook	
JavaScript skripts	nē
JavaScript skripta parametra nosaukums	nē
JavaScript skripta parametra vērtība	jā
Paneļi	
Datu tipa *Text* kolonna paneļa logrīkā *Top hosts*	jā
Parametrs *Description* paneļa logrīkos *Item value* un *Gauge*	jā
Primārās/sekundārās etiķetes parametrs *Text* paneļa logrīkā *Honeycomb*	jā
Parametrs *URL*[3] paneļa logrīkā *URL*	jā
Lietotāji → *Lietotāji* → *Mediji*	
Kad aktīvs	jā
Administrēšana → *Vispārīgi* → *GUI*	
Darba laiks	nē
Administrēšana → *Vispārīgi* → *Noildzes*	
Noildzes vienumu tipiem	nē

Atrašanās vieta	Vairāki makrosi/sajaukums ar tekstu[1]
Administrēšana → *Vispārīgi* → *Savienotāji*	
URL	jā
Lietotājvārds	jā
Parole	jā
Bearer token	jā
Noildze	nē
HTTP starpniekserveris	jā
SSL sertifikāta fails	jā
SSL atslēgas fails	jā
SSL atslēgas parole	jā
Brīdinājumi → *Mediju tipi* → *Ziņojumu veidnes*	
Temats	jā
Ziņojums	jā
Brīdinājumi → *Mediju tipi* → *Skripts*	
Skripta parametri	jā
Brīdinājumi → *Mediju tipi* → *Medija tips*	
Lauki *Username* un *Password* medija tipam *Email* (ja *Authentication* ir iestatīts uz ”Username and password”; ieteicami slepenie makrosi)	jā

Pilnu visu Zabbix atbalstīto makrosu sarakstu skatiet sadaļā atbalstītie makrosi.

Piezīmes

1 Ja atrašanās vietai netiek atbalstīti vairāki makrosi laukā vai makrosi, kas sajaukti ar tekstu, vienam makrosam ir jāaizpilda viss lauks.

2 Makrosi, kas izmantoti birku nosaukumos un vērtībās, tiek atrisināti tikai notikuma ģenerēšanas procesa laikā.

3 URL, kas satur slepeno makrosu, nedarbosies, jo tajos esošais makross tiks atrisināts kā ”******”.

7 Vienību simboli

Pārskats

Darbs ar lielām vērtībām, piemēram, 86400, 104857600 vai 1000000, var būt sarežģīts un var izraisīt kļūdas. Lai vienkāršotu konfigurēšanu un uzlabotu lasāmību, Zabbix atbalsta mērvienību simbolus (sufiksus), kas darbojas kā vērtību reizinātāji.

Trigeru izteiksmes bez sufiksiem:

```
last(/host/system.uptime)<86400
avg(/host/system.cpu.load,600s)<10
last(/host/vm.memory.size[available])<20971520
```

Trigeru izteiksmes ar sufiksiem:

```
last(/host/system.uptime)<1d
avg(/host/system.cpu.load,10m)<10
last(/host/vm.memory.size[available])<20M
```

Sufiksi var arī vienkāršot citu entītiju, piemēram, vienumu, logrīku u.c., konfigurēšanu un palīdzēt attēlot vienumu vērtības cilvēkam saprotamā formātā.

Note:
Lai noskaidrotu, vai entītijas konfigurācijas lauks atbalsta sufiksus, vienmēr skatiet attiecīgo lapu par konfigurējamo entītiju.

Laika sufiksi

Zabbix lietotāja saskarne atbalsta šādus laika sufiksus entītiju konfigurācijā:

- **s** - sekundes *(lietojot, darbojas tāpat kā neapstrādātā vērtība)*
- **m** - minūtes
- **h** - stundas
- **d** - dienas
- **w** - nedēļas
- **M** - mēneši *(trendu funkcijas tikai)*
- **y** - gadi *(trendu funkcijas tikai)*

Laika sufiksi tiek atbalstīti tikai ar veseliem skaitļiem. Piemēram, 1h tiek atbalstīts, bet 1,5h vai 1.5h netiek atbalstīts; tā vietā izmantojiet 90m.

Atmiņas lieluma sufiksi

Zabbix atbalsta šādus atmiņas lieluma sufiksus:

- **K** - kilobaiti
- **M** - megabaiti
- **G** - gigabaiti
- **T** - terabaiti

Vienuma vērtību sufiksi

Sufiksus var izmantot arī, lai attēlotu skaitliskas vienuma vērtības cilvēkam saprotamā formātā.

Lai to iespējotu, izmantojiet vienu no tālāk norādītajiem sufiksiem laukā *Units*, konfigurējot vienumu:

- **B** - baiti
- **Bps** - baiti sekundē
- **s** - sekundes, attēlotas, izmantojot līdz trim lielākajām nenulles laika vienībām
- **uptime** - pagājušais laiks formātā hh:mm:ss vai N days, hh:mm:ss
- **unixtime** - Unix laikspiedols, formatēts kā yyyy.mm.dd hh:mm:ss

Tālāk norādītie papildu noteikumi attiecas uz to, kā šie sufiksi tiek interpretēti un attēloti:

- Attiecībā uz B un Bps Zabbix izmanto bāzes-2 konvertēšanu (1K = 1024B), ievērojot JEDEC standartu.
- Citām vienībām (piemēram, Hz, W utt.) Zabbix izmanto bāzes-10 konvertēšanu (1K = 1000).
- Attiecībā uz s (sekundēm):
  - Formāts ietver yyy mmm ddd hhh mmm sss ms; tiek attēlotas tikai līdz trim lielākajām nenulles laika vienībām (piemēram, 1M 10d 4h).
  - Ja vienība ir nulle un atrodas starp divām nenulles vienībām, tā tiek izlaista (piemēram, 10d 56m, nevis 10d 0h 56m).

Ja tiek izmantots *Units*, vienuma vērtībām automātiski tiek piemēroti šādi reizinātāju sufiksi:

- **K**, **M**, **G**, **T** - kilo, mega, giga, tera
- **P**, **E**, **Z**, **Y** - peta, eksa, zeta, jota *(tie tiek piemēroti tikai lietotāja saskarnē)*

Lai novērstu vienību konvertēšanu, izmantojiet prefiksu ! (piemēram, !B vai !s).

Tālāk sniegtie piemēri parāda, kā saņemtās vienuma vērtības tiek konvertētas atbilstoši norādītajām vienībām:

```
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:
Pirms Zabbix 4.0 bija stingri definēts vienību izņēmumu saraksts, kas sastāvēja no `ms`, `rpm`, `RPM`, `%`. Šis izņēmumu saraksts ir novecojis, tādēļ pareizais veids, kā novērst šādu vienību konvertēšanu, ir `!ms`, `!rpm`, `!RPM`, `!%`.

8 Laika perioda sintakse

Pārskats

Lai iestatītu laika periodu, jāizmanto šāds formāts:

`d-d,hh:mm-hh:mm`

kur simboli nozīmē:

Symbol	Apraksts
d	Nedēļas diena: 1 - pirmdiena, 2 - otrdiena, ... , 7 - svētdiena
hh	Stundas: 00-24
mm	Minūtes: 00-59

Varat norādīt vairāk nekā vienu laika periodu, izmantojot semikolu (;) kā atdalītāju:

`d-d,hh:mm-hh:mm;d-d,hh:mm-hh:mm...`

Atstājot laika periodu tukšu, tas ir ekvivalents vērtībai 1-7,00:00-24:00, kas ir noklusējuma vērtība.

Attention:
Laika perioda augšējā robeža netiek iekļauta. Tādēļ, ja norādāt 09:00-18:00, pēdējā sekunde, kas ir iekļauta laika periodā, ir 17:59:59.

Piemēri

Darba laiks. No pirmdienas līdz piektdienai no 9:00 līdz 18:00:

```
1-5,09:00-18:00
```

Darba laiks, ieskaitot nedēļas nogali. No pirmdienas līdz piektdienai no 9:00 līdz 18:00 un sestdien, svētdien no 10:00 līdz 16:00:

```
1-5,09:00-18:00;6-7,10:00-16:00
```

9 Komandu izpilde

Zabbix izmanto kopīgu funkcionalitāti ārējām pārbaudēm, lietotāja parametriem, system.run vienumiem, pielāgotiem brīdinājumu skriptiem, attālinātām komandām un globālajiem skriptiem.

Izpildes soļi

Note:
Pēc noklusējuma visi skripti Zabbix tiek izpildīti, izmantojot *sh* čaulu, un noklusējuma čaulu nav iespējams mainīt. Lai izmantotu citu čaulu, var izmantot apiešanas risinājumu: izveidot skripta failu un komandas izpildes laikā izsaukt šo skriptu.

Komanda/skripts tiek izpildīts līdzīgi gan Unix, gan Windows platformās:

1. Zabbix (vecākprocess) izveido kanālu saziņai
2. Zabbix iestata šo kanālu kā izvadi izveidojamajam bērnprocesam
3. Zabbix izveido bērnprocesu (palaiž komandu/skriptu)
4. Bērnprocesam tiek izveidota jauna procesu grupa (Unix) vai uzdevums (Windows)
5. Zabbix lasa no kanāla, līdz iestājas noildze vai arī neviens vairs neraksta otrā galā (visi turi/failu deskriptori ir aizvērti). Ņemiet vērā, ka bērnprocess var izveidot papildu procesus un beigt darbu pirms tie beidz darbu vai aizver turi/failu deskriptoru.
6. Ja noildze vēl nav sasniegta, Zabbix gaida, līdz sākotnējais bērnprocess beidz darbu vai iestājas noildze
7. Ja sākotnējais bērnprocess ir beidzis darbu un noildze vēl nav sasniegta, Zabbix pārbauda sākotnējā bērnprocesa izejas kodu un salīdzina to ar 0 (vērtība, kas nav nulle, tiek uzskatīta par izpildes kļūmi, tikai pielāgotajiem brīdinājumu skriptiem, attālinātajām komandām un lietotāju skriptiem, kas izpildīti uz Zabbix serveris un Zabbix starpniekserveris)
8. Šajā brīdī tiek pieņemts, ka viss ir pabeigts un viss procesu koks (t.i., procesu grupa vai uzdevums) tiek pārtraukts

Attention:
Zabbix pieņem, ka komanda/skripts ir pabeidzis apstrādi, kad sākotnējais bērnprocess ir beidzis darbu UN neviens cits process vairs netur atvērtu izvades turi/failu deskriptoru. Kad apstrāde ir pabeigta, visi izveidotie procesi tiek pārtraukti.

Visas dubultpēdiņas un atpakaļvērstās slīpsvītras komandā tiek ekrānētas ar atpakaļvērstajām slīpsvītrām, un komanda tiek ietverta dubultpēdiņās.

Izejas koda pārbaude

Izejas kodi tiek pārbaudīti saskaņā ar šādiem nosacījumiem:

- Tikai pielāgotiem brīdinājumu skriptiem, attālinātām komandām un lietotāju skriptiem, kas tiek izpildīti Zabbix serverī un Zabbix starpniekserverī.
- Jebkurš izejas kods, kas atšķiras no 0, tiek uzskatīts par izpildes kļūmi.
- Neveiksmīgu izpildes reižu standarta kļūdu izvades un standarta izvades saturs tiek apkopots un ir pieejams lietotāja saskarnē (kur tiek parādīts izpildes rezultāts).
- Papildu žurnāla ieraksts var tikt izveidots attālinātām komandām, kas izpildītas Zabbix aģentā/starpniekserverī, iespējojot parametru LogRemoteCommands aģenta/starpniekservera konfigurācijā.

Iespējamie ziņojumi lietotāja saskarnē un žurnāla ieraksti neveiksmīgām komandām/skriptiem:

- Neveiksmīgu izpildes reižu standarta kļūdu izvades un standarta izvades saturs (ja tāds ir).
- "Process exited with code: N." (ja izvade ir tukša un izejas kods nav vienāds ar 0).
- "Process killed by signal: N." (ja procesu pārtrauc signāls, tikai Linux vidē).
- "Process terminated unexpectedly." (ja process tiek pārtraukts nezināmu iemeslu dēļ).

Skatīt arī

- Ārējās pārbaudes
- Lietotāja parametri
- system.run vienumi
- Pielāgoti brīdinājumu skripti

- Attālās komandas
- Globālie skripti

10 Versiju saderība

Atbalstītie aģenti

Lai būtu saderīgs ar Zabbix 8.0, Zabbix aģents nedrīkst būt vecāks par versiju 1.4 un nedrīkst būt jaunāks par 8.0.

Iespējams, būs jāpārskata vecāku aģentu konfigurācija, jo daži parametri ir mainīti, piemēram, parametri, kas saistīti ar žurnalēšanu versijām pirms 3.0.

Lai pilnībā izmantotu jaunāko funkcionalitāti, metriku, uzlaboto veiktspēju un samazinātu atmiņas lietojumu, izmantojiet jaunāko atbalstīto aģentu.

Piezīmes par Windows XP

- 32 bitu Windows XP sistēmā neizmantojiet Zabbix aģentus, kas ir jaunāki par 6.0.x;
- Windows XP/Server 2003 sistēmā neizmantojiet aģentu veidnes, kas ir jaunākas par Zabbix 4.0.x. Jaunākās veidnes izmanto angļu valodas veiktspējas skaitītājus, kas tiek atbalstīti tikai kopš Windows Vista/Server 2008.

Atbalstītie aģenti 2

Vecākas Zabbix aģenta 2 versijas, sākot no 4.4, ir saderīgas ar Zabbix 8.0; Zabbix aģents 2 nedrīkst būt jaunāks par 8.0.

Ņemiet vērā, ka, izmantojot Zabbix aģenta 2 versijas 4.4 un 5.0, pēc noklusējuma neatbalstīto vienumu atsvaidzināšanai tiek izmantots 10 minūšu intervāls.

Lai pilnībā izmantotu jaunāko funkcionalitāti, metriku, uzlaboto veiktspēju un samazinātu atmiņas lietojumu, izmantojiet jaunāko atbalstīto aģentu 2.

Atbalstītie Zabbix starpniekserveri

Pilnīgai saderībai ar Zabbix 8.0 starpniekserveriem ir jāatbilst servera galvenajai versijai. Tikai Zabbix 8.0.x starpniekserveri ir pilnībā saderīgi ar Zabbix 8.0.x serveri.

Novecojuši starpniekserveri tiek atbalstīti daļēji: tie joprojām var vākt datus un izpildīt skriptus, taču nevar saņemt konfigurācijas atjauninājumus, piemēram, jaunus vienumus.

Attiecībā pret Zabbix serveri starpniekserveri var būt:

- *Pašreizējie* (starpniekserverim un serverim ir viena un tā pati galvenā versija);
- *Novecojuši* (starpniekservera versija ir vecāka par servera versiju, bet tiek daļēji atbalstīta);
- *Neatbalstīti* (starpniekservera versija ir vecāka par servera iepriekšējā LTS laidiena versiju *vai* starpniekservera versija ir jaunāka par servera galveno versiju).

Piemēri:

Servera versija	*Pašreizējā* starpniekservera versija	*Novecojusi* starpniekservera versija	*Neatbalstīta* starpniekservera versija
6.4	6.4	6.0, 6.2	Vecāka par 6.0; jaunāka par 6.4
7.0	7.0	6.0, 6.2, 6.4	Vecāka par 6.0; jaunāka par 7.0
7.2	7.2	7.0	Vecāka par 7.0; jaunāka par 7.2
7.4	7.4	7.0	Vecāka par 7.0; jaunāka par 7.4
8.0	8.0	7.0, 7.2, 7.4	Vecāka par 7.0; jaunāka par 8.0

Starpniekserveru atbalstītā funkcionalitāte:

Starpniekservera versija	Datu atjaunināšana	Konfigurācijas atjaunināšana	Uzdevumi
Pašreizējā	Jā	Jā	Jā
Novecojusi	Jā	Nē	Attālinātās komandas (piemēram, čaulas skripti); Tūlītējas vienuma vērtības pārbaudes (t. i., *Izpildīt tagad*); Piezīme: priekšapstrādes testi ar reālu vērtību netiek atbalstīti.

Starpniekservera versija	Datu atjaunināšana	Konfigurācijas atjaunināšana	Uzdevumi
Neatbalstīta	Nē	Nē	Nē

Brīdinājumi par nesaderīgu Zabbix dēmonu versiju izmantošanu tiek reģistrēti žurnālā.

Atbalstītie XML faili

XML faili, kas nav vecāki par versiju 1.8, tiek atbalstīti importēšanai Zabbix 8.0.

Attention:
XML eksporta formātā trigeru atkarības tiek saglabātas tikai pēc nosaukuma. Ja ir vairāki trigeri ar vienādu nosaukumu (piemēram, ar atšķirīgām nopietnības pakāpēm un izteiksmēm), starp kuriem ir definēta atkarība, tos nav iespējams importēt. Šādas atkarības ir manuāli jānoņem no XML faila un pēc importēšanas jāpievieno no jauna.

12 Zabbix sender dinamisko saišu bibliotēka operētājsistēmai Windows

Pārskats

Windows vidē lietojumprogrammas var sūtīt datus uz Zabbix serveri/starpniekserveri, izmantojot Zabbix sender dinamisko saišu bibliotēku (zabbix_sender.dll), nevis palaižot ārēju procesu (zabbix_sender.exe).

Lai kompilētu lietotāju lietojumprogrammas ar zabbix_sender.dll, ir nepieciešami zabbix_sender.h un zabbix_sender.lib.

Iegūšana

Ir divi veidi, kā iegūt zabbix_sender.dll.

1. Lejupielādējiet failus zabbix_sender.h, zabbix_sender.lib un zabbix_sender.dll ZIP arhīva veidā.

Izvēloties lejupielādes opcijas, pārliecinieties, ka sadaļā *Encryption* ir atlasīts "No encryption" un sadaļā *Packaging* ir atlasīts "Archive". Pēc tam lejupielādējiet Zabbix aģentu (nevis Zabbix aģentu 2).

Faili zabbix_sender.h, zabbix_sender.lib un zabbix_sender.dll atradīsies lejupielādētajā ZIP arhīvā direktorijā `bin\dev`. Atarhivējiet failus vajadzīgajā vietā.

2. Izveidojiet zabbix_sender.dll no pirmkoda (skatiet norādījumus).

Dinamisko saišu bibliotēka kopā ar izstrādes failiem atradīsies direktorijā `bin\winXX\dev`. Lai to izmantotu, iekļaujiet zabbix_sender.h galvenes failu un veiciet sasaisti ar zabbix_sender.lib bibliotēku.

Skatiet arī

- vienkāršas Zabbix sender utilītas piemērs, kas izstrādāta, izmantojot Zabbix sender dinamisko saišu bibliotēku, lai ilustrētu bibliotēkas lietošanu;
- fails zabbix_sender.h Zabbix sender dinamisko saišu bibliotēkas saskarnes funkcijām. Šajā failā ir dokumentācija, kas izskaidro katras saskarnes funkcijas mērķi, tās argumentus un atgriezto vērtību.

13 Pakalpojumu uzraudzības jaunināšana

Pārskats Zabbix 6.0 versijā pakalpojumu uzraudzības funkcionalitāte ir būtiski pārstrādāta (izmaiņu sarakstu skatiet sadaļā Kas jauns Zabbix 6.0.0).

Šajā lapā ir aprakstīts, kā pakalpojumi un SLA, kas definēti iepriekšējās Zabbix versijās, tiek mainīti jaunināšanas laikā uz Zabbix 6.0 vai jaunāku versiju.

Pakalpojumi Vecākās Zabbix versijās pakalpojumiem bija divu veidu atkarības: mīkstās un stingrās. Pēc jaunināšanas visas atkarības kļūs vienādas.

Ja pakalpojums "Child service" iepriekš bija saistīts ar "Parent service 1", izmantojot stingro atkarību, un papildus ar "Parent service 2", izmantojot mīksto atkarību, tad pēc jaunināšanas pakalpojumam "Child service" būs divi vecākpakalpojumi: "Parent service 1" un "Parent service 2".

Uz trigeriem balstītā problēmu un pakalpojumu kartēšana ir aizstāta ar uz birkām balstītu kartēšanu. Zabbix 6.0 un jaunākās versijās pakalpojuma konfigurācijas formai ir jauns parametrs *Problem tags*, kas ļauj norādīt vienu vai vairākus birkas nosaukuma un vērtības pārus problēmu atbilstības noteikšanai. Trigeri, kas ir saistīti ar pakalpojumu, saņems jaunu birku `ServiceLink : <trigera ID>:<trigera nosaukums>` (birkas vērtība tiks saīsināta līdz 32 rakstzīmēm). Saistītie pakalpojumi saņems `ServiceLink` problēmas birku ar to pašu vērtību.

Statusa aprēķināšanas noteikumi

'Status calculation algorithm' tiks uzlabots, izmantojot šādus noteikumus:

- Neaprēķināt → Iestatīt statusu uz OK
- Problēma, ja vismaz vienam bērnservisam ir problēma → Bērnservisu kritiskākais statuss
- Problēma, ja visiem bērnservisiem ir problēmas → Kritiskākais statuss, ja visiem bērnservisiem ir problēmas

SLA Iepriekš SLA mērķi bija jādefinē katram pakalpojumam atsevišķi. Sākot ar Zabbix 6.0, SLA ir kļuvis par atsevišķu entītiju, kas satur informāciju par pakalpojuma grafiku, sagaidāmo pakalpojuma līmeņa mērķi (SLO) un dīkstāves periodiem, kas jāizslēdz no aprēķina. Pēc konfigurēšanas SLA var piešķirt vairākiem pakalpojumiem, izmantojot pakalpojumu birkas.

Atjaunināšanas laikā:

- Identiski SLA, kas definēti katram pakalpojumam, tiks grupēti, un katrai grupai tiks izveidots viens SLA.
- Katrs ietekmētais pakalpojums saņems īpašu birku `SLA:<ID>`, un tā pati birka tiks norādīta atbilstošā SLA parametrā *Pakalpojumu birkas*.
- Pakalpojuma izveides laiks — jauna metrika SLA atskaitēs — esošajiem pakalpojumiem tiks iestatīts uz 01/01/2000 00:00.

14 Citas problēmas

Pieteikšanās un systemd

Mēs iesakām izveidot *zabbix* lietotāju kā sistēmas lietotāju, tas ir, bez iespējas pieteikties sistēmā. Daži lietotāji ignorē šo ieteikumu un izmanto to pašu kontu, lai pieteiktos (piemēram, izmantojot SSH) hostā, kurā darbojas Zabbix. Tas var izraisīt Zabbix dēmona avāriju, izrakstoties no sistēmas. Šādā gadījumā Zabbix servera žurnālā redzēsiet kaut ko līdzīgu šim:

```
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
```

un Zabbix aģenta žurnālā:

```
zabbix_agentd [27796]: [file:'log.c',line:238] lock failed: [22] Invalid argument
```

Tas notiek noklusējuma systemd iestatījuma `RemoveIPC=yes` dēļ, kas konfigurēts failā `/etc/systemd/logind.conf`. Kad izrakstāties no sistēmas, iepriekš Zabbix izveidotie semafori tiek noņemti, kas izraisa avāriju.

Citāts no systemd dokumentācijas:

```
RemoveIPC=

Nosaka, vai lietotājam piederošie System V un POSIX IPC objekti ir jānoņem, kad
lietotājs pilnībā izrakstās. Pieņem būla argumentu. Ja iespējots, lietotājs nevar izmantot IPC resursus
pēc tam, kad ir beigusies pēdējā lietotāja sesija. Tas attiecas uz System V semaforiem, koplietojamo atmiņ
un ziņojumu rindām, kā arī uz POSIX koplietojamo atmiņu un ziņojumu rindām. Ņemiet vērā, ka
root lietotāja un citu sistēmas lietotāju IPC objekti nav pakļauti šī iestatījuma ietekmei. Pēc noklusējum
```

Šai problēmai ir 2 risinājumi:

1. (ieteicams) Pārtrauciet izmantot *zabbix* kontu jebkam citam, izņemot Zabbix procesus; citām vajadzībām izveidojiet atsevišķu kontu.
2. (nav ieteicams) Iestatiet `RemoveIPC=no` failā `/etc/systemd/logind.conf` un pārstartējiet sistēmu. Ņemiet vērā, ka `RemoveIPC` ir visas sistēmas parametrs, un tā maiņa ietekmēs visu sistēmu.

Zabbix lietotāja saskarnes izmantošana aiz starpniekservera

Ja Zabbix lietotāja saskarne darbojas aiz starpniekservera, sīkdatnes ceļš starpniekservera konfigurācijas failā ir jāpārraksta, lai tas atbilstu apgrieztā starpniekservera ceļam. Skatiet tālāk sniegtos piemērus. Ja sīkdatnes ceļš netiek pārrakstīts, lietotājiem var rasties autorizācijas problēmas, mēģinot pieteikties Zabbix lietotāja saskarnē.

Nginx konfigurācijas piemērs

```
# ..
location / {
# ..
proxy_cookie_path /zabbix /;
proxy_pass http://192.168.0.94/zabbix/;
# ..
```

Apache konfigurācijas piemērs

```
# ..
ProxyPass "/" http://host/zabbix/
ProxyPassReverse "/" http://host/zabbix/
ProxyPassReverseCookiePath /zabbix /
ProxyPassReverseCookieDomain host zabbix.example.com
# ..
```

16 Ekranēšanas piemēri

Pārskats

Šajā lapā ir sniegti piemēri pareizas atsoļošanas izmantošanai, lietojot regulārās izteiksmes dažādos kontekstos.

> **Note:**
> Izmantojot trigeris izteiksmes konstruktoru, pareiza atsoļošana regulārajās izteiksmēs tiek pievienota automātiski.

Piemēri

Lietotāja makross ar kontekstu

Regulārā izteiksme: `\.+\"[a-z]+<br>` Lietotāja makross ar kontekstu: `{$MACRO:regex:"\.+\\"[a-z]+"}`

Ņemiet vērā:

- atpakaļvērstās slīpsvītras nav atdalītas;
- pēdiņas ir atdalītas.

Makro funkcija vienuma atslēgas parametrā

Regulārā izteiksme: `.+:(\d+)$<br>` Vienuma atslēga: `net.tcp.service[tcp,,"{{$ENDPOINT}.regsub(\".+:(\d+)$\",\1)}"]`

Ņemiet vērā:

- regulārā izteiksme `regsub` makro funkcijā ir ievietota dubultpēdiņās (jo satur aizverošo iekavu);
- pēdiņas ap regulāro izteiksmi ir atdalītas (jo viss trešais vienuma parametrs ir ievietots dubultpēdiņās);
- trešais vienuma atslēgas parametrs ir ievietots dubultpēdiņās, jo tas satur komatu.

LLD makro funkcija

Regulārā izteiksme: `\.+\"([a-z]+)<br>` LLD makross: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}`

Ņemiet vērā:

- atpakaļvērstās slīpsvītras nav atdalītas;
- pēdiņas ir atdalītas.

LLD makro funkcija lietotāja makrosa kontekstā

Regulārā izteiksme: `\.+\"([a-z]+)<br>` LLD makross: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Lietotāja makross ar kontekstu: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"[a-z]+\", \1)}"}`

Ņemiet vērā:

- atpakaļvērsto slīpsvītru atdalīšana LLD gadījumā nemainās;
- ievietojot LLD makrosu lietotāja makrosa kontekstā, tas jāievieto virknē:

1. ap makro izteiksmi tiek pievienotas pēdiņas;
2. pēdiņas tiek atdalītas; kopumā tiek ieviestas 3 jaunas atpakaļvērstās slīpsvītras.

Funkcijas virknes parametrs (jebkurš)

Kā piemērs izmantots `concat`.

Virknes saturs: `\.+\"[a-z]+<br>` Izteiksme: `concat("abc", "\\.\\\"[a-z]+")`

Ņemiet vērā:

- virknes parametros jāatdala gan atpakaļvērstās slīpsvītras, gan pēdiņas.

LLD makro funkcija funkcijas virknes parametrā

Regulārā izteiksme: `\.+\"([a-z]+)<br>` LLD makross: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Izteiksme: `concat("abc, "{{#MACRO}.iregsub(\"\\.+\\\\\"([a-z]+)\", \\1)}")`

Ņemiet vērā:

- virknes parametros jāatdala gan atpakaļvērstās slīpsvītras, gan pēdiņas;
- tiek pievienots vēl viens atdalīšanas slānis, jo makross tiks atrisināts tikai pēc tam, kad virkne būs noņemta no pēdiņām;

Lietotāja makross ar kontekstu funkcijas virknes parametrā

Regulārā izteiksme: `\.+\"[a-z]+<br>` Lietotāja makross ar kontekstu: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Izteiksme: `concat("abc, "{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}")`

Ņemiet vērā:

- tāpat kā iepriekšējā piemērā ir nepieciešams papildu atdalīšanas slānis;
- atpakaļvērstās slīpsvītras un pēdiņas tiek atdalītas tikai augstākā līmeņa atdalīšanai (jo tas ir virknes parametrs).

LLD makro funkcija lietotāja makrosa kontekstā funkcijā

Regulārā izteiksme: `\.+\"([a-z])+<br>` LLD makross: `{{#MACRO}.iregsub("\.+\\"([a-z]+)", \1)}<br>` Lietotāja makross ar kontekstu: `{$MACRO:"{{#MACRO}.iregsub(\"\.+\\\"([a-z]+)\", \1)}"}<br>` Izteiksme: `concat("abc, "{$MACRO:\"{{#MACRO}.iregsub(\\\"\.+\\\\\\\"([a-z]+)\\\", \\1)}\"}")`

Ņemiet vērā trīs atdalīšanas slāņus:

1. LLD makro funkcijai bez atpakaļvērsto slīpsvītru atdalīšanas;
2. Lietotāja makrosam ar kontekstu bez atpakaļvērsto slīpsvītru atdalīšanas;
3. Funkcijas virknes parametram ar atpakaļvērsto slīpsvītru atdalīšanu.

Lietotāja makross ar kontekstu tieši virknē

Regulārā izteiksme: `\.+\"[a-z]+<br>` Lietotāja makross ar kontekstu: `{$MACRO:regex:"\.+\\"[a-z]+"}<br>` Kādas izteiksmes virknes iekšpusē, piemēram: `func(arg1, arg2, arg3)="{$MACRO:regex:\"\\.+\\\\\"[a-z]+\"}"`

Ņemiet vērā:

- virknēs arī jāatdala atpakaļvērstās slīpsvītras;
- virknēs arī jāatdala pēdiņas;
- atkal gadījums ar 2 atdalīšanas līmeņiem:

1. atdalīšana lietotāja makrosa kontekstam bez atpakaļvērsto slīpsvītru atdalīšanas;
2. atdalīšana tāpēc, ka tā ir virkne, ar atpakaļvērsto slīpsvītru atdalīšanu.

21 Īsās atsauces rokasgrāmatas

Pārskats

Šajā dokumentācijas sadaļā ir iekļautas īsas receptes Zabbix iestatīšanai dažiem bieži nepieciešamiem uzraudzības mērķiem.

Tā ir izstrādāta, domājot par jauno Zabbix lietotāju, un to var izmantot kā ceļvedi pa citām dokumentācijas sadaļām, kas satur informāciju, kura nepieciešama uzdevuma atrisināšanai.

Ir pieejamas šādas īsās atsauces rokasgrāmatas:

- Linux uzraudzība ar Zabbix aģentu
- Windows uzraudzība ar Zabbix aģentu
- Apache uzraudzība, izmantojot HTTP
- MySQL uzraudzība ar Zabbix aģentu 2
- VMware uzraudzība ar Zabbix
- Tīkla datplūsmas uzraudzība ar Zabbix
- Tīkla datplūsmas uzraudzība ar Zabbix, izmantojot aktīvās pārbaudes
- Tīmekļa vietņu uzraudzība ar pārlūka vienumiem

- Tīmekļa vietņu sertifikātu uzraudzība ar Zabbix aģentu 2 (pasīvā)
- Tīkla komutatora vai maršrutētāja uzraudzība ar Zabbix
- Windows notikumu žurnāla uzraudzība, izmantojot aktīvās pārbaudes

1. Monitor Linux with Zabbix agent

Ievads Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai ar Zabbix sāktu Linux datoru pamata uzraudzību.
Šajā pamācībā aprakstītās darbības var piemērot jebkurai uz Linux balstītai operētājsistēmai.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu jūsu Linux datora pamata uzraudzību. Ja meklējat padziļinātas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt ar šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu OS paredzētajām instrukcijām.

Instalēt Zabbix aģentu Zabbix aģents ir process, kas ir atbildīgs par datu vākšanu.

Pārbaudiet savu Zabbix servera versiju:

```
zabbix_server -V
```

Instalējiet tās pašas versijas Zabbix aģentu (ieteicams) Linux datorā, kuru vēlaties uzraudzīt. Atkarībā no jūsu uzraudzības vajadzībām tas var būt tas pats dators, kurā ir instalēts Zabbix serveris, vai arī pilnīgi cits dators.

Izvēlieties vispiemērotāko instalēšanas metodi:

- Palaidiet kā Docker konteineru — skatiet pieejamo attēlu sarakstu Zabbix Docker repozitorijā.
- Instalējiet no Zabbix pakotnēm (pieejamas Alma Linux, CentOS, Debian, Oracle Linux, Raspberry Pi OS, RHEL, Rocky Linux, SUSE Linux Enterprise Server, Ubuntu).
- Kompilējiet no pirmkoda.

Konfigurējiet Zabbix uzraudzībai Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus).

> **Note:**
> Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa dažus datus (piemēram, CPU slodzi), un Zabbix aģents nosūta rezultātu atpakaļ serverim. Aktīvajām pārbaudēm ir nepieciešama sarežģītāka apstrāde. Aģentam vispirms jāiegūst no servera(-iem) vienumu saraksts neatkarīgai apstrādei un pēc tam dati paketēs jānosūta atpakaļ. Plašāku informāciju skatiet sadaļā Pasīvās un aktīvās aģenta pārbaudes.

Zabbix nodrošinātās uzraudzības veidnes parasti piedāvā divas alternatīvas — veidni Zabbix aģentam un veidni Zabbix aģentam (aktīvs). Izmantojot pirmo iespēju, aģents vāks metriku pasīvajā režīmā. Šādas veidnes nodrošinās identiskus uzraudzības rezultātus, taču izmantos atšķirīgus saziņas protokolus.

Turpmākā Zabbix konfigurēšana ir atkarīga no tā, vai izvēlaties veidni aktīvām vai pasīvām Zabbix aģenta pārbaudēm.

Pasīvās pārbaudes Zabbix aģents

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents.

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

2. Pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligāto portu) server parameter parametram. Piemēram:

```
Server=192.0.2.0:10051
```

Zabbix aģents izmantos šo adresi, lai pieņemtu ienākošos savienojumus tikai no norādītajiem Zabbix serveriem vai starpniekserveriem datu iegūšanai.

3. Restartējiet Zabbix aģentu.

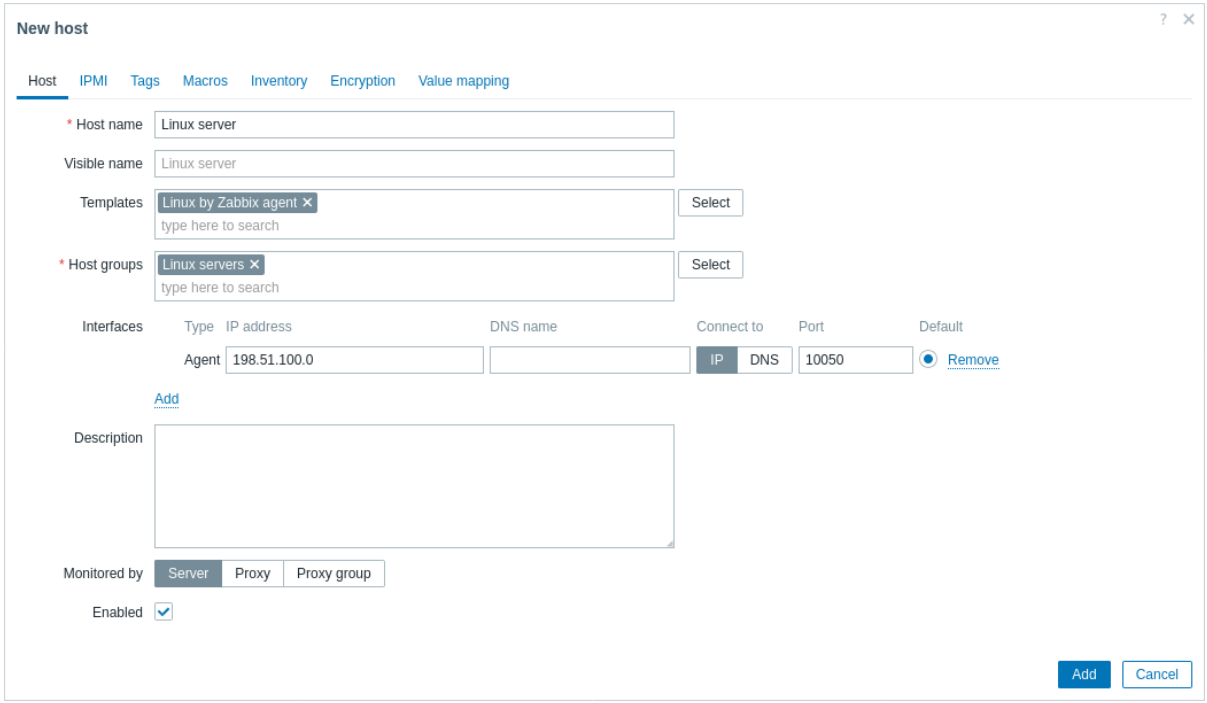
```
systemctl restart zabbix-agent
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”Linux serveris”).
- Laukā *Templates* ierakstiet vai atlasiet veidni ”Linux by Zabbix aģents”, kas tiks saistīta ar hostu.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”Linux serveri”).
- Parametrā *Interfaces* pievienojiet *Agent* saskarni un norādiet tās Linux mašīnas IP adresi vai DNS nosaukumu, kurā ir instalēts aģents.



3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraugāmo Linux mašīnu.

Aktīvās pārbaudes Zabbix aģents

1. Atveriet aģenta konfigurācijas failu datorā, kurā aģents ir instalēts.

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

2. Pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligāto portu) parametram ServerActive. Piemēram:

```
ServerActive=192.0.2.0:10051
```

Zabbix aģents izmantos šo adresi, lai izveidotu savienojumu ar Zabbix servera trapper portu (noklusējums: 10051) un pieprasītu aktīvo pārbaužu konfigurācijas datus.

3. Definējiet parametru Hostname, kam jāsakrīt ar hosta nosaukumu, kas tiks definēts Zabbix lietotāja saskarnē. Mūsu piemērā tas ir:

```
Hostname=Linux server
```

Hostname vērtībai jāsakrīt, jo aktīvajām pārbaudēm Zabbix aģents to izmanto, lai no servera saņemtu pareizo hosta konfigurāciju. Precīzāk, aģents iniciē savienojumu ar serveri un identificē sevi, izmantojot Hostname vērtību. Pēc tam serveris nodrošina šī hosta uzraudzības konfigurāciju. Ja šīs vērtības atšķiras, aģents nesaņems atbilstošo konfigurāciju, kā rezultātā trūks metriku vai radīsies uzraudzības problēmas.

4. Restartējiet Zabbix aģentu.

```
systemctl restart zabbix-agent
```

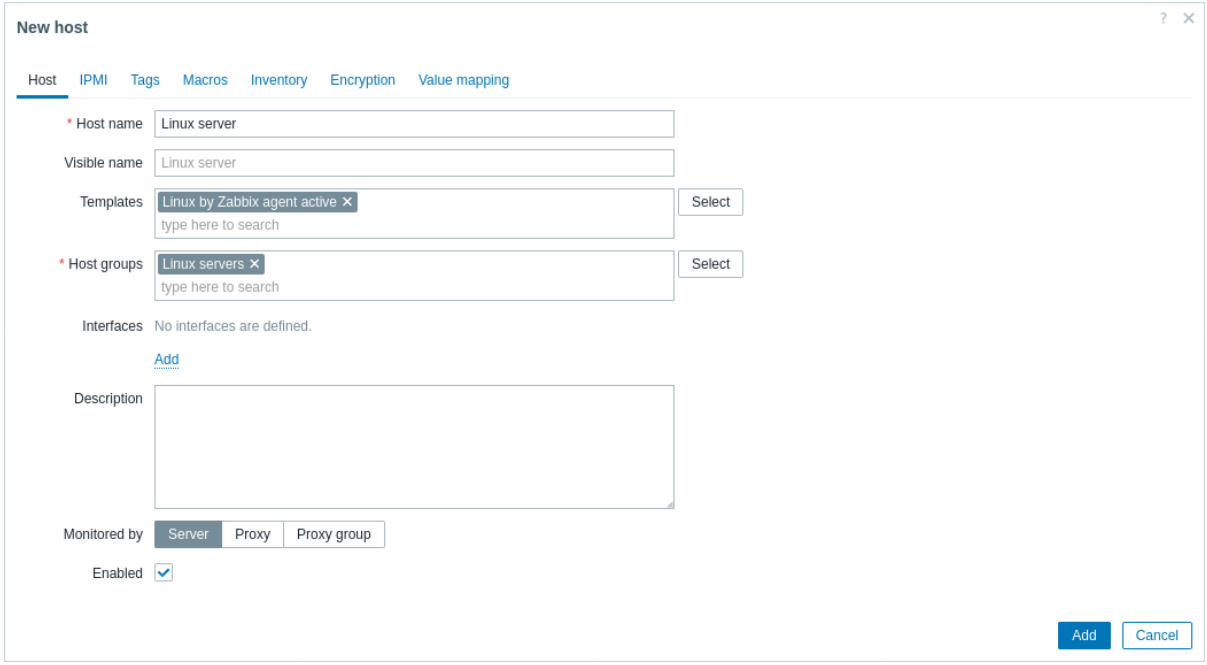
Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”Linux server”), kas atbilst iepriekš aģenta konfigurācijas failā definētajai parametra *Hostname* vērtībai.

- Laukā *Templates* ierakstiet vai atlasiet veidni ”Linux by Zabbix agent active”, kas tiks saistīta ar hostu.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”Linux servers”).
- Atstājiet *Interfaces* nedefinētu; saskarne nav nepieciešama aktīvajām pārbaudēm, jo aģents pats iniciē savienojumu ar serveri, nevis gaida savienojumu no servera.

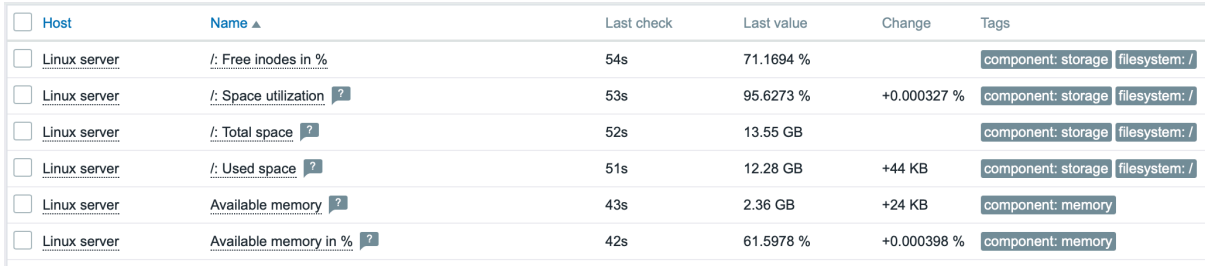


3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraugāmo Linux datoru.

Skatīt savāktos metriku datus Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Linux datoru.

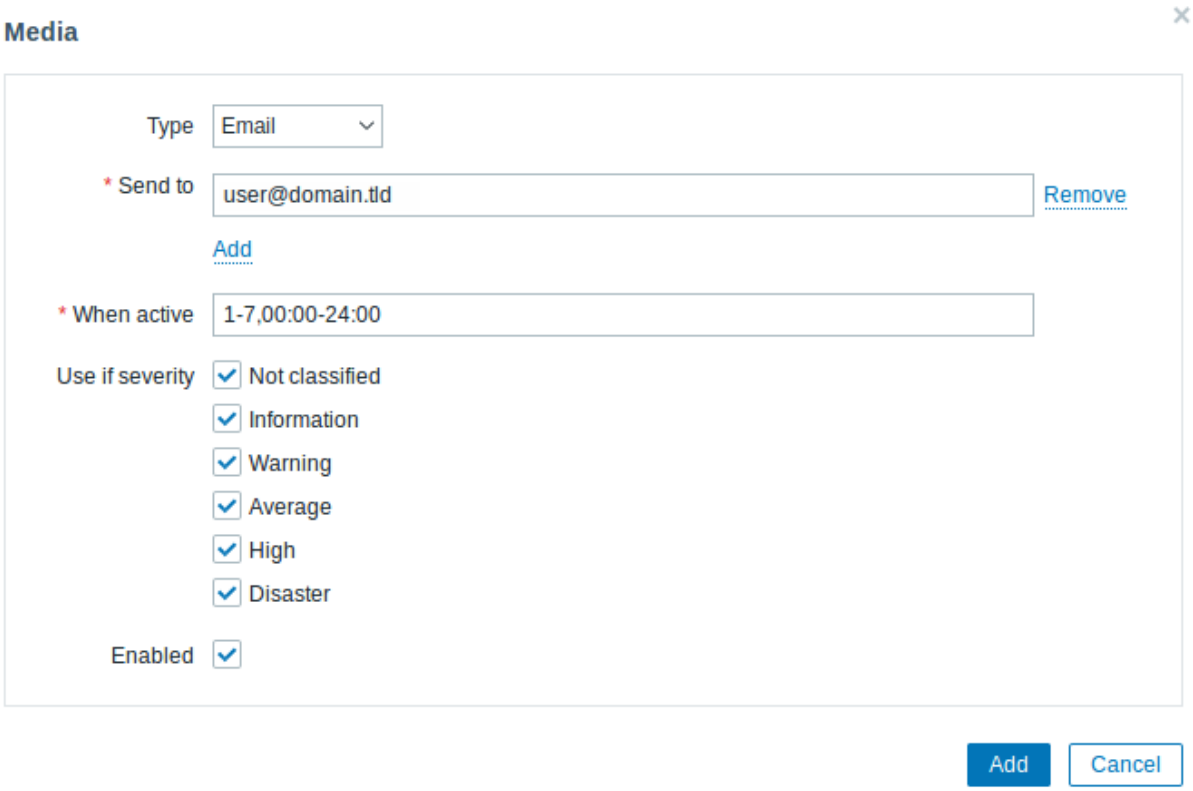
Lai skatītu savāktos metriku datus, atveriet *Monitoring->Hosts* izvēlnes sadaļu un noklikšķiniet uz *Latest data* blakus hostam.

Šī darbība atvērs visu jaunāko no Linux servera hosta savākto metriku sarakstu.



Iestatiet problēmu brīdinājumus Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti konfigurēšanas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi -> Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

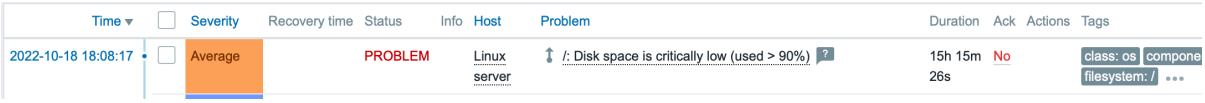
Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju Linux vidē varat simulēt augstu CPU noslodzi un rezultātā saņemt problēmas brīdinājumu, izpildot:

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

Iespējams, būs jāpalaiž vairāki md5sum procesi, lai CPU noslodze pārsniegtu slieksni.

Kad Zabbix konstatēs problēmu, tā parādīsies sadaļā Monitoring->Problems.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī paziņojumu par problēmu.

Skatiet arī:

- Vienuma izveide - kā sākt uzraudzīt papildu metriku (pielāgota uzraudzība bez veidnēm).
- Zabbix aģenta vienumi, Zabbix aģenta vienumi operētājsistēmai Windows - pilns metrikas saraksts, ko varat uzraudzīt, izmantojot Zabbix aģentu operētājsistēmā Windows.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).
- Instalēšana no pakotnēm - kā instalēt Zabbix komponentus, izmantojot oficiālās RPM un DEB pakotnes dažādām Linux distribūcijām, nodrošinot piekļuvi jaunākajām iespējām un kļūdu labojumiem.

2 Windows uzraudzība ar Zabbix aģentu

Ievads Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu Windows datoru pamata uzraudzību ar Zabbix.

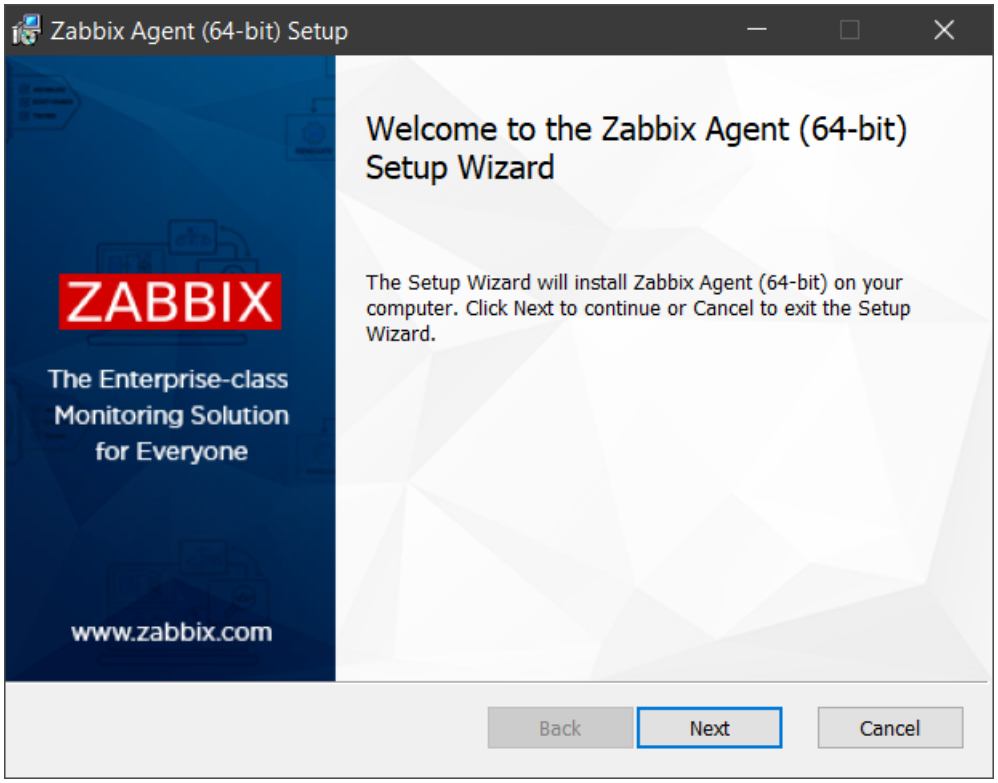
Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu jūsu Windows datora pamata uzraudzību. Ja meklējat plašākas pielāgošanas iespējas vai jums ir nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo instalēšanas rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu OS paredzētajām instrukcijām.

Instalēt Zabbix aģentu Zabbix aģents ir process, kas atbild par datu vākšanu. Tas ir jāinstalē Windows datorā, kuru vēlaties uzraudzīt. Sekojiet Zabbix aģenta instalēšanas norādījumiem Windows.



Konfigurējiet Zabbix uzraudzībai Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus).

Note:
Pasīvā pārbaude ir vienkāršs datu pieprasījums. Zabbix serveris vai starpniekserveris pieprasa dažus datus (piemēram, CPU slodzi), un Zabbix aģents nosūta rezultātu atpakaļ serverim. Aktīvajām pārbaudēm ir nepieciešama sarežģītāka apstrāde. Aģentam vispirms jāiegūst no servera(-iem) vienumu saraksts neatkarīgai apstrādei un pēc tam dati paketēs jānosūta atpakaļ. Plašāku informāciju skatiet sadaļā Passive and active agent checks.

Zabbix nodrošinātās uzraudzības veidnes parasti piedāvā divas alternatīvas — veidni Zabbix aģentam un veidni Zabbix aģentam (active). Izmantojot pirmo opciju, aģents vāks metriku pasīvajā režīmā. Šādas veidnes nodrošinās identiskus uzraudzības rezultātus, taču izmantos atšķirīgus saziņas protokolus.

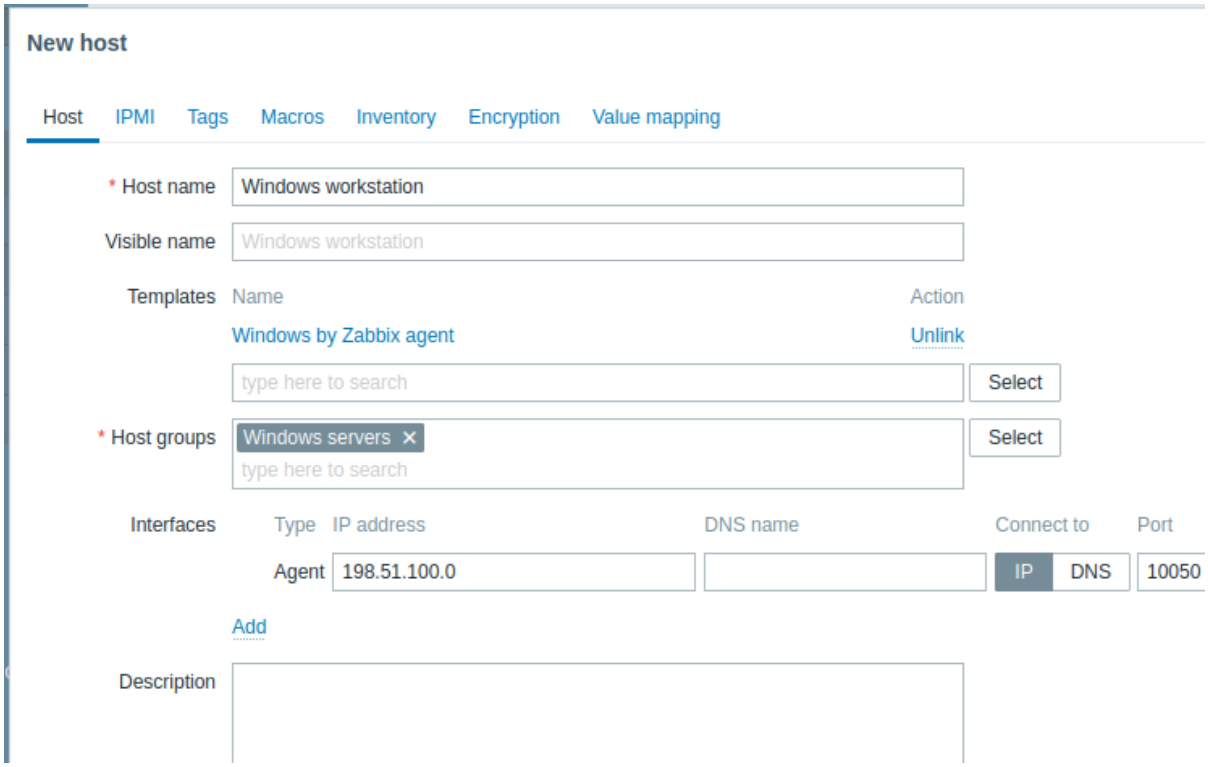
Turpmākā Zabbix konfigurācija ir atkarīga no tā, vai izvēlaties veidni active vai passive Zabbix aģenta pārbaudēm.

Pasīvās pārbaudes Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.
2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

Šis hosts attēlos jūsu Windows datoru.

3. Parametrā *Interfaces* pievienojiet *Aģenta* saskarni un norādiet tā Windows datora IP adresi vai DNS nosaukumu, kurā ir instalēts aģents.
4. Parametrā *Templates* ierakstiet vai atlasiet *Windows by Zabbix agent*.



Zabbix aģents

Pasīvajām pārbaudēm Zabbix aģentam ir jāzina Zabbix servera IP adrese vai DNS nosaukums. Ja aģenta instalēšanas laikā esat norādījis pareizu informāciju, konfigurācijas fails jau ir atjaunināts. Pretējā gadījumā tā ir jānorāda manuāli. Atveriet mapi `C:\Program files\Zabbix Agent`, atveriet failu *zabbix_agentd.conf* un pievienojiet sava Zabbix servera IP/DNS parametram *Server*.

Piemērs:

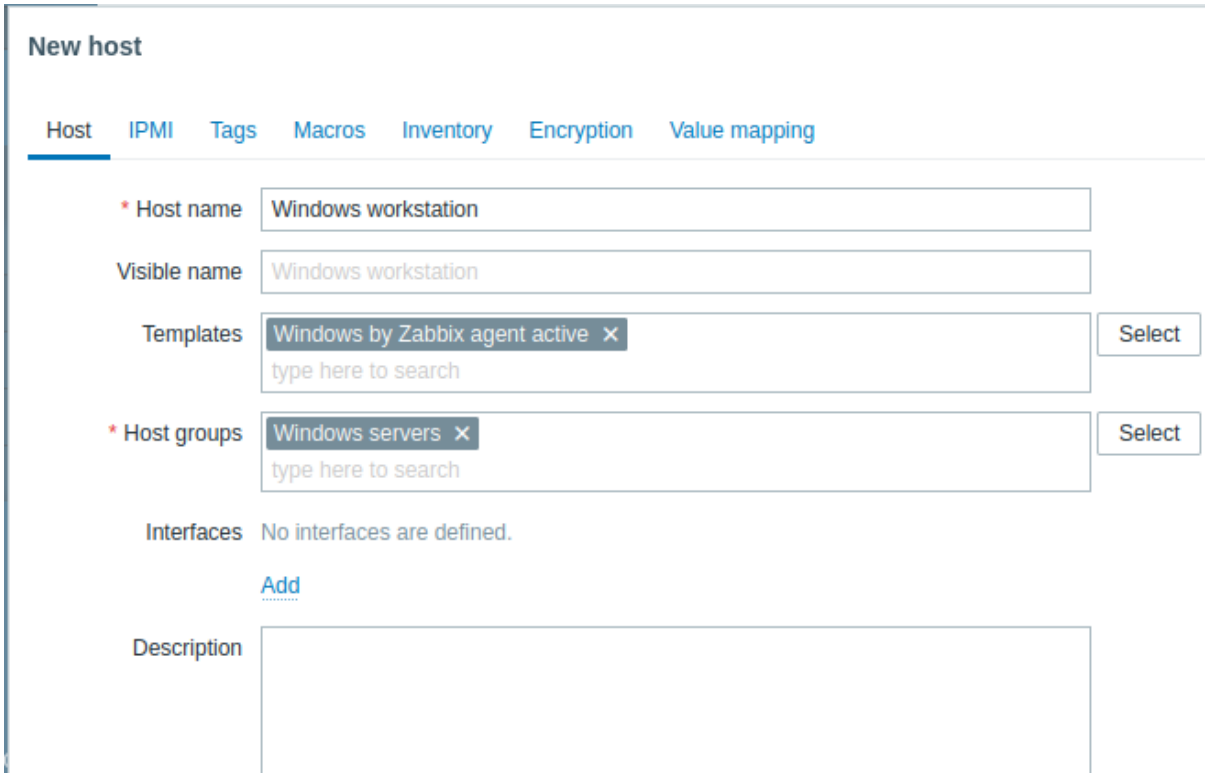
```
Server=192.0.2.22
```

Aktīvās pārbaudes Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.
2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

Šis hosts attēlos jūsu Windows datoru.

3. Parametrā *Templates* ierakstiet vai atlasiet *Windows by Zabbix agent active*.



Zabbix aģents

Mapē `C:\Program files\Zabbix Agent` atveriet failu *zabbix_agentd.conf* un pievienojiet:

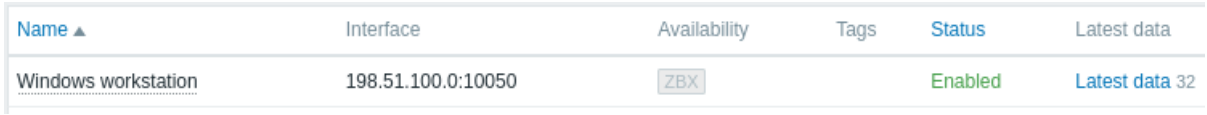
- Jūsu Zabbix tīmekļa saskarnē izveidotā hosta nosaukumu parametram *Hostname*.
- Jūsu Zabbix servera datora IP adresi vai DNS nosaukumu parametram *ServerActive* (tas var būt jau aizpildīts, ja to norādījāt Zabbix aģenta iestatīšanas laikā).

Piemērs:

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

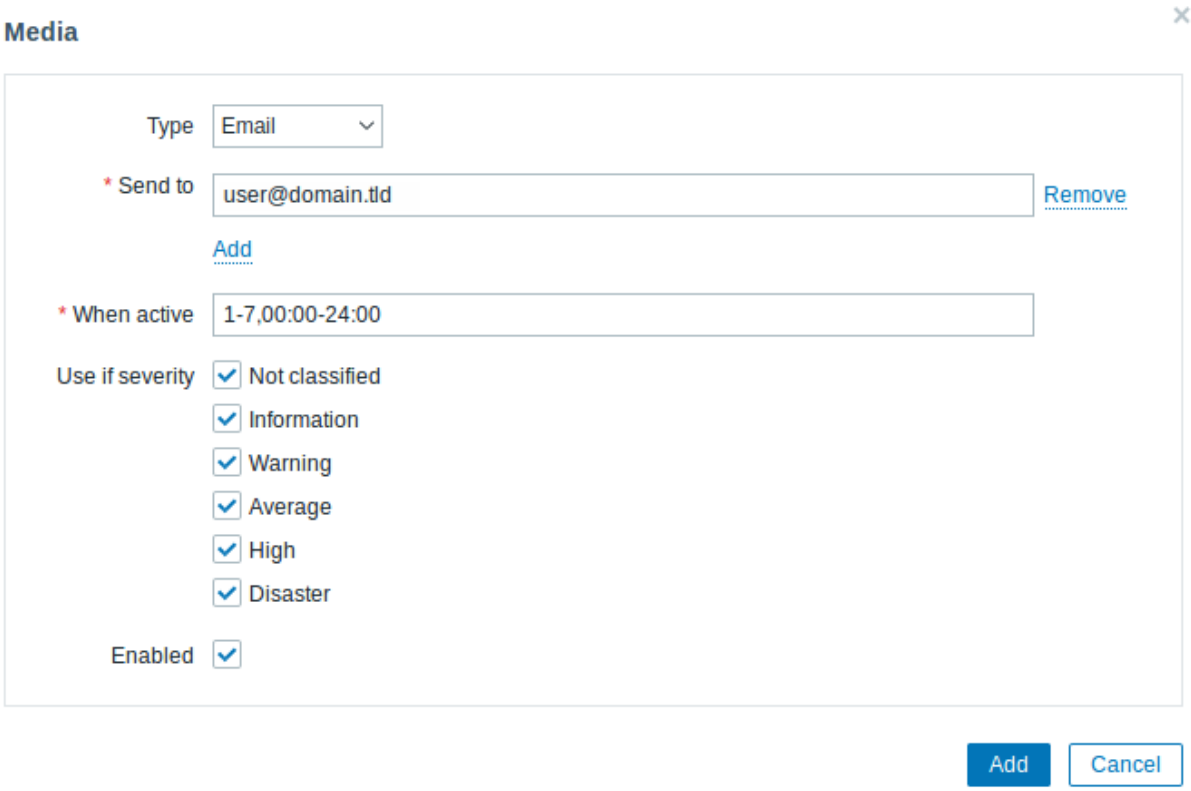
Skatīt savāktos metriku datus Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Windows datoru.

Lai skatītu savāktos metriku datus, atveriet *Monitoring->Hosts* izvēlnes sadaļu un noklikšķiniet uz *Latest data* blakus hostam.



Problēmu brīdinājumu iestatīšana Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti konfigurēšanas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi -> Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums būtu jāsaņem brīdinājums pa e-pastu.

Note:
Sistēmā Windows varat izmantot utilītu CpuStres, lai simulētu augstu CPU noslodzi un rezultātā saņemtu problēmas brīdinājumu.

Skatiet arī:

- Vienuma izveide - kā sākt uzraudzīt papildu metriku (pielāgota uzraudzība bez veidnēm).
- Zabbix aģenta vienumi, Zabbix aģenta vienumi operētājsistēmai Windows - pilns metriku saraksts, ko varat uzraudzīt, izmantojot Zabbix aģentu sistēmā Windows.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

3. Monitor Apache via HTTP

Ievads Šajā lapā ir parādīts ātrs un vienkāršs veids, kā sākt Apache tīmekļa servera uzraudzību, neinstalējot nekādu papildu programmatūru.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu jūsu Apache instalācijas pamata uzraudzību. Ja meklējat padziļinātas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo instalēšanas ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu OS paredzētajām instrukcijām.

Sagatavojiet Apache 1. Pārbaudiet, kuru Apache versiju jūs izmantojat:

RHEL bāzētā sistēmā izpildiet:

```
httpd -v
```

Debian/Ubuntu sistēmā izpildiet:

```
apache2 -v
```

2. Pārliecinieties, ka jūsu Apache instancē ir iespējots Status modulis.

RHEL bāzētā sistēmā izpildiet:

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

Debian/Ubuntu sistēmā izpildiet:

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

Ja sarakstā neredzat status_module, iespējojiet moduli, izpildot:

RHEL bāzētā sistēmā izpildiet:

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

Debian/Ubuntu sistēmā izpildiet:

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

3. Rediģējiet Apache konfigurācijas failu, lai atļautu piekļuvi statusa atskaitēm no Zabbix servera IP adreses.

RHEL bāzētā sistēmā: `/etc/httpd/conf.modules.d/status.conf`:

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

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

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

Pievienojiet failam šādas rindas (**aizstājiet 198.51.100.255** ar sava Zabbix servera IP adresi):

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

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

4. Restartējiet Apache

RHEL bāzētā sistēmā izpildiet:

```
sudo systemctl restart httpd
```

Debian/Ubuntu sistēmā izpildiet:

```
sudo systemctl restart apache2
```

5. Lai pārbaudītu, vai viss ir konfigurēts pareizi, izpildiet (**aizstājiet 198.51.100.255** ar sava Zabbix servera IP adresi):

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

Atbildē jābūt ietvertai Apache tīmekļa servera statistikai.

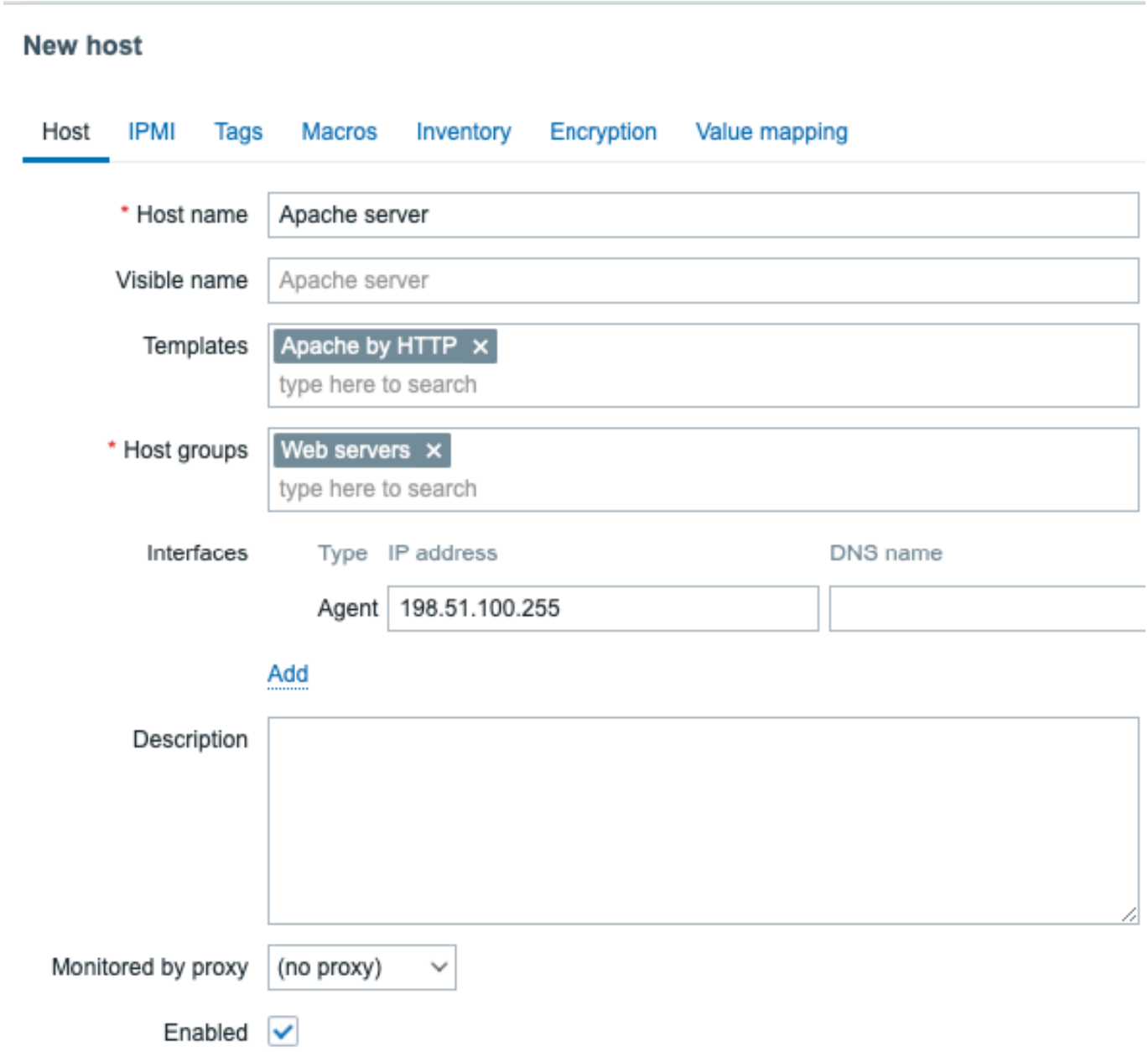
Konfigurējiet Zabbix uzraudzībai 1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

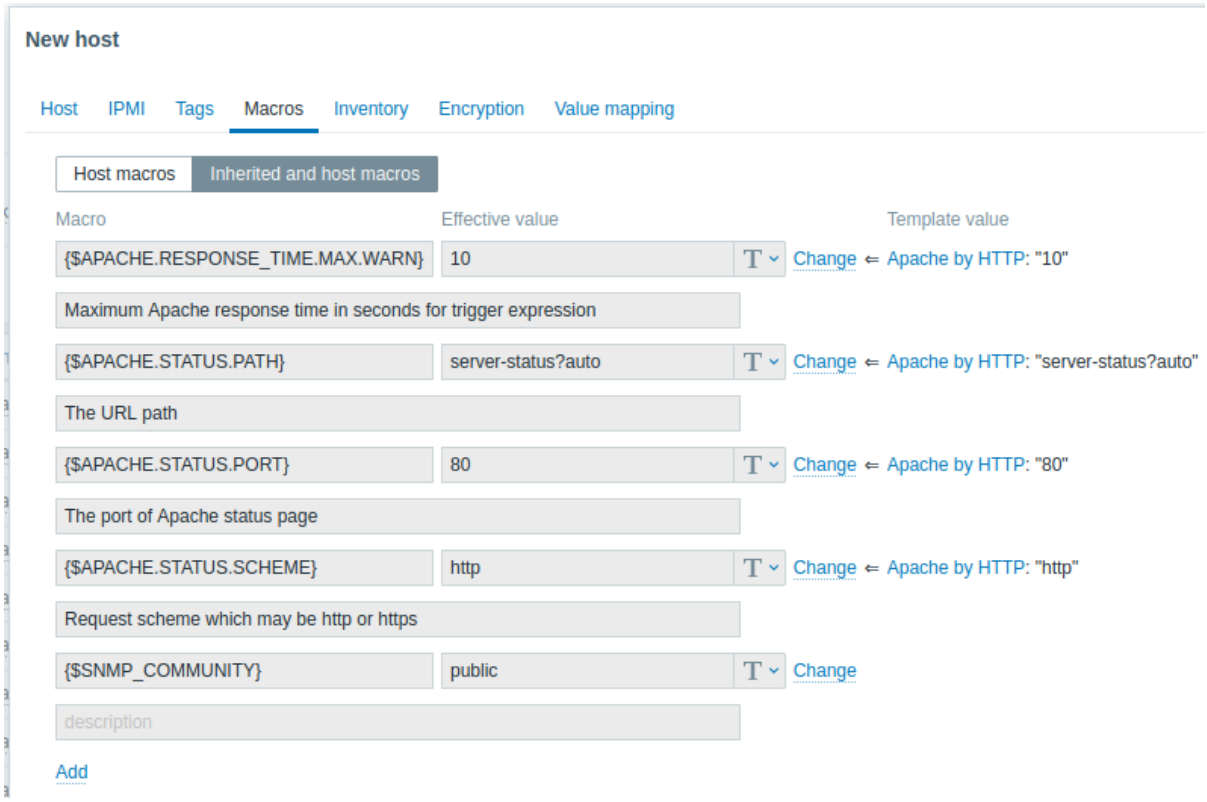
Šis hosts attēlos jūsu Apache serveri.

3. Parametrā *Interfaces* pievienojiet *Agent* saskarni un norādiet savas Apache instances IP adresi. **Jums nav nepieciešams instalēt Zabbix aģentu šajā datorā**, saskarne tiks izmantota tikai {HOST.CONN} makrosa atrisināšanai. Šis makross tiek izmantots veidnes vienumos, lai noteiktu Apache instances atrašanās vietu.

4. Parametrā *Templates* ierakstiet vai atlasiet *Apache by HTTP*.



5. Pārslēdzieties uz cilni **Macros** un atlasiet režīmu *Inherited and host macros*. Pārbaudiet, vai makrosu {$APACHE.STATUS.PORT} un {$APACHE.STATUS.SCHEME} vērtības atbilst jūsu instalācijas iestatījumiem. Pēc noklusējuma ports ir 80 un shēma ir http. Mainiet makrosu vērtības, ja izmantojat citu portu un/vai shēmu.

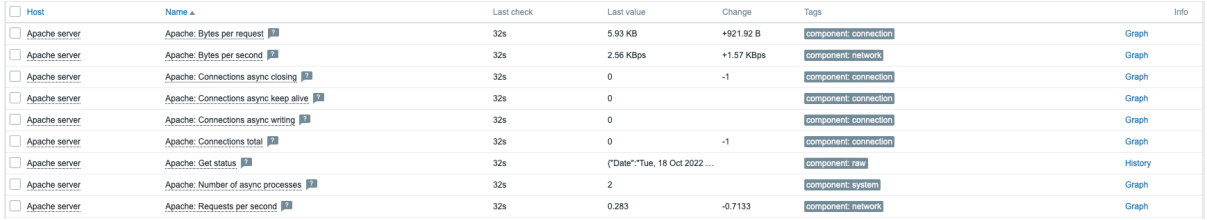


Skatīt savāktos metriku datus Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu Apache tīmekļa serveri.

Lai skatītu savāktos metriku datus, atveriet *Monitoring->Hosts* izvēlnes sadaļu un noklikšķiniet uz *Dashboards* blakus hostam.

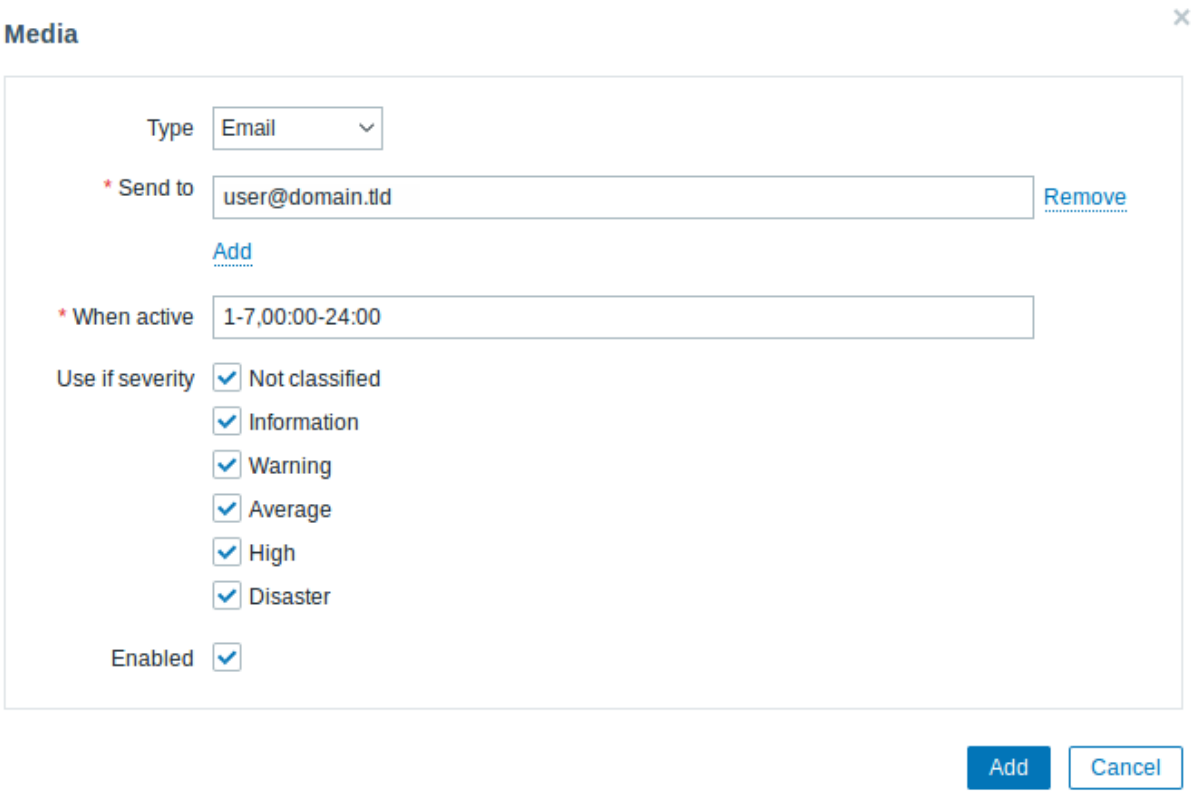
Šī darbība jūs novirzīs uz hosta paneli ar svarīgākajiem metriku datiem, kas savākti no Apache /server-status lapas.

Alternatīvi, sadaļā *Monitoring->Hosts* varat noklikšķināt uz *Latest data*, lai sarakstā skatītu visus jaunākos savāktos metriku datus.



Problēmu brīdinājumu iestatīšana Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti konfigurēšanas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi* -> *Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.

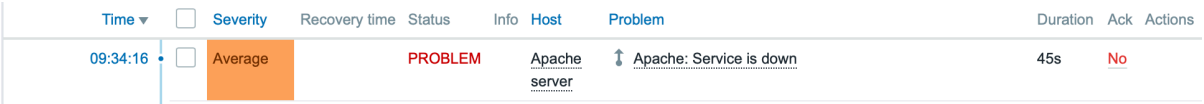


2. Izpildiet ceļvedī Problēmas paziņojuma saņemšana aprakstītās darbības.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju Lai simulētu reālu problēmu un saņemtu testa problēmas brīdinājumu:

1. Atveriet *Apache servera* hosta konfigurāciju Zabbix.
2. Pārslēdzieties uz cilni Makrosi un atlasiet *Mantotie un hosta makrosi*.
3. Nospiediet *Mainīt* blakus makrosam {$APACHE.STATUS.PORT} un iestatiet citu portu.
4. Nospiediet Update, lai saglabātu hosta konfigurāciju.
5. Pēc dažām minūtēm Zabbix noteiks problēmu *Apache service is down*, jo tagad tas nevar izveidot savienojumu ar instanci. Tā parādīsies sadaļā Uzraudzība->Problēmas.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī problēmas paziņojumu.

6. Mainiet makrosa vērtību atpakaļ, lai novērstu problēmu un turpinātu Apache uzraudzību.

Skatiet arī:

- Tīmekļa servera drošības stiprināšana - ieteicamie iestatījumi lielākai tīmekļa servera drošībai.
- Vienuma izveide - kā sākt uzraudzīt papildu metriku.
- HTTP vienumi - kā uzraudzīt pielāgotu metriku, izmantojot HTTP aģentu.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

4. Monitor MySQL with Zabbix agent 2

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu MySQL servera pamata uzraudzību.

Lai uzraudzītu MySQL serveri, ir pieejamas vairākas pieejas: Zabbix aģents, Zabbix aģents 2 vai Open Database Connectivity (ODBC) standarts. Šīs rokasgrāmatas galvenais uzsvars ir uz MySQL servera uzraudzību ar Zabbix aģentu 2, kas ir **ieteicamā** pieeja, pateicoties tās vienkāršajai konfigurēšanai dažādās vidēs. Tomēr šajā lapā ir sniegtas arī instrukcijas par citām pieejām, tāpēc droši izvēlieties to, kas vislabāk atbilst jūsu prasībām.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu MySQL servera pamata uzraudzību. Ja meklējat padziļinātas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt ar šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents 2 saskaņā ar jūsu OS paredzētajām instrukcijām.

Atkarībā no jūsu vides daži šajā ceļvedī aprakstītie soļi var nedaudz atšķirties. Šis ceļvedis ir balstīts uz vidi, kurā darbojas Ubuntu.

Izveidojiet MySQL lietotāju

Lai uzraudzītu MySQL serveri, Zabbix ir nepieciešama piekļuve tam un tā procesiem. Jūsu MySQL instalācijā jau ir lietotājs ar nepieciešamo piekļuves līmeni (lietotājs "zabbix", kas tika izveidots Zabbix instalēšanas laikā), tomēr šim lietotājam ir vairāk privilēģiju, nekā nepieciešams vienkāršai uzraudzībai (privilēģijas DROP datubāzēm, DELETE ierakstu dzēšanai no tabulām u.c.). Tāpēc ir jāizveido MySQL lietotājs, kas paredzēts *tikai* MySQL servera uzraudzībai.

1. Pieslēdzieties MySQL klientam, izveidojiet lietotāju "zbx_monitor" (aizstājiet <*password*> lietotājam "zbx_monitor" ar pašu izvēlētu paroli) un piešķiriet lietotājam nepieciešamās GRANT privilēģijas:

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

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

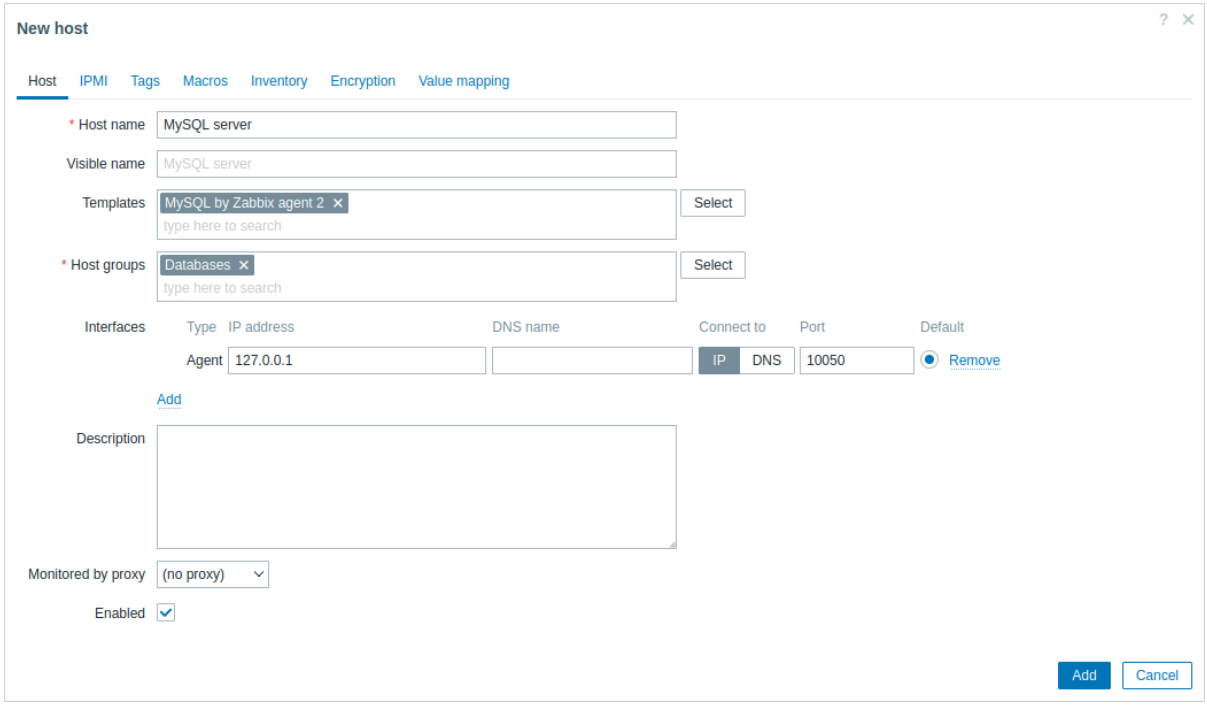
Kad lietotājs ir izveidots, varat pāriet pie nākamā soļa.

Konfigurējiet Zabbix lietotāja saskarni

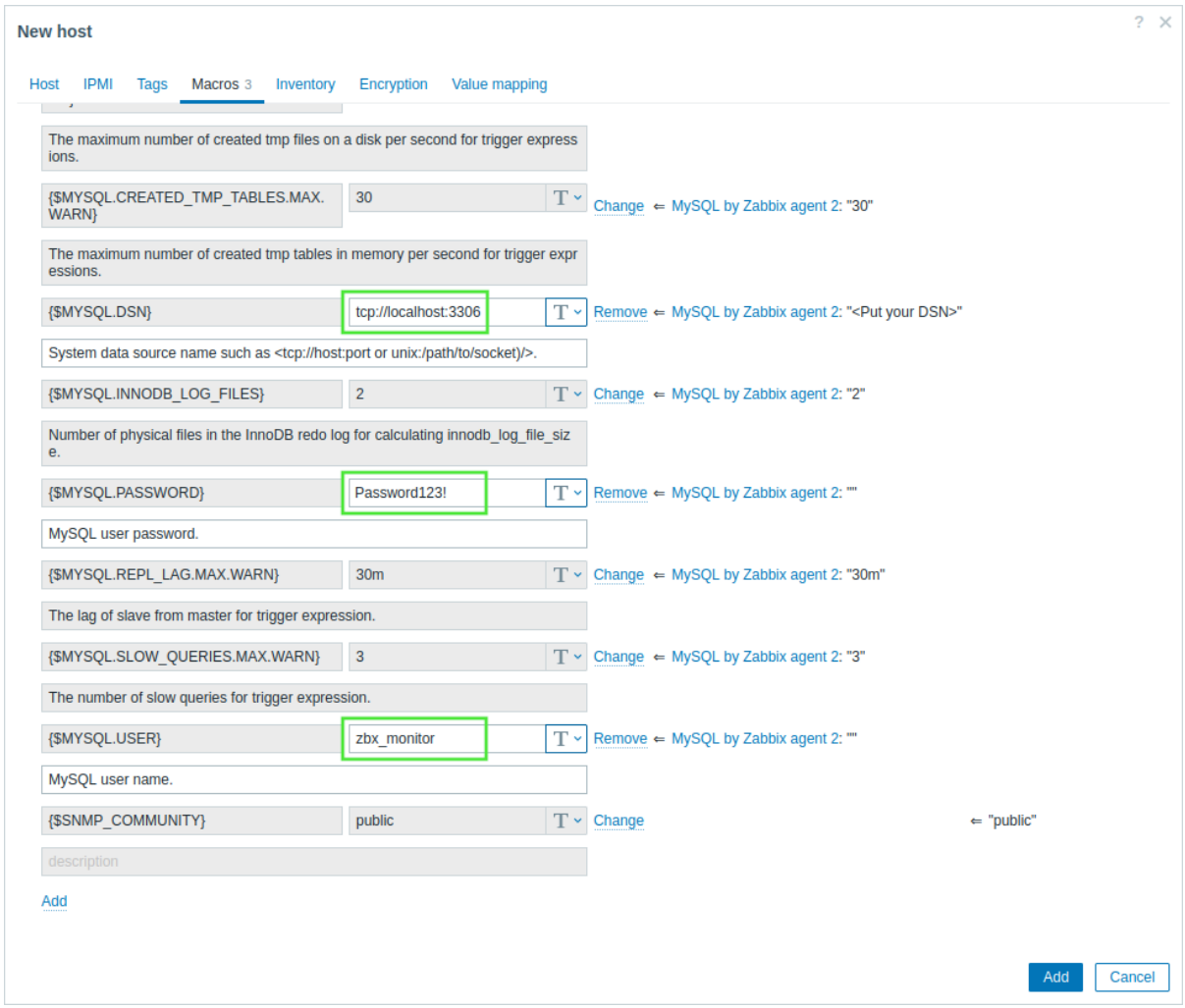
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, "MySQL server").
- Laukā *Templates* ierakstiet vai atlasiet veidni "MySQL by Zabbix aģents 2", kas tiks saistīta ar hostu.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, "Databases").
- Laukā *Interfaces* pievienojiet saskarni ar tipu "Agent" un norādiet sava MySQL servera IP adresi. Šajā ceļvedī tiek izmantots "127.0.0.1" (localhost), lai uzraudzītu MySQL serveri, kas ir instalēts tajā pašā datorā, kur Zabbix serveris un Zabbix aģents 2.



- Cilnē *Macros* pārslēdzieties uz *Inherited and host macros*, atrodiet šādus makrosus un noklikšķiniet uz *Change* blakus makrosa vērtībai, lai to atjauninātu:
  - {$MYSQL.DSN} - iestatiet MySQL servera datu avotu (MySQL Zabbix aģenta 2 spraudņa konfigurācijas faila nosauktās sesijas savienojuma virkni). Šajā ceļvedī tiek izmantots noklusējuma datu avots "tcp://localhost:3306", lai uzraudzītu MySQL serveri, kas ir instalēts tajā pašā datorā, kur Zabbix serveris un Zabbix aģents 2.
  - {$MYSQL.PASSWORD} - iestatiet iepriekš izveidotā MySQL lietotāja "zbx_monitor" paroli.
  - {$MYSQL.USER} - iestatiet iepriekš izveidotā MySQL lietotāja "zbx_monitor" vārdu.

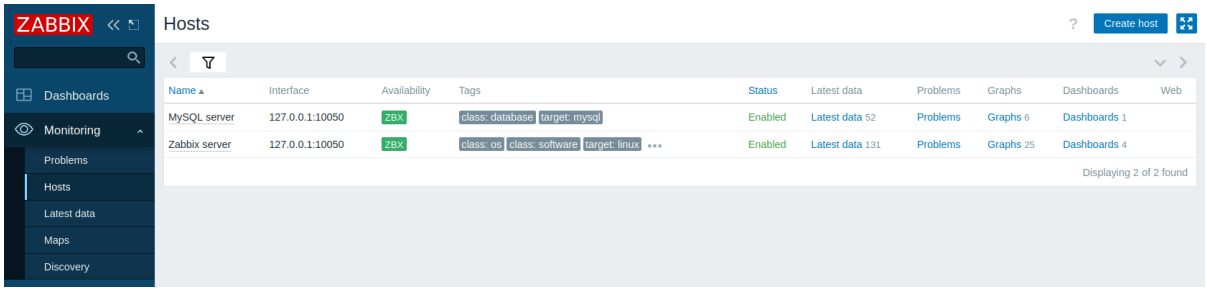


3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos jūsu MySQL serveri.

Skatīt savāktos metriku datus

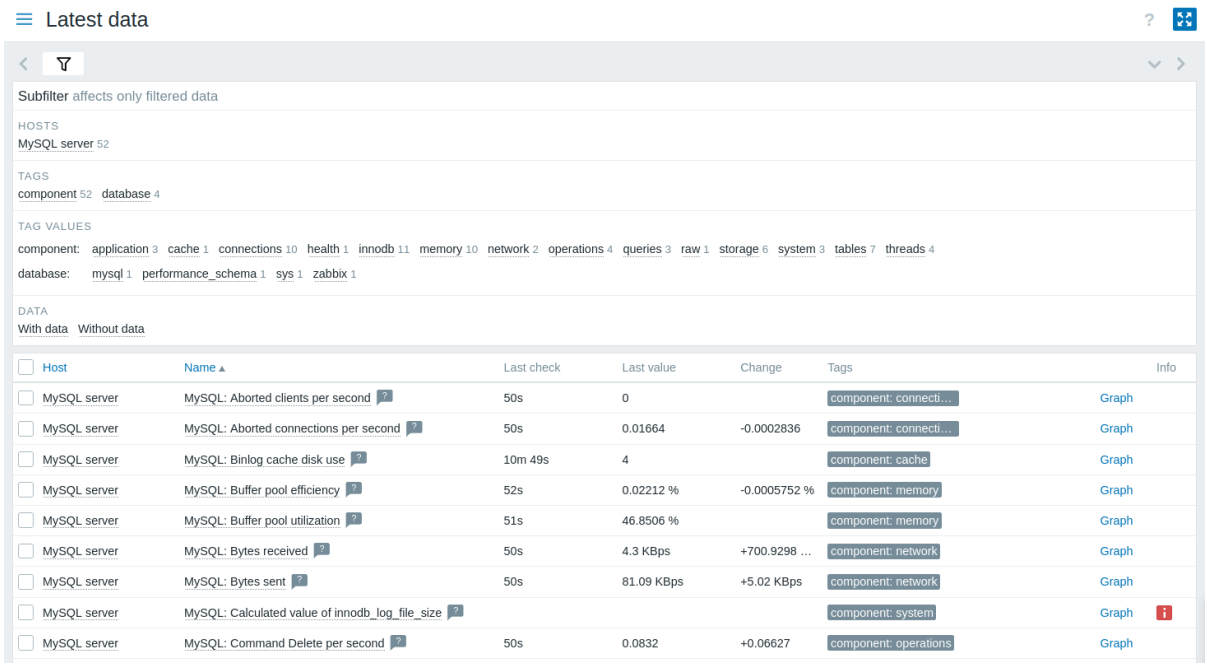
Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu MySQL serveri.

Lai skatītu savāktos metriku datus, dodieties uz izvēlnes sadaļu *Uzraudzība → Hosti* un noklikšķiniet uz *Paneļi* blakus hostam.



Šī darbība jūs novirzīs uz hosta paneli (konfigurētu veidnes līmenī), kurā redzami svarīgākie no MySQL servera savāktie metriku dati.

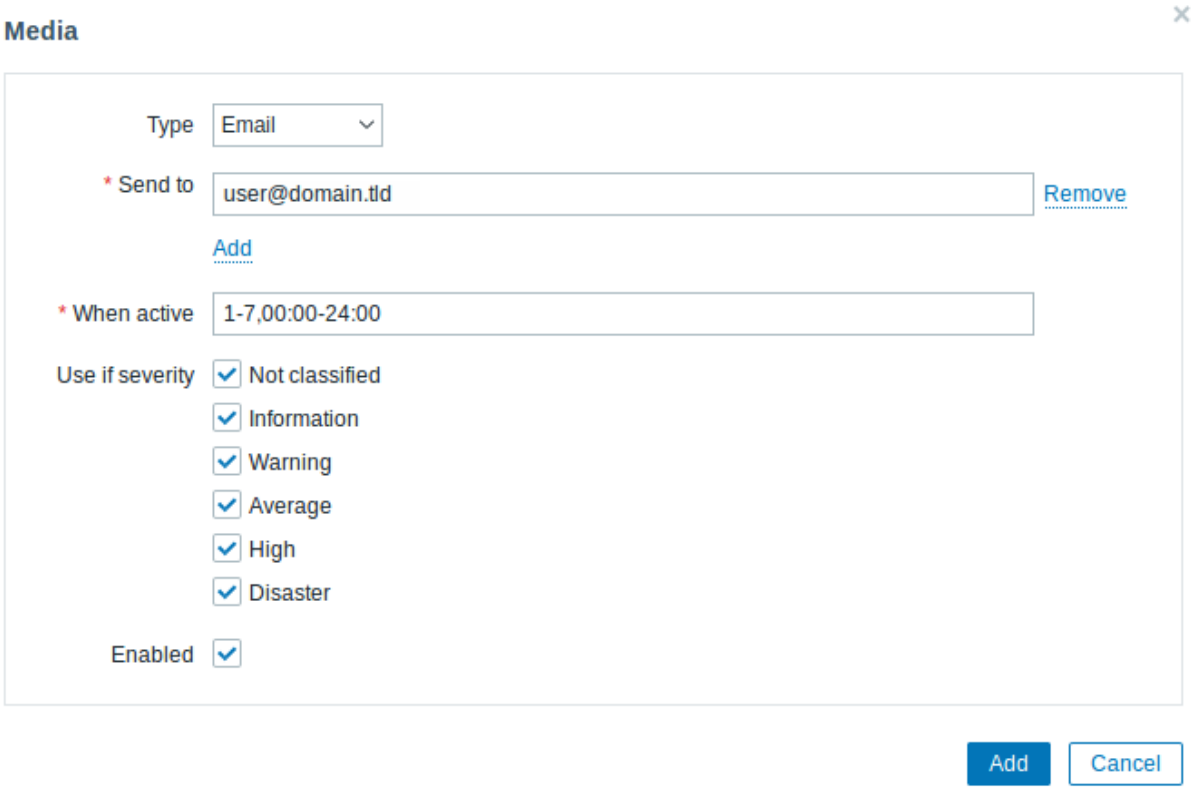
Alternatīvi, izvēlnes sadaļā *Uzraudzība → Hosti* varat noklikšķināt uz *Jaunākie dati*, lai sarakstā skatītu visus jaunākos savāktos metriku datus. Ņemiet vērā, ka vienumam *MySQL: Calculated value of innodb_log_file_size* nav paredzami dati, jo vērtība tiks aprēķināta no pēdējās stundas datiem.



Problēmu brīdinājumu iestatīšana

Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi → Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, mēs varam simulēt reālu problēmu, atjauninot hosta konfigurāciju Zabbix lietotāja saskarnē.

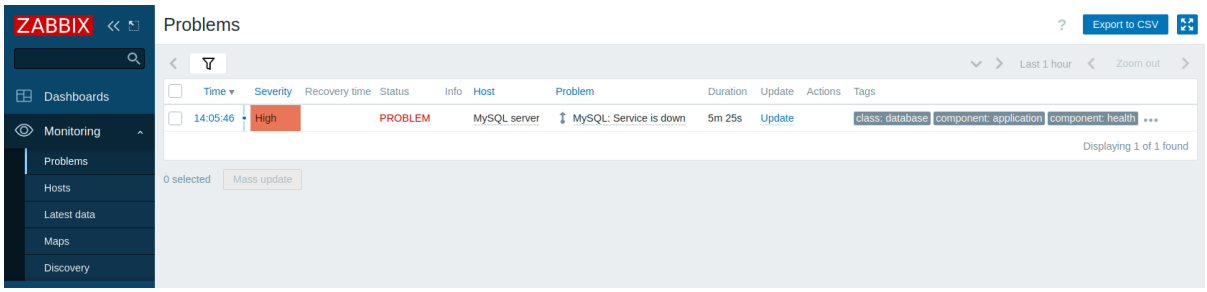
1. Atveriet sava MySQL servera hosta konfigurāciju Zabbix.

2. Pārslēdzieties uz cilni *Macros* un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus, piemēram, iepriekš konfigurētajai {$MYSQL.USER} makro vērtībai un iestatiet citu MySQL lietotājvārdu.

4. Noklikšķiniet uz *Update*, lai atjauninātu hosta konfigurāciju.

5. Pēc dažiem mirkļiem Zabbix noteiks problēmu "MySQL: Service is down", jo tas nevarēs izveidot savienojumu ar MySQL serveri. Problēma parādīsies sadaļā *Monitoring → Problems*.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī problēmas paziņojumu.

6. Mainiet makro vērtību atpakaļ uz iepriekšējo vērtību, lai novērstu problēmu un turpinātu MySQL servera uzraudzību.

Citas pieejas MySQL uzraudzībai

Tā vietā, lai uzraudzītu MySQL serveri ar Zabbix aģents 2, varat izmantot arī Zabbix aģents vai Open Database Connectivity (ODBC) standartu. Lai gan ir ieteicams izmantot Zabbix aģents 2, var būt konfigurācijas, kas neatbalsta Zabbix aģents 2 vai kurām nepieciešama pielāgota pieeja.

Galvenā atšķirība starp Zabbix aģents un ODBC ir datu ievākšanas metodē — Zabbix aģents tiek instalēts tieši MySQL serverī un ievāc datus, izmantojot tā iebūvēto funkcionalitāti, savukārt ODBC izmanto ODBC draiveri, lai izveidotu savienojumu ar MySQL serveri un iegūtu datus, izmantojot SQL vaicājumus.

Lai gan daudzi konfigurācijas soļi ir līdzīgi MySQL servera uzraudzībai ar Zabbix aģents 2, pastāv dažas būtiskas atšķirības — jums jākonfigurē Zabbix aģents vai ODBC, lai varētu uzraudzīt MySQL serveri. Tālāk sniegtās instrukcijas izskaidro šīs **atšķirības**.

MySQL uzraudzība ar Zabbix aģentu

Lai uzraudzītu MySQL serveri ar Zabbix aģentu, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents saskaņā ar jūsu OS paredzētajām instrukcijām.

Kad esat veiksmīgi instalējis nepieciešamos Zabbix komponentus, jums ir jāizveido MySQL lietotājs, kā aprakstīts sadaļā *MySQL lietotāja izveide*.

Pēc MySQL lietotāja izveides jums ir jākonfigurē Zabbix aģents, lai tas varētu izveidot savienojumu ar MySQL serveri un to uzraudzīt. Tas ietver vairāku lietotāja parametru konfigurēšanu pielāgotu aģenta pārbaužu izpildei, kā arī nepieciešamo akreditācijas datu nodrošināšanu Zabbix aģentam, lai tas varētu izveidot savienojumu ar MySQL serveri, izmantojot iepriekš izveidoto lietotāju "zbx_monitor".

Konfigurējiet Zabbix aģentu

1. Pārejiet uz Zabbix aģenta papildu konfigurāciju direktoriju.

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

> **Attention:**
> Zabbix aģenta papildu konfigurāciju direktorijam jāatrodas tajā pašā direktorijā, kur atrodas jūsu Zabbix aģenta konfigurācijas fails (*zabbix_agentd.conf*). Atkarībā no jūsu OS un Zabbix instalācijas šai direktorijai var būt cita atrašanās vieta, nekā norādīts šajā rokasgrāmatā. Noklusējuma atrašanās vietas skatiet parametrā `Include` Zabbix aģenta konfigurācijas failā.

Tā vietā, lai visus MySQL servera uzraudzībai nepieciešamos lietotāja parametrus definētu Zabbix aģenta konfigurācijas failā, šie parametri tiks definēti atsevišķā failā papildu konfigurāciju direktorijā.

2. Izveidojiet failu *template_db_mysql.conf* Zabbix aģenta papildu konfigurāciju direktorijā.

```
vi template_db_mysql.conf
```

3. Nokopējiet saturu no faila *template_db_mysql.conf* (kas atrodas Zabbix repozitorijā) uz jūsu izveidoto failu *template_db_mysql.conf* un saglabājiet to.

4. Restartējiet Zabbix aģentu, lai atjauninātu tā konfigurāciju.

```
systemctl restart zabbix-agent
```

Kad esat konfigurējis Zabbix aģenta lietotāja parametrus, varat turpināt ar akreditācijas datu konfigurēšanu, kas ļaus Zabbix aģentam piekļūt MySQL serverim.

5. Pārejiet uz Zabbix aģents mājas direktoriju (ja tā jūsu sistēmā nepastāv, tā ir jāizveido; noklusējums: */var/lib/zabbix*).

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

6. Izveidojiet *.my.cnf* failu Zabbix aģents mājas direktorijā.

```
vi .my.cnf
```

7. Iekopējiet tālāk norādīto saturu failā *.my.cnf* (aizstājiet <*password*> ar lietotāja "zbx_monitor" paroli).

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

Konfigurējiet Zabbix lietotāja saskarni un pārbaudiet savu konfigurāciju

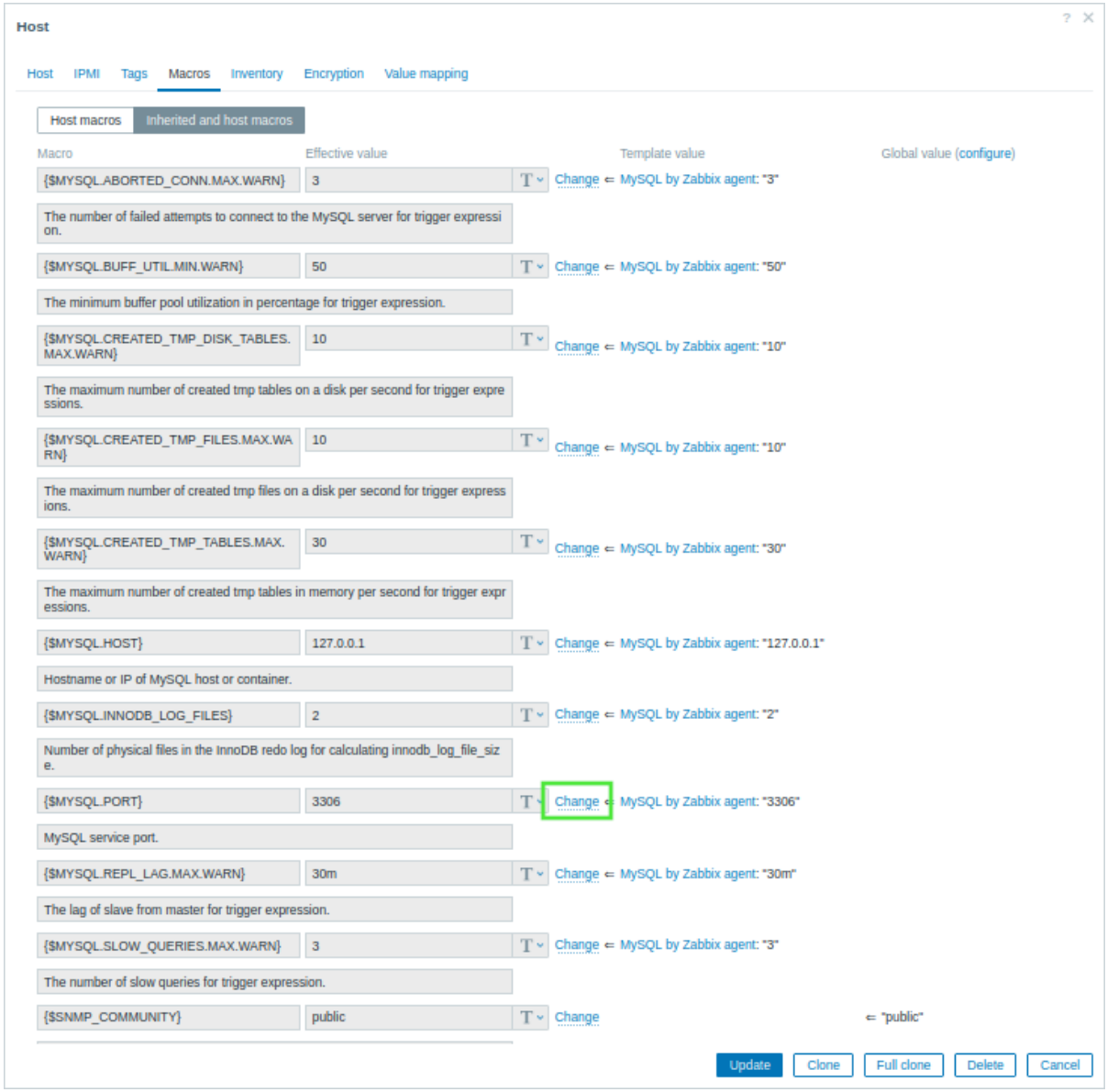
Lai konfigurētu Zabbix lietotāja saskarni, izpildiet norādījumus sadaļā *Configure Zabbix frontend*, veicot šādas izmaiņas:

- Laukā *Templates* ierakstiet vai atlasiet veidni "MySQL by Zabbix agent", kas tiks saistīta ar hostu.
- *Macros* konfigurēšana nav nepieciešama.

Kad esat konfigurējis Zabbix lietotāja saskarni, varat skatīt savāktos metriku datus un iestatīt problēmu brīdinājumus.

Lai pārbaudītu savu konfigurāciju, izpildiet norādījumus sadaļā *Test your configuration*, veicot šādas izmaiņas:

- MySQL servera hosta konfigurācijas sadaļā *Inherited and host macros* noklikšķiniet uz *Change* blakus makrosa {$MYSQL.PORT} vērtībai un iestatiet citu portu (piemēram, "6033").



MySQL uzraudzība ar ODBC

Lai uzraudzītu MySQL serveri ar ODBC, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne.

Kad esat veiksmīgi instalējis nepieciešamos Zabbix komponentus, jums ir jāizveido MySQL lietotājs, kā aprakstīts sadaļā *MySQL lietotāja izveide*.

Pēc MySQL lietotāja izveides jums ir jākonfigurē ODBC.
Tas ietver vienas no visbiežāk izmantotajām atvērtā pirmkoda ODBC API implementācijām — unixODBC — un unixODBC draivera instalēšanu, kā arī ODBC draivera konfigurācijas faila rediģēšanu.

ODBC konfigurēšana

1. Instalējiet unixODBC. Ieteicamais unixODBC instalēšanas veids ir izmantot Linux operētājsistēmas noklusējuma pakotņu repozitorijus.

```
apt install unixodbc
```

2. Instalējiet MariaDB unixODBC datubāzes draiveri. Lai gan jums ir MySQL datubāze, saderības apsvērumu dēļ tiek izmantots MariaDB unixODBC draiveris.

```
apt install odbc-mariadb
```

3. Pārbaudiet ODBC konfigurācijas failu *odbcinst.ini* un *odbc.ini* atrašanās vietu.

```
odbcinst -j
```

Šīs komandas izpildes rezultātam jābūt līdzīgam šādam.

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

4. Lai konfigurētu ODBC draiveri MySQL datubāzes uzraudzībai, jums ir nepieciešams draivera nosaukums, kas atrodas failā *odbcinst.ini*. Tālāk redzamajā *odbcinst.ini* faila piemērā draivera nosaukums ir "MariaDB Unicode".

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

5. Iekopējiet tālāk norādīto saturu failā *odbc.ini* (aizstājiet <*password*> ar lietotāja "zbx_monitor" paroli). Šajā rokasgrāmatā tiek izmantota "127.0.0.1" (localhost) kā MySQL servera adrese, lai uzraudzītu MySQL serveri, kas ir instalēts tajā pašā datorā, kur ODBC draiveris. Pievērsiet uzmanību datu avota nosaukumam (DSN) "test", kas būs nepieciešams, kad konfigurēsiet Zabbix lietotāja saskarne.

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

Konfigurējiet Zabbix lietotāja saskarni un pārbaudiet savu konfigurāciju

Lai konfigurētu Zabbix lietotāja saskarni, izpildiet norādījumus sadaļā *Konfigurēt Zabbix lietotāja saskarni*, veicot šādas izmaiņas:

- Laukā *Templates* ierakstiet vai atlasiet veidni "MySQL by ODBC", kas tiks saistīta ar hostu.
- *Interfaces* konfigurēšana nav nepieciešama.
- MySQL servera hosta konfigurācijas sadaļā *Inherited and host macros* makro {$MYSQL.DSN} vērtībai jābūt iestatītai uz DSN nosaukumu no *odbc.ini* faila.

Kad esat konfigurējis Zabbix lietotāja saskarni, varat skatīt savāktos metriku datus, iestatīt problēmu brīdinājumus un pārbaudīt savu konfigurāciju.

Skatiet arī

- Vienuma izveide — kā sākt papildu metriku uzraudzību.
- Problēmu eskalācijas — kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

- ODBC uzraudzība — kā iestatīt ODBC citās Linux distribūcijās un kā sākt ar ODBC uzraudzīt papildu ar datubāzi saistītas metrikas.
- Veidne *MySQL by Zabbix agent* — papildu informācija par veidni *MySQL by Zabbix agent*.
- Veidne *MySQL by Zabbix agent 2* — papildu informācija par veidni *MySQL by Zabbix agent 2*.
- Veidne *MySQL by ODBC* — papildu informācija par veidni *MySQL by ODBC*.

5 VMware uzraudzība ar Zabbix

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu VMware pamata uzraudzību.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu VMware pamata uzraudzību. Ja meklējat padziļinātas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Virtuālo mašīnu uzraudzība vai sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt ar šo rokasgrāmatu, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu OS paredzētajām instrukcijām.

Atkarībā no jūsu vides daži šīs rokasgrāmatas soļi var nedaudz atšķirties. Šī rokasgrāmata ir balstīta uz vidi, kurā darbojas Ubuntu.

Tiek pieņemts, ka VMware jau ir konfigurēts. Šī rokasgrāmata neaptver VMware konfigurēšanu.

Konfigurējiet Zabbix serveri

Lai uzraudzītu VMware, ir jāiespējo Zabbix *vmware collector* procesi. Plašāku informāciju par to, kā tiek veikta VMware uzraudzība, skatiet Virtuālo mašīnu uzraudzība.

1. Atveriet Zabbix servera konfigurācijas failu.

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

2. Atrodiet un iestatiet Zabbix servera konfigurācijas failā parametru `StartVMwareCollectors` uz 2 vai vairāk (noklusējuma vērtība ir 0).

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

StartVMwareCollectors=2
```

3. Restartējiet Zabbix serveri.

```
systemctl restart zabbix-server
```

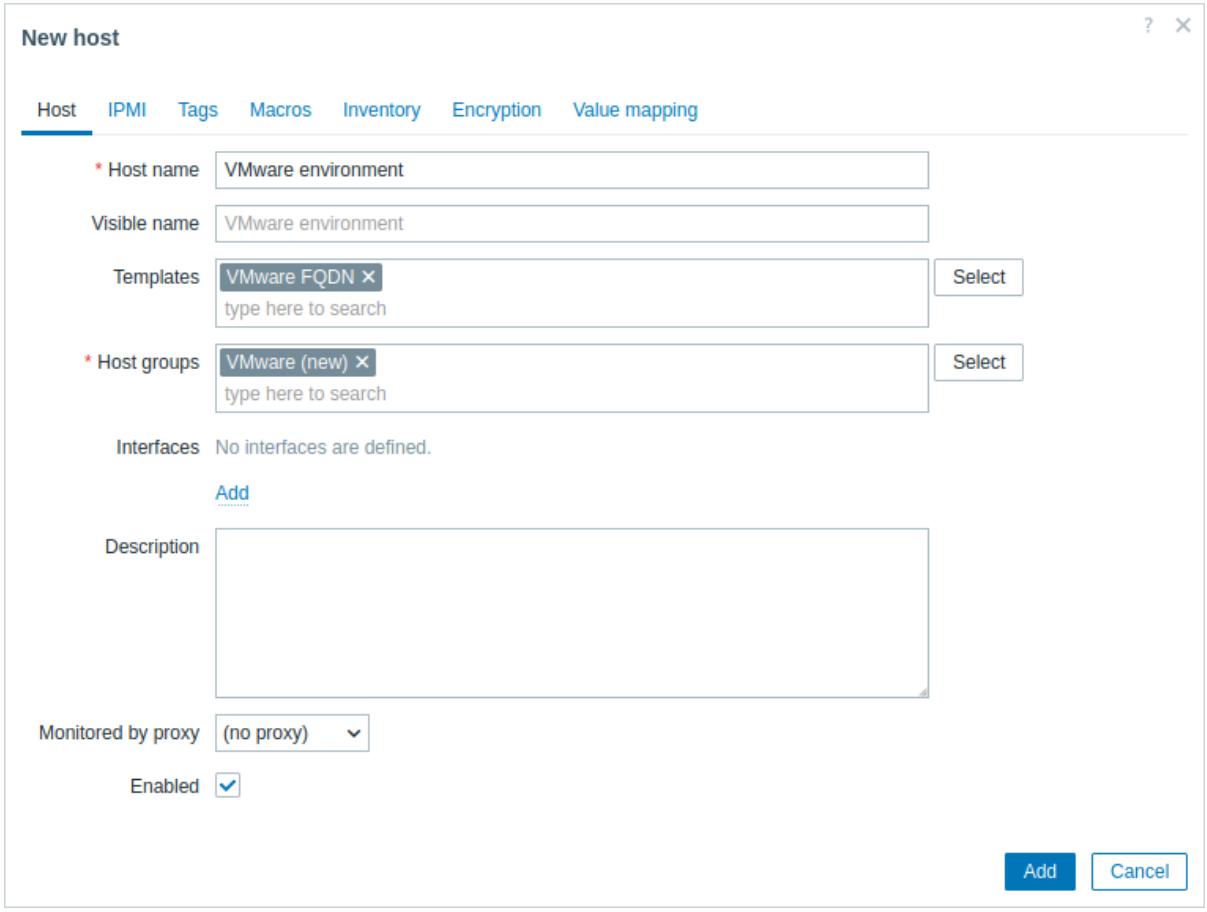
Kad *vmware collector* procesi ir palaisti, pārejiet pie nākamā soļa.

Konfigurējiet Zabbix lietotāja saskarni

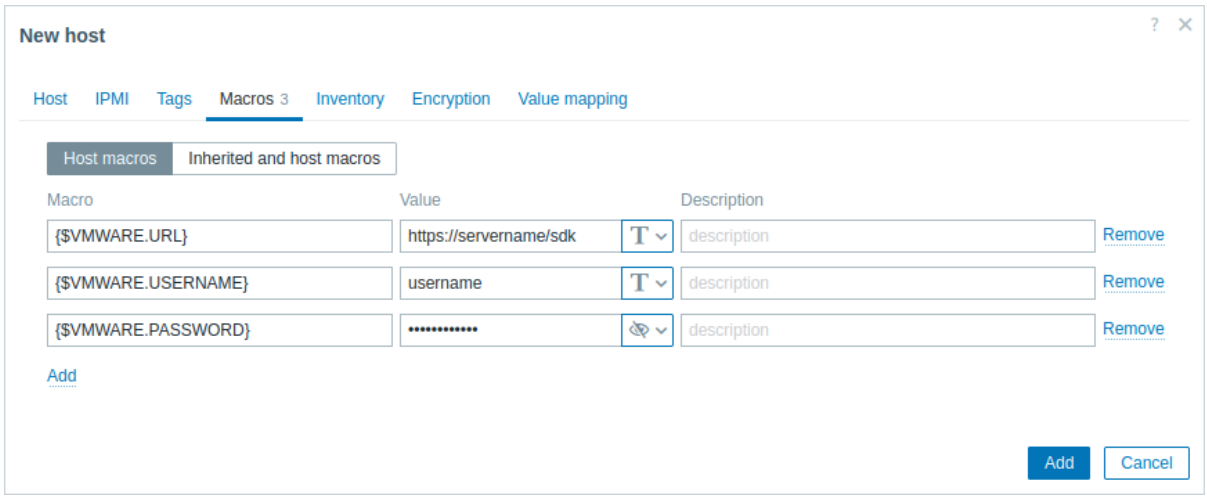
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”VMware environment”).
- Laukā *Templates* ierakstiet vai atlasiet veidni ”VMware FQDN” (vai ”VMware”). Papildinformāciju par šīm veidnēm skatiet sadaļā Virtuālo mašīnu uzraudzība.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, jaunu hostu grupu ”VMware”).



- Cilnē *Macros* iestatiet šādus hosta makrosus:
  - {$VMWARE.URL} - VMware pakalpojuma (vCenter vai ESXi hipervizora) SDK URL (https://servername/sdk)
  - {$VMWARE.USERNAME} - VMware pakalpojuma lietotājvārds
  - {$VMWARE.PASSWORD} - VMware pakalpojuma lietotāja {$VMWARE.USERNAME} parole



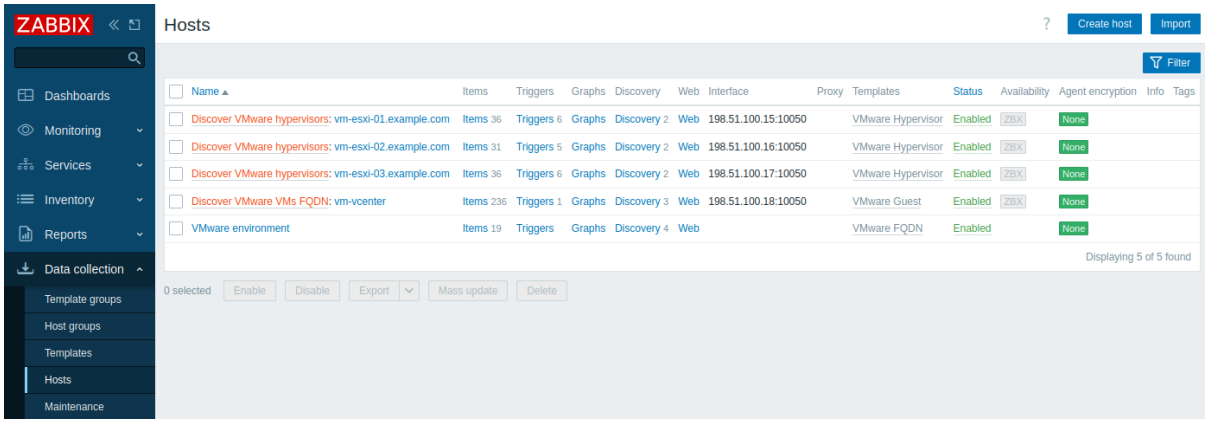
3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts attēlos jūsu VMware vidi.

Skatīt savāktos metriku datus

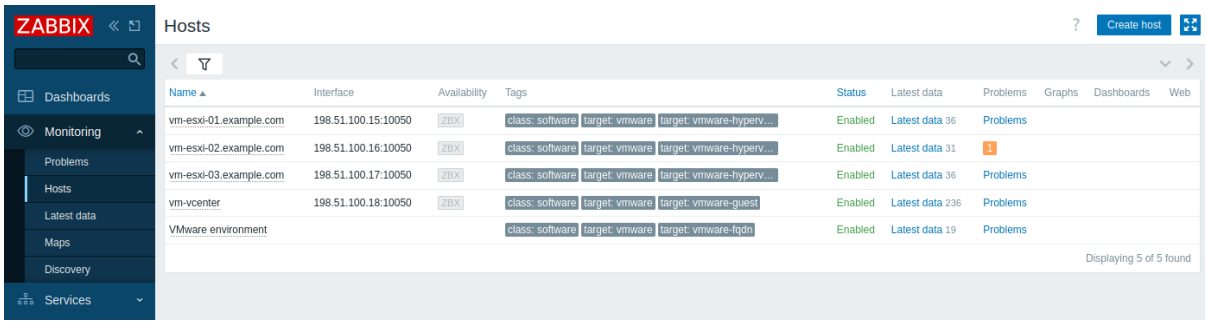
Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu VMware vidi.

Atkarībā no jūsu VMware vides konfigurācijas, Zabbix var atklāt un pēc tam izveidot hostus atklātajām entītijām. Ņemiet vērā, ka atklāšanu un hostu izveidi vajadzības gadījumā var arī izpildīt manuāli.

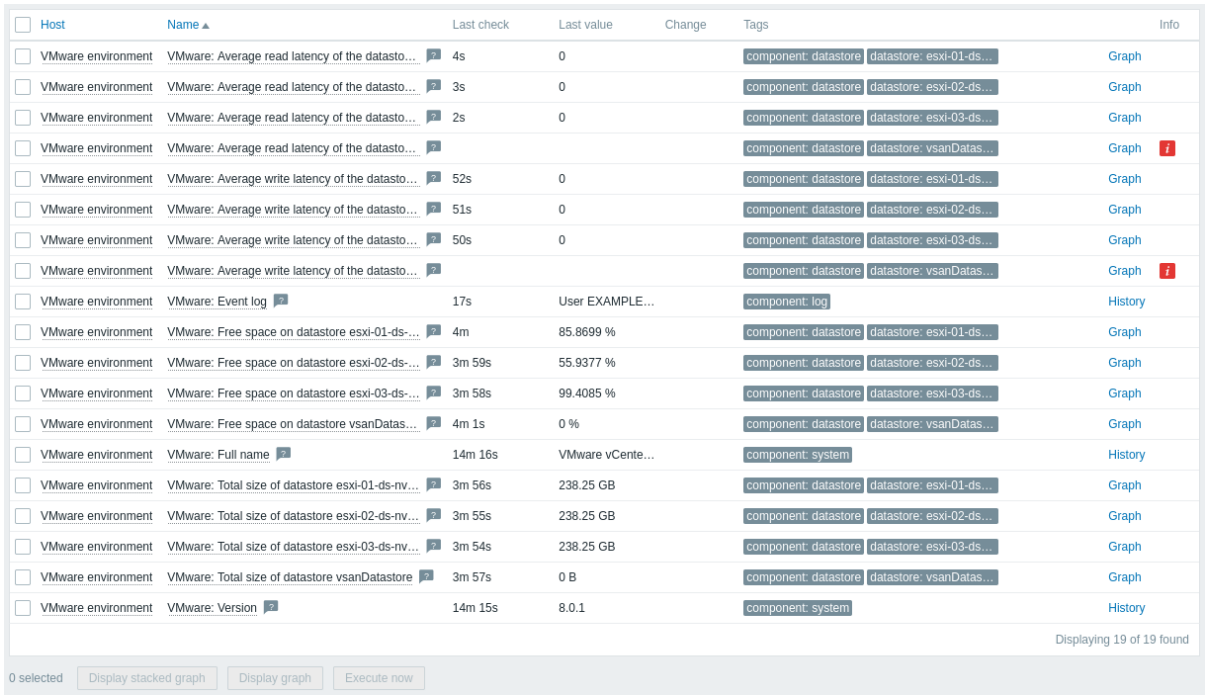
Lai skatītu izveidotos hostus, dodieties uz izvēlnes sadaļu *Datu ievākšana → Hosti*.



Lai skatītu savāktos metriku datus, dodieties uz izvēlnes sadaļu *Uzraudzība → Hosti* un noklikšķiniet uz *Jaunākie dati* blakus izveidotajam "VMware environment" hostam vai vienam no hostiem, kas tika izveidoti atklātajām entītijām.



Šī darbība atvērs visu no atlasītā hosta savākto jaunāko metriku sarakstu.

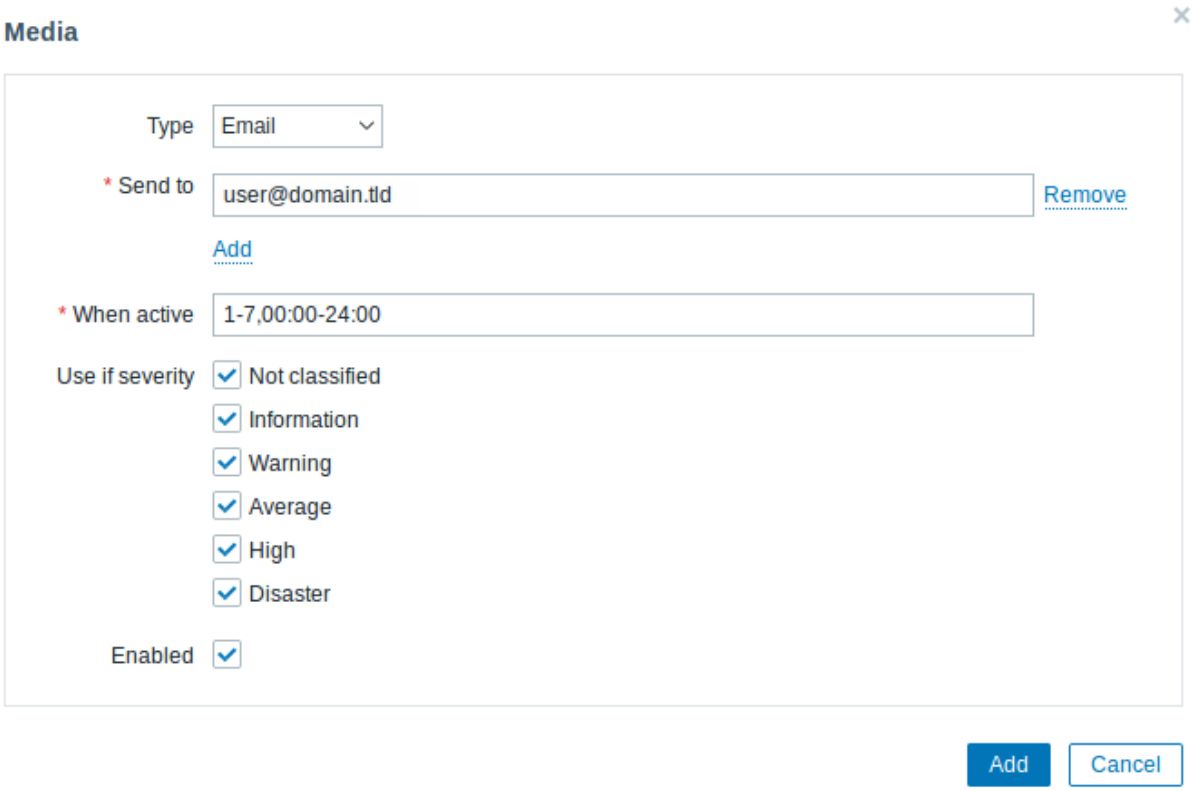


Ņemiet vērā, ka dažiem vienumiem nav datu un tiem ir stāvoklis *Nav atbalstīts*. Tas ir tāpēc, ka Zabbix nevar atrast derīgus veiktspējas skaitītājus konkrētajā datu krātuvē, jo tie nav iespējoti uzraugāmajā VMware vidē.

Problēmu brīdinājumu iestatīšana

Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu nosūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi → Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Skatiet arī

- Vienuma izveide - kā sākt papildu metriku uzraudzību.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).
- Virtuālo mašīnu uzraudzība - papildu informācija par VMware uzraudzību (datu vākšanas process, servera konfigurācijas opcijas, problēmu novēršanas norādījumi u.c.).
- VMware uzraudzības vienumu atslēgas - pilns VMware metriku saraksts, kuras var uzraudzīt, izmantojot Zabbix.
- Veidne *VMware* - papildu informācija par *VMware* veidni.
- Veidne *VMware FQDN* - papildu informācija par *VMware FQDN* veidni.

6 Tīkla datplūsmas uzraudzība ar Zabbix

Ievads Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai ar Zabbix sāktu jūsu tīkla datplūsmas pamata uzraudzību.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu jūsu tīkla datplūsmas pamata uzraudzību. Ja meklējat plašākas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents saskaņā ar jūsu OS paredzētajām instrukcijām. Ņemiet vērā, ka Zabbix aģents ir jāinstalē datorā, kuram nepieciešama datplūsmas uzraudzība. Tas var būt tas pats hosts, kurā ir instalēts Zabbix serveris, vai arī cits hosts.

Šajā ceļvedī ir sniegtas instrukcijas, kā konfigurēt *eth0* saskarnes tīkla datplūsmas uzraudzību atsevišķā datorā ar nosaukumu *Remote host*.

Konfigurējiet Zabbix uzraudzībai Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus). Plašāku informāciju skatiet sadaļā Passive and active agent checks. Šajā rokasgrāmatā tiks aprakstīta uzraudzība, izmantojot pasīvās pārbaudes.

Zabbix aģenta konfigurēšana

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents (pēc noklusējuma ceļš ir */usr/local/etc/zabbix_agentd.conf*):

```
sudo vi /usr/local/etc/zabbix_agentd.conf
```

2. Parametram *Server* pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu. Piemēram:

```
Server=192.0.2.22
```

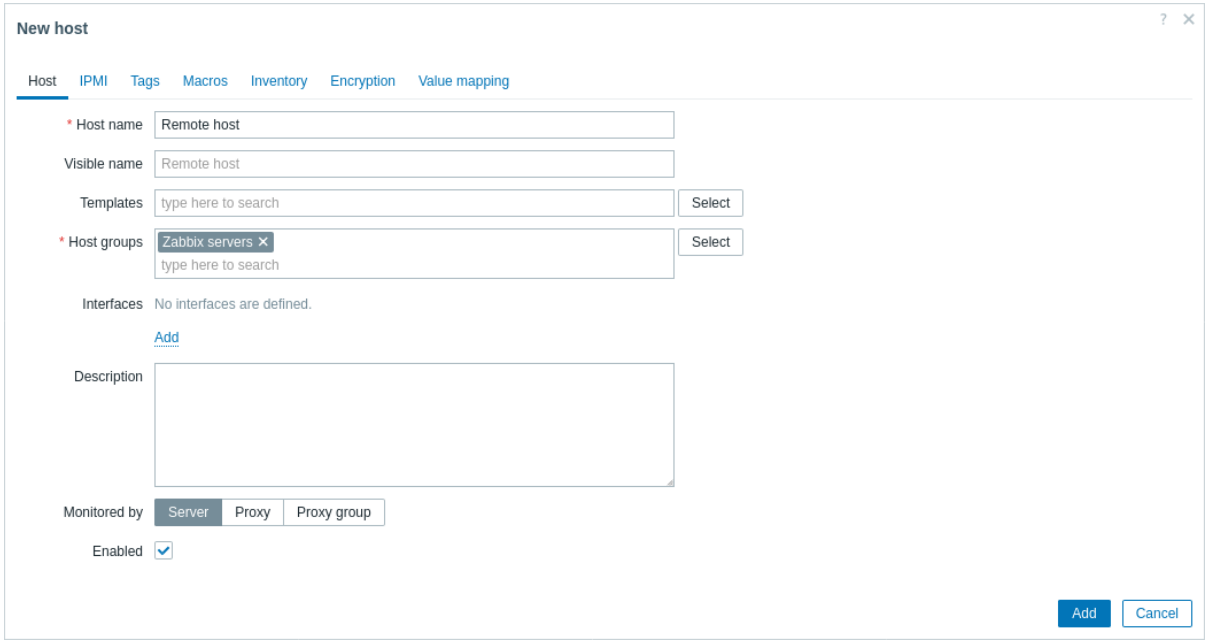
3. Restartējiet Zabbix aģentu:

```
systemctl restart zabbix-agent
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

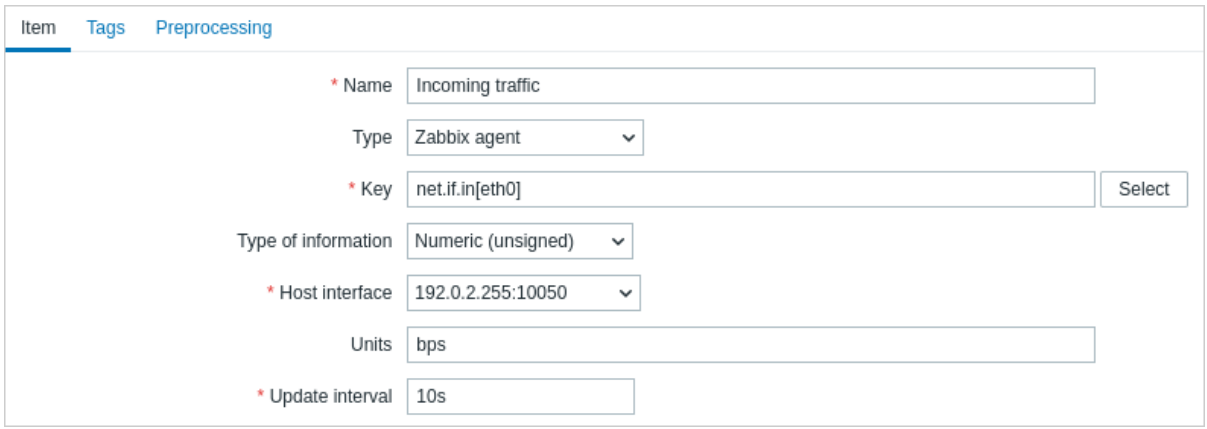
2. Izveidojiet hostu Zabbix tīmekļa saskarnē, norādot tās iekārtas IP adresi vai DNS nosaukumu, kurā ir instalēts aģents.



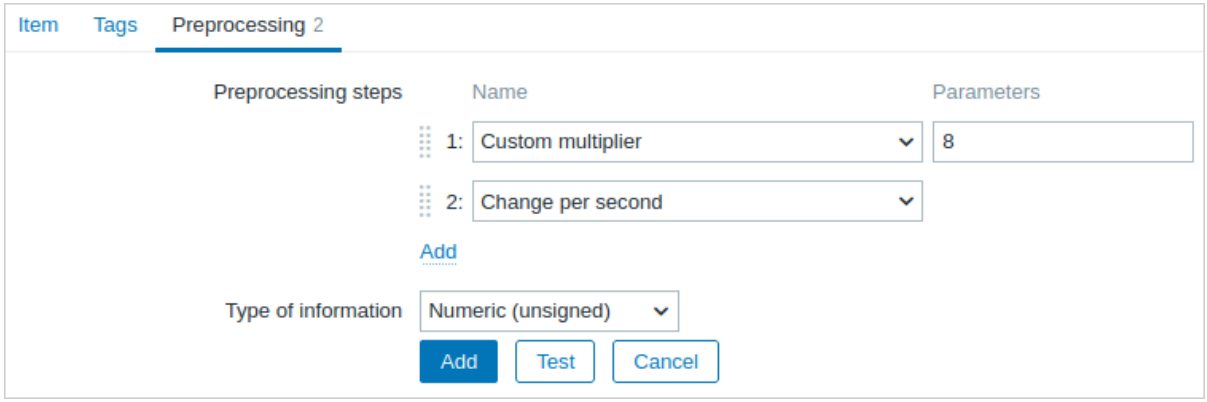
Izveidot vienumus Izpildiet norādījumus sadaļā vienuma izveide, lai pievienotu vienumus datplūsmas uzraudzībai, proti:

- Ienākošā datplūsma
- Izejošā datplūsma
- Kopējā datplūsma

Vienkārša ienākošās datplūsmas uzraudzības vienuma konfigurācija varētu izskatīties šādi:

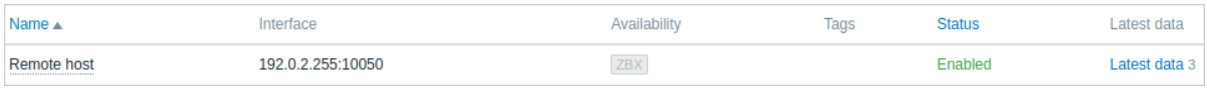


Lai savāktie dati būtu piemēroti praktiskai izmantošanai, veidojot vienumus, varat iestatīt dažus priekšapstrādes soļus. Šajā gadījumā tie var būt reizināšana ar 8 (lai pārvērstu baitus bitos) un attēlošana kā izmaiņas sekundē.

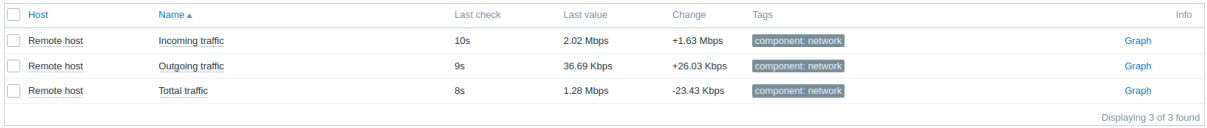


Skatīt savāktos datus Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu tīkla datplūsmu.

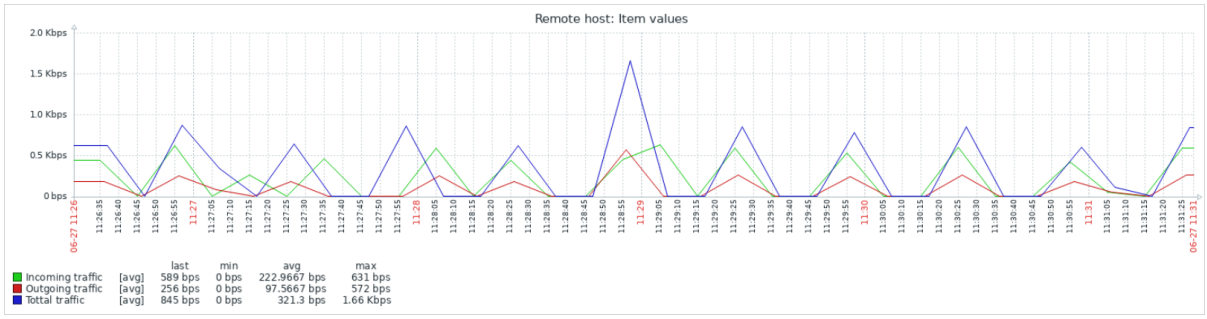
Lai skatītu savāktās metrikas, atveriet izvēlnes sadaļu *Monitoring → Hosts* un noklikšķiniet uz *Latest data* attiecīgā hosta rindā.



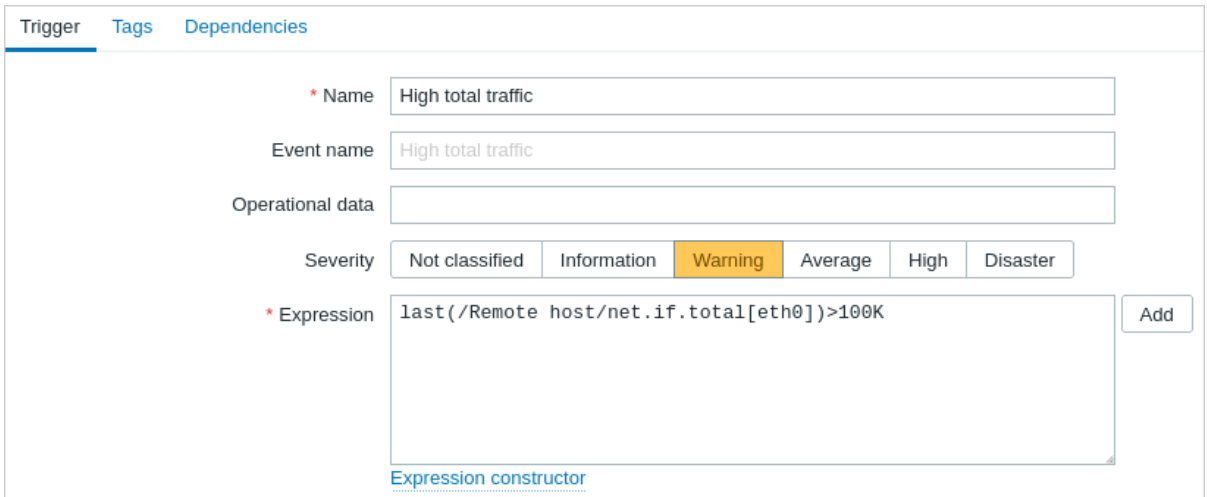
Jūs redzēsiet datplūsmas datus.



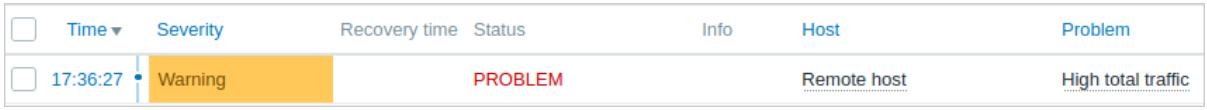
Skatīt grafikus Savāktos datus var attēlot kā grafikus. Lai tos skatītu, sadaļā *Jaunākie dati* noklikšķiniet uz *Grafiks* attiecīgā vienuma rindā vai atlasiet nepieciešamos vienumus un zemāk noklikšķiniet uz *Parādīt grafiku*.



Konfigurējiet trigerus Varat iestatīt trigerus, lai noteiktu neparastu tīkla datplūsmu. Skatiet norādījumus par trigera konfigurēšanu un pievienojiet trigeri, kas signalizē, ka kopējā datplūsma ir pārāk liela, piemēram:



Tagad panāciet, lai datplūsma pārsniegtu slieksni, ko esat iestatījis trigera izteiksmē, un dodieties uz *Uzraudzība → Problēmas*, lai pārbaudītu, ka problēma tur ir norādīta.



Problēmu brīdinājumu iestatīšana Ir vairāki veidi, kā saņemt paziņojumus par problēmu. E-pasts ir vispopulārākais, tāpēc sekojiet norādījumiem, kā iestatīt paziņojumu par problēmu pa e-pastu. Varat arī izvēlēties citus mediju tipus, ko izmantot paziņojumu piegādei.

Skatiet arī:

- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).
- Problēmas apstiprināšana - kā norādīt, ka problēma ir zināma, pievienot komentārus par tās risināšanu, apspiest vai aizvērt problēmu.
- Linux uzraudzība ar Zabbix aģentu - kā sākt svarīgāko vienumu pamata uzraudzību, piesaistot iepriekš konfigurētu veidni.

7 Tīkla datplūsmas uzraudzība, izmantojot aktīvās pārbaudes

Ievads Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai ar Zabbix, izmantojot aktīvās pārbaudes, sāktu jūsu tīkla datplūsmas pamata uzraudzību.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu jūsu tīkla datplūsmas pamata uzraudzību, izmantojot aktīvās pārbaudes. Ja meklējat plašākas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents atbilstoši jūsu OS paredzētajām instrukcijām. Ņemiet vērā, ka Zabbix aģents ir jāinstalē datorā, kuram nepieciešama datplūsmas uzraudzība. Tas var būt tas pats hosts, kurā ir instalēts Zabbix serveris, vai arī cits hosts.

Šajā ceļvedī tiks sniegtas instrukcijas, kā konfigurēt *eth0* saskarnes tīkla datplūsmas uzraudzību atsevišķā datorā ar nosaukumu *Remote host*.

Konfigurējiet Zabbix uzraudzībai Zabbix aģents var vākt metriku aktīvajā vai pasīvajā režīmā (vienlaikus). Plašāku informāciju skatiet sadaļā Pasīvās un aktīvās aģenta pārbaudes. Šajā ceļvedī tiks aprakstīta uzraudzība, izmantojot **aktīvās pārbaudes**.

Zabbix aģenta konfigurēšana

1. Atveriet aģenta konfigurācijas failu datorā, kurā ir instalēts aģents.

Ja izmantojat Zabbix aģentu:

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

Ja izmantojat Zabbix aģentu 2:

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

2. Pievienojiet sava Zabbix servera IP adresi vai DNS nosaukumu (un neobligātu portu) parametram ServerActive. Piemēram:

```
ServerActive=192.0.2.0:10051
```

Zabbix aģents izmantos šo adresi, lai izveidotu savienojumu ar Zabbix servera trapper portu (noklusējums: 10051) un pieprasītu aktīvo pārbaužu konfigurācijas datus.

3. Definējiet parametru Hostname, kam jāsakrīt ar hosta nosaukumu, kas tiks definēts Zabbix lietotāja saskarnē. Mūsu piemērā tas ir:

```
Hostname=Remote host
```

Hostname vērtībai jāsakrīt, jo aktīvo pārbaužu gadījumā Zabbix aģents to izmanto, lai no servera saņemtu pareizo hosta konfigurāciju. Precīzāk, aģents iniciē savienojumu ar serveri un identificē sevi, izmantojot Hostname vērtību. Pēc tam serveris nodrošina šī hosta uzraudzības konfigurāciju. Ja šīs vērtības atšķiras, aģents nesaņems atbilstošo konfigurāciju, kā rezultātā trūks metriku vai radīsies uzraudzības problēmas.

4. Restartējiet Zabbix aģentu.

Ja izmantojat Zabbix aģentu:

```
systemctl restart zabbix-agent
```

Ja izmantojat Zabbix aģentu 2:

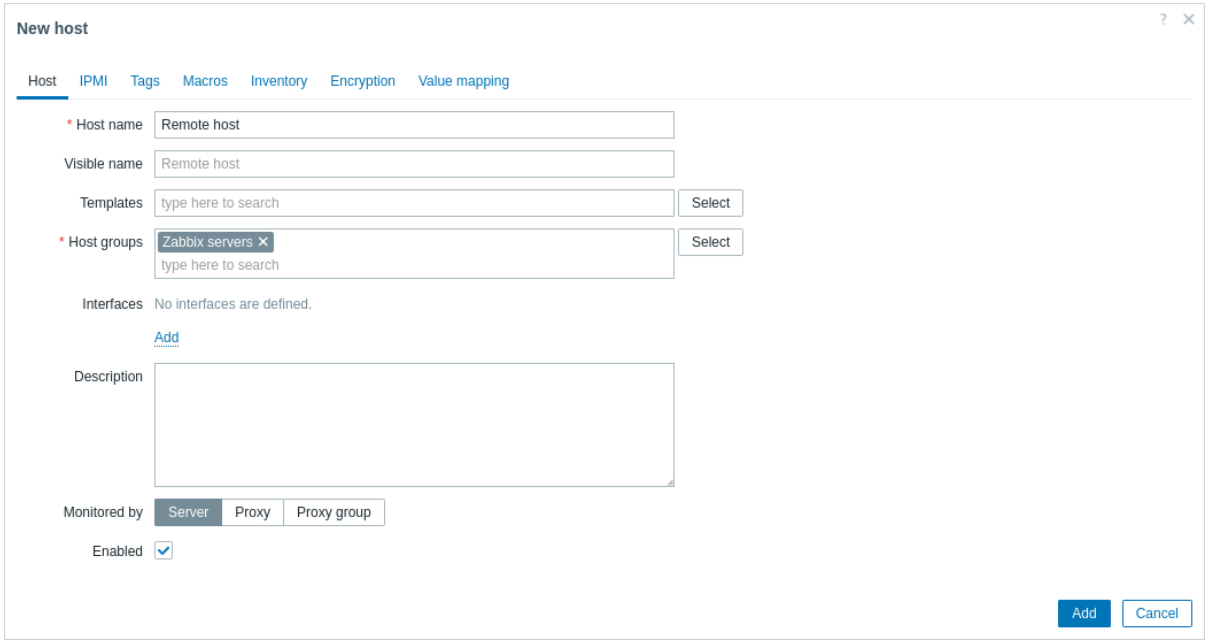
```
systemctl restart zabbix-agent2
```

Zabbix lietotāja saskarne

1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē.

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”Remote host”), kas atbilst iepriekš aģenta konfigurācijas failā definētajai parametra Hostname vērtībai.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”Zabbix servers”).
- Atstājiet *Interfaces* nedefinētu; saskarne nav nepieciešama aktīvajām pārbaudēm, jo aģents pats iniciē savienojumu ar serveri, nevis gaida savienojumu no servera.

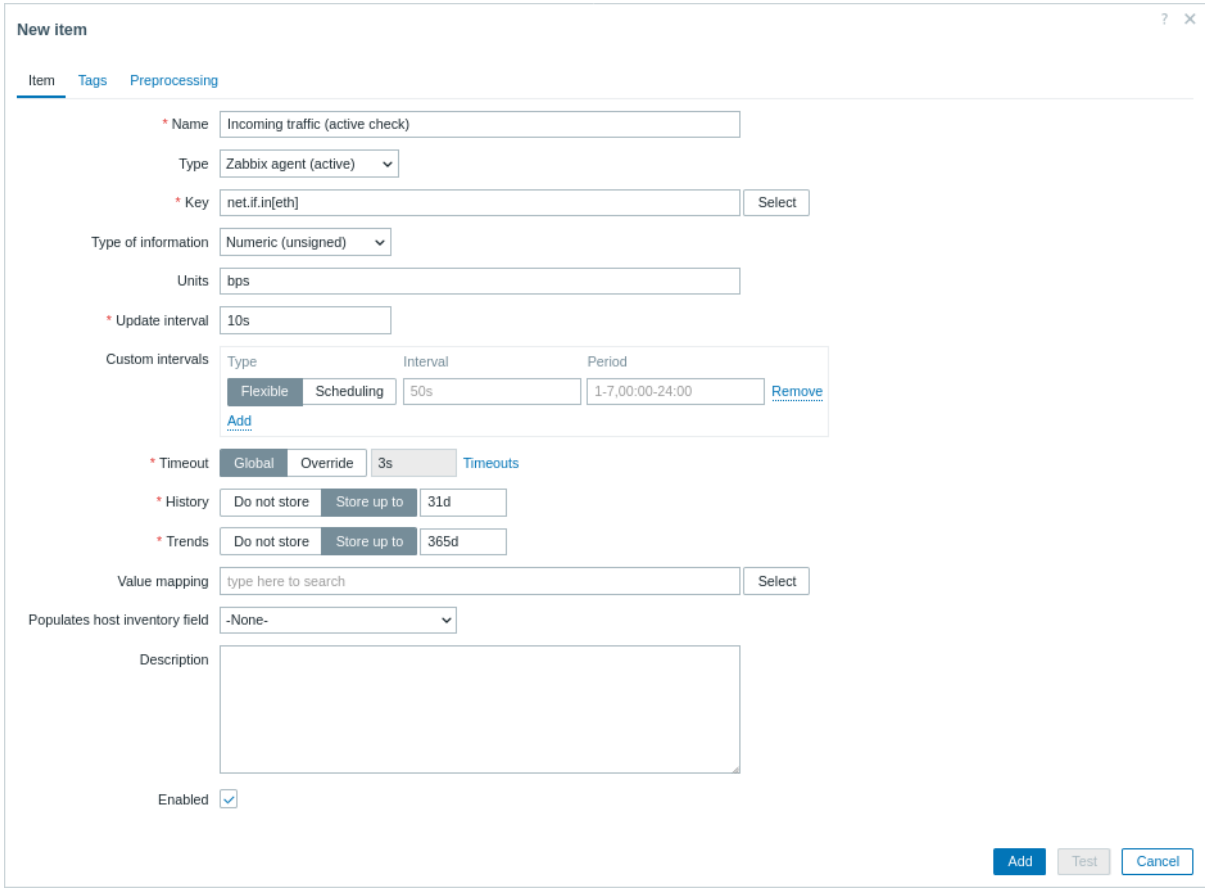


3. Noklikšķiniet uz *Add*, lai pievienotu hostu. Šis hosts attēlos uzraudzīto Linux datoru.

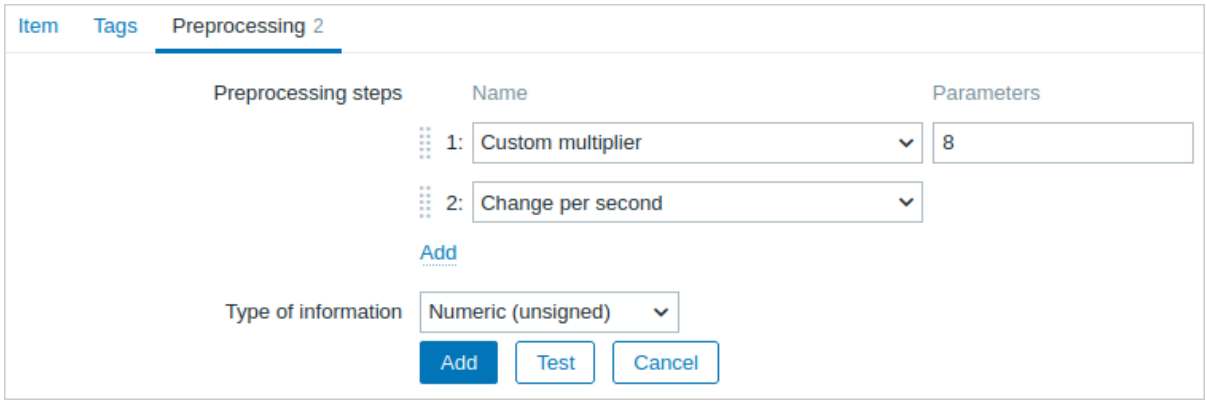
Izveidojiet vienumus Izpildiet norādījumus sadaļā vienuma izveide, lai pievienotu vienumus datplūsmas uzraudzībai, proti:

- Ienākošā datplūsma
- Izejošā datplūsma
- Kopējā datplūsma

Vienkārša ienākošās datplūsmas uzraudzības vienuma konfigurācija ar aktīvo pārbaudi izskatītos šādi:

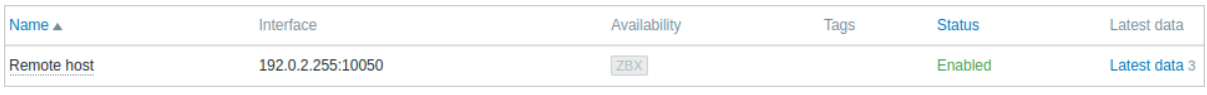


Lai savāktie dati būtu piemēroti praktiskai izmantošanai, veidojot vienumus, varat iestatīt dažus priekšapstrādes soļus. Šajā gadījumā tie var būt reizināšana ar 8 (lai pārvērstu baitus bitos) un attēlošana kā izmaiņas sekundē.

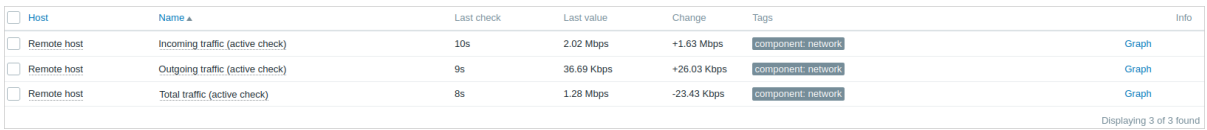


Skatīt savāktos datus Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu tīkla datplūsmu.

Lai skatītu savāktās metrikas, atveriet izvēlnes sadaļu *Monitoring → Hosts* un noklikšķiniet uz *Latest data* attiecīgā hosta rindā.



Jūs redzēsiet datplūsmas datus.



Skatiet arī:

- Skatīt grafikus - kā attēlot savāktos datus grafiku veidā.
- Konfigurēt trigerus - kā iestatīt trigerus, lai noteiktu neparastu tīkla datplūsmu.
- Iestatīt problēmu brīdinājumus - kā iestatīt paziņojumus par problēmsituācijām.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).

- Problēmas apstiprināšana - kā norādīt, ka problēma ir zināma, pievienot komentārus par tās risināšanu, apspiest vai aizvērt problēmu.
- Linux uzraudzība ar Zabbix aģentu - kā sākt svarīgāko vienumu pamata uzraudzību, piesaistot iepriekš konfigurētu veidni.
- Instalēšana no pakotnēm - kā instalēt Zabbix komponentus, izmantojot oficiālās RPM un DEB pakotnes dažādām Linux distribūcijām, nodrošinot piekļuvi jaunākajām iespējām un kļūdu labojumiem.

8 Uzraugiet tīmekļa vietnes ar pārlūka vienumiem

Ievads

Šajā lapā ir aprakstītas darbības, kas nepieciešamas, lai sāktu vietņu pamata uzraudzību ar pārlūka vienumiem.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu vietņu pamata uzraudzību ar pārlūka vienumiem. Ja meklējat padziļinātas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Pārlūka vienumi lapu vai Zabbix rokasgrāmatas sadaļu Konfigurācija.

Priekšnosacījumi

Pirms turpināt ar šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne saskaņā ar jūsu OS paredzētajām instrukcijām.

Atkarībā no jūsu vides daži šajā ceļvedī aprakstītie soļi var nedaudz atšķirties. Šis ceļvedis ir balstīts uz vidi, kurā darbojas Ubuntu.

Konfigurēt WebDriver

Pārlūka vienumiem ir nepieciešams automatizācijas ietvars (vai nu Selenium Server, vai vienkāršs WebDriver, piemēram, ChromeDriver) kā tīmekļa testēšanas galapunkts, kas kontrolē un mijiedarbojas ar pārlūku, izpildot testu komandas, piemēram, pogu klikšķināšanu vai teksta ievadi. Kā piemēru šajā rokasgrāmatā tiks izmantots Selenium Server ar Chrome Docker konteinerā.

Tiek pieņemts, ka Docker jau ir konfigurēts. Šajā rokasgrāmatā Docker konfigurēšana nav aplūkota. Instalēšanas norādījumus skatiet šeit: Install Docker Engine on Ubuntu.

1. Palaidiet Selenium Server ar Chrome Docker konteinerā, izmantojot šādas opcijas:

- **docker run --name browser** — palaiž jaunu Docker konteineru ar nosaukumu "browser";
- **-p 4444:4444** — kartē 4444. portu jūsu hostā uz 4444. portu konteinerā (šis ir ports, ko Selenium Server izmanto komandu pieņemšanai);
- **-p 7900:7900** — kartē 7900. portu jūsu hostā uz 7900. portu konteinerā (šis ir ports, ko izmanto Virtual Network Computing (VNC) serveris, ļaujot jums attālināti skatīt pārlūka GUI; nepieciešams VNC klients);
- **--shm-size="2g"** — piešķir konteineram 2 GB koplietojamās atmiņas (tas ir svarīgi, lai Chrome darbotos pareizi, jo tam var būt nepieciešams ievērojams koplietojamās atmiņas apjoms, lai izvairītos no avārijām);
- **-d** — palaiž konteineru atvienotajā režīmā, kas nozīmē, ka tas darbosies fonā;
- **selenium/standalone-chrome:latest** — norāda izmantojamo Docker attēlu; šajā gadījumā tā ir jaunākā Selenium Server with Chrome versija.

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

2. Pārliecinieties, ka `browser` Docker konteiners darbojas un ir pieejams.

- Iegūstiet konteinera IP adresi (šajā piemērā 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
###     ...
```

- Pārbaudiet savienojumu ar konteineru, izmantojot Ncat:

```
nc -zv 192.0.2.1 4444

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

- Iegūstiet tīmekļlapas saturu no Selenium Server, izmantojot 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:
Problēmu novēršanai skatiet Docker documentation.

Konfigurējiet Zabbix serveri

Pārlūka vienumi tiek izpildīti un apstrādāti ar *browser poller* Zabbix procesiem, kas ir jāiespējo, pielāgojot servera konfigurācijas parametru StartBrowserPollers. Papildus tam parametram WebDriverURL jānorāda iepriekš konfigurētais tīmekļa testēšanas galapunkts.

Pēc noklusējuma parametrs StartBrowserPollers ir iestatīts uz 1, tāpēc jums ir jānorāda tikai tīmekļa testēšanas galapunkts.

1. Atveriet Zabbix servera konfigurācijas failu.

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

2. Atrodiet un iestatiet parametru WebDriverURL Zabbix servera konfigurācijas failā:

```
##### Option: WebDriverURL
###WebDriver saskarnes HTTP[S] URL. Piemēram, http://localhost:4444 tiek izmantots ar Selenium WebDriver s
###
### Mandatory: no
### Default:
### WebDriverURL=

WebDriverURL=192.0.2.1:4444
```

3. Restartējiet Zabbix serveri.

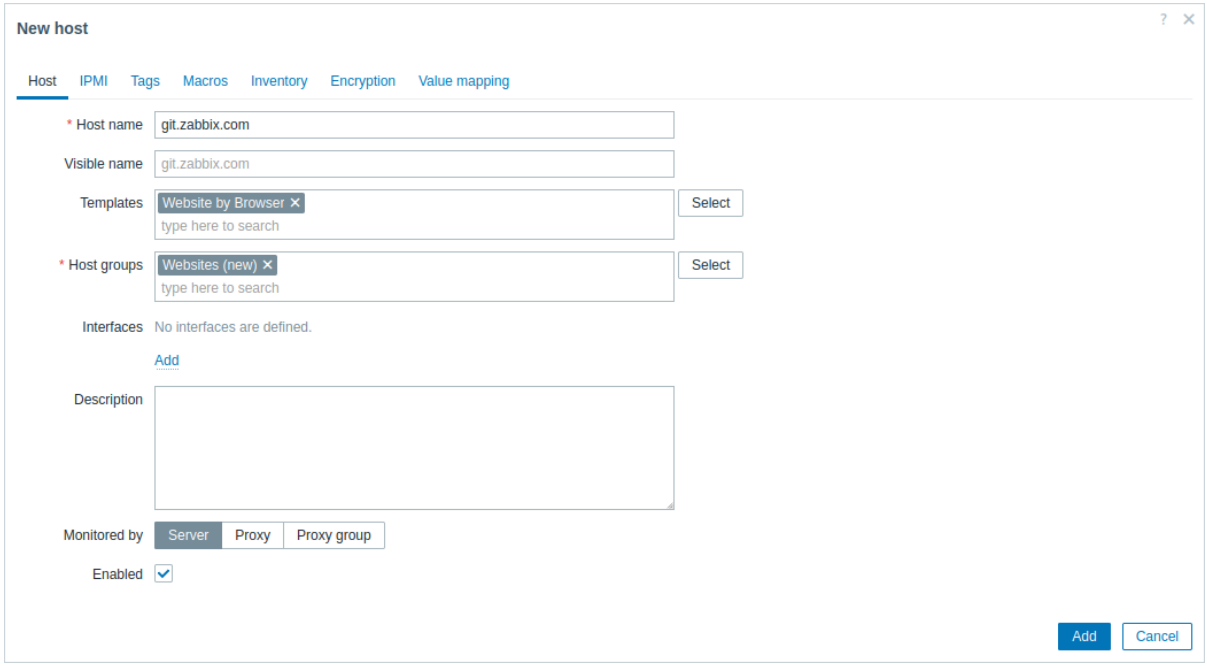
```
systemctl restart zabbix-server
```

Konfigurējiet Zabbix lietotāja saskarni

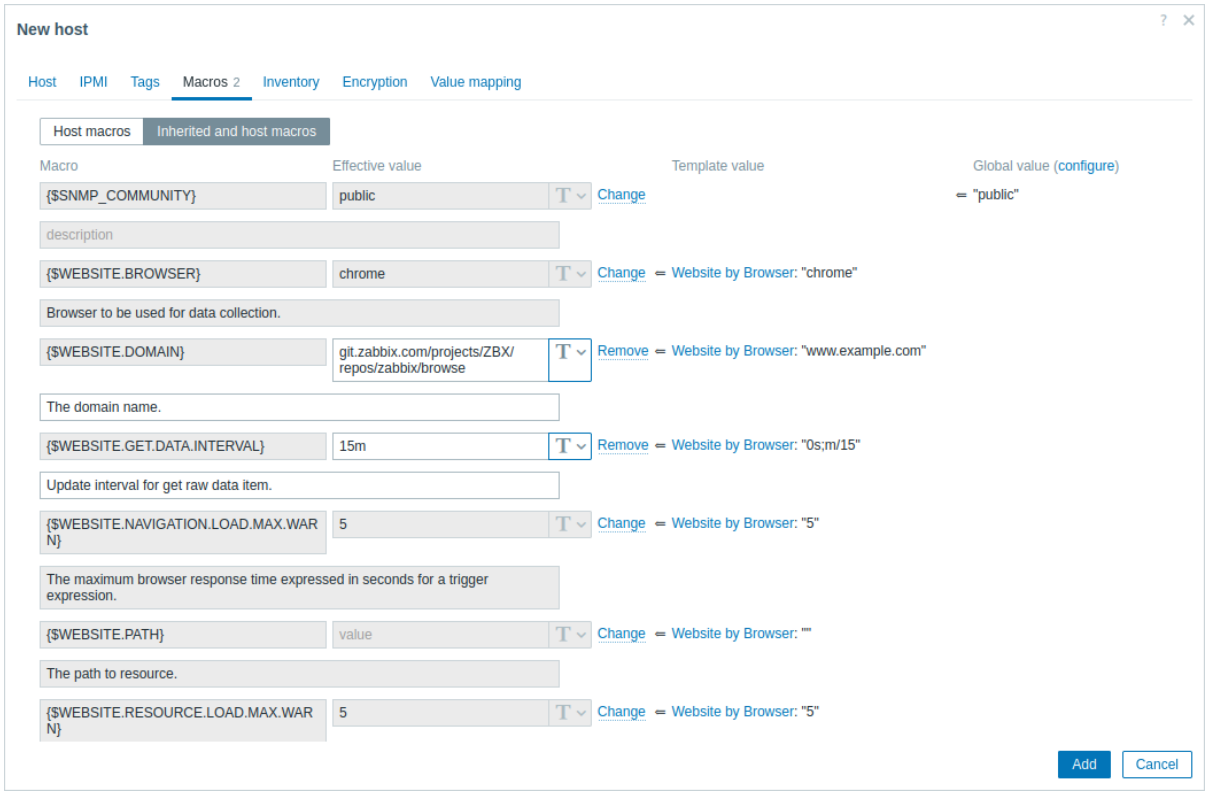
1. Piesakieties Zabbix lietotāja saskarnē.

2. Izveidojiet hostu Zabbix tīmekļa saskarnē:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”git.zabbix.com”).
- Laukā *Templates* ierakstiet vai atlasiet veidni ”Website by Browser”. Papildinformāciju par šo veidni skatiet šeit: Website by Browser.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, jaunu hostu grupu ”Websites”).



- Cilnē *Macros* pārslēdzieties uz *Inherited and host macros*, atrodiet tālāk norādītos makrosus un noklikšķiniet uz *Change* blakus makrosa vērtībai, lai to atjauninātu:
  - {$WEBSITE.DOMAIN} - Domēna nosaukums (piemēram, git.zabbix.com/projects/ZBX/repos/zabbix/browse)
  - {$WEBSITE.GET.DATA.INTERVAL} - Vienuma datu atjaunināšanas intervāls (piemēram, 15m)

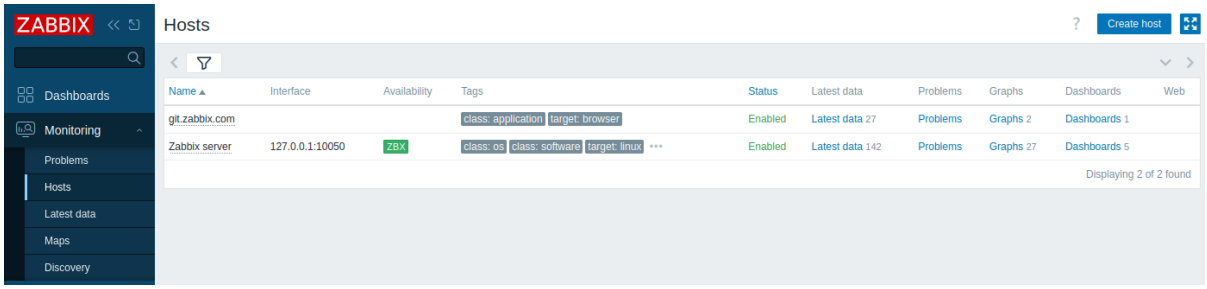


3. Noklikšķiniet uz pogas *Add*, lai izveidotu hostu. Šis hosts attēlos vietni, kuru vēlaties uzraudzīt.

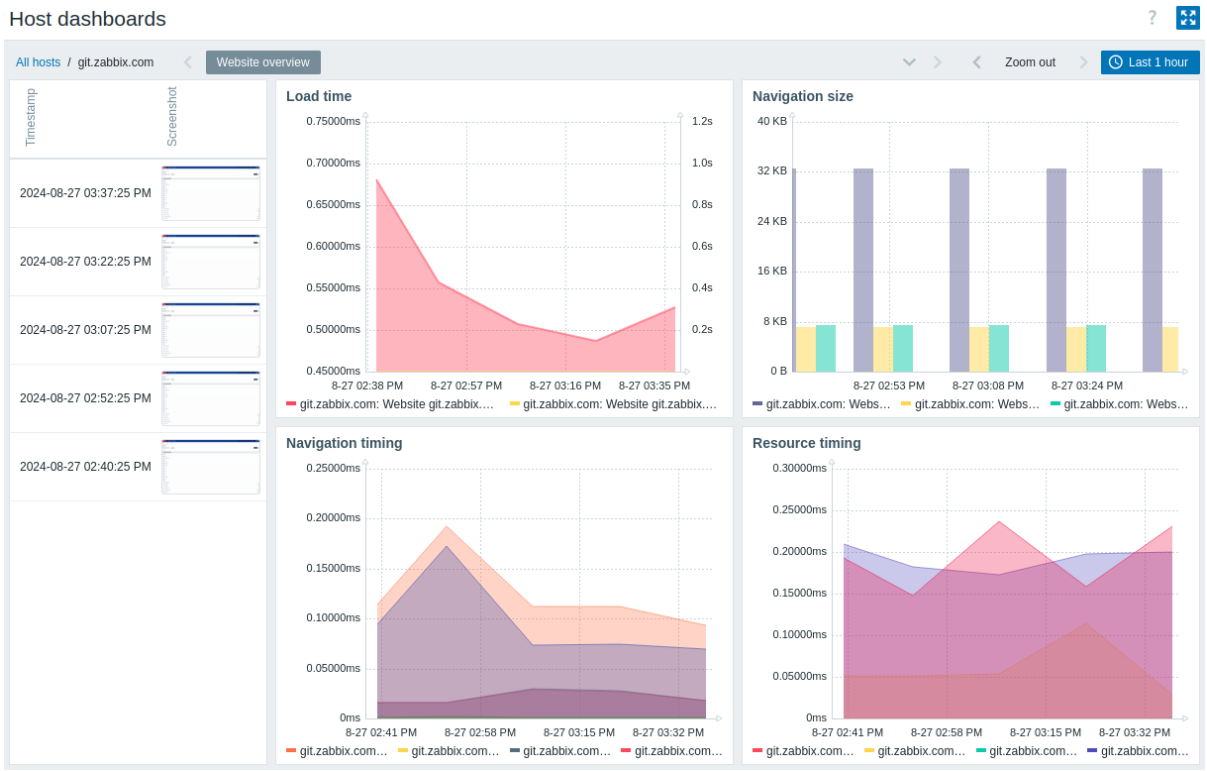
Skatīt savāktos metriku datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga jūsu norādīto tīmekļa vietni.

Lai skatītu savāktos metriku datus, atveriet izvēlnes sadaļu *Monitoring → Hosts* un noklikšķiniet uz *Dashboards* blakus hostam.



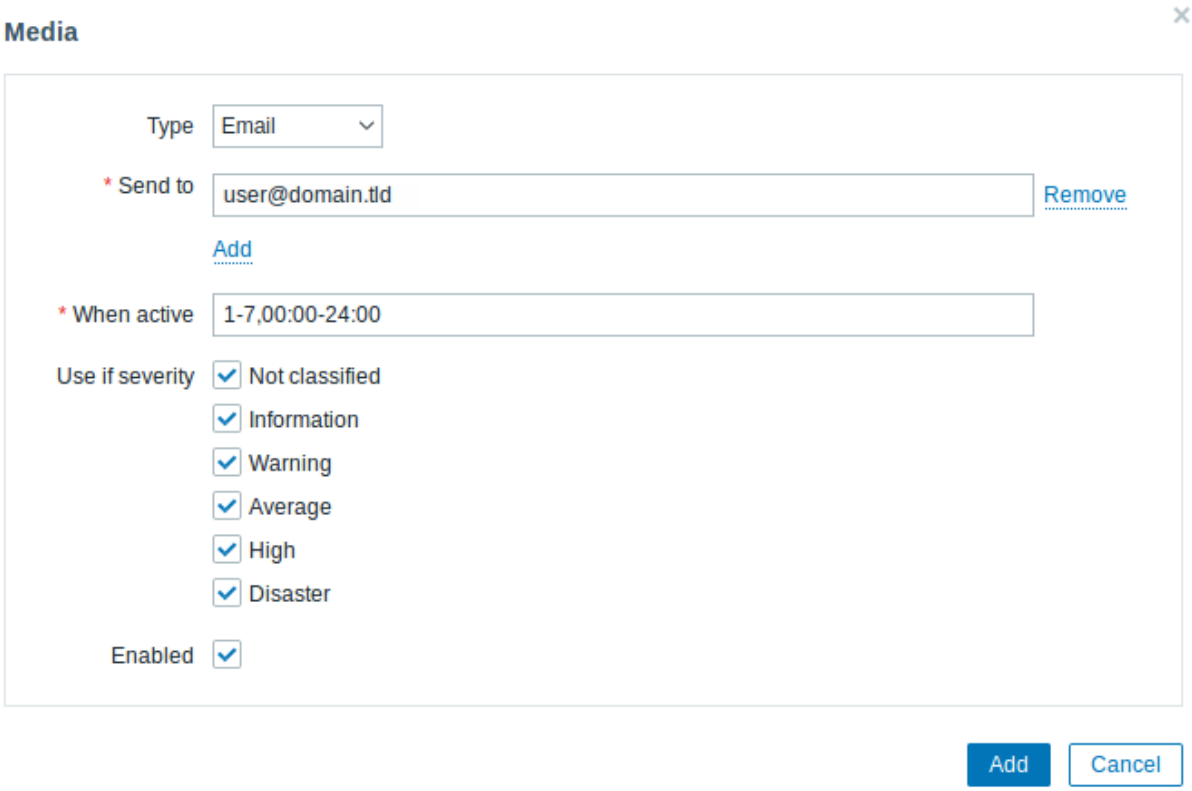
Šī darbība jūs novirzīs uz hosta paneli (konfigurētu veidnes līmenī), kurā redzami svarīgākie no tīmekļa vietnes savāktie metriku dati.



Iestatiet problēmu brīdinājumus

Zabbix var jūs informēt par problēmu jūsu infrastruktūrā, izmantojot dažādas metodes. Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi → Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, mēs varam simulēt reālu problēmu, atjauninot hosta konfigurāciju Zabbix lietotāja saskarnē.

1. Atveriet sava tīmekļa vietnes hosta konfigurāciju Zabbix.

2. Pārslēdzieties uz cilni *Macros* un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus, piemēram, iepriekš konfigurētajai {$WEBSITE.DOMAIN} makro vērtībai un iestatiet nepareizu domēna nosaukumu (piemēram, /git.zabbix.com/projects/ZBX/repos/zabbix/browse).

4. Noklikšķiniet uz *Update*, lai atjauninātu hosta konfigurāciju.

5. Pēc neilga brīža Zabbix noteiks problēmu "Failed to get JSON of the requested website", jo tas nevarēs izveidot savienojumu ar norādīto tīmekļa vietni. Problēma parādīsies sadaļā *Monitoring → Problems*.

Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī paziņojumu par problēmu.

6. Mainiet makro vērtību atpakaļ uz iepriekšējo vērtību, lai novērstu problēmu un turpinātu tīmekļa vietnes uzraudzību.

Skatiet arī

- Vienuma izveide - kā sākt uzraudzīt papildu metriku.
- Problēmu eskalācijas - kā izveidot daudzpakāpju brīdinājumu scenārijus (piemēram, vispirms nosūtīt ziņojumu sistēmas administratoram, pēc tam, ja problēma netiek atrisināta 45 minūšu laikā, nosūtīt ziņojumu datu centra vadītājam).
- Pārlūka vienumi - kā konfigurēt pārlūka vienumus.
- Veidne Website by Browser - papildu informācija par *Website by Browser* veidni.

9 Uzraugiet vietņu sertifikātus ar Zabbix aģents 2 (pasīvais)

Ievads

Šī rokasgrāmata sniedz visaptverošu pārskatu par to, kā iestatīt un uzraudzīt SSL/TLS sertifikātus, izmantojot atslēgu `web.certificate.get` Zabbix aģents 2. Tā ir izstrādāta, lai vienkāršotu sertifikātu uzraudzību vienai vai vairākām tīmekļa vietnēm, nodrošinot, ka administratori var ātri identificēt iespējamās problēmas, piemēram, sertifikātus ar beigušos derīguma termiņu vai nederīgus sertifikātus.

Kam šī rokasgrāmata ir paredzēta

Šī rokasgrāmata ir paredzēta jauniem Zabbix lietotājiem un ietver minimālo darbību kopumu, kas nepieciešams, lai iespējotu vietņu sertifikātu pamata uzraudzību. Ja meklējat plašākas pielāgošanas iespējas vai jums nepieciešama sarežģītāka konfigurācija, skatiet Zabbix rokasgrāmatas sadaļu Configuration.

Priekšnosacījumi

Pirms turpināt šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris, Zabbix lietotāja saskarne un Zabbix aģents 2 atbilstoši jūsu OS instrukcijām. Šajā pamācībā tiek pieņemts, ka gan Zabbix serveris, gan aģents ir instalēti vienā un tajā pašā datorā; tādēļ konfigurācijā tiek izmantota `127.0.0.1`.

Konfigurējiet Zabbix aģents 2

1. Atveriet Zabbix aģents konfigurācijas failu (noklusējuma ceļš: `/etc/zabbix/zabbix_agent2.conf`):

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

2. Iestatiet parametru Server uz 127.0.0.1, jo aģents un serveris darbojas vienā un tajā pašā datorā:

```
Server=127.0.0.1
```

3. Saglabājiet failu un restartējiet Zabbix aģents 2 servisu:

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

4. Pēc Zabbix aģents 2 iestatīšanas un konfigurēšanas pārbaudiet tā pieejamību ar:

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

Zabbix aģents 2 pēc noklusējuma ietver WebCertificate spraudni, tāpēc nav nepieciešama atsevišķa instalēšana vai konfigurēšana.

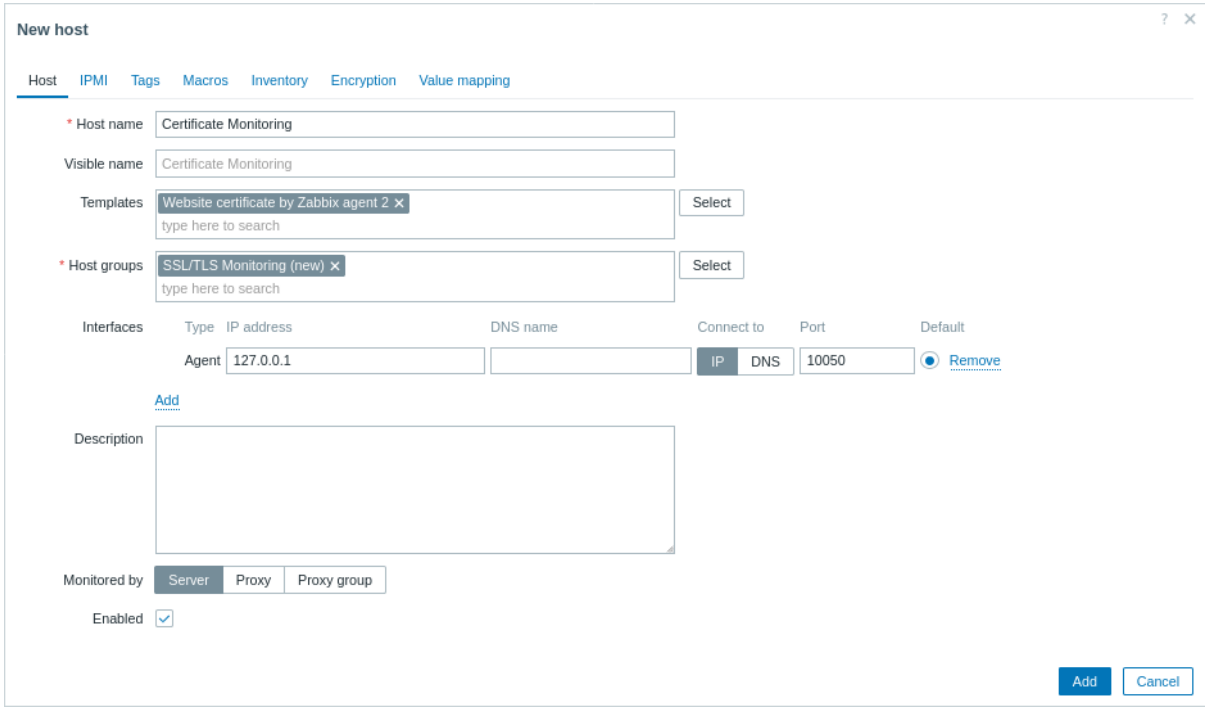
Konfigurējiet Zabbix lietotāja saskarni

1. Piesakieties Zabbix lietotāja saskarnē.

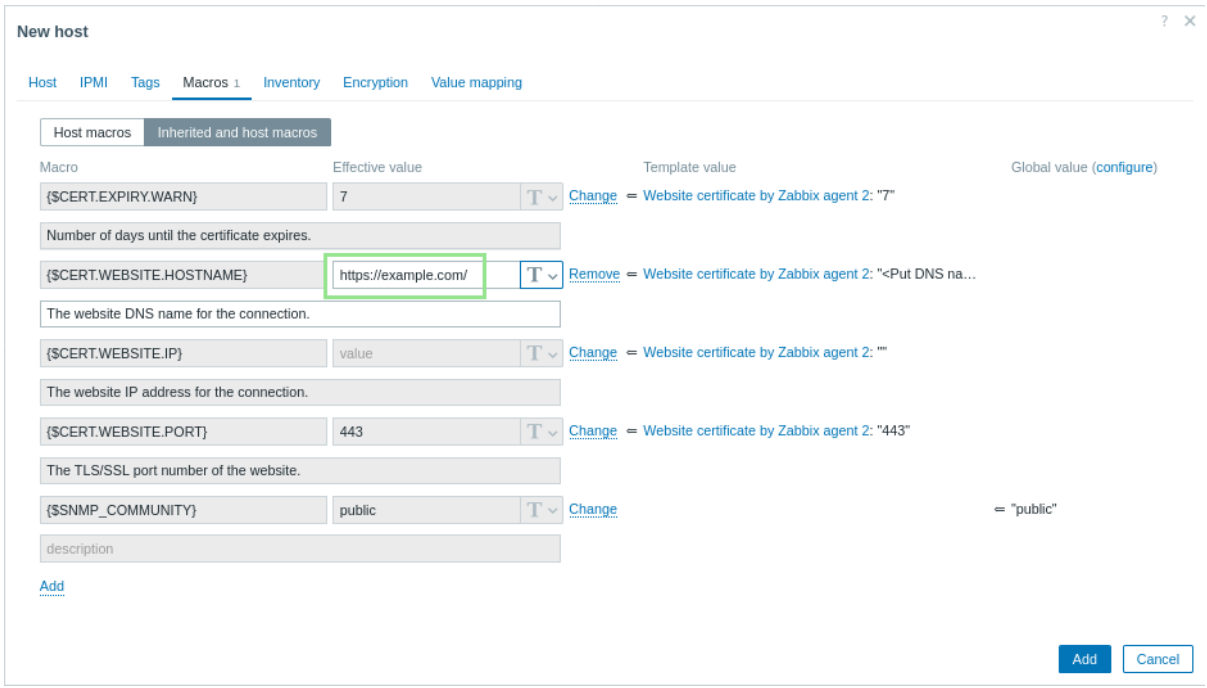
2. Dodieties uz *Monitoring* > *Hosts*.

3. Noklikšķiniet uz esoša hosta, kurā vēlaties uzraudzīt vietņu sertifikātus, vai izveidojiet hostu, ja nepieciešams:

- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”Certificate Monitoring”).
- Laukā *Templates* ierakstiet vai atlasiet veidni ”Website certificate by Zabbix aģents 2”, kas tiks piesaistīta hostam.
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”SSL/TLS Monitoring”).
- Laukā *Interfaces* pievienojiet saskarni ar tipu ”Agent” un norādiet IP adresi. Šajā piemērā tiek izmantota ”127.0.0.1”.



- Cilnē *Macros* pārslēdzieties uz *Inherited and host macros*, atrodiet šādus makrosus un noklikšķiniet uz *Change* blakus makrosa vērtībai, lai to atjauninātu:
  - `{$CERT.WEBSITE.HOSTNAME}` - ievadiet vēlamo vietnes DNS nosaukumu kā vērtību.



4. Noklikšķiniet uz *Add*, lai pievienotu hostu.

Lai uzraudzītu vairākas vietnes, makrosā `{$CERT.WEBSITE.HOSTNAME}` varat norādīt ar komatiem atdalītu to hostu nosaukumu sarakstu. Pēc izvēles makrosā `{$CERT.WEBSITE.PORT}` varat norādīt arī ar komatiem atdalītu portu sarakstu, kur katrs ports secībā atbilst attiecīgajam hosta nosaukumam. Piemēram, lai uzraudzītu example.com un example.org portā 8443:

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

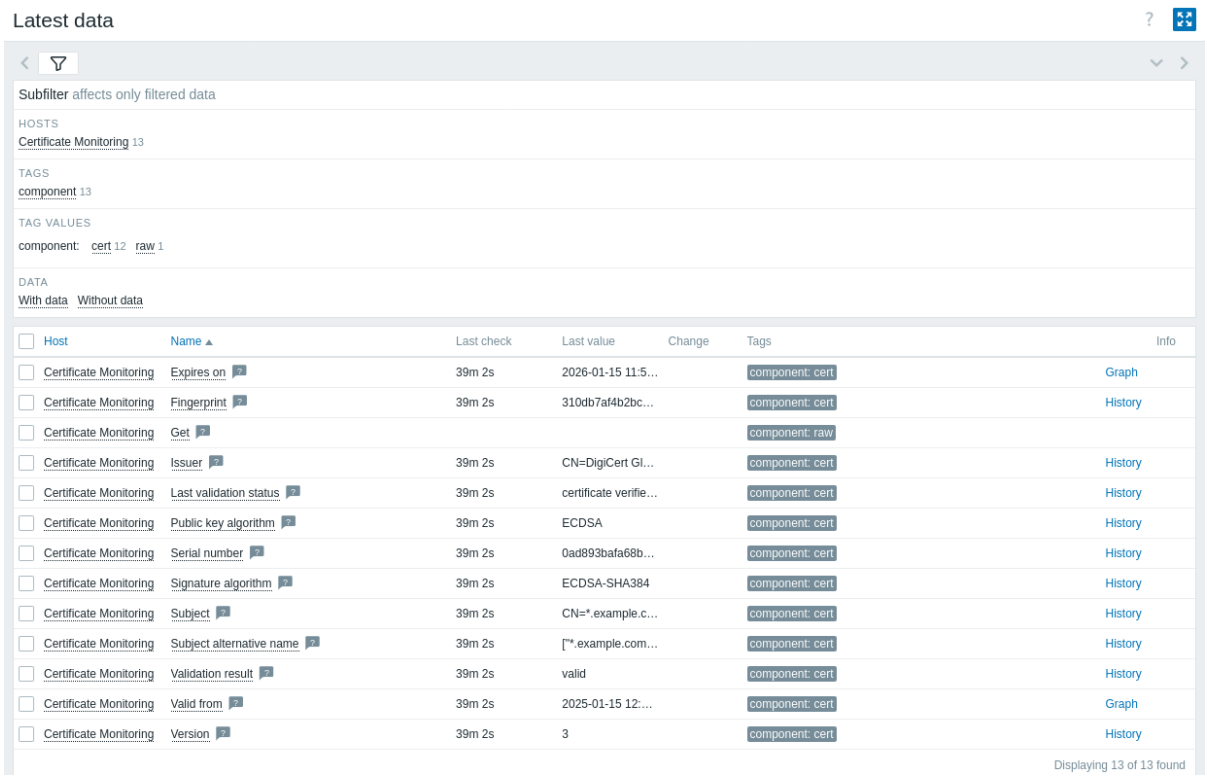
Katram vietnes ierakstam, kas norādīts makrosā `{$CERT.WEBSITE.HOSTNAME}`, Zabbix izveidos atbilstošu vienumu un trigeru kopu. Tas ļauj veikt atsevišķu uzraudzību un brīdināšanu katras vietnes SSL sertifikātam.

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:…

Skatīt savāktos metriku datus

Apsveicam! Šajā brīdī Zabbix jau uzrauga vēlamo tīmekļa sertifikātu.

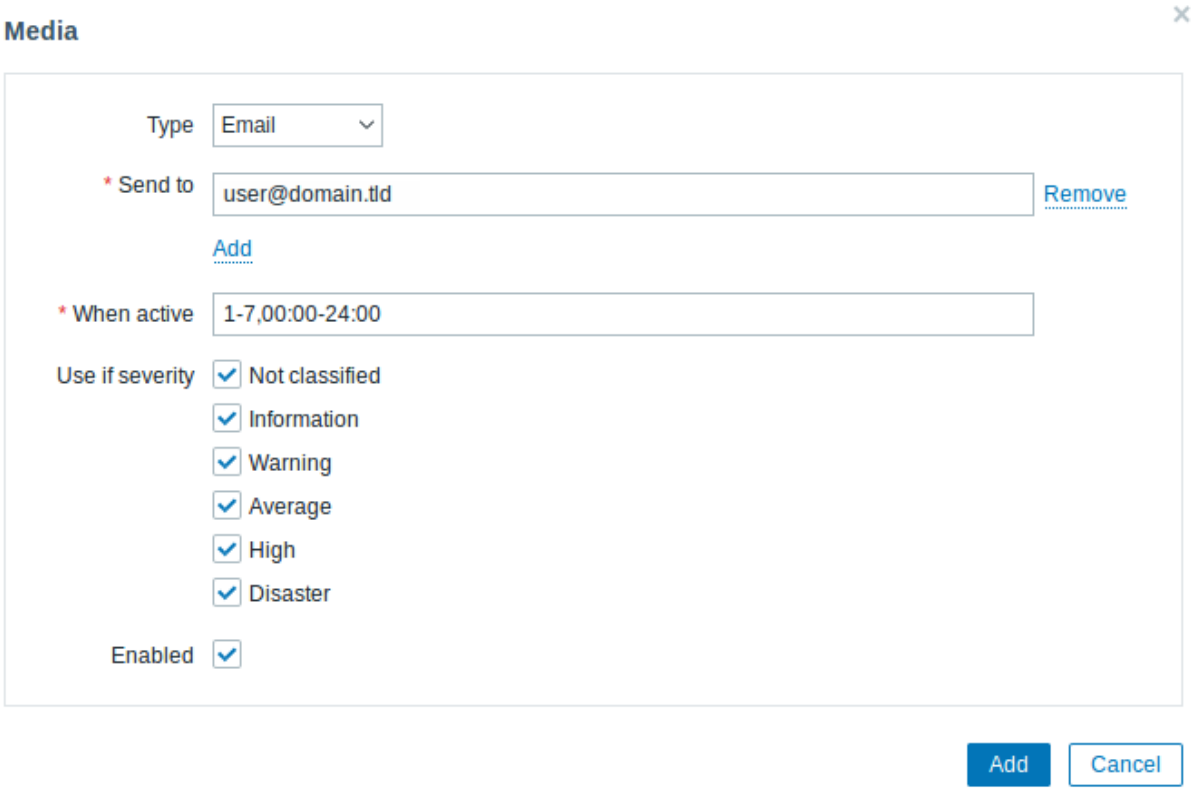
Lai skatītu savāktos metriku datus, dodieties uz izvēlnes sadaļu *Uzraudzība > Hosti* un noklikšķiniet uz *Jaunākie dati* blakus hostam, lai sarakstā skatītu visus jaunākos savāktos metriku datus, piemēram, derīguma termiņu, izdevēju un subjektu.



Iestatiet problēmu brīdinājumus

Zabbix var jūs informēt par infrastruktūras problēmām. Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.



2. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju

Lai pārbaudītu savu konfigurāciju, varam simulēt reālu problēmu, atjauninot hosta konfigurāciju Zabbix lietotāja saskarnē.

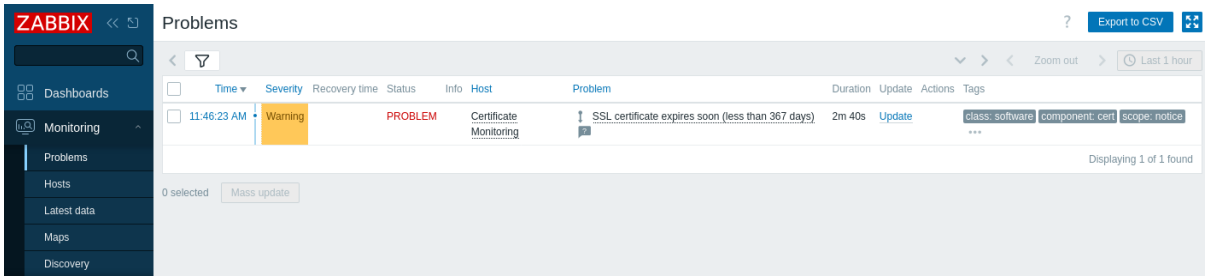
1. Atveriet sava ”Certificate Monitoring” hosta konfigurāciju Zabbix.

2. Pārslēdzieties uz cilni *Macros* un atlasiet *Inherited and host macros*.

3. Noklikšķiniet uz *Change* blakus iepriekš konfigurētajai makro `{$CERT.EXPIRY.WARN}` vērtībai un iestatiet ļoti lielu dienu skaitu (vajadzētu pietikt ar vairāk nekā 365 dienām), lai saņemtu brīdinājumu pirms sertifikāta derīguma termiņa beigām.

4. Noklikšķiniet uz *Update*, lai atjauninātu hosta konfigurāciju.

5. Pēc neilga brīža Zabbix noteiks problēmu "SSL certificate expires soon", norādot dienu skaitu līdz derīguma termiņa beigām. Problēma būs redzama sadaļā *Monitoring > Problems*.



Ja brīdinājumi ir konfigurēti, jūs saņemsiet arī problēmas paziņojumu.

6. Mainiet makro vērtību atpakaļ uz iepriekšējo vērtību, lai novērstu problēmu un turpinātu sertifikāta vērtību uzraudzību.

Skatiet arī

- Zabbix aģents 2 - uzskaita vienumu atslēgas.
- Veidne *Website certificate by Zabbix agent 2* - papildu informācija par veidni *Website certificate by Zabbix agent 2*.
- Veidne *Website certificate by Zabbix agent 2 active* - papildu informācija par veidni *Website certificate by Zabbix agent 2 active*.
- Vietņu uzraudzība ar Browser vienumiem - kā sākt vietņu pamata uzraudzību ar Browser vienumiem.

10 Tīkla komutatora vai maršrutētāja uzraudzība ar Zabbix

Ievads Šajā ceļvedī ir aprakstītas darbības, kas nepieciešamas, lai sāktu jūsu tīkla komutatora vai maršrutētāja pamata uzraudzību, izmantojot Zabbix. Kā piemērs ir izmantots Cisco maršrutētājs, taču procedūra attiecas uz jebkuru tīkla ierīci ar iespējotu SNMP.

Kam šis ceļvedis ir paredzēts

Šis ceļvedis ir paredzēts jauniem Zabbix lietotājiem un tīkla administratoriem, kuri vēlas ātri iespējot tīkla ierīču pamata uzraudzību. Ja jums nepieciešama padziļināta pielāgošana vai papildu konfigurācijas iespējas, lūdzu, skatiet SNMP aģents lapu vai Zabbix rokasgrāmatas Konfigurācija sadaļu.

Priekšnosacījumi

Pirms turpināt darbu ar šo ceļvedi, pārliecinieties, ka jums ir:

- instalēts Zabbix serveris un Zabbix lietotāja saskarne: instalējiet tos atbilstoši jūsu operētājsistēmas instrukcijām (skatiet Instalēšana no pakotnēm un Tīmekļa saskarnes instalēšana).
- instalēts Zabbix aģents, ja uzraugāt lokālā tīkla metriku.
- ierīce ar iespējotu SNMP: tīkla komutators vai maršrutētājs (piemēram, Cisco maršrutētājs) ar iespējotu SNMP.
- instalēti MIB faili: MIB failu instalēšana ļauj Zabbix pārvērst skaitliskos OID cilvēkiem saprotamos nosaukumos un aprakstos. Bez atbilstoša MIB atbalsta jūs varat redzēt tikai skaitliskas vērtības, kas apgrūtina vienumu konfigurēšanu un problēmu novēršanu.

Lai instalētu MIB failus Ubuntu sistēmā:

1. Instalējiet MIB lejupielādētāja pakotni:

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

Ja nepieciešams pievienot ražotājam specifiskus MIB (piemēram, no Cisco, Juniper), ievietojiet tos atbilstošajā MIB direktorijā:

- Linux sistēmās biežāk izmantotās atrašanās vietas ir /usr/share/snmp/mibs/ vai /usr/local/share/snmp/mibs/.
- Zabbix instalācijās MIB failus var glabāt direktorijā /var/lib/zabbix/mibs/.

Pārliecinieties, ka vides mainīgais MIBDIRS vai fails snmp.conf ietver pareizo ceļu.

Lai pārbaudītu, vai jūsu sistēma atpazīst jaunos MIB, izmantojiet:

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

Detalizētākas instrukcijas skatiet savas SNMP bibliotēkas dokumentācijā:

- Cisco MIBs
- Juniper MIBs

2. Rediģējiet `/etc/snmp/snmp.conf` un izkomentējiet rindu, kas sākas ar `mibs :`, lai sistēma ielādētu visus pieejamos MIB.

3. Pārbaudiet, izpildot `snmpwalk` (piemēram, `snmpwalk -v 2c -c <your_community_string> <device_IP>`), un pārliecinieties, ka OID tiek attēloti ar aprakstošiem nosaukumiem.

Atkarībā no jūsu vides daži šī ceļveža soļi var nedaudz atšķirties. Šis ceļvedis ir balstīts uz vidi, kurā darbojas Ubuntu un tiek uzraudzīta Cisco Catalyst 3750V2-24FS tīkla ierīce.

Tiek pieņemts, ka jūsu tīkla ierīce jau ir fiziski uzstādīta un pievienota.

Konfigurējiet tīkla ierīci (Cisco maršrutētāja piemērs) Lai uzraudzītu, izmantojot SNMP, jums jākonfigurē tīkla ierīce tā, lai tā atļautu SNMP vaicājumus. Tālāk redzamais piemērs ir paredzēts SNMPv2 un neņem vērā esošos iestatījumus. Uzmanību: šo komandu piemērošana var pārrakstīt pašreizējās SNMP konfigurācijas.

Cisco maršrutētāja konfigurēšana parasti ietver tālāk aprakstītās darbības.

SNMPv2 piemērs 1. Iespējojiet SNMP un iestatiet kopienas virkni.

Piesakieties sava Cisco maršrutētāja konsolē un ieejiet konfigurēšanas režīmā:

```
configure terminal
```

Pēc tam iespējojiet SNMP, norādot tikai lasāmu kopienas virkni. Piemēram:

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

Aizstājiet `<your_community_string>` ar savu drošo kopienas virkni. Piezīme: opcija RO (Read-Only) ļauj SNMP iegūt datus no ierīces, bet neļauj veikt nekādas konfigurācijas izmaiņas.

Drošības apsvērumu dēļ ieteicams ierobežot SNMP piekļuvi tikai nepieciešamajām ierīcēm. Papildu norādījumus par piekļuves kontroles sarakstu (ACL) konfigurēšanu skatiet Cisco oficiālajā dokumentācijā.

2. Saglabājiet konfigurāciju.

Saglabājiet izmaiņas, lai nodrošinātu, ka SNMP iestatījumi saglabājas arī pēc pārstartēšanas:

```
write memory
```

SNMPv3 piemērs SNMPv3 nodrošina uzlabotu drošību ar autentifikāciju un šifrēšanu. Tā konfigurācija ir drošāka nekā SNMPv2, un tā jāpārbauda atbilstoši jūsu ierīces specifiskajai dokumentācijai.

1. Izveidojiet SNMP grupu.

Konfigurējiet SNMPv3 grupu ar ieslēgtu privātumu (šifrēšanu):

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

2. Izveidojiet SNMP lietotāju.

Pievienojiet SNMPv3 lietotāju ar autentifikāciju un privātumu. Aizstājiet vietturus ar vēlamajām vērtībām:

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

3. Saglabājiet konfigurāciju:

```
write memory
```

Papildinformācijai vai konkrētam modelim paredzētām instrukcijām varat skatīt ārējās Cisco SNMP konfigurācijas pamācības. Tomēr šajā ceļvedī ir sniegti pamata soļi SNMP uzraudzības iespējošanai.

Konfigurēt Zabbix lietotāja saskarni

Izveidojiet hostu Zabbix lietotāja saskarnē 1. Piesakieties Zabbix lietotāja saskarnē.

2. Pievienojiet jaunu hostu.

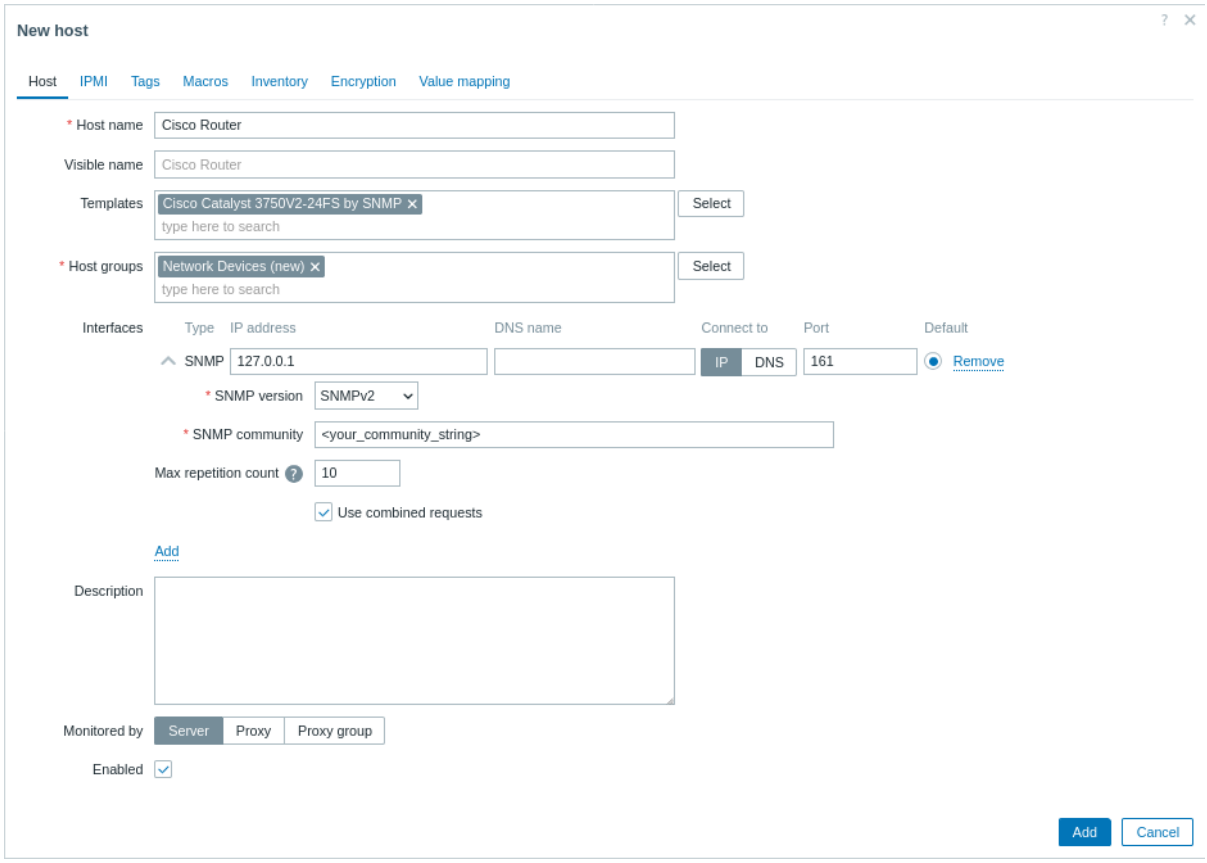
Dodieties uz *Datu ievākšana* > *Hosti* un noklikšķiniet uz *Izveidot hostu*.

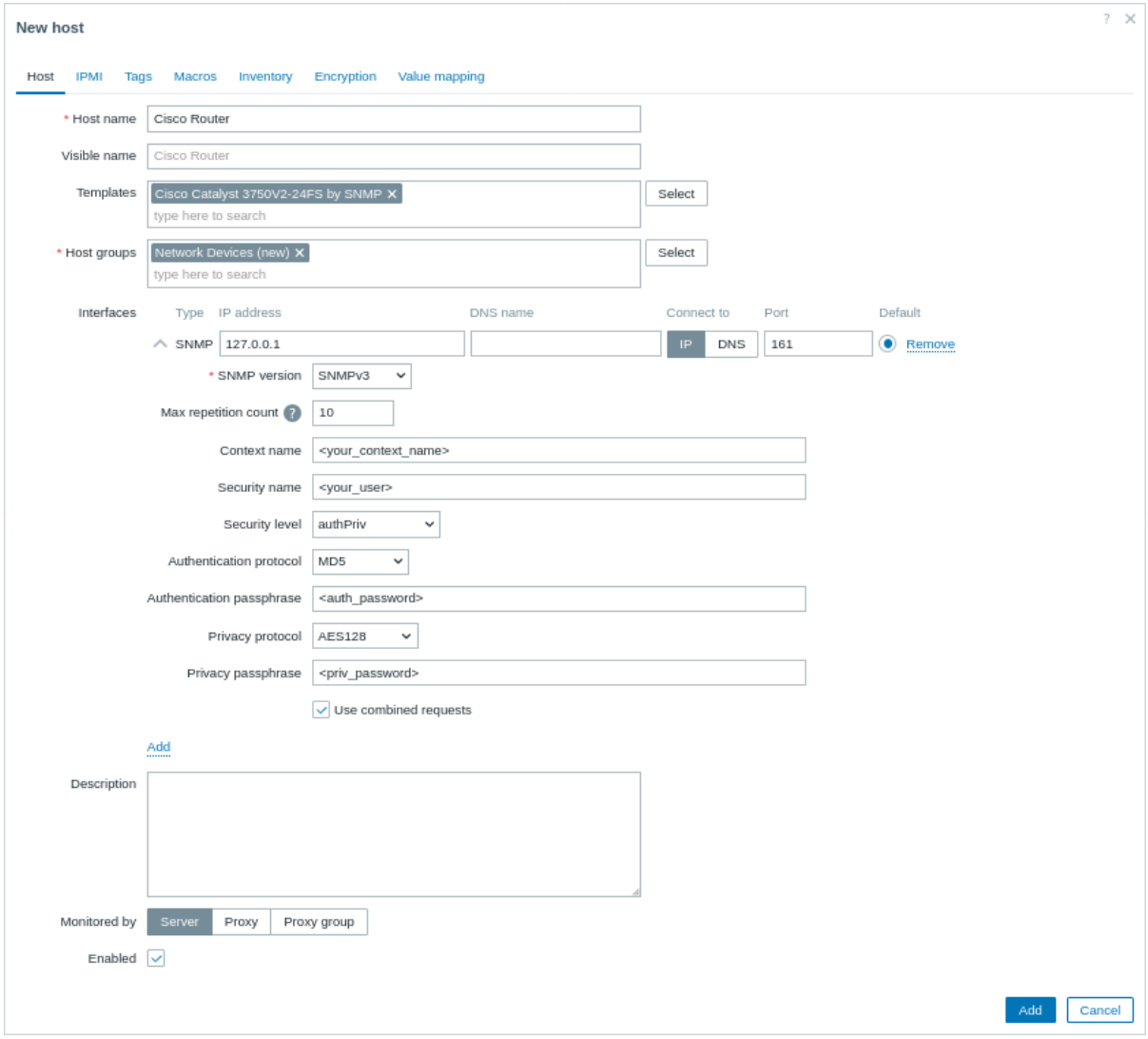
- *Hosta nosaukums*: ievadiet savas ierīces nosaukumu (piemēram, ”Cisco maršrutētājs”).
- Hosta grupas: atlasiet esošu grupu vai izveidojiet jaunu grupu, piemēram, ”Tīkla ierīces”.
- Saskarnes:
  - Noklikšķiniet uz *Pievienot* sadaļā Saskarnes.
  - Izvēlieties *SNMP* kā saskarnes tipu.
  - Ievadiet sava Cisco maršrutētāja IP adresi vai DNS nosaukumu.
  - Iestatiet noklusējuma SNMP portu (parasti 161).
  - Izmantojiet nolaižamo izvēlni, lai atlasītu atbilstošo SNMP versiju (piem., SNMPv2).
  - SNMPv1/v2 gadījumā ievadiet kopienas virkni laukā *SNMP kopiena*. SNMPv3 gadījumā tiks pieprasīti papildu akreditācijas dati (*Konteksta nosaukums*, *Drošības nosaukums* un *Drošības līmenis* u.c.).

3. Saistiet veidnes

Laukā *Veidnes* atlasiet SNMP veidni, kas vislabāk atbilst jūsu ierīcei. Zabbix nodrošina plašu iepriekš izveidotu SNMP veidņu klāstu daudzām ierīču saimēm. Piemēram, ja uzraugāt Cisco ierīci, izvēlieties veidni, kas atbilst jūsu ierīces OS vai modelim (piemēram, Cisco IOS SNMP vai Cisco Catalyst 3750<ierīces modelis> SNMP).

4. Noklikšķiniet uz *Pievienot*, lai saglabātu hostu.

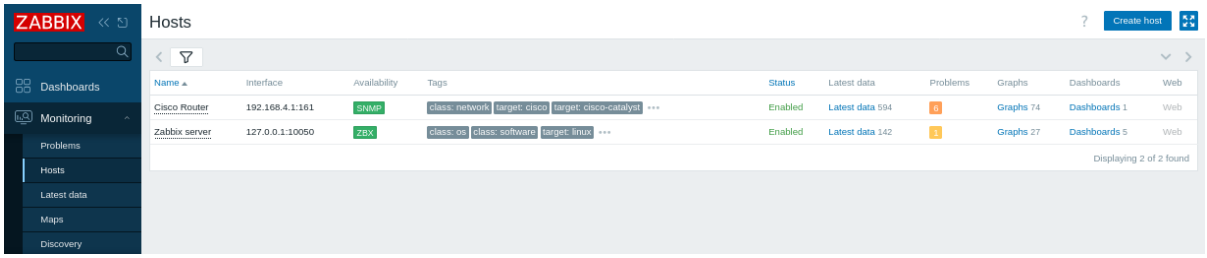




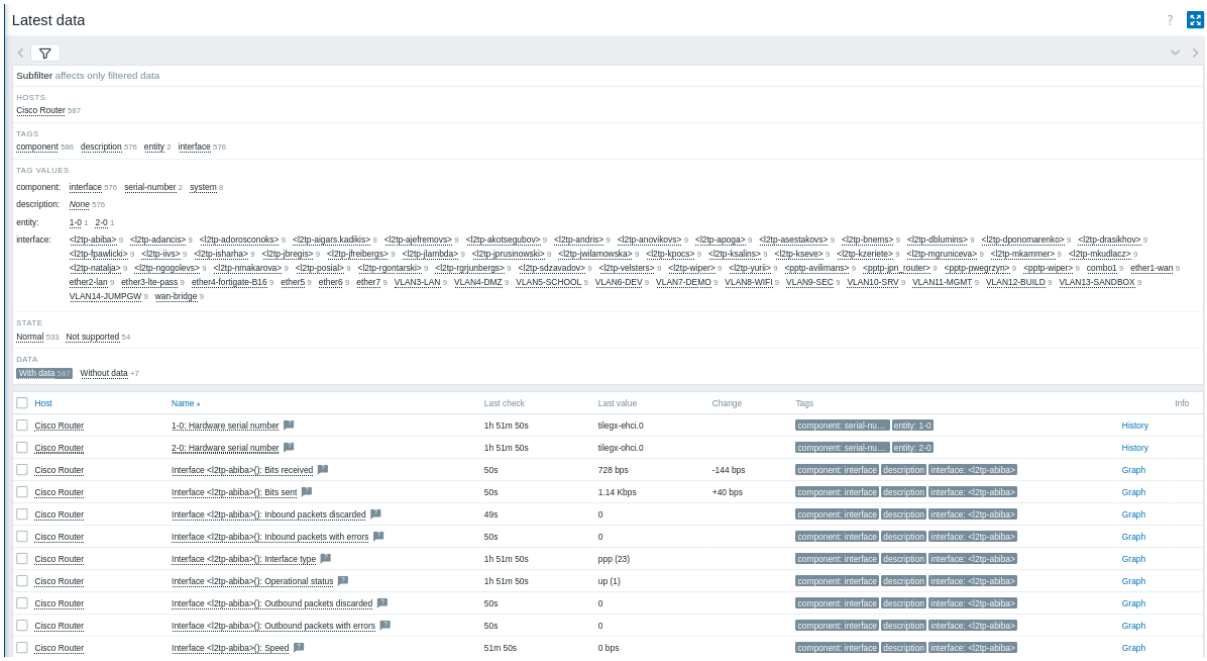
Skatīt savāktos metriku datus Apsveicam! Zabbix tagad ir iestatīts, lai uzraudzītu jūsu tīkla ierīci.

Jaunākie dati:

- Dodieties uz Monitoring > Latest data Zabbix lietotāja saskarnē.



- Atlasiet savu “Cisco Router” hostu (vai atklātos hostus), lai skatītu tādus metriku datus kā aparatūras un tīkla darbības laiks, ICMP zudumi, ping un atbildes laiks u.c.



- Grafiki un ekrāni:

Lai vizualizētu veiktspējas datus, noklikšķiniet uz *Graphs* blakus SNMP vienumiem, lai skatītu detalizētus metriku datus.

Kā nākamo soli varat:

- Pievienot pielāgotus SNMP vienumus, lai uzraudzītu papildu metriku datus.
- Iestatīt problēmu brīdinājumus, lai saņemtu paziņojumus par iespējamām problēmām.

Izveidojiet SNMP vienumus Kad hosts ir iestatīts, varat izveidot vienumus, lai uzraudzītu konkrētus rādītājus. Piezīme: šī darbība nav obligāta, ja izmantojat veidni, jo veidnes jau satur noklusējuma vienumu kopas.

1. Nosakiet SNMP OID:

Izmantojiet komandu `snmpwalk`, lai uzskaitītu jūsu ierīcē pieejamos OID. Piemēram:

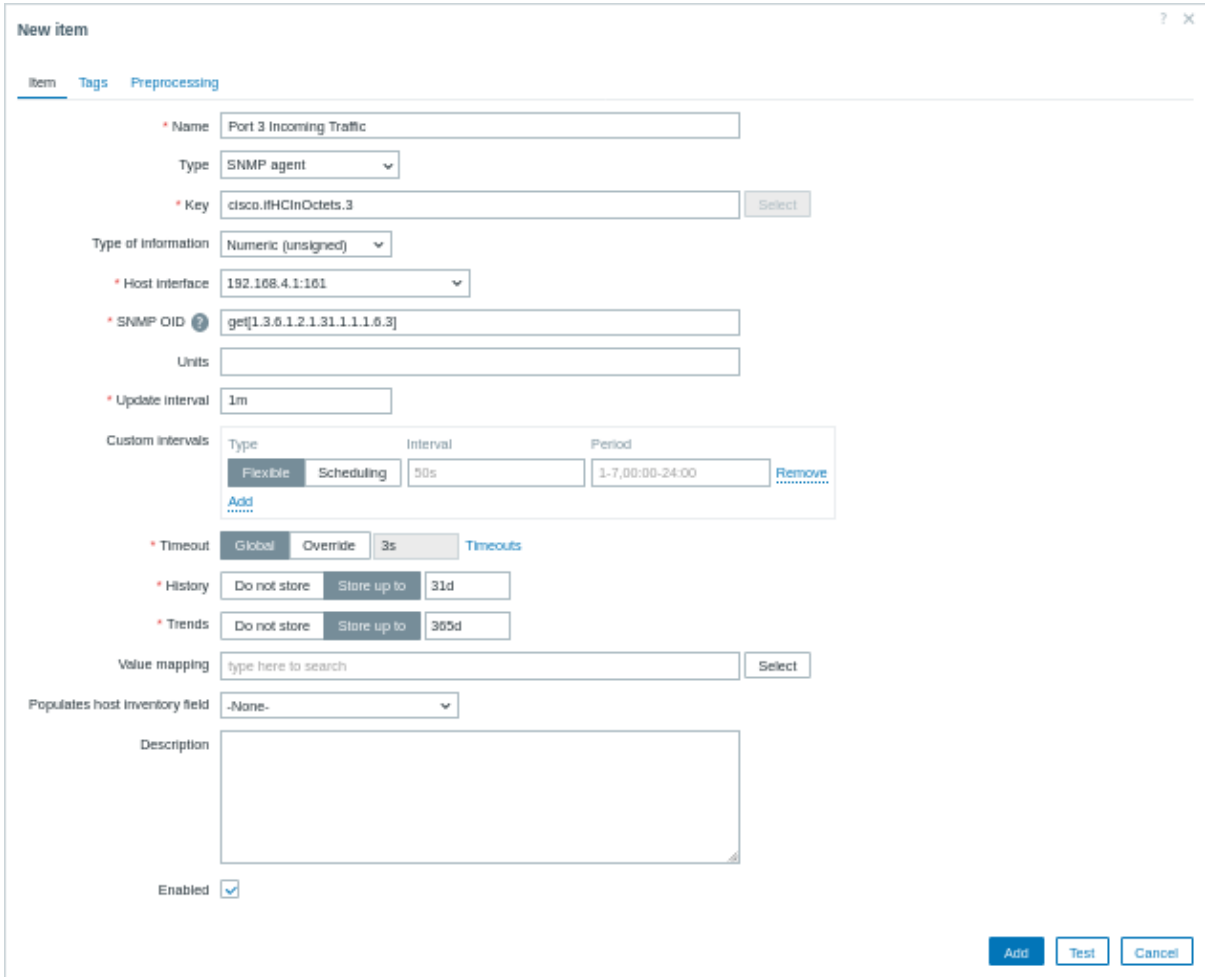
```
snmpwalk -v 2c -c <your_community_string> <device_IP> .
```

Atrodiet OID rādītājam, kuru vēlaties uzraudzīt (piemēram, IF-MIB::ifHCInOctets.3 ienākošajai datplūsmai 3. portā). Lai iegūtu skaitlisko OID, varat izmantot:

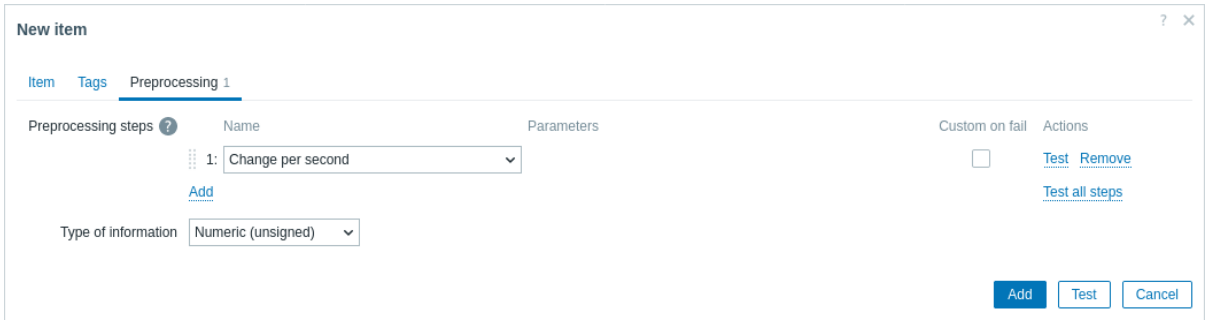
```
snmpget -v 2c -c <your_community_string> -On <device_IP> IF-MIB::ifHCInOctets.3
```

2. Izveidojiet SNMP vienumu:

- Atveriet *Data collection* > *Hosts* un savam SNMP hostam noklikšķiniet uz cilnes *Items*, pēc tam noklikšķiniet uz *Create item*.
- *Name*: ievadiet aprakstošu nosaukumu (piemēram, "3. porta ienākošā datplūsma").
- *Type*: atlasiet *SNMP agent*.
- *Key*: norādiet saprotamu atslēgu (piemēram, `cisco.ifHCInOctets.3`).
- *Host interface*: pārliecinieties, ka ir atlasīta SNMP saskarne.
- *SNMP OID*: ievadiet OID, izmantojot vienu no atbalstītajiem formātiem, piemēram:
  - `get[1.3.6.1.2.1.31.1.1.1.6.3]` vienai vērtībai;
  - `walk[1.3.6.1.2.1.31.1.1.1.6.3]` lai asinhroni izgūtu vērtību apakškoku.



- *Preprocessing* (ja nepieciešams): ja vienums atgriež kumulatīvu skaitītāju (piemēram, saskarnes datplūsmu), atveriet cilni *Preprocessing* un pievienojiet priekšapstrādes soli, piemēram, “Change per second”, lai aprēķinātu ātrumu.



Lai vienā SNMP transakcijā izgūtu vairākas vērtības, varat norādīt vairākus OID, izmantojot sintaksi `walk[OID1,OID2,...]`.

OID tulkošana starp skaitliskajiem un MIB nosaukumiem Strādājot ar SNMP, jums var būt nepieciešams konvertēt starp skaitliskajiem OID un tiem atbilstošajiem MIB nosaukumiem. Šī tulkošana palīdz vieglāk identificēt un novērst problēmas ar metrikām.

- MIB nosaukuma tulkošana uz skaitlisku OID: izmantojiet komandu `snmptranslate` ar opciju `-On`. Piemēram, lai pārtulkotu MIB nosaukumu `IF-MIB::ifHCInOctets.3` uz tā skaitlisko OID, izpildiet:

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

Šī komanda var izvadīt:

```
.1.3.6.1.2.1.31.1.1.1.6.3
```

- Skaitliska OID tulkošana uz tā MIB nosaukumu: izmantojiet komandu `snmptranslate` ar opciju `-IR` (vai `-m ALL`), lai veiktu apgriezto tulkošanu. Piemēram, lai pārtulkotu skaitlisko OID `.1.3.6.1.2.1.31.1.1.1.6.3` atpakaļ uz tā MIB nosaukumu, izpildiet:

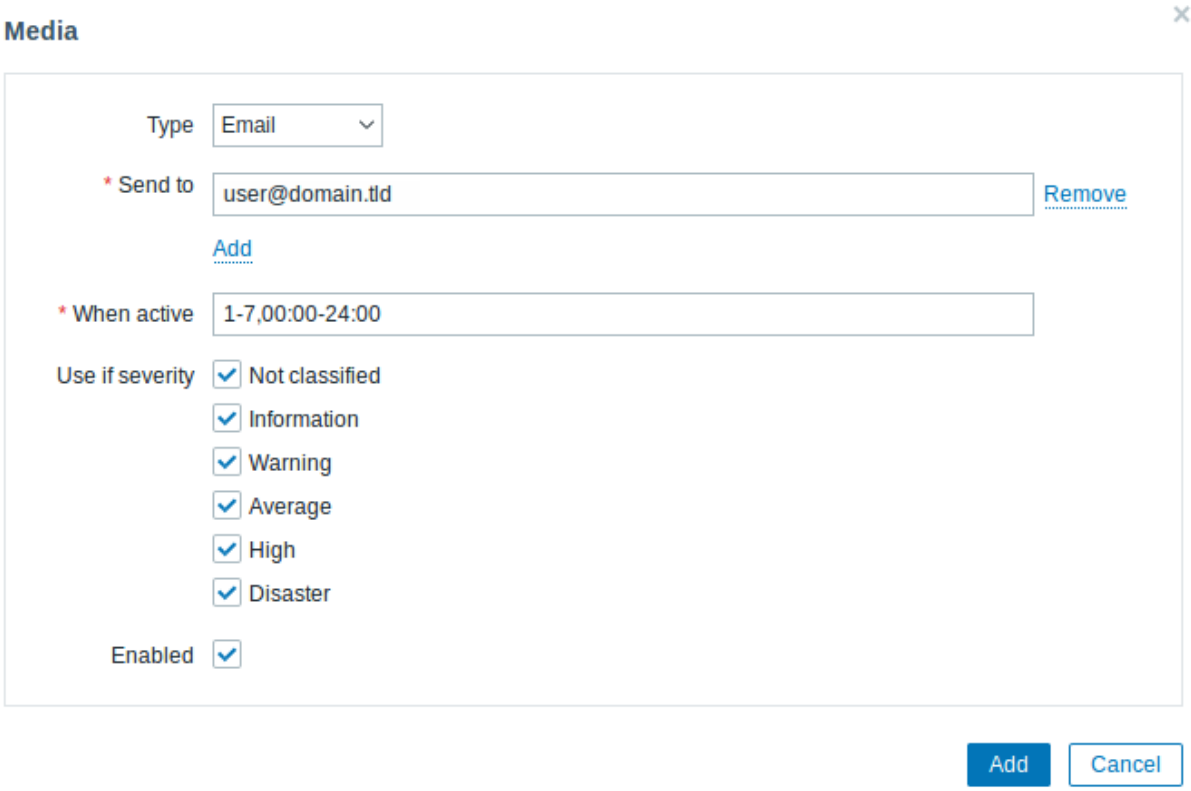
```
snmptranslate -IR -On .1.3.6.1.2.1.31.1.1.1.6.3
```

Šī komanda var izvadīt:

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

Iestatiet problēmu brīdinājumus Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu sūtīšanai.

1. Atveriet *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pasta adresi.

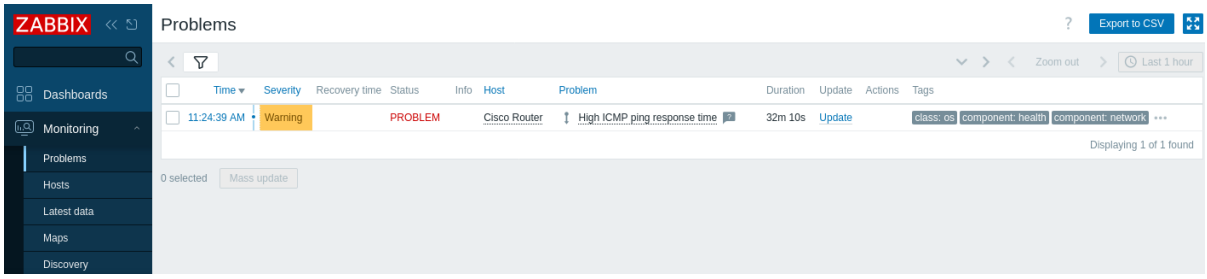


2. Izpildiet ceļvedi par problēmas paziņojuma saņemšanu.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Pārbaudiet savu konfigurāciju Lai nodrošinātu, ka Zabbix pareizi nosaka tīkla veiktspējas problēmas, simulējiet reālu problēmu, palielinot ICMP ping atbildes laika slieksni.

1. Atveriet sava "Cisco Router" hosta konfigurāciju Zabbix.
2. Dodieties uz cilni *Macros* un atlasiet *Inherited and host macros*.
3. Atrodiet makrosu `{$ICMP_RESPONSE_TIME_WARN}` (vai līdzīgu atbildes laika sliekšņa makrosu).
4. Iestatiet ļoti zemu vērtību (piemēram, 0.001), lai aktivizētu brīdinājumu, kad ping atbildes laiks pārsniedz šo vērtību.
5. Noklikšķiniet uz *Update*, lai piemērotu izmaiņas.
6. Uzgaidiet dažus mirkļus, līdz Zabbix nosaka simulēto problēmu.
7. Dodieties uz *Monitoring > Problems*, lai pārbaudītu, vai parādās brīdinājums (piemēram, "High ICMP ping response time").



Ja brīdinājumi ir konfigurēti, jums būtu jāsaņem arī problēmas brīdinājums.

8. Atjaunojiet makrosa vērtību uz tās sākotnējo iestatījumu un noklikšķiniet uz *Update*, lai saglabātu izmaiņas.
9. Pārliecinieties, ka problēma ir novērsta un pazūd no sadaļas *Problems*.

SNMP uzraudzības problēmu novēršana Ja pamanāt, ka SNMP ikona Zabbix lietotāja saskarnē ir SARKANA vai dati netiek vākti, izmēģiniet tālāk norādītās darbības:

1. Pārbaudiet SNMP savienojamību.

SNMPv2 gadījumā palaidiet šādu komandu no sava Zabbix servera:

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

Šī komanda pārbauda, vai ierīce atbild uz SNMP vaicājumiem.

SNMPv3 gadījumā iekļaujiet atbilstošos SNMPv3 akreditācijas datus:

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

Tas pārbauda, vai SNMPv3 akreditācijas dati ir pareizi un ierīce droši atbild.

2. Pārliecinieties, ka MIB faili ir instalēti un iespējoti, kā aprakstīts priekšnosacījumos. Lai to pārbaudītu, šādai komandai, vaicājot tīkla ierīci, nevajadzētu atgriezt kļūdu:

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

Tai būtu jāatgriež tulkoti OID bez kļūdām.

3. Apstipriniet, ka Zabbix konfigurētā SNMP versija un akreditācijas dati atbilst tiem, kas iestatīti jūsu ierīcē. Piemēram, pārskatiet SNMP iestatījumus Zabbix hosta konfigurācijā un salīdziniet tos ar jūsu ierīces konfigurāciju. Cisco ierīcē SNMP iestatījumus var pārbaudīt, izpildot:

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

Tas nodrošina, ka kopienas virkne (SNMPv2 gadījumā) vai SNMPv3 lietotāja informācija ir pareiza.

4. Pārbaudiet, vai SNMP ir pareizi iespējots jūsu tīkla ierīcē. Cisco maršrutētājā piesakieties konsolē un izpildiet:

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

Šī komanda parāda aktīvo SNMP konfigurāciju un palīdz apstiprināt, ka SNMP ir pareizi konfigurēts.

5. Pārliecinieties, ka ne ugunsmūri, ne tīkla problēmas nebloķē SNMP datplūsmu (parasti 161. portā) starp Zabbix serveri un ierīci. Savienojamību var pārbaudīt, izmantojot:

nc -zv <device_IP> 161

`nc -zv` pārbauda, vai 161. ports ierīcē ir atvērts un klausās.

Papildus, ja Ubuntu izmantojat UFW, pārbaudiet ugunsmūra statusu:

```
sudo ufw status
```

Vai arī iptables gadījumā:

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

6. Pārskatiet Zabbix servera žurnālfailus, lai atrastu ar SNMP saistītas kļūdas un palīdzētu precīzāk noteikt problēmu:

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

`tail -f` ļauj reāllaikā uzraudzīt žurnāla atjauninājumus.

Skatiet arī:

- Vienuma izveide — uzziniet, kā pievienot papildu metriku.
- SNMP aģents — papildu informācija par SNMP uzraudzību ar Zabbix.
- Standartizētas veidnes tīkla ierīcēm — informācija par pieejamajām SNMP veidnēm.
- SNMP OID atklāšana — papildu informācija par SNMP atklāšanu komutatorā.
- Tīkla atklāšanas kārtulas konfigurēšana — papildu informācija par to, kā konfigurēt tīkla atklāšanas kārtulu, ko Zabbix izmanto, lai atklātu hostus un pakalpojumus.

11 Windows notikumu žurnāla uzraudzība, izmantojot aktīvās pārbaudes

Ievads

Šajā ceļvedī ir skaidrots, kā ar Zabbix palīdzību uzraudzīt Windows notikumu žurnālus, izmantojot aktīvās pārbaudes. Izmantojot Zabbix Windows specifiskās vienumu atslēgas, varat reāllaikā vākt un analizēt kritiskus notikumus (piemēram, neveiksmīgus pieteikšanās mēģinājumus, sistēmas kļūdas u.c.).

Kam šis ceļvedis ir paredzēts

Šis ceļvedis ir paredzēts jauniem Zabbix lietotājiem un tīkla administratoriem, kuri vēlas uzraudzīt Windows notikumu žurnālus. Papildu konfigurācijas iespējām skatiet dokumentāciju par Windows specifiskajām vienumu atslēgām.

Priekšnosacījumi

Pirms turpināt darbu ar šo ceļvedi, jums ir jālejupielādē un jāinstalē Zabbix serveris un Zabbix lietotāja saskarne atbilstoši jūsu OS paredzētajām instrukcijām. Jums arī jābūt lejupielādētam un instalētam Zabbix aģentam Windows datorā, kuru vēlaties uzraudzīt.

Konfigurējiet Zabbix aģents Windows notikumu žurnāla uzraudzībai

1. Atveriet `zabbix_agentd.conf` (noklusējuma ceļš `C:\Program Files\Zabbix Agent\zabbix_agentd.conf`) savā Windows hostā un pārliecinieties, ka parametrs ServerActive ir iestatīts uz jūsu Zabbix serveris IP adresi un parametrs Hostname atbilst hosta nosaukumam, kas tiks definēts Zabbix lietotāja saskarne. Tas ļauj aģents pieprasīt aktīvās pārbaudes savam hostam no norādītā Zabbix serveris. Piemēram:

```
ServerActive=192.0.2.0
Hostname=MyWindowsHost
```

2. Restartējiet Zabbix aģents servisu, lai piemērotu izmaiņas:

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

3. Pārbaudiet, vai Windows hosts darbojas:

- Pārliecinieties, ka Zabbix aģents serviss darbojas Windows hostā.
- Pārbaudiet, vai Windows hosts var izveidot savienojumu ar Zabbix serveris 10051. portā. Lai pārbaudītu savienojamību no Windows hosta, atveriet PowerShell un izpildiet šādu komandu:

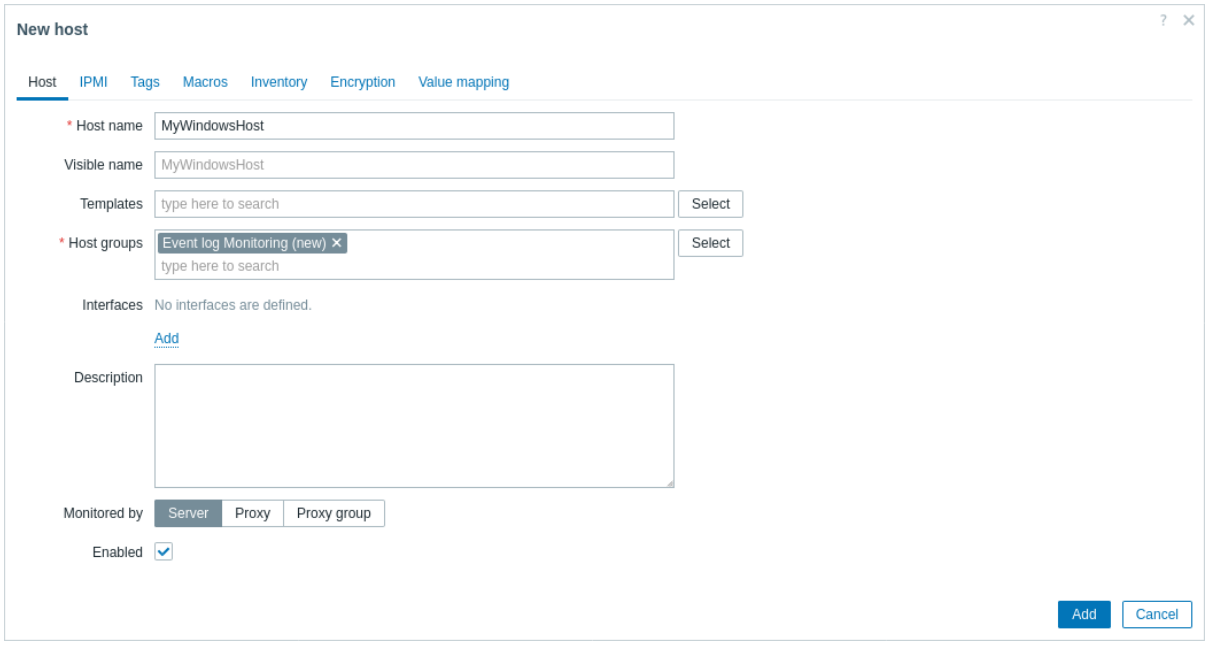
```
Test-NetConnection -ComputerName <Zabbix-server-IP> -Port 10051
```

Konfigurējiet Zabbix lietotāja saskarni

1. Dodieties uz *Datu vākšana* > *Hosts* un izveidojiet hostu:

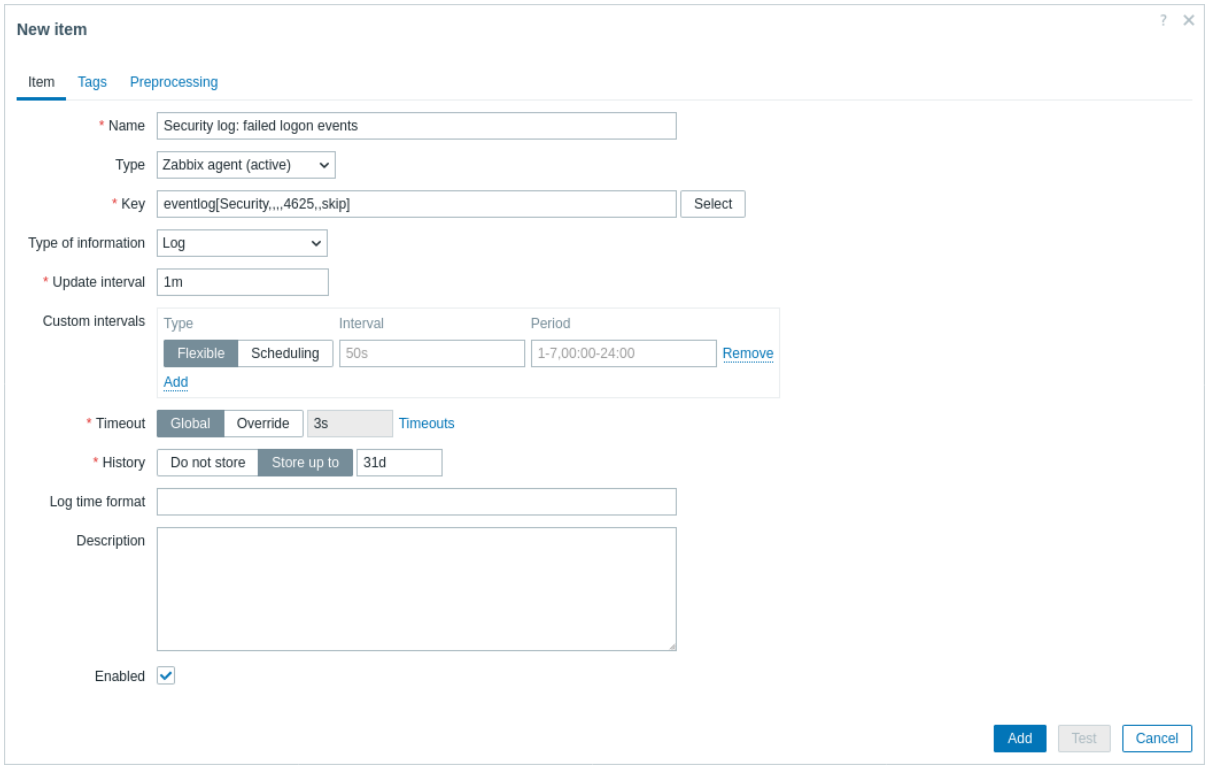
- Laukā *Host name* ievadiet hosta nosaukumu (piemēram, ”MyWindowsHost”).
- Laukā *Host groups* ierakstiet vai atlasiet hostu grupu (piemēram, ”Event log Monitoring”).
- Nospiediet *Add*, lai saglabātu konfigurēto hostu.

Note:
Laukā *Templates* varat pievienot veidni ”Windows by Zabbix aģents active”, lai palīdzētu novērst problēmas, novērojot, vai citi aktīvie vienumi tajā pašā hostā tiek atjaunināti.



2. Izveidojiet jaunu vienumu ar šādiem parametriem:

- Laukā *Name* ievadiet aprakstošu vienuma nosaukumu (piemēram, ”Security log: failed logon events”).
- Nolaižamajā sarakstā *Type* atlasiet ”Zabbix aģents (active)” (nepieciešams Event log uzraudzībai).
- Laukā *Key* izmantojiet vienuma atslēgu eventlog. Piemēram, lai uzraudzītu neveiksmīgus pieteikšanās mēģinājumus (Event ID: 4625) Security žurnālā un ignorētu ierakstus, kas ir vecāki par vienuma pēdējo pārbaudi (izmantojot parametru `skip`), ievadiet šādu vienuma atslēgu: `eventlog[Security,,,,4625,,skip]`
- Nolaižamajā sarakstā *Type of information* atlasiet ”Log”.



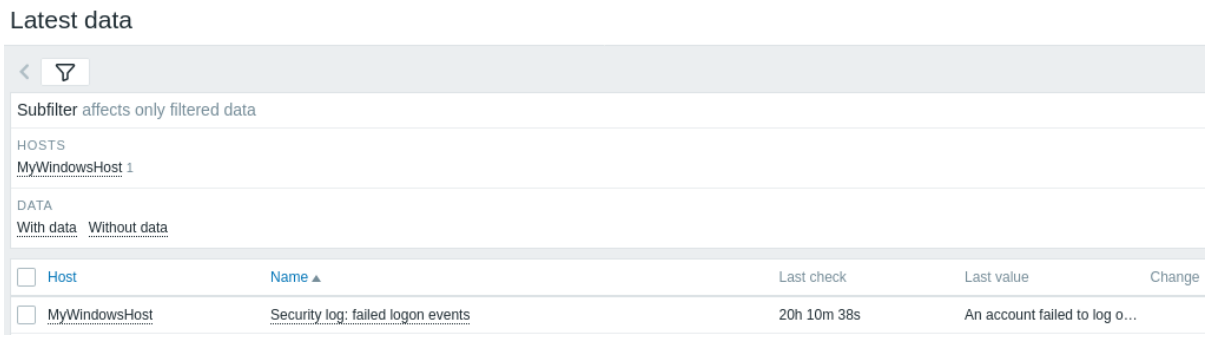
3. Noklikšķiniet uz *Add*, lai saglabātu vienumu.

Pārbaudiet un skatiet savāktos metriku datus

Apsveicam! Zabbix tagad ir iestatīts, lai vāktu jūsu Windows notikumu žurnālus. Lai pārbaudītu, vai notikumu žurnāli tiek vākti, varat pārbaudīt vienumu ”Drošības žurnāls: neveiksmīgi pieteikšanās notikumi”, izrakstoties no sava Windows konta un mēģinot pieteikties, izmantojot nepareizus akreditācijas datus.

Pēc tam skatiet savāktos žurnālus Zabbix lietotāja saskarnē:

1. Atveriet *Monitoring* > *Latest data* Zabbix lietotāja saskarnē.



2. Laukā *Name* filtrējiet pēc sava hosta ”MyWindowsHost”.

3. Noklikšķiniet uz *History*, lai skatītu reģistrētās žurnāla vērtības.

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. Ja žurnāla vērtības nav redzamas, pārejiet uz rokasgrāmatas sadaļu Traucējummeklēšana.

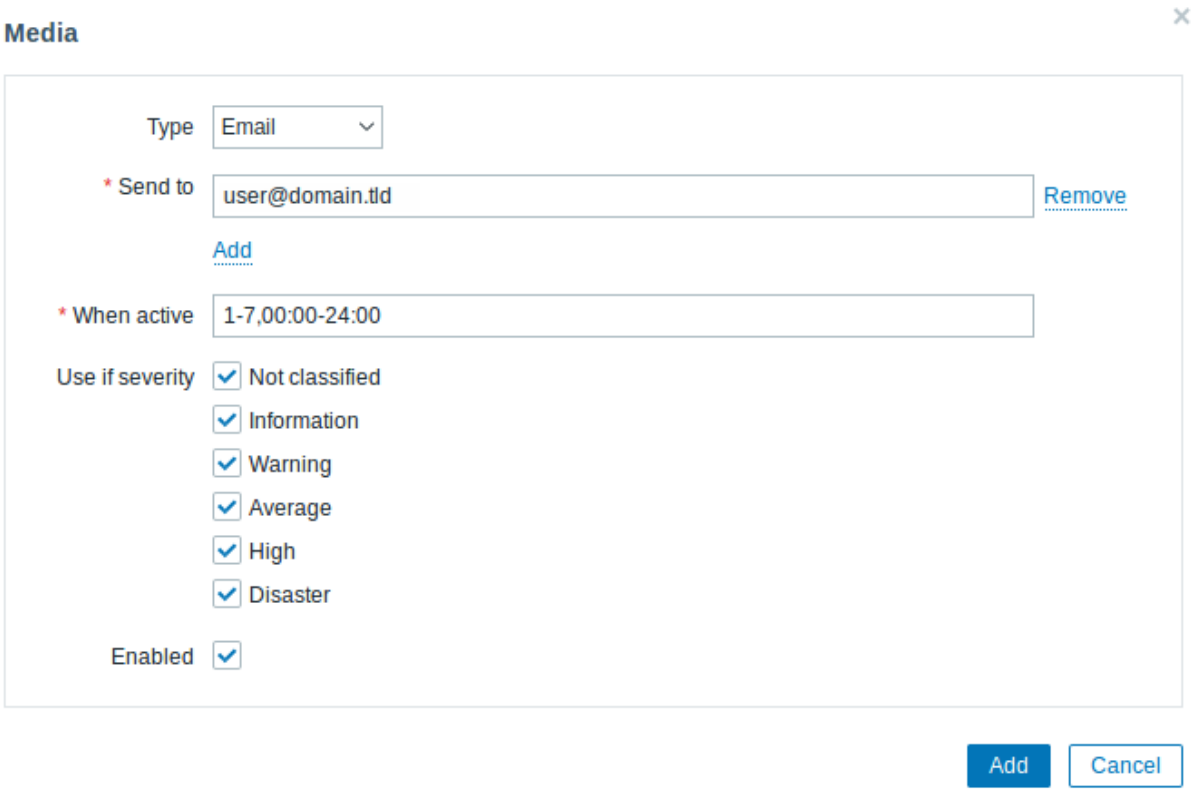
Iestatiet problēmu brīdinājumus

Šajā ceļvedī ir sniegti pamata konfigurācijas soļi e-pasta brīdinājumu sūtīšanai.

1. Dodieties uz Datu ievākšana > Hosti, lai definētu trigeri, kas nostrādā, kad jūsu notikumu žurnāla vienums reģistrē jūs interesējošo šablonu. Piemēram, lai uztvertu neveiksmīgus pieteikšanās mēģinājumus Security žurnālā, izmantojiet funkciju find():

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

2. Dodieties uz *Lietotāja iestatījumi > Profils*, pārslēdzieties uz cilni *Mediji* un pievienojiet savu e-pastu.



3. Izpildiet ceļvedi Problēmas paziņojuma saņemšana.

Nākamreiz, kad Zabbix konstatēs problēmu, jums vajadzētu saņemt brīdinājumu pa e-pastu.

Problēmu novēršana

Ja rodas problēmas ar Windows notikumu žurnālu vākšanu vai skatīšanu, izmantojiet tālāk sniegtos padomus, lai noteiktu un novērstu biežāk sastopamās problēmas:

1. Zabbix serverī (Linux) uzskaitiet savus iptables noteikumus ar šādu komandu:

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

un pārbaudiet, vai ir ACCEPT noteikums TCP portam 10051.

2. Pārliecinieties, ka jūsu `eventlog[...]` atslēga izmanto precīzu žurnāla nosaukumu (reģistrjutīgs), notikuma ID, režīmu (piemēram, skip) un citus parametrus tieši tā, kā parādīts sadaļā Windows-specific item keys.

Skatiet arī:

- Vienuma izveide - uzziniet, kā pievienot papildu metriku.
- Zabbix aģents Microsoft Windows vidē - detalizētas instalēšanas instrukcijas.
- Windows uzraudzība ar Zabbix aģentu - visaptveroša rokasgrāmata par pamata uzraudzības iestatīšanu Windows datoriem, izmantojot Zabbix aģentu.
- Windows specifiskās vienumu atslēgas - detalizēta informācija par Windows specifiskajām vienumu atslēgām, ko atbalsta Zabbix aģenti, tostarp notikumu žurnāla uzraudzībai paredzētajām atslēgām.
- Žurnālfailu uzraudzība - norādījumi par Zabbix konfigurēšanu centralizētai žurnālfailu uzraudzībai un analīzei, kas attiecas arī uz Windows notikumu žurnāliem.

Kas ir Zabbix

Zabbix izveidoja Aleksejs Vladiševs, un pašlaik to aktīvi izstrādā un atbalsta Zabbix SIA.

Zabbix ir uzņēmuma līmeņa atvērtā pirmkoda izkliedētas uzraudzības risinājums.

Zabbix ir programmatūra, kas uzrauga daudzus tīkla parametrus un serveru, virtuālo mašīnu, lietojumprogrammu, pakalpojumu, datubāzu, tīmekļvietņu, mākoņpakalpojumu un citu resursu darbspēju un integritāti. Zabbix izmanto elastīgu paziņojumu mehānismu, kas ļauj lietotājiem konfigurēt uz e-pastu balstītus brīdinājumus praktiski jebkuram notikumam. Tas ļauj ātri reaģēt uz servera problēmām. Zabbix piedāvā izcilas atskaišu un datu vizualizācijas iespējas, kas balstītas uz saglabātajiem datiem. Tas padara Zabbix ideāli piemērotu jaudas plānošanai.

Zabbix atbalsta gan aptaujāšanu, gan slazdošanu. Visām Zabbix atskaitēm un statistikai, kā arī konfigurācijas parametriem piekļūst, izmantojot tīmekļa lietotāja saskarne. Tīmekļa lietotāja saskarne nodrošina, ka jūsu tīkla statusu un serveru darbspēju var novērtēt no jebkuras vietas. Pareizi konfigurēts, Zabbix var spēlēt svarīgu lomu IT infrastruktūras uzraudzībā. Tas vienlīdz attiecas gan uz mazām organizācijām ar dažiem serveriem, gan uz lieliem uzņēmumiem ar daudziem serveriem.

Zabbix ir bez maksas. Zabbix ir rakstīts un izplatīts saskaņā ar AGPL-3.0 licenci. Tas nozīmē, ka tā pirmkods tiek brīvi izplatīts un ir pieejams plašai sabiedrībai.

Ir pieejams komerciālais atbalsts, ko nodrošina Zabbix Company un tās partneri visā pasaulē.

Uzziniet vairāk par Zabbix iespējām.

Zabbix lietotāji Daudzas dažāda lieluma organizācijas visā pasaulē paļaujas uz Zabbix kā galveno uzraudzības platformu.

Arhitektūra Zabbix sastāv no vairākiem galvenajiem programmatūras komponentiem. To atbildības jomas ir izklāstītas tālāk.

Serveris

Zabbix serveris ir centrālā komponente, kurai aģenti ziņo par pieejamības un integritātes informāciju un statistiku. Serveris ir centrālā krātuve, kurā tiek glabāti visi konfigurācijas, statistikas un darbības dati.

Datubāzes glabāšana

Visa konfigurācijas informācija, kā arī Zabbix apkopotie dati tiek glabāti datubāzē.

Tīmekļa saskarne

Lai nodrošinātu ērtu piekļuvi Zabbix no jebkuras vietas un no jebkuras platformas, ir pieejama tīmekļa lietotāja saskarne. Saskarne ir daļa no Zabbix servera un parasti (bet ne obligāti) darbojas tajā pašā fiziskajā datorā, kurā darbojas serveris.

Starpniekserveris

Zabbix starpniekserveris var vākt veiktspējas un pieejamības datus Zabbix servera vārdā. Starpniekserveris ir izvēles Zabbix izvietošanas daļa; tomēr tas var būt ļoti noderīgs, lai sadalītu viena Zabbix servera slodzi.

Aģents

Zabbix aģenti tiek izvietoti uzraudzības mērķos, lai aktīvi uzraudzītu vietējos resursus un lietojumprogrammas un ziņotu savāktos datus Zabbix serverim. Kopš Zabbix 4.4 ir pieejami divu veidu aģenti: Zabbix aģents (viegls, atbalstīts daudzās platformās, rakstīts C valodā) un Zabbix agent 2 (īpaši elastīgs, viegli paplašināms ar spraudņiem, rakstīts Go valodā).

Datu plūsma Turklāt ir svarīgi atkāpties soli atpakaļ un aplūkot kopējo datu plūsmu Zabbix ietvaros. Lai izveidotu vienumu, kas apkopo datus, vispirms jāizveido hosts. Pārejot uz otru Zabbix spektra galu, lai izveidotu trigeri, vispirms jābūt vienumam. Lai izveidotu darbību, jābūt trigerim. Tādējādi, ja vēlaties saņemt brīdinājumu, ka jūsu CPU slodze uz *Server X* ir pārāk augsta, vispirms jāizveido hosta ieraksts priekš *Server X*, pēc tam vienums tā CPU uzraudzībai, tad trigeris, kas aktivizējas, ja CPU slodze ir pārāk augsta, un pēc tam darbība, kas nosūta jums e-pasta ziņojumu. Lai gan tas var šķist daudz soļu, izmantojot veidnes, tā patiesībā nav. Tomēr šī dizaina dēļ ir iespējams izveidot ļoti elastīgu iestatījumu.

Iespējas Zabbix ir augsti integrēts tīkla uzraudzības risinājums, kas vienā pakotnē piedāvā daudzas iespējas.

Datu vākšana

- pieejamības un veiktspējas pārbaudes
- SNMP (gan trapu, gan aptauju), IPMI, JMX, VMware uzraudzības atbalsts
- pielāgotas pārbaudes
- nepieciešamo datu vākšana ar pielāgotiem intervāliem
- veic serveris/starpniekserveris un aģenti

Elastīgas sliekšņu definīcijas

- varat definēt ļoti elastīgus problēmu sliekšņus, ko sauc par trigeriem, atsaucoties uz vērtībām no aizmugursistēmas datubāzes

Plaši konfigurējama brīdināšana

- paziņojumu sūtīšanu var pielāgot eskalācijas grafikam, saņēmējam, medija tipam
- paziņojumus var padarīt jēgpilnus un noderīgus, izmantojot makro mainīgos
- automātiskās darbības ietver attālinātās komandas

Grafiku veidošana reāllaikā

- uzraudzītie vienumi tiek nekavējoties attēloti grafikos, izmantojot iebūvēto grafiku veidošanas funkcionalitāti

Tīmekļa uzraudzības iespējas

- Zabbix var sekot simulētu peles klikšķu ceļam tīmekļa vietnē un pārbaudīt funkcionalitāti un atbildes laiku

Plašas vizualizācijas iespējas

- iespēja izveidot pielāgotus grafikus, kuros var apvienot vairākus vienumus vienā skatā
- tīkla kartes
- slaidrādes informācijas paneļa stila pārskatā
- atskaites
- uzraudzīto resursu augsta līmeņa (biznesa) skats

Vēsturisko datu glabāšana

- dati tiek glabāti datubāzē
- konfigurējama vēsture
- iebūvēta novecojušo datu tīrīšanas procedūra

Vienkārša konfigurēšana

- pievienojiet uzraudzītās ierīces kā hostus
- hosti tiek iekļauti uzraudzībā, tiklīdz tie ir datubāzē
- piemērojiet veidnes uzraudzītajām ierīcēm

Veidņu izmantošana

- pārbaužu grupēšana veidnēs
- veidnes var mantot citas veidnes

Tīkla atklāšana

- automātiska tīkla ierīču atklāšana
- aģenta automātiskā reģistrācija
- failu sistēmu, tīkla saskarņu un SNMP OID atklāšana

Ātra tīmekļa saskarne

- uz PHP balstīta lietotāja saskarne
- pieejama no jebkuras vietas
- varat pārvietoties, klikšķinot pa sadaļām
- audita žurnāls

Zabbix API

- Zabbix API nodrošina programmējamu saskarni ar Zabbix masveida izmaiņu veikšanai, trešo pušu programmatūras integrācijai un citiem mērķiem.

Atļauju sistēma

- droša lietotāju autentifikācija
- noteiktiem lietotājiem var ierobežot piekļuvi noteiktiem skatiem

Pilnvērtīgs un viegli paplašināms aģents

- izvietots uzraudzības mērķos
- var tikt izvietots gan Linux, gan Windows vidē

Binārie dēmoni

- rakstīti C valodā, lai nodrošinātu veiktspēju un mazu atmiņas patēriņu
- viegli pārnesami

Piemērots sarežģītām vidēm

- attālinātā uzraudzība ir vienkārša, izmantojot Zabbix starpniekserveri

Glosārijs Šajā sadaļā varat uzzināt dažu Zabbix bieži lietotu terminu nozīmi.

hosts

- jebkura fiziska vai virtuāla ierīce, lietojumprogramma, pakalpojums vai jebkura cita loģiski saistīta uzraugāmu parametru kopa.

hostu grupa

- hostu loģiska grupēšana. Hostu grupas tiek izmantotas, piešķirot piekļuves tiesības hostiem dažādām lietotāju grupām.

vienums

- konkrēta datu vienība, ko vēlaties saņemt no hosta, datu metrika.

vērtības priekšapstrāde

- saņemtās metrikas vērtības pārveidošana pirms tās saglabāšanas datubāzē.

trigeris

- loģiska izteiksme, kas nosaka problēmas slieksni un tiek izmantota, lai "novērtētu" vienumos saņemtos datus.

Ja saņemtie dati pārsniedz slieksni, trigeri pāriet no 'Ok' stāvokļa uz 'Problem' stāvokli. Ja saņemtie dati ir zem sliekšņa, trigeri paliek/atgriežas 'Ok' stāvoklī.

veidne

- entītiju kopa (vienumi, trigeri, grafiki, zema līmeņa atklāšanas kārtulas, tīmekļa scenāriji), kas ir gatava piemērošanai vienam vai vairākiem hostiem.

Veidņu uzdevums ir paātrināt uzraudzības uzdevumu izvēršanu hostā, kā arī atvieglot masveida izmaiņu piemērošanu uzraudzības uzdevumiem. Veidnes tiek tieši piesaistītas atsevišķiem hostiem.

veidņu grupa

- veidņu loģiska grupēšana. Veidņu grupas tiek izmantotas, piešķirot piekļuves tiesības veidnēm dažādām lietotāju grupām.

notikums

- vienreizēja kāda uzmanības vērta notikuma parādīšanās, piemēram, trigera stāvokļa maiņa vai atklāšanas/aģenta automātiskās reģistrācijas izpilde.

notikuma tags

- *iepriekš definēts notikuma marķieris.* To var izmantot notikumu korelācijā, atļauju detalizācijā u.c.

notikumu korelācija

- *metode, kas ļauj elastīgi un precīzi sasaistīt problēmas ar to atrisinājumu.*

Piemēram, varat definēt, ka viena trigera ziņotu problēmu var atrisināt cits trigeris, kas var pat izmantot citu datu vākšanas metodi.

problēma

- *trigeris, kas atrodas "Problem" stāvoklī.*

problēmas atjaunināšana

- *Zabbix nodrošinātās problēmu pārvaldības iespējas, piemēram, komentāra pievienošana, apstiprināšana, būtiskuma maiņa vai manuāla aizvēršana.*

darbība

- *iepriekš definēts veids, kā reaģēt uz notikumu.*

Darbība sastāv no operācijām (piemēram, paziņojuma nosūtīšanas) un nosacījumiem (*kad* operācija tiek izpildīta)

eskalācija

- *pielāgots scenārijs operāciju izpildei darbības ietvaros; paziņojumu sūtīšanas/attālināto komandu izpildes secība.*

makross

- *mainīgā vietturis, kas atkarībā no konteksta tiek aizstāts ar noteiktu vērtību.*

medijs

- *paziņojumu piegādes līdzeklis; piegādes kanāls.*

paziņojums

- *ziņojums par kādu notikumu, kas lietotājam nosūtīts, izmantojot izvēlēto medija kanālu.*

attālinātā komanda

- *iepriekš definēta komanda, kas tiek automātiski izpildīta uzraugāmajā hostā, iestājoties noteiktam nosacījumam.*

tīmekļa scenārijs

- *viens vai vairāki HTTP pieprasījumi, lai pārbaudītu tīmekļa vietnes pieejamību.*

lietotāja saskarne

- *kopā ar Zabbix nodrošinātā tīmekļa saskarne.*

panelis

- *pielāgojama tīmekļa saskarnes sadaļa, kurā tiek attēloti svarīgas informācijas kopsavilkumi un vizualizācijas* vizuālās vienībās, ko sauc par logrīkiem.

logrīks

- *vizuāla vienība, kas attēlo noteikta veida un avota informāciju* (kopsavilkumu, karti, grafiku, pulksteni u.c.), un tiek izmantota panelī.

Zabbix API

- *Zabbix API ļauj izmantot JSON RPC protokolu, lai izveidotu, atjauninātu un izgūtu Zabbix objektus (piemēram, hostus, vienumus, grafikus un citus) vai veiktu jebkurus citus pielāgotus uzdevumus.*

Zabbix serveris

- *centrālais Zabbix programmatūras process, kas veic uzraudzību, mijiedarbojas ar Zabbix starpniekserveriem un aģentiem, aprēķina trigerus, sūta paziņojumus; centrālā datu krātuve.*

Zabbix starpniekserveris

- *process, kas var vākt datus Zabbix servera vārdā, noņemot daļu apstrādes slodzes no servera.*

Zabbix aģents

- *process, kas izvietots uzraudzības mērķos, lai aktīvi uzraudzītu vietējos resursus un lietojumprogrammas.*

Zabbix aģents 2

- *jaunās paaudzes Zabbix aģents vietējo resursu un lietojumprogrammu aktīvai uzraudzībai, kas ļauj uzraudzībā izmantot pielāgotus spraudņus.*

> **Attention:**
> Tā kā Zabbix aģentam 2 ir daudz kopīgas funkcionalitātes ar Zabbix aģentu, termins "Zabbix aģents" dokumentācijā apzīmē abus — Zabbix aģentu un Zabbix aģentu 2, ja funkcionālā uzvedība ir vienāda. Zabbix aģents 2 tiek īpaši nosaukts tikai tur, kur tā funkcionalitāte atšķiras.

šifrēšana

- *šifrētas saziņas atbalsts starp Zabbix komponentiem (serveris, starpniekserveris, aģents, zabbix_sender un zabbix_get utilītas)*, izmantojot Transport Layer Security (TLS) protokolu.

aģenta automātiskā reģistrācija

- *automatizēts process, kurā Zabbix aģents pats tiek reģistrēts* kā hosts un tiek sākta tā uzraudzība.

tīkla atklāšana

- *automatizēta tīkla ierīču atklāšana*.

zema līmeņa atklāšana

- *automatizēta zema līmeņa entītiju atklāšana konkrētā ierīcē* (piemēram, failu sistēmas, tīkla saskarnes u.c.).

zema līmeņa atklāšanas kārtula

- *definīciju kopa zema līmeņa entītiju automatizētai atklāšanai* ierīcē.

vienuma prototips

- *metrika ar noteiktiem parametriem kā mainīgajiem, gatava zema līmeņa atklāšanai*. Pēc zema līmeņa atklāšanas mainīgie tiek automātiski aizstāti ar reāli atklātajiem parametriem, un metrika automātiski sāk vākt datus.

trigera prototips

- *trigeris ar noteiktiem parametriem kā mainīgajiem, gatavs zema līmeņa atklāšanai*. Pēc zema līmeņa atklāšanas mainīgie tiek automātiski aizstāti ar reāli atklātajiem parametriem, un trigeris automātiski sāk novērtēt datus.

Dažu citu Zabbix entītiju prototipi arī tiek izmantoti zema līmeņa atklāšanā — grafiku prototipi, hostu prototipi, hostu grupu prototipi.

Zabbix Cloud

Zabbix Cloud

Iepazīstiet pieejamās konfigurācijas opcijas un piekļuves kontroles organizācijām un mezgliem Zabbix Cloud vidē.

Lūdzu, ņemiet vērā, ka Zabbix Cloud mezgli pašlaik darbojas ar **Zabbix 7.0**.
 Padziļinātai informācijai par Zabbix 7.0 skatiet oficiālo dokumentāciju.

Iestatiet un pārvaldiet Zabbix Cloud

Sāciet lietot Zabbix Cloud Norādījumi soli pa solim, lai ātri palaistu un sagatavotu darbam savu Zabbix Cloud instanci, tostarp sākotnējo konfigurēšanu un pārraugāmo ierīču pievienošanu.

Mezgla konfigurācija Uzziniet, kā iestatīt savu Zabbix Cloud mezglu, koncentrējoties uz piekļuvi, datu apstrādi un resursu iestatījumiem.

Lietotāju pievienošana Uzziniet, kā pievienot lietotājus un piešķirt tiem lomas Zabbix Cloud organizācijā un tās mezglos.

Uzziniet par Zabbix Cloud

Zabbix Cloud salīdzinājumā ar lokālu izvietojumu Salīdziniet atšķirības pārvaldībā, mērogojamībā un uzturēšanā starp abiem izvietošanas veidiem un uzziniet, kurš no tiem var visvienkāršāk atbalstīt jūsu darbplūsmu.

Iepazīstiet Zabbix Cloud Iegūstiet pārskatu par Zabbix Cloud, tostarp tā galvenajām iespējām, priekšrocībām un to, kas ir iekļauts abonementā. Biežāk uzdotos jautājumus par iestatīšanu un lietošanu skatiet BUJ.

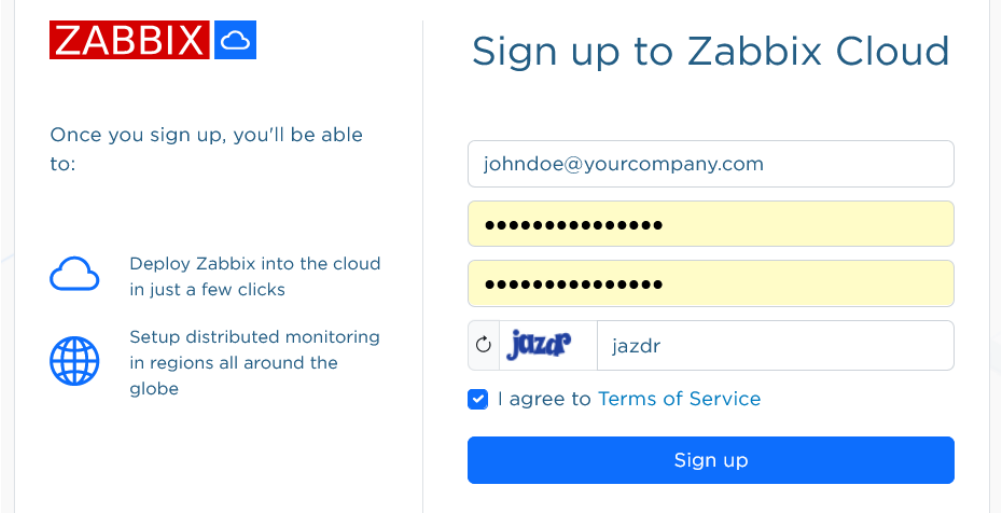
Zabbix izvietošana mākonī

Lai sāktu izmantot Zabbix mākonī, izpildiet šo procesu:

- Piesakieties vai reģistrējieties vietnē cloud.zabbix.com
- Izveidojiet mākoņa mezglu
- Sāciet izmantot Zabbix mākonī

Pieteikšanās vai reģistrēšanās Jūs varat pieteikties vietnē cloud.zabbix.com, izmantojot esošu Google, GitHub vai Microsoft kontu.

Alternatīvi atveriet reģistrēšanās lapu, lai reģistrētos.



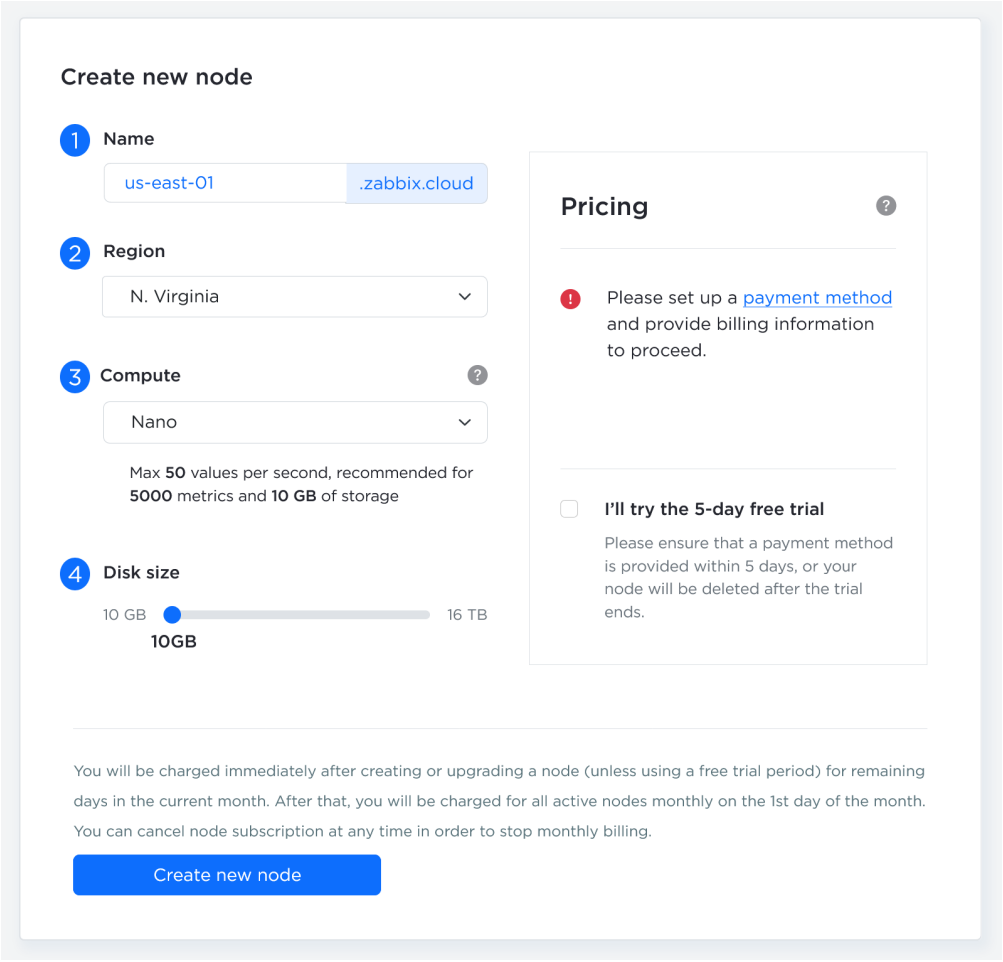
Pēc reģistrēšanās jūs saņemsiet e-pastā apstiprinājumu par savu jauno kontu.

Dodieties uz cloud.zabbix.com, lai pieteiktos. Piesakoties jums tiks prasīta vienreizēja parole. Šī parole jums tiks nosūtīta e-pastā.

Mākoņa mezgla izveide Mākoņa mezgls ir Zabbix instance, kas darbojas mākonī. Tajā ietilpst Zabbix serveris, datubāze un tīmekļa lietotāja saskarne.

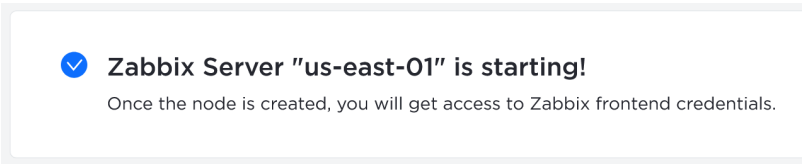
Pēc pieteikšanās jūs redzēsiet, ka jūsu kontam pēc noklusējuma jau pastāv *Private* organizācija. Noklusējuma organizāciju vēlāk varat pārdēvēt.

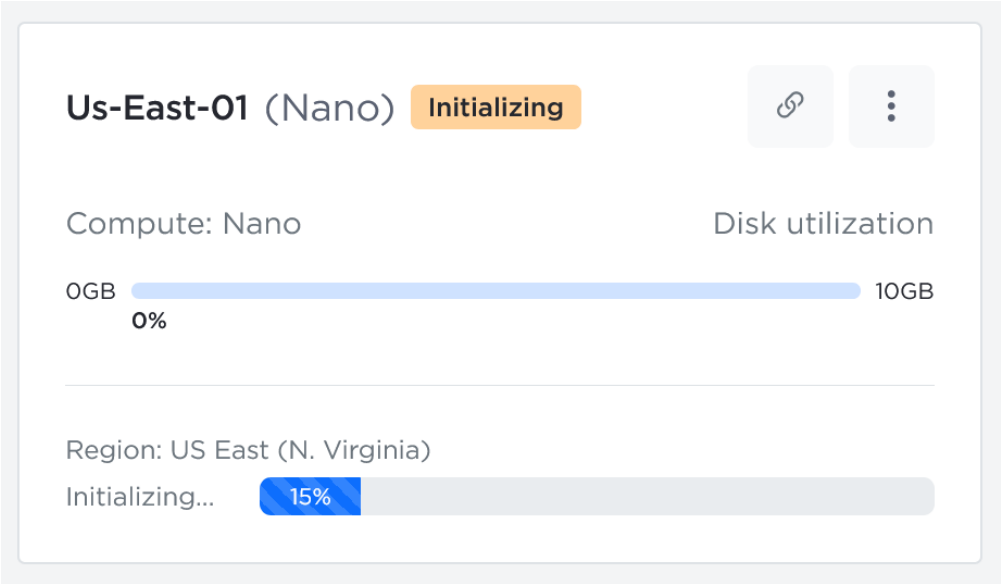
Noklikšķiniet uz *Create New Node*. Tiks atvērta mezgla konfigurācijas forma.

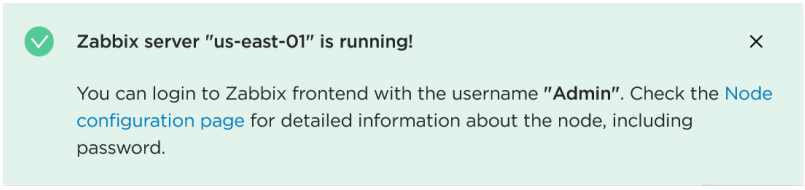


Parametrs	Apraksts
Name	Izvēlieties unikālu mezgla nosaukumu. Mezgla nosaukums veido unikālu URL mākoņa serverim/lietotāja saskarnei formātā `<your-name>.zabbix.cloud`.
Region	Izvēlieties tuvāko datu centra atrašanās vietu, lai nodrošinātu minimālu tīkla aizturi.
Compute	Izvēlieties mezglam nepieciešamo skaitļošanas jaudu. Lai noteiktu, kāds resursu līmenis jums ir nepieciešams, skatiet cenu informāciju. Ja vēlāk būs nepieciešama lielāka jauda, vienmēr varēsiet veikt jaunināšanu. Pirms netesta mezgla izveides būs jānorāda norēķinu metode. Pievienojiet norēķinu informāciju, izmantojot izvēlni *Billing*. *Nano* līmeņa mezgla izmēģināšanai ir pieejams 5 dienu **bezmaksas izmēģinājums**. Izmēģinājuma periodā no jums netiks iekasēta maksa, taču pēc tā beigām mezgls tiks noņemts, ja tas netiks pārveidots par maksas mezglu.
Disk size	Slīdnis pēc noklusējuma ir iestatīts uz ieteicamo minimumu atbilstoši līmenim. Palieliniet to, ja glabājat lielu vēstures vai tendenču datu apjomu. Ņemiet vērā, ka lielāki diski rada augstākas ikmēneša izmaksas un ka diska izmēru pats samazināt nevarēsiet — lai to samazinātu, būs jāsazinās ar atbalsta dienestu.

Pēc formas aizpildīšanas noklikšķiniet uz *Create New Node*. Kamēr mezgls tiek inicializēts, tiek parādīts atbilstošs ziņojums, un mezgls jau ir redzams sadaļā Organizations.





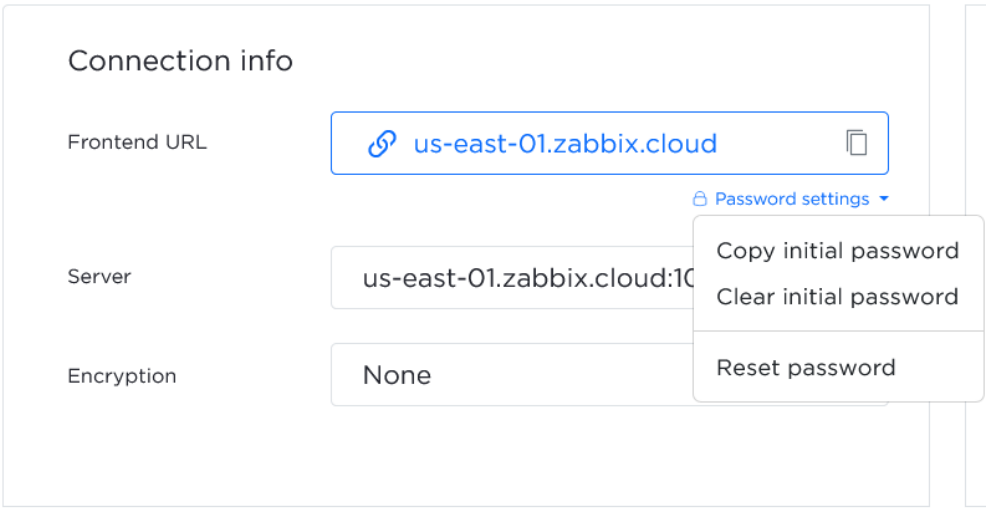


Sāciet lietot Zabbix

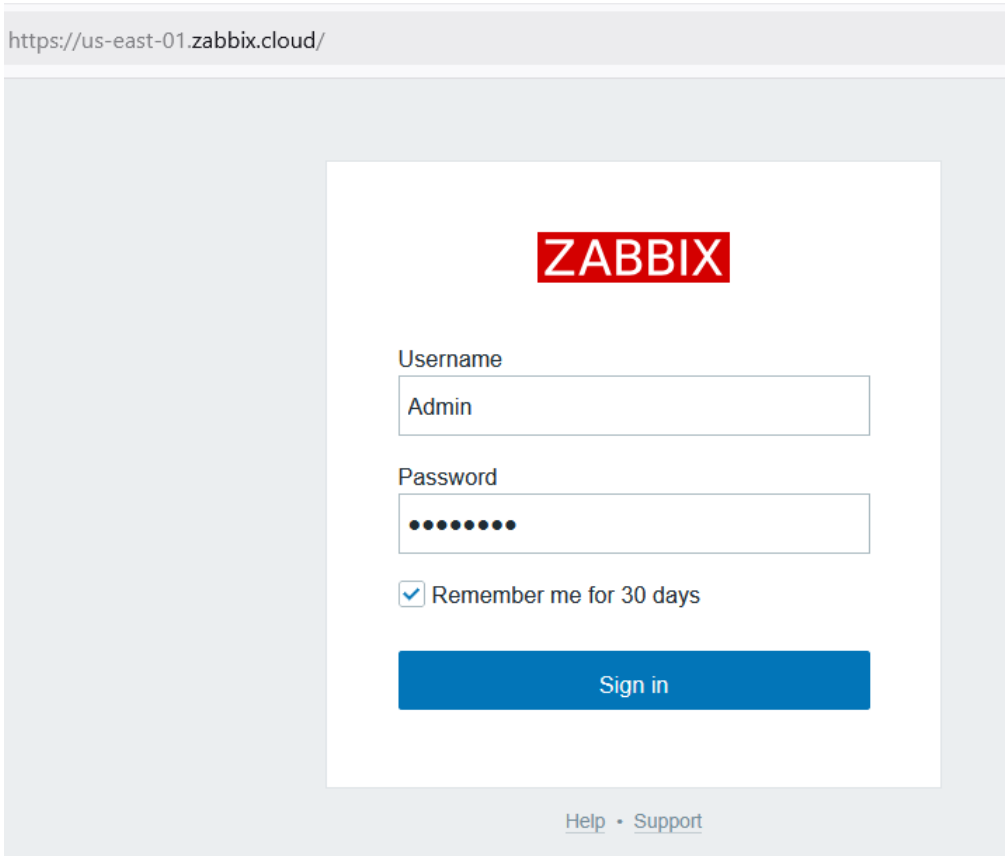
Kad mezgla statuss mainās no *Initializing* uz *Running*, varat pieteikties Zabbix lietotāja saskarnē.

Lai piekļūtu sākotnējai lietotāja saskarnes parolei, noklikšķiniet uz izveidotā mezgla.

Jūs tiksiet novirzīts uz mezgla iestatījumiem (pārskats, piekļuves filtri, šifrēšana, dublējumi utt.). Sadaļā **Connection info** noklikšķiniet uz *Password settings* un nolaižamajā izvēlnē atlasiet *Copy initial password*.



Noklikšķiniet uz lietotāja saskarnes URL, lai atvērtu Zabbix lietotāja saskarni (`us-east-01.zabbix.cloud`).



Piesakieties, izmantojot šādus akreditācijas datus:

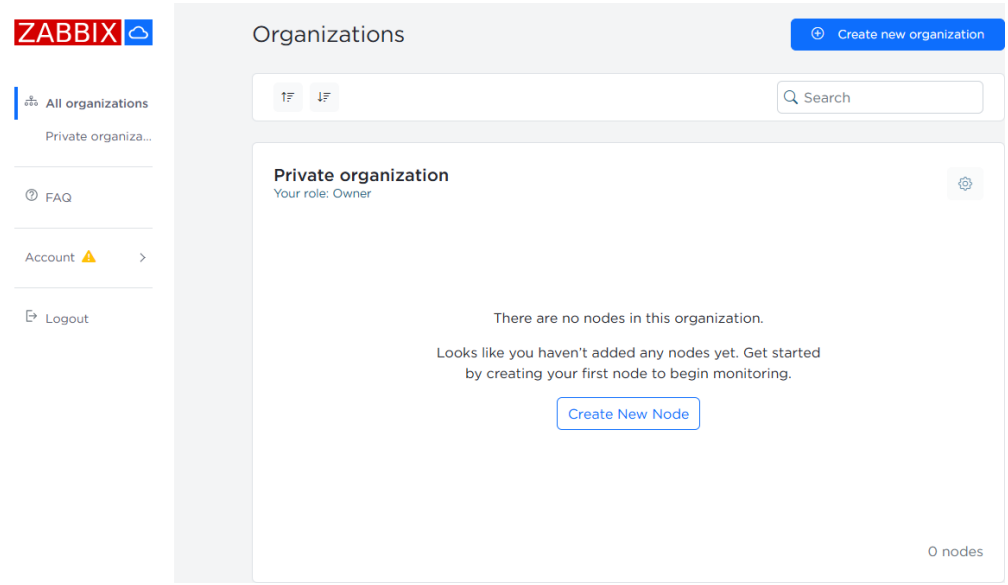
- *Username* - `Admin`
- *Password* - norādiet sākotnējo paroli (piemēram, HzTG9t7Y)

Lai sāktu uzraudzību, jums jādefinē piekļuves filtri — IP adrese vai adreses, kurām ir atļauts izveidot savienojumu ar Zabbix mākoņserveri. Jābūt definētam vismaz vienam filtram.

Organizācijas pārvaldība Organizācija ir konteiners vienam vai vairākiem mezgliem. Organizācijas arī palīdz strukturēt jūsu resursus un deleģēt atbildību, piemēram:

- Mezglu pārvaldība
- Norēķini

Piesakoties Zabbix Cloud, tiek parādīta lapa *Visas organizācijas*. Pēc noklusējuma jūsu kontam tiek izveidota privāta organizācija.



Kā šīs organizācijas īpašnieks jūs varat pārvaldīt lietotājus, administratorus, mezglus un organizācijas nosaukumu. Noklikšķiniet uz iestatījumu pogas , lai piekļūtu pārvaldības opcijām.

Mezglu konfigurācija

Pārskats

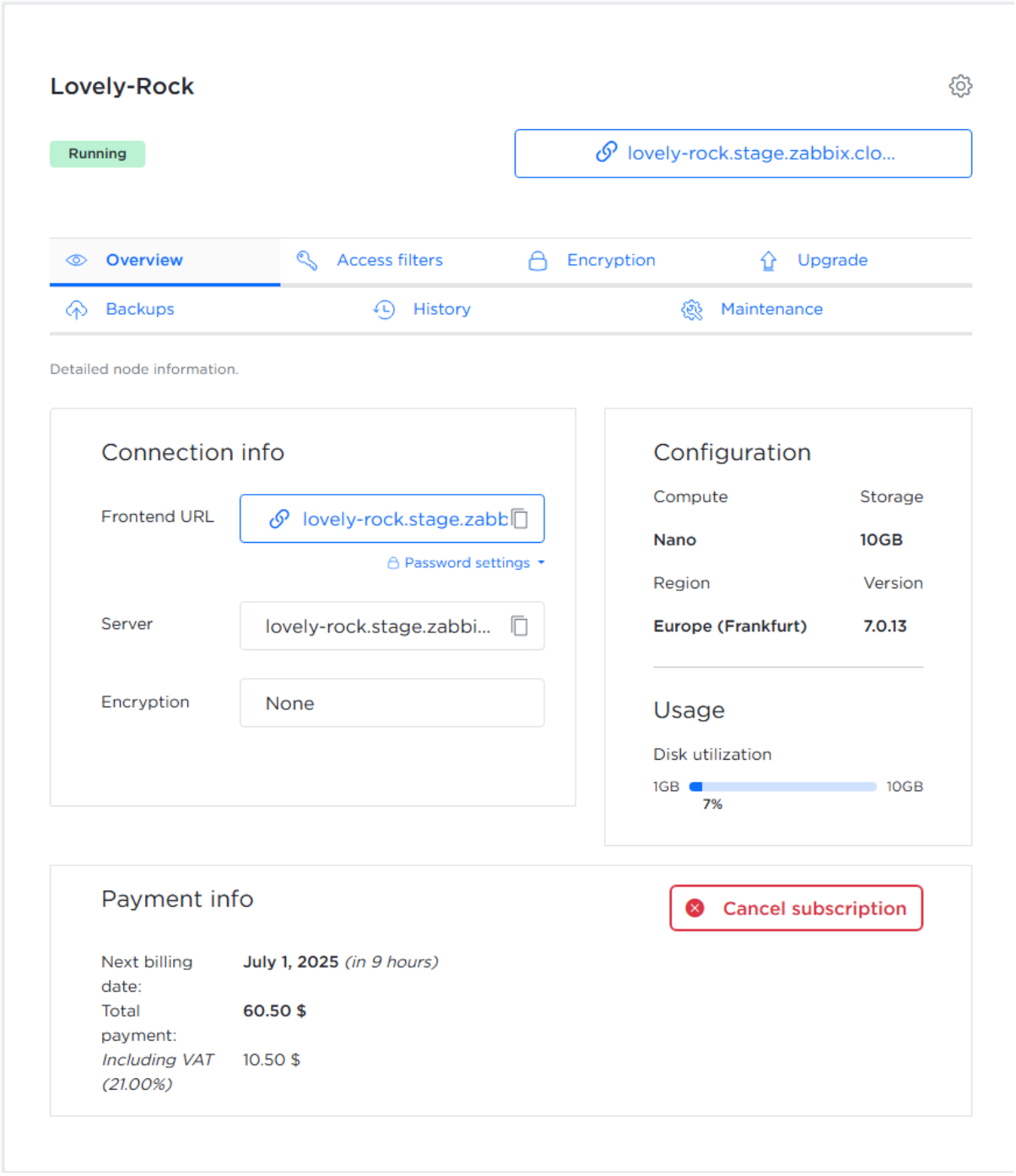
Mākoņa mezgls ir Zabbix instances ekvivalents, kas izvietots mākonī. Tas ietver Zabbix serveri, datubāzi un tīmekļa lietotāja saskarni. Mākoņa mezglam ir tam paredzēti skaitļošanas resursi (CPU, atmiņa) un diska krātuve.

Lietotāji var izveidot vienu vai vairākus mākoņa mezglus. Mākoņa mezglus var grupēt **organizācijās** (strukturālās, reģionālās u.c.), tādējādi izveidojot sadalītu uzraudzības vidi.

Mākoņa mezgli un organizācijas tiek izveidoti, izmantojot Zabbix Cloud lietotāja saskarni. Lai piekļūtu mezgla konfigurācijas lapai, noklikšķiniet uz mezgla sadaļā *Visas organizācijas*, savā organizācijas izvēlnē vai izmantojot saiti *Mezgla konfigurācijas lapa*, kas tiek parādīta pēc mezgla palaišanas.

Mezgla konfigurācijas skatā katra cilne ļauj pārvaldīt noteiktu jūsu mezgla aspektu:

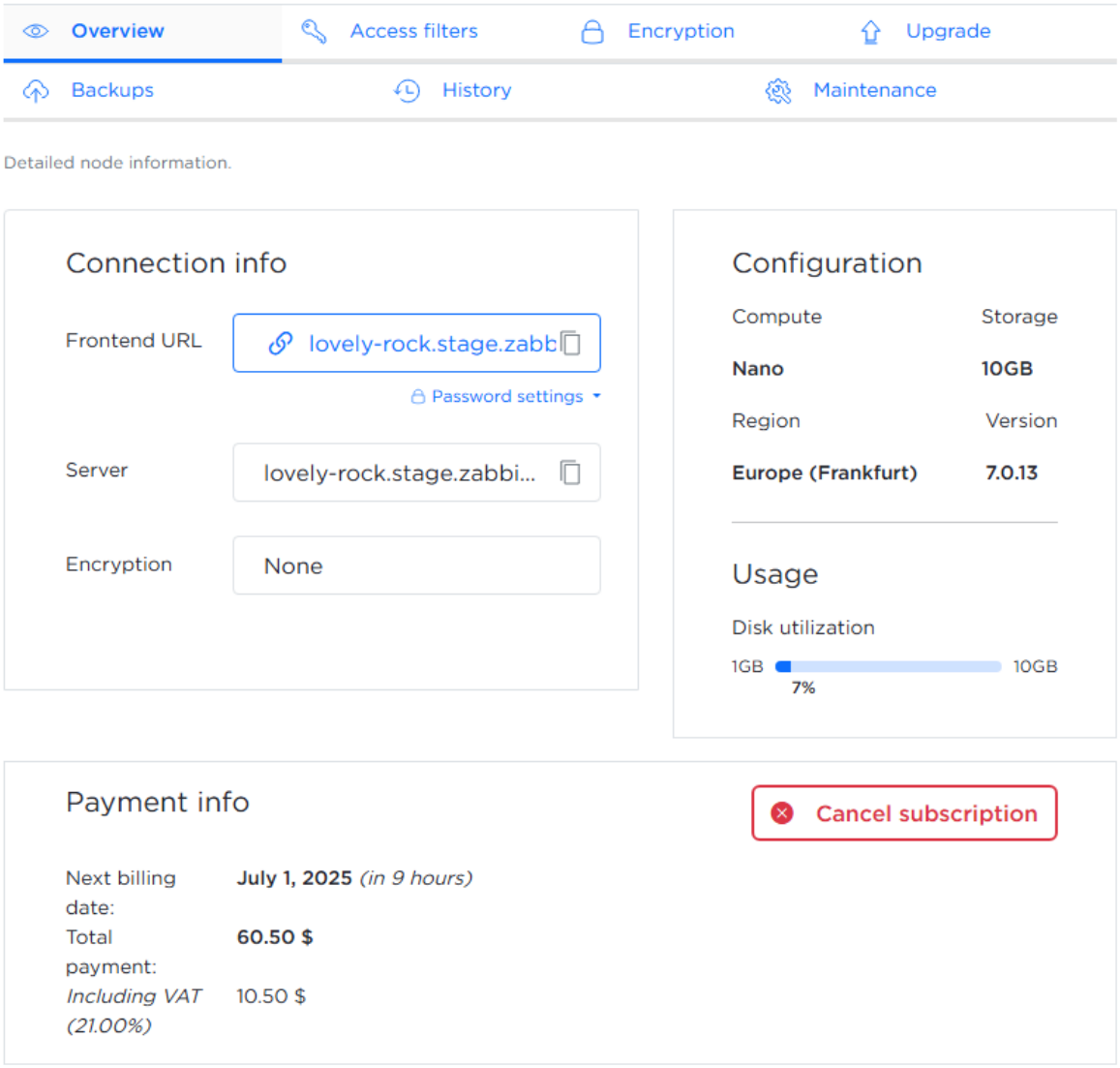
- Pārskats — skatīt mezgla informāciju, atvērt tā lietotāja saskarni un pārvaldīt mezgla lietotājus.
- Piekļuves filtri — kontrolēt piekļuvi pēc IP adreses vai apakštīkla.
- Šifrēšana — konfigurēt TLS, augšupielādējot sertifikātus, atslēgas un izvēles CRL.
- Dublējumkopijas — izveidot, atjaunot un dzēst mezgla momentuzņēmumus; skatīt nākamo ieplānoto dublējumkopiju.
- Vēsture — iestatīt audita žurnālu, vēstures un tendenču glabāšanas periodus, izmantojot iepriekš definētas vienības un atjaunināšanas ierobežojumu 3 reizes dienā.
- Uzturēšana — definēt iknedēļas uzturēšanas logus pēc laika joslas, dienas un stundas.
- Jaunināšana — mērogot skaitļošanas/krātuves resursus vai pārvaldīt izmēģinājuma/maksājumu statusu.



Note:
Dažas cilnes var nebūt pieejamas atkarībā no lapu skatošā lietotāja tipa un mezgla statusa (izmēģinājuma/neizmēģinājuma).

Pārskata cilne

Cilnē *Pārskats* ir pieejama detalizēta informācija par mezglu.



Tā ir sadalīta trīs sadaļās:

Savienojuma informācija:

- Noklikšķināms lietotāja saskarnes URL ar iespēju kopēt starpliktuvē
- Paroles iestatījumi
- Pašreizējā šifrēšanas metode

Konfigurācija un lietojums:

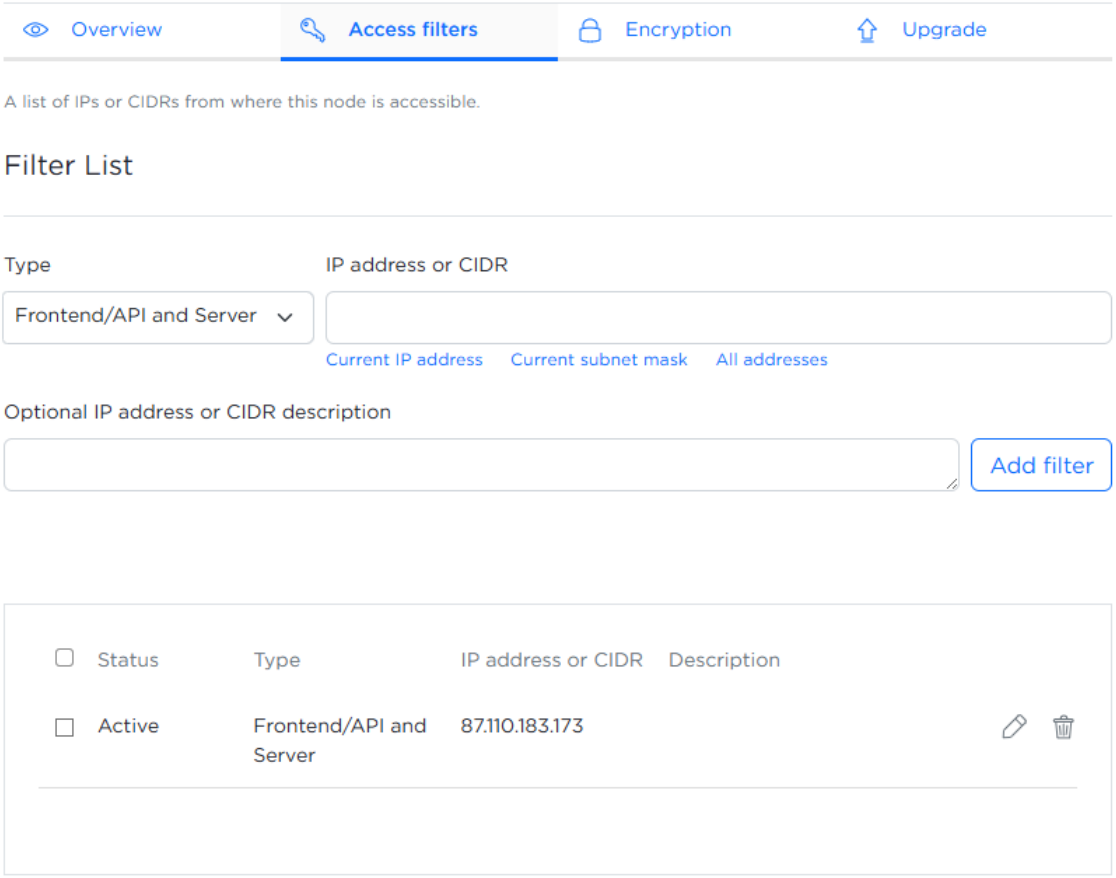
- Skaitļošanas līmenis, maksimālā krātuve, reģions
- Zabbix komponentu versija
- Diska izmantojums

Mezgla informācija/Maksājumu informācija:

- Derīguma termiņa beigu datums (izmēģinājuma un priekšapmaksas mezgliem)
- Pēc mezgla inicializācijas pabeigšanas atkarībā no mezgla tipa tiek parādīta kontekstam atbilstoša poga: -*Jaunināt uz maksas plānu* - pieejama mezgliem bezmaksas izmēģinājuma periodā - *Atcelt abonementu* - pieejama maksas mezgliem ar aktīvu abonementu - *Atkārtoti abonēt maksas plānu* - tiek parādīta, ja abonements ir atcelts - Priekšapmaksas mezgliem darbības poga netiek parādīta, jo tiem ir ilgs derīguma termiņš un tos nevar atcelt

Piekļuves filtru cilne

Cilne *Piekļuves filtri* ļauj ierobežot piekļuvi jūsu Zabbix mezglam pēc IP adreses vai apakštīkla.



Lai konfigurētu piekļuves filtru, aizpildiet šādus laukus:

- *Tips* — izmantojiet nolaižamo sarakstu, lai izvēlētos, kuru Zabbix komponentu ierobežot: tikai lietotāja saskarne/API, tikai serveris vai abus.
- *IP adrese vai CIDR* — ievadiet IP adresi vai CIDR manuāli vai aizpildiet lauku automātiski, atlasot atbilstošo opciju.
- *Neobligāts IP adreses vai CIDR apraksts* — pievienojiet neobligātu aprakstu katrai IP adresei vai apakštīklam.

Pēc lauku aizpildīšanas noklikšķiniet uz *Pievienot filtru*.
Pievienotie filtri tiek parādīti zemāk esošajā tabulā, norādot statusu, tipu, IP adresi vai apakštīklu un aprakstu.

Warning:
Mezgla inicializācijas laikā piekļuves filtrs tiek automātiski izveidots tai IP adresei, kuru izmantojāt. Ja vēlāk pieslēgsieties no citas IP adreses vai tīkla, šī adrese šeit ir jāpievieno, pretējā gadījumā piekļuve tiks bloķēta.

Jūs varat:

- — rediģēt filtru.
- — dzēst filtru vai vairākus filtrus, izmantojot izvēles rūtiņas.

Šifrēšanas cilne

Cilne *Šifrēšana* ļauj konfigurēt TLS šifrēšanu starp Zabbix komponentiem, augšupielādējot nepieciešamos sertifikātu failus.

Lai iespējotu drošu saziņu, izmantojot sertifikātus, atlasiet *CERTIFICATE* un augšupielādējiet no sava datora šādus failus:

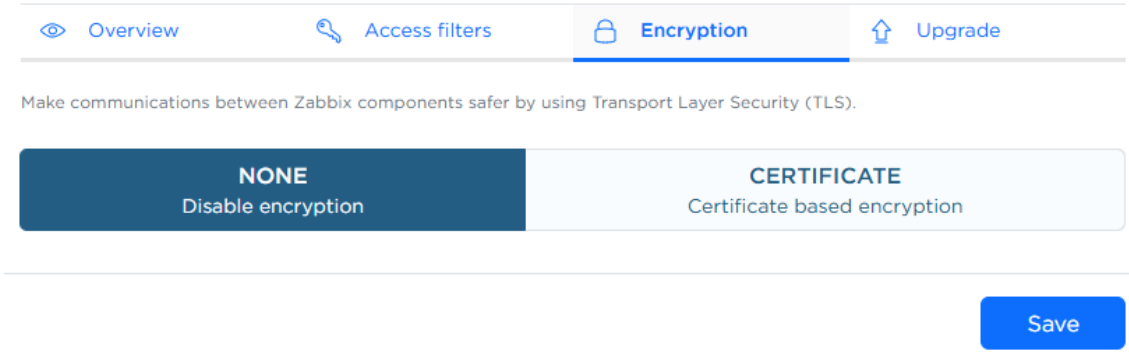
- *Saknes CA sertifikāti* — tiek izmantoti, lai pārbaudītu otras puses sertifikātu autentiskumu.
  Noklikšķiniet uz *Choose file* un atlasiet failu, kurā ir uzticami saknes CA sertifikāti (parasti `.crt` vai `.pem` fails PEM kodējumā).

- *Sertifikātu ķēde* — attēlo uzticamības ķēdi jūsu sertifikātam.
  Noklikšķiniet uz *Choose file* un atlasiet PEM kodētu failu (piemēram, `.crt` vai `.pem`), kas ietver jūsu sertifikātu un visus starpniek-CA sertifikātus.

- *Privātā atslēga* — privātā atslēga, kas atbilst jūsu sertifikātam.
  Noklikšķiniet uz *Choose file* un atlasiet PEM kodētu `.key` failu.

- *Atsauktie sertifikāti* (neobligāti) — ja uzturat sertifikātu atsaukšanas sarakstu (CRL), noklikšķiniet uz *Choose file* un atlasiet CRL failu, lai norādītu sertifikātus, kuriem vairs nevajadzētu uzticēties.

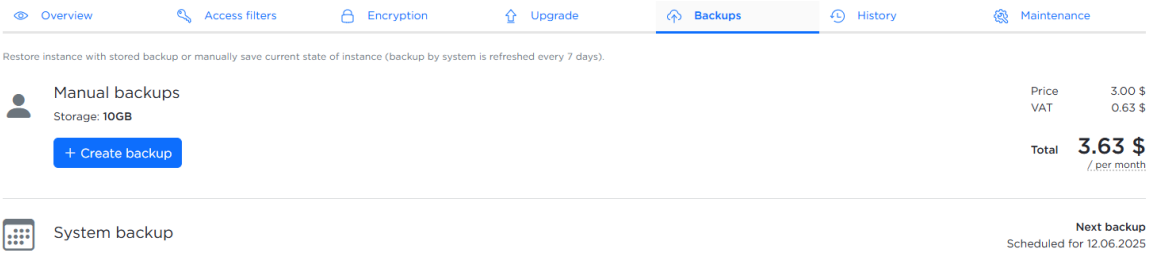
Pēc failu atlasīšanas noklikšķiniet uz *Save*. Faili tiks augšupielādēti, pārbaudīti un piemēroti, lai konfigurētu TLS šifrēšanu starp Zabbix komponentiem.



Dublējumu cilne

Cilne *Dublējumi* ļauj skatīt, izveidot, atjaunot un dzēst jūsu Zabbix mezgla dublējumus.

Jūs varat atjaunot savu instanci no jebkura saglabāta dublējuma vai jebkurā laikā izveidot jaunu manuālu dublējumu. Cilnē tiek parādīts arī nākamā ieplānotā dublējuma datums.



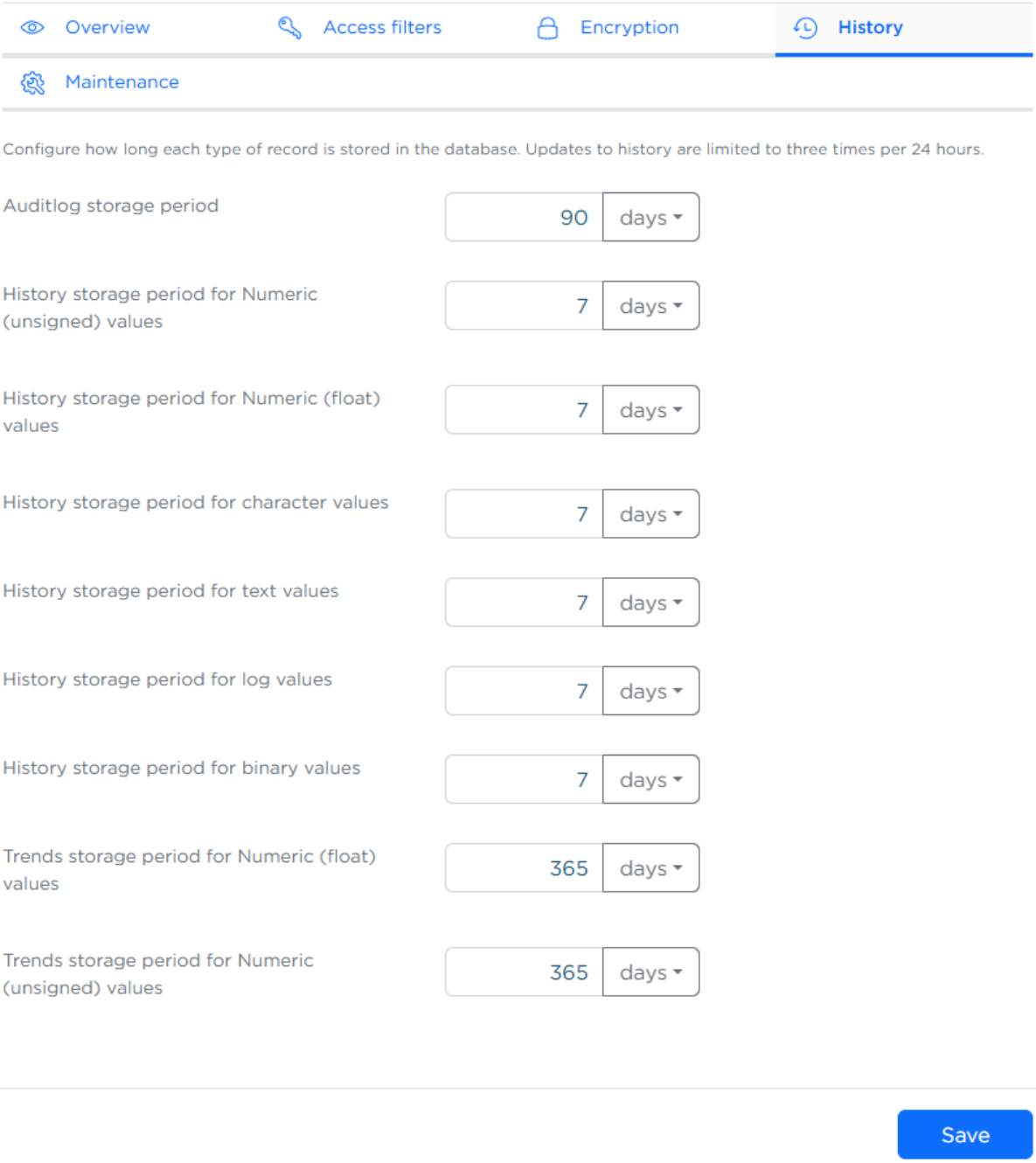
Lai izveidotu dublējumu:

1. Noklikšķiniet uz pogas *Izveidot dublējumu*.
2. Uznirstošajā logā ievadiet neobligātu aprakstu.
3. Noklikšķiniet uz *Izveidot*, lai apstiprinātu.

Warning:
Par papildu dublējumiem tiek piemērota ikmēneša maksa atbilstoši izmantotajai krātuves vietai.

Vēstures cilne

Cilne *Vēsture* ļauj konfigurēt, cik ilgi tiek glabāti dažādu tipu vēsturiskie dati.



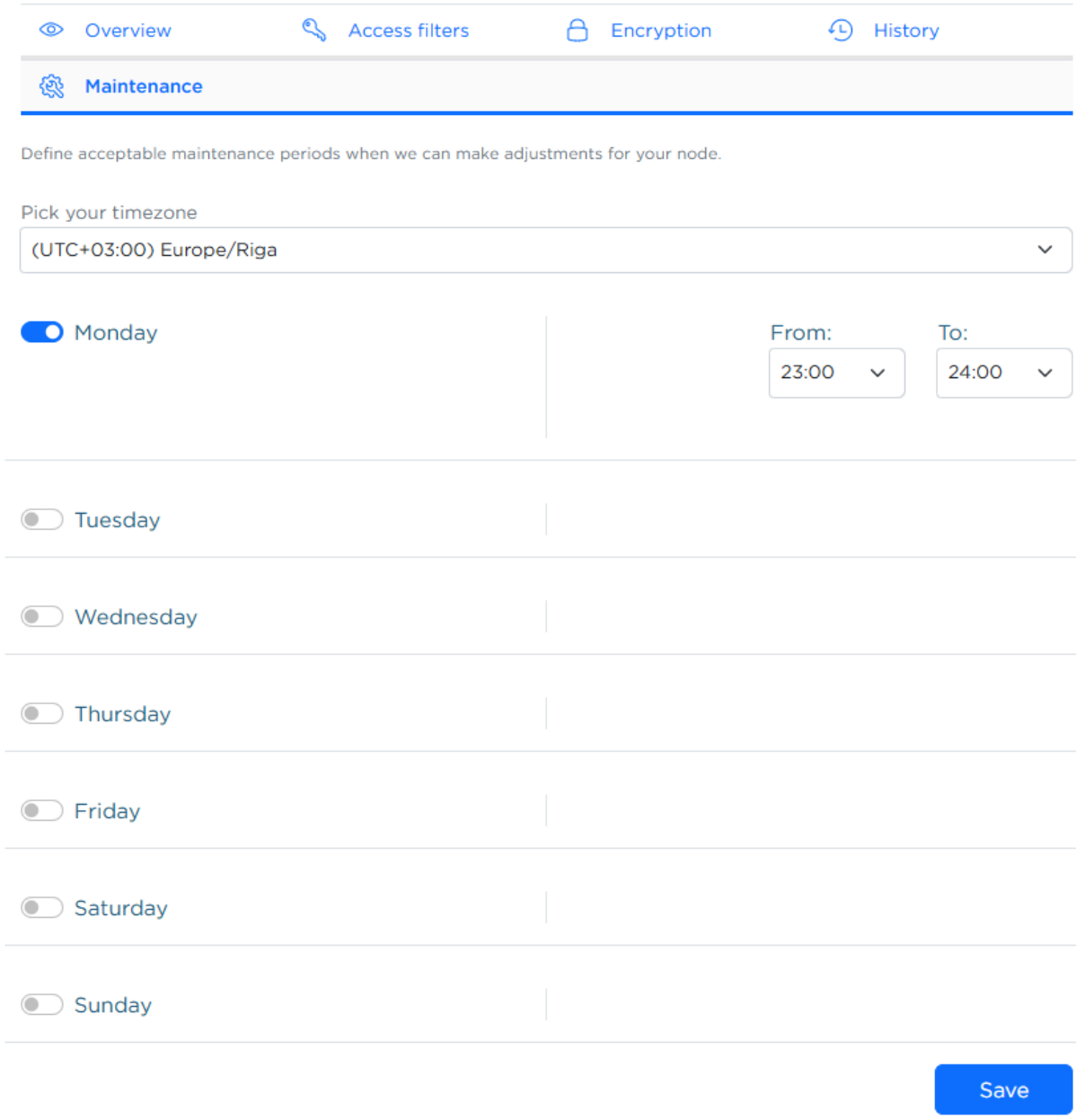
Varat iestatīt glabāšanas periodu katram datu tipam (piemēram, skaitliskie dati, teksts, žurnāli) un izvēlēties mērvienību (dienas, mēneši, gadi). Izmaiņas var veikt ne vairāk kā trīs reizes 24 stundu laikā.

Pēc vērtību atjaunināšanas noklikšķiniet uz *Saglabāt*, lai piemērotu izmaiņas.

Warning:
Ilgāki glabāšanas periodi var palielināt diska vietas izmantošanu un ietekmēt veiktspēju. Ieteicams konfigurēt glabāšanas periodus atbilstoši jūsu faktiskajiem datiem un analīzes vajadzībām.
 Zabbix Cloud datu glabāšanas pārvaldībai izmanto datubāzes particionēšanu tradicionālās Zabbix housekeepinga vietā. Tas nodrošina ātrāku veiktspēju un paredzamāku glabāšanas periodu pārvaldību.

Uzturēšanas cilne

Cilne *Uzturēšana* ļauj definēt pieļaujamos uzturēšanas logus, kuros jūsu mezglā var veikt atjauninājumus vai pielāgojumus.

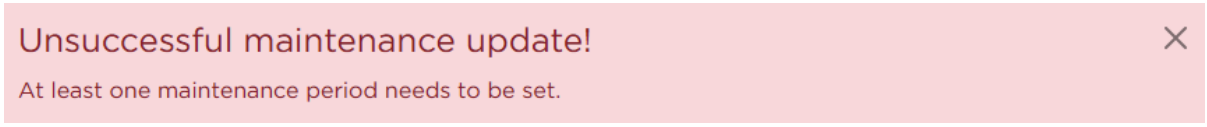


Jums ir jāveic šādas darbības:

- Nolaižamajā sarakstā atlasiet savu laika joslu.
- Izvēlieties vismaz vienu dienu un vienu stundas logu uzturēšanai.

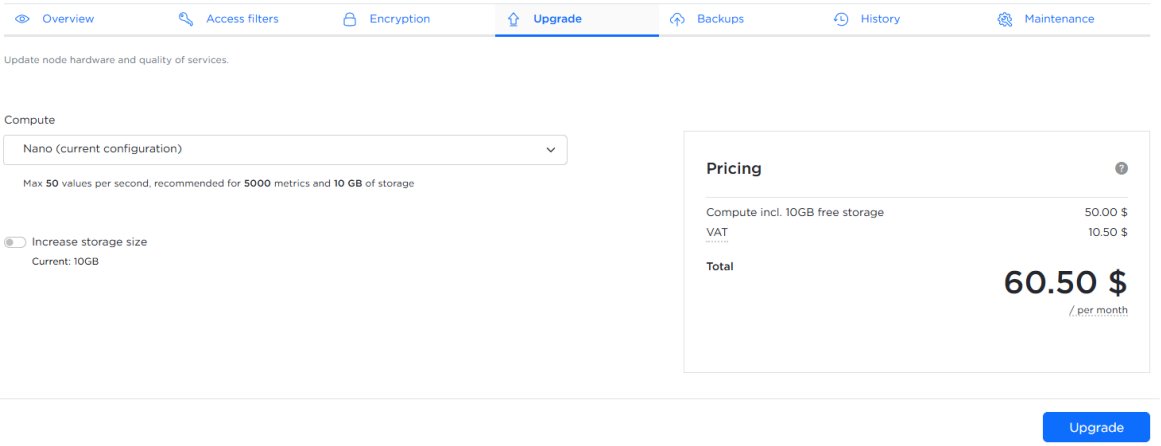
Noklikšķiniet uz *Saglabāt*, lai piemērotu izmaiņas.

Ja nav atlasīta neviena diena vai laika logs, tiks parādīts kļūdas ziņojums.



Jauninašanas cilne

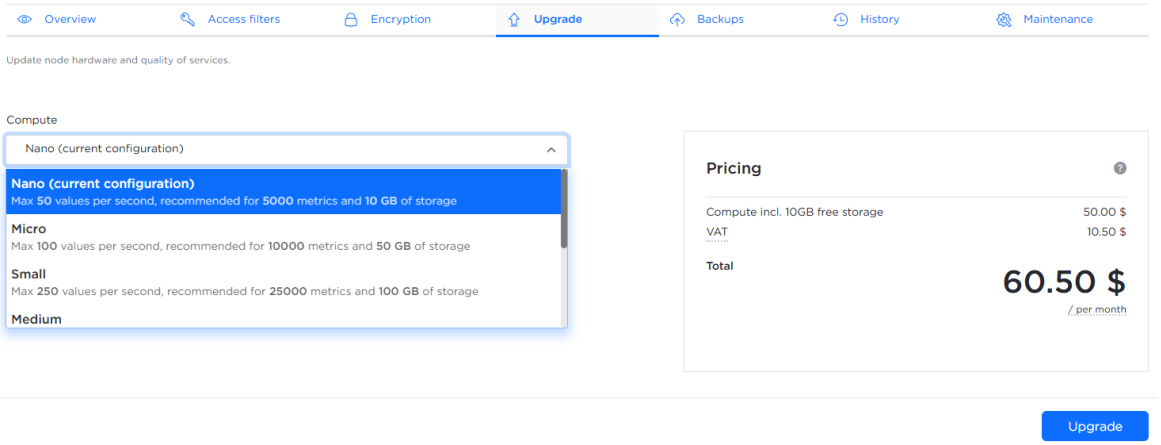
Cilne *Jaunināšana* ļauj mērogot jūsu mezgla resursus vai pārvaldīt tā abonementu.



Varat palielināt skaitļošanas līmeni vai diska izmēru — ņemiet vērā, ka diska izmēru var palielināt tikai vienu reizi 24 stundās, un to nevar samazināt, izmantojot lietotāja saskarni (sazinieties ar Zabbix atbalstu, ja vēlaties samazināt diska izmēru). Bezmaksas izmēģinājuma mezglus var pārveidot par maksas abonementiem.

Lai palielinātu skaitļošanas līmeni:

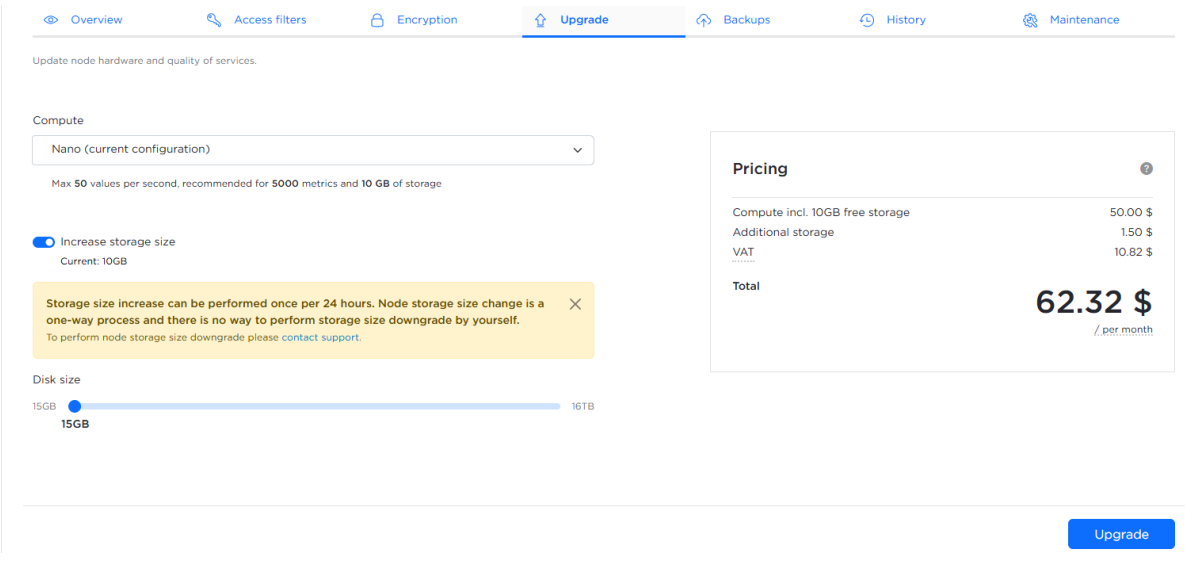
1. Noklikšķiniet uz nolaižamā saraksta *Compute*.



2. Atlasiet vēlamo konfigurācijas līmeni.
3. Noklikšķiniet uz pogas *Upgrade*.

Lai palielinātu diska izmēru:

1. Noklikšķiniet uz radio pogas *Increase storage size*.
2. Pārvietojiet slīdni *Disk size* līdz vēlamajai vērtībai.



3. Noklikšķiniet uz pogas *Upgrade*.

Note:
Atjauninātā kopējā mēneša cena tiek parādīta tabulā *Pricing* pirms izmaiņu apstiprināšanas.

Lietotāju pievienošana

Ievads

Tālāk sniegtajos piemēros ir aprakstīts, kā pievienot lietotājus jūsu organizācijai un mezgliem.

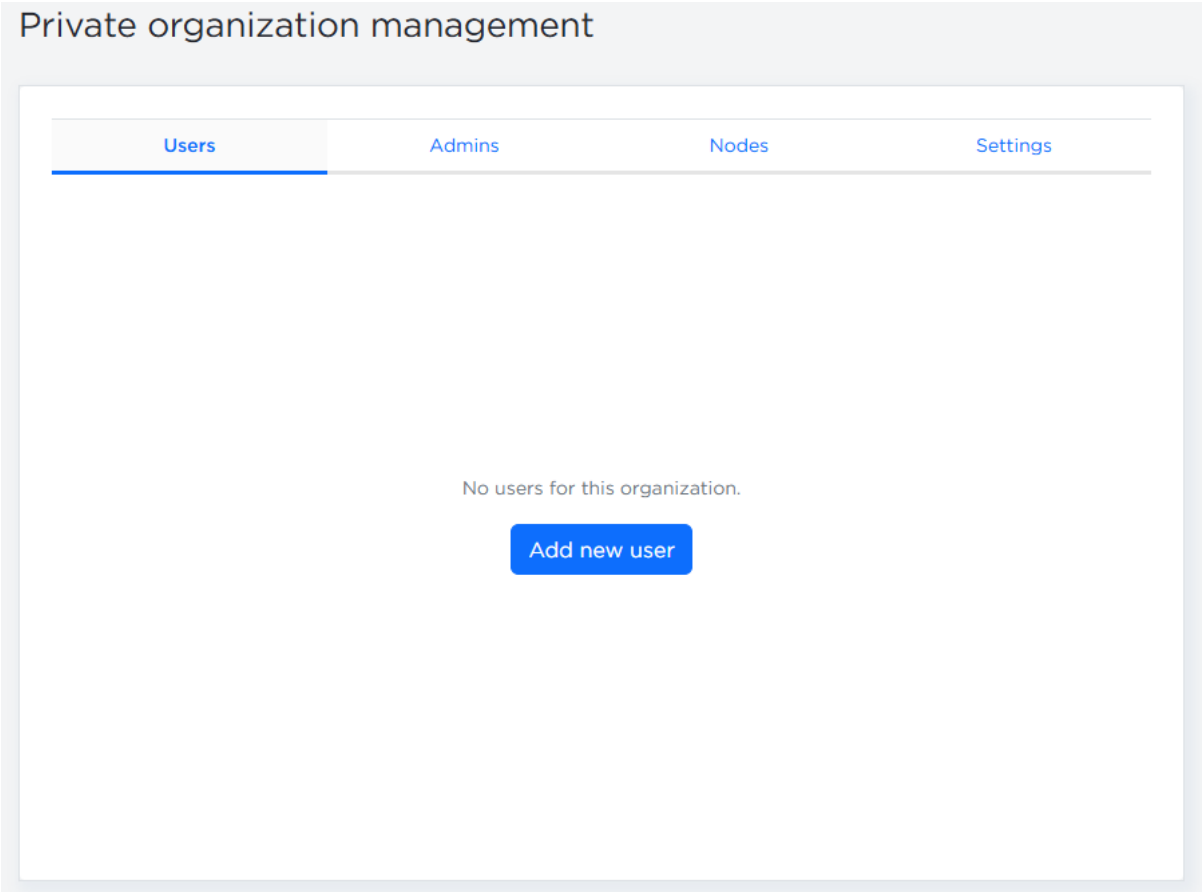
Note:
Lietotāja saskarnes paroles kopēšanu, notīrīšanu un atiestatīšanu var veikt tikai īpašnieks.

Lietotāju pievienošana organizācijām

Līdzīgi kā lietotāji Zabbix vidē, jūs varat izveidot lietotājus Zabbix Cloud un piešķirt tiem lomas, lai kontrolētu, ko viņi var darīt organizācijā un tās mezglos.

Lai pievienotu lietotājus savai organizācijai:

1. Atveriet lapu *All organizations*.
2. Noklikšķiniet uz konkrētās organizācijas iestatījumu pogas , lai atvērtu tās pārvaldības izvēlni.



3. Cilnē *Users* vai *Admins* noklikšķiniet uz *Add new user*.
4. Ievadiet lietotāja e-pasta adresi un izvēlieties lomu: *Administrator* vai *User*.

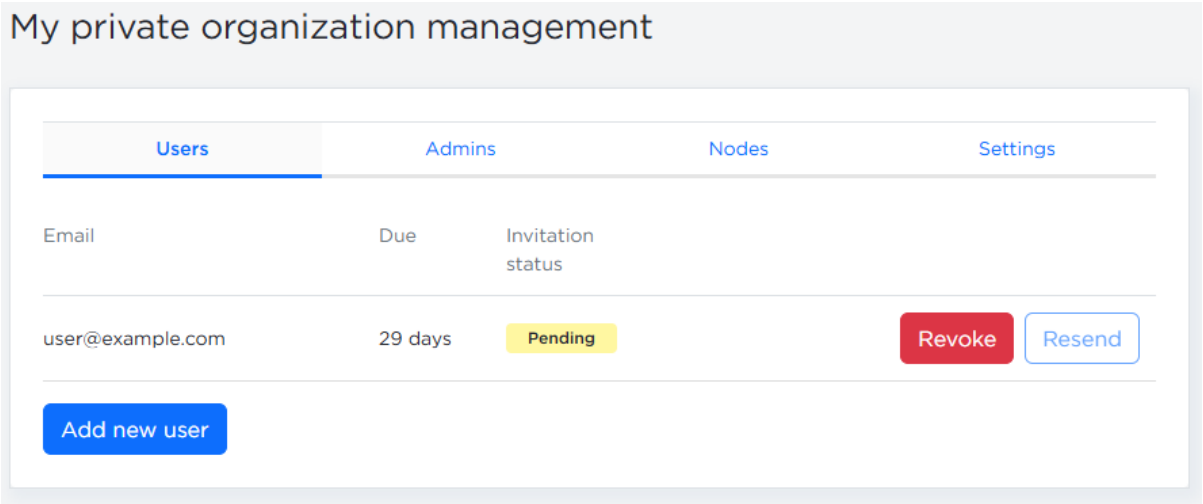
Ņemiet vērā, ka, izvēloties lomu *User*, lietotājs tiek pievienots organizācijai, taču viņam netiek piešķirta piekļuve nevienam mezglam.

Katra loma piešķir atšķirīgu piekļuves līmeni:

- **User** — var piekļūt mezgliem, kuriem tas ir piešķirts, un var veikt konfigurēšanas uzdevumus, kas nerada papildu izmaksas, piemēram:
  - Pārvaldīt piekļuves filtrus
  - Konfigurēt šifrēšanu
  - Iestatīt vēstures/tendenču glabāšanas periodus
  - Plānot apkopes
- **Administrator** — ir visas User atļaujas un var arī veikt darbības, kas var radīt papildu izmaksas, piemēram:
  - Izveidot un atjaunot rezerves kopijas
  - Palielināt vai samazināt mezgla resursus (skaitļošanas līmeni un diska izmēru)
  - Atcelt abonementus
- **Owner** — ir pilnīga kontrole pār organizāciju. Papildus visām Administrator atļaujām Owner var:
  - Izveidot un dzēst mezglus
  - Pievienot vai noņemt Users un Administrators
  - Pārdēvēt vai dzēst organizāciju
  - Pārvaldīt norēķinu informāciju

> **Attention:**
> Par visiem pakalpojumiem maksa tiek iekasēta no Owner maksājuma metodes. Ja ir nepieciešams bankas apstiprinājums, Owner ir jāautorizē darījums, pat ja to ir iniciējis Administrator.

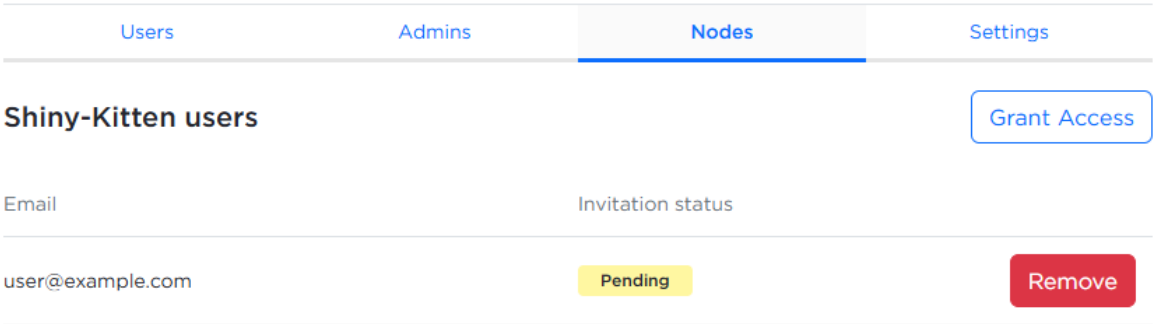
Uzaicinātais lietotājs tiks pievienots ar statusu *Pending*.
Jūs varat atsaukt uzaicinājumu vai nosūtīt to atkārtoti (atkārtota nosūtīšana ir pieejama reizi 10 minūtēs — poga būs atspējota, ja šis ierobežojums būs sasniegts).



Lietotāju pievienošana mezgliem

Lai pievienotu lietotāju mezglam:

1. Atveriet cilni *Mezgli*.
2. Noklikšķiniet uz *Piešķirt piekļuvi*.
3. Nolaižamajā sarakstā atlasiet lietotāju, kas pievienots organizācijai.
4. Noklikšķiniet uz *Piešķirt piekļuvi*.



Alternatīvi, piekļuvi var piešķirt logā *Mezgla konfigurācija*:

1. Noklikšķiniet uz iestatījumu pogas .
2. Atlasiet *Pievienot lietotāju*.
3. Izvēlieties lietotāju un noklikšķiniet uz *Piešķirt piekļuvi*.

Uzaicinātajiem lietotājiem ir jāpieņem uzaicinājums, izveidojot Zabbix Cloud kontu ar to pašu e-pasta adresi, uz kuru tika nosūtīts uzaicinājums. Ja lietotājam jau ir Zabbix Cloud konts ar šo e-pasta adresi, viņam jāpierakstās un jāapstiprina uzaicinājums. Uzaicinājumi ir piesaistīti konkrētai e-pasta adresei, un tos nevar pieņemt no cita konta.

Zabbix Cloud galvenās atšķirības

Ievads

Zabbix Cloud ir mākoņdatošanas platforma, kas ļauj lietotājiem darbināt atsevišķas Zabbix instances mākonī (mākoņa mezglus).

Salīdzinot ar parastu lokālu Zabbix izvietojumu:

- Nav nepieciešama instalēšana vai uzturēšana.
- Mākoņa mezglus pilnībā pārvalda Zabbix.
- Mākoņa mezglus var palaist dažu minūšu laikā.

Šis pakalpojums tiek piedāvāts pēc pay-as-you-go principa (ir pieejams **bezmaksas izmēģinājums**).

Salīdzinājuma tabula Lai gan lielākā daļa funkcionalitātes pārklājas starp lokālām Zabbix instalācijām un mākoņa mezgliem, pastāv arī atšķirības. Tālāk ir sniegta kopsavilkuma salīdzinājuma tabula. Turpmākajās sadaļās ir detalizēti aprakstīti atlasītie vienumi.

Funkcija / Aspekts	Zabbix Cloud	Lokāls Zabbix
Izlaidumu cikls	Atbilstoši LTS izlaidumiem	Jūs izvēlaties savu Zabbix versiju (stabila, beta, pielāgoti būvējumi)
Lietošanas ērtums	Lietošanai gatava platforma, ko pilnībā pārvalda Zabbix	Nepieciešama instalēšana, konfigurēšana un uzturēšana
Ievainojamību ielāpošana	Automātiska OS/Zabbix ielāpošana	Manuāla ielāpošana, ko veic lietotājs
HTTPS sertifikāts	Derīgs sertifikāts uzreiz pēc izvietošanas (bez Let's Encrypt)	Pēc noklusējuma pašparakstīts (var konfigurēt Let's Encrypt vai citu CA)
DB veiktspēja (INSERT/UPDATE/SELECT)	Automātiska particionēšana maksimālam ātrumam; nav ilgstoši izpildāmu DELETE darbību	Manuāla particionēšana un tīrīšana (iespējamas ilgstošas DELETE transakcijas)
Veiktspējas regulēšana	Iepriekš noregulē Zabbix Cloud komanda (bez izmaiņu žurnāliem)	Pilnīga pārskatāmība un kontrole pār regulēšanas parametriem
Ugunsmūra kontrole	Pārvalda Zabbix Cloud komanda; GUI/trapper porti nav publiski pieejami	Pilnīga kontrole pār ugunsmūra noteikumiem un portu pieejamību
DB izolācija un aizsardzība	Augsta izolācija, nav tiešas DB/SSH piekļuves	Atkarīgs no jūsu tīkla un hosta drošības
Augsta pieejamība (HA)	Darbojas augstas pieejamības mākoņa platformā, augstākai datu vākšanas noturībai var izmantot Zabbix starpniekserverus	Lietotāja pārvaldīta HA
SNMP slazdi	Tikai caur specializētu starpniekserveri (bez HA/slodzes balansēta slazdu uztvērēja)	Tieši vai caur starpniekserveri, iespējama HA/slodzes balansēšana
SNMP aptauja	Sarežģīti bez starpniekserveriem (katrai ierīcei nepieciešams NAT/pielāgots ports)	Vietējā SNMP aptauja; starpniekserveris nav obligāts
Glabāšanas iestatījumi	Vēsture/Tendences/Audits tikai caur UI (bez API vai konfigurācijas faila)	Konfigurējams Zabbix servera konfigurācijas failā vai caur API
Pielāgoti skripti	`AlertScriptsPath`, `ExternalScripts`, lietotāja saskarnes un kopienas moduļi netiek atbalstīti	Pilnībā atbalstīts (skriptu ceļš, moduļi, integrācijas)

Funkcija / Aspekts	Zabbix Cloud	Lokāls Zabbix
ODBC uzraudzība	Tikai PostgreSQL (draiveris {postgresql}); MariaDB spraudnis ir pieejams kā vietturis	ODBC PostgreSQL, MySQL, Oracle u.c. - pilnībā konfigurējams
ODBC izsaukumu ierobežošana	Nav iespējams (StartODBCPollers=1 tikai; smagi sinhronie vaicājumi bloķē citus darbus)	Pilnībā regulējams aptaujātāju skaits un plānošana
SAML sertifikāta augšupielāde	Netiek atbalstīta	Atbalstīta caur UI vai API
Plānotie pārskati	Jāizveido savs e-pasta medijs (skripta medijs netiek atbalstīts)	Atbalsta gan skripta, gan e-pasta mediju uzreiz pēc izvietošanas
Hosta saskarne aktīvajām pārbaudēm	Platformas piešķirta IP adrese (izveidota automātiski)	Saskarnes pārvaldāt jūs; IP adreses ir jūsu kontrolē

Funkciju atšķirības SNMP trapi

SNMP trapi tiek atbalstīti tikai ar īpašu Zabbix starpniekserveri. Ja ir nepieciešama SNMP trapu uzraudzība, starpniekserverim nevar izmantot automatizētu slodzes līdzsvarošanu vai augstu pieejamību, jo SNMP trapi ir jānosūta uz vienu IP adresi.

SNMP aptauja

Bez starpniekserveriem SNMP aptaujai nepieciešams, lai katra ierīce būtu pieejama caur NAT un pielāgotiem portiem.

Glabāšanas iestatījumi

History, trend un auditācijas žurnāla glabāšanas periodus mākonī var konfigurēt tikai, izmantojot tīmekļa saskarni. Šos iestatījumus nevar konfigurēt, izmantojot `zabbix_server.conf` vai API.

Manuālas katra vienuma vēstures pārrakstīšanas nav atbalstītas (particionēšana tiek kontrolēta globāli). Mākoņa mezglu URL nepieņem tādus pašus vaicājuma parametrus kā lokālās instalācijas.

Pielāgoti skripti

Zabbix Cloud neatbalsta šādus pielāgoto skriptu tipus:

- Brīdinājumu skripti (`AlertScriptsPath`)
- Ārējie skripti (`ExternalScripts`)
- Lietotāja saskarnes skripti Zabbix serverī

Kopienas izstrādātos lietotāja saskarnes moduļus nevar instalēt.

ODBC monitorings

Zabbix Cloud atbalsta ODBC monitoringu tikai PostgreSQL. Izmantojiet oficiālo Zabbix ODBC veidni un definējiet savienojuma virkni veidnē, izmantojot:

```
Driver={postgresql}
```

Ir instalēts arī MariaDB spraudnis, taču tas pašlaik nedarbojas. Ja to izmanto, definējiet:

```
Driver={mysql}
```

Attention:
Nav zināma neviena strādājoša Oracle MySQL 8.0 konfigurācija Zabbix Cloud vidē. Lai gan vienkārši vaicājumi (piemēram, `SELECT 1`) var izdoties, sarežģītāki vaicājumi atgriež `SQL_ERROR`.

 Nav iespējams ierobežot ODBC izsaukumus (`StartODBCPollers=1` only). Intensīva sinhronā atskaišu veidošana var ietekmēt veiktspēju, un vienlaikus var tikt izpildīts tikai viens `SELECT`.

Piekļuve infrastruktūrai

Zabbix Cloud nenodrošina SSH piekļuvi pamatā esošajiem mezgliem, kā arī neatļauj tiešus datubāzes savienojumus (piemēram, ar portu 3306). Visas konfigurēšanas, uzraudzības un problēmu novēršanas darbības jāveic, izmantojot mākoņa lietotāja saskarni vai API, nodrošinot, ka OS un datubāzes slāņi paliek izolēti un aizsargāti.

Aktīvo pārbaužu saskarne

Mākonī aktīvās pārbaudes automātiski izveido hosta saskarni ar IP adresi, kas nav saistīta ar jūsu lokālo tīklu. Pēc noklusējuma šī IP adrese var būt ārēja. Lai saglabātu konsekvenci, pēc izveides varat manuāli dzēst vai pielāgot hosta saskarni.

SAML sertifikāta augšupielāde

SAML autentifikācija netiek atbalstīta, jo tikai daži pakalpojumu sniedzēji pieņem nešifrētus vai neparakstītus SAML ziņojumus.

Plānotie pārskati

Noklusējuma *Cloud Email* paziņojumu veids izmanto skripta transportu un to nevar izmantot plānotajiem pārskatiem. Lai nosūtītu pārskatus pa e-pastu:

1. Izveidojiet jaunu Email paziņojumu veidu.
2. Piešķiriet e-pasta paziņojumu savam lietotājam sadaļā *Lietotāja iestatījumi* > *Paziņojumu veidi*.
3. Konfigurējot plānotos pārskatus, atlasiet šo paziņojumu veidu.

Izstrādātāju centrs

Izstrādātāju centrs

Sāciet izstrādāt pielāgotus Zabbix paplašinājumus, izmantojot norādes par spraudņiem, moduļiem un logrīkiem.

Veidojiet uz Zabbix bāzes

Spraudņi Pārskats par to, kā izstrādāt un pārvaldīt Zabbix spraudņus, lai paplašinātu funkcionalitāti vai integrētu ar ārējām sistēmām.

Python bibliotēka Zabbix vajadzībām Resursi oficiālās `zabbix_utils` bibliotēkas izmantošanai, lai mijiedarbotos ar Zabbix API un veidotu integrācijas.

Lietotāja saskarnes moduļi Rokasgrāmatas un atsauces pielāgotu moduļu izveidei, kas paplašina vai modificē Zabbix lietotāja saskarni, lai tā atbilstu konkrētiem lietošanas gadījumiem.

Logrīki Logrīku struktūras un loģikas pārskats, kā arī norādījumi pielāgotu paneļa elementu izveidei atbilstoši jūsu vajadzībām.

Spraudņa izveide

Šī lapa ir paredzēta tiem, kurus interesē sava spraudņa (neatkarīga datu savācēja) izstrāde Zabbix aģents 2.

Pielāgots spraudnis ļauj paplašināt Zabbix aģents 2 funkcionalitāti ārpus piedāvātajiem iebūvētajiem spraudņiem un ielādējamajiem spraudņiem.

Katrs spraudnis ir *Go* pakotne, kas definē struktūru un ievieš vienu vai vairākas saskarnes (*Exporter*, *Configurator*, *Runner*). Papildinformāciju skatiet sadaļās Spraudņu saskarnes un Savienojumu diagramma.

Šī rokasgrāmata palīdzēs jums izveidot pielāgotu ielādējamu spraudni.

Papildu norādījumus skatiet arī šo repozitoriju saturā:

- Spraudņa piemērs
- Esošie ielādējamie spraudņi

Ko jūs izveidosiet Šī ir soli pa solim apmācība, kā izveidot vienkāršu ielādējamu spraudni ar nosaukumu **MyIP**. Šis spraudnis ieviesīs vienu vienuma atslēgu (`myip`), kas atgriež Zabbix aģenta hosta ārējo IP adresi.

1. darbība: Iestatīšana

1. Spraudnis ir standarta Go modulis. Sāciet ar `go.mod` faila inicializēšanu spraudņa direktorijā, lai izsekotu spraudņa atkarības:

```
cd path/to/plugins/myip # Pārslēgties uz jūsu spraudņa direktoriju
go mod init myip
```

2. Instalējiet obligāto atkarību Zabbix Go SDK (`golang.zabbix.com/sdk`):

```
go get golang.zabbix.com/sdk@$LATEST_COMMIT_HASH
```

Aizstājiet `$LATEST_COMMIT_HASH` ar jaunāko `HEAD` izmaiņu hash no `golang.zabbix.com/sdk` repozitorija atbilstošajā laidiena zarā. Piemēram:

```
go get golang.zabbix.com/sdk@af85407
```

Ņemiet vērā, ka `golang.zabbix.com/sdk` versiju pārvaldība pašlaik netiek atbalstīta, taču nākotnē tas var mainīties.

Papildu atkarības pēc vajadzības var instalēt, izmantojot `go get`.

3. Izveidojiet tukšu `main.go` failu spraudņa pirmkodam:

```
touch main.go
```

Tagad sākotnējā iestatīšana ir pabeigta, un spraudnis ir gatavs izstrādei.

2. darbība: Spraudņa struktūra Iepriekšējā darbībā instalētais modulis `golang.zabbix.com/sdk` nodrošina spraudņu izstrādei nepieciešamo ietvaru un garantē, ka visiem spraudņiem ir vienota struktūra.

1. Iestatiet pamata izpildes plūsmu.

Sāciet ar spraudņa galvenās izpildes plūsmas definēšanu. Pievienojiet `main.go` failam šādu kodu:

```
package main

func main() {
    err := run()
    if err != nil {
        panic(err)
    }
}

func run() error {
    return nil
}
```

Tas izveido spraudņa pamata izpildes plūsmu. Funkcija `run` vēlāk saturēs spraudņa pamatloģiku.

2. Izpētiet spraudņa saskarnes.

Zabbix aģents 2 spraudnis ir jāattēlo ar struktūru, kas ievieš saskarnes no pakotnes `golang.zabbix.com/sdk/plugin`:

- *Accessor* — definē būtiskās metodes, kas jāievieš visiem spraudņiem, piemēram, spraudņa nosaukuma iestatīšanu un vienuma atslēgas noildzes apstrādi.
- Viena vai vairākas no šīm funkcionālajām spraudņa saskarnēm:
  - *Exporter* — veic aptauju un atgriež vērtību (vai vērtības), neko vai kļūdu; bieži tiek izmantota kopā ar *Collector* saskarni.
  - *Collector* — pārvalda periodisku datu vākšanu.
  - *Runner* — definē spraudņa palaišanas un apturēšanas procedūras.
  - *Watcher* — ļauj ieviest neatkarīgu metriku aptaujāšanu, apejot aģenta iekšējo plānotāju; noder uz slazdiem vai notikumiem balstītai uzraudzībai.
  - *Configurator* — definē, kā spraudnis nolasa un piemēro savus konfigurācijas iestatījumus.

Jūs varat vai nu ieviest šīs saskarnes pats, vai izmantot Zabbix Go SDK nodrošinātās noklusējuma saskarnes, vajadzības gadījumā tās pielāgojot.
Šajā pamācībā tiek izmantotas noklusējuma ieviešanas.

3. Izveidojiet spraudņa struktūru.

Tagad importējiet pakotni *plugin* un izveidojiet `myIP` struktūru, kas iegulda `plugin.Base` struktūru:

```
import "golang.zabbix.com/sdk/plugin"

type myIP struct {
    plugin.Base
}
```

Struktūra `myIP` pašlaik atbilst *Accessor* saskarnei.
Metode vienas no funkcionālajām spraudņa saskarnēm — `Exporter` — ieviešanai tiks pievienota vēlāk šajā pamācībā.

3. darbība: Definējiet vienumu atslēgas Jūsu spraudnim ir nepieciešamas vienumu atslēgas, lai vāktu datus un nodotu tos Zabbix serverim vai starpniekserverim.

1. Importējiet *errs* pakotni kļūdu apstrādei:

```
import "golang.zabbix.com/sdk/errs"
```

2. Reģistrējiet vienumu atslēgas, izmantojot funkciju `plugin.RegisterMetrics()` funkcijā `run()`:

```
func run() error {
    p := &myIP{}

    // Register the `myip` item key.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Plugin name
        "myip",                             // Item key name
        "Returns the host's IP address.", // Item key description
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    return nil
}
```

Lai reģistrētu vairākas vienumu atslēgas, atkārtojiet parametrus *metrikas nosaukums* un *apraksts* katrai metrikai.
Piemēram:

```
plugin.RegisterMetrics(&impl, "Myip", "metric.one", "Metric one description.", "metric.two", "Metric two d
```

4. darbība: Iestatiet apstrādātāju Apstrādātājs nodrošina saziņu starp aģents un spraudni.

1. Importējiet *container* pakotni:

```
import "golang.zabbix.com/sdk/plugin/container"
```

2. Funkcijā `run()` pievienojiet kodu, lai izveidotu un iestatītu apstrādātāju:

```
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
}
```

5. darbība: Datu vākšanas ieviešana Datu vākšana tiek veikta, izmantojot Exporter saskarni, kas apraksta `Export` metodi:

```
func Export(
  key string,             // Vācamā vienuma atslēga.
  params []string,        // Argumenti, kas nodoti vienuma atslēgai (`myip[arg1, arg2]`).
  context ContextProvider // Metadati par vienuma atslēgas datu vākšanu.
) (any, error)
```

1. Importējiet nepieciešamās pakotnes HTTP pieprasījumiem un atbildes nolasīšanai:

```
import (
    "io"
    "net/http"
)
```

2. Ieviesiet `Export` metodi `myIP` struktūrai:

```
func (p *myIP) Export(
    key string, params []string, context plugin.ContextProvider,
) (any, error) {
    // Spraudnis var izmantot atšķirīgu datu vākšanas loģiku atkarībā no parametra `key`.
    // Šī ieviešana tikai pārbauda, vai norādītā `key` vērtība tiek atbalstīta.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Žurnāla ieraksts tiks pārsūtīts uz aģents 2 žurnālu.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Savāciet datus un atgrieziet tos.

    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
}
```

6. solis: Izveidojiet un konfigurējiet spraudni

1. Lai izveidotu spraudni, izpildiet:

```
go mod tidy
go build
```

Tādējādi pašreizējā direktorijā tiks izveidots izpildāms fails `myip`.

2. Konfigurējiet Zabbix aģents 2, lai izmantotu spraudni:

```
echo "Plugins.MyIP.System.Path=$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE" > /etc/zabbix_agent2.d/plugins.d/myip.
```

Aizstājiet `$PATH_TO_THE_MYIP_PLUGIN_EXECUTABLE` ar ceļu uz `myip`, kas izveidots 5. solī.

Spraudņa nosaukumam konfigurācijas parametra nosaukumā (šajā pamācībā *MyIP*) jāsakrīt ar spraudņa nosaukumu, kas definēts funkcijā *plugin.RegisterMetrics()*.

3. Lai pārbaudītu spraudni un tā `myip` vienumu, izpildiet:

```
zabbix_agent2 -c /etc/zabbix_agent2.conf -t myip
```

Izvadē jābūt jūsu hosta ārējai IP adresei, un tai vajadzētu izskatīties līdzīgi šādi:

```
myip                                          [s|192.0.2.0]
```

Ar to jūs esat izveidojis vienkāršu ielādējamu spraudni Zabbix aģents 2. Apsveicam!

```
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{}

    // Reģistrē `myip` vienuma atslēgu.
    err := plugin.RegisterMetrics(
        p,
        "MyIP",                             // Spraudņa nosaukums
        "myip",                             // Vienuma atslēgas nosaukums
        "Returns the host's IP address.", // Vienuma atslēgas apraksts
    )
    if err != nil {
        return errs.Wrap(err, "failed to register metrics")
    }

    // Izveido jaunu apstrādātāju.
    h, err := container.NewHandler("MyIP") // Spraudņa nosaukums
    if err != nil {
        return errs.Wrap(err, "failed to create new handler")
    }

    // Iestata žurnalēšanu, lai pārsūtītu žurnālus no spraudņa uz aģentu.
    // Pieejams, izmantojot p.Logger.Infof, p.Logger.Debugf utt.
    p.Logger = h

    // Sāk spraudņa izpildi.
    // Bloķē, līdz tiek saņemts izbeigšanas pieprasījums no aģenta.
    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) {
    // Spraudnis var izmantot atšķirīgu datu vākšanas loģiku atkarībā no `key` parametra.
    // Šī implementācija tikai pārbauda, vai norādītais `key` tiek atbalstīts.
    if key != "myip" {
        return nil, errs.Errorf("unknown item key %q", key)
    }

    // Žurnāls tiks pārsūtīts uz aģenta 2 žurnālu.
    p.Infof(
        "received request to handle %q key with %d parameters",
        key,
        len(params),
    )

    // Savāc datus un atgriež tos.

    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
}
```

Pilns pirmkods

Spraudņu saskarnes

Šajā sadaļā ir aprakstītas pieejamās spraudņu saskarnes.

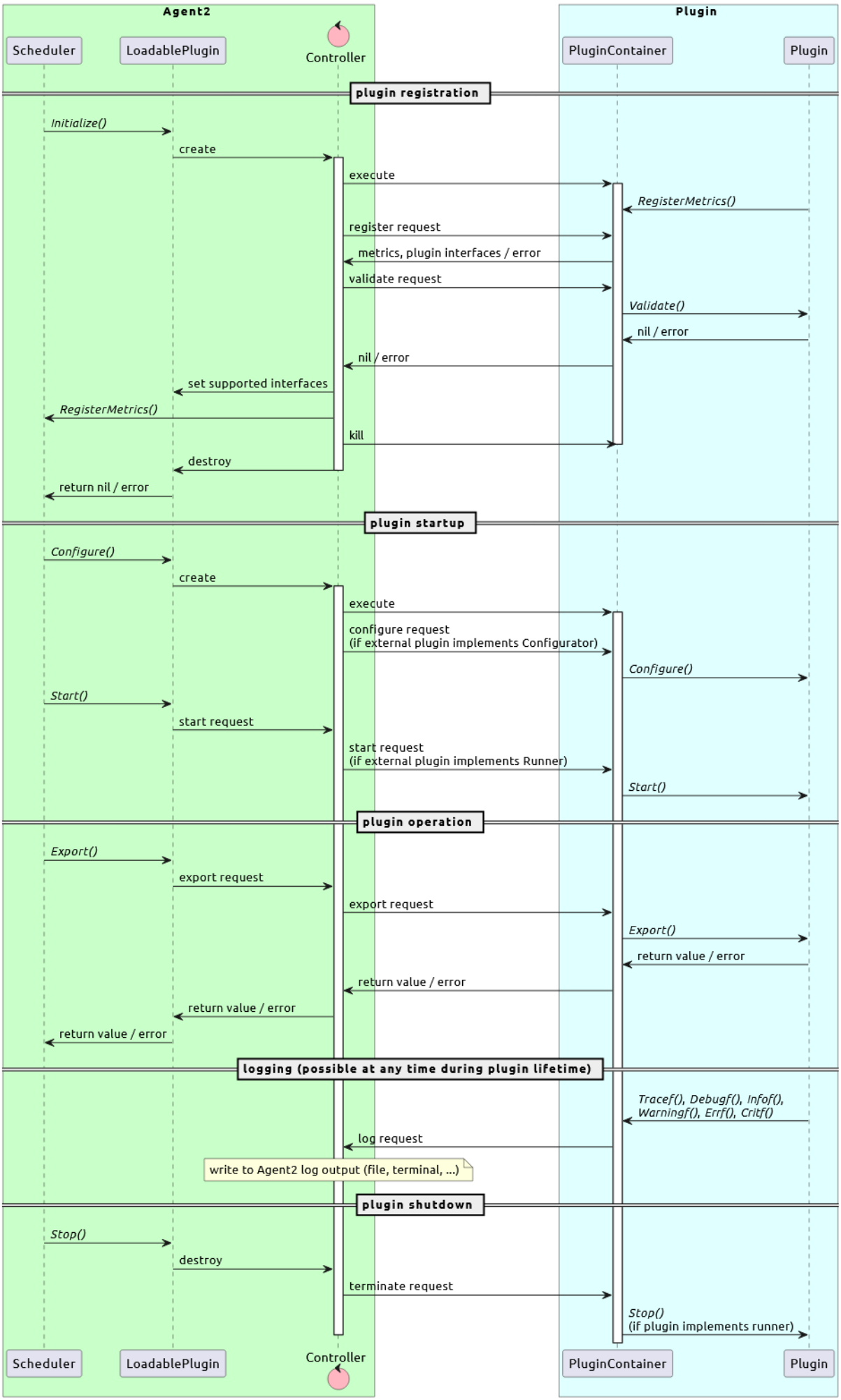
plugin.Exporter *Exporter* ir vienkāršākais interfeiss, kas veic aptauju un atgriež vērtību (vērtības), neko vai kļūdu. Tas pieņem iepriekš parsētu vienuma atslēgu, parametrus un kontekstu. Piekļuve visiem pārējiem spraudņa interfeisiem ir ekskluzīva, un nevienu metodi nevar izsaukt, ja spraudnis jau izpilda uzdevumu. Turklāt katram spraudnim ir noteikts maksimālais vienlaicīgo *Export()* izsaukumu skaits — 100, ko var samazināt atbilstoši katra spraudņa prasībām.

plugin.Configurator *Configurator* saskarne nodrošina spraudņa konfigurācijas parametrus no Zabbix aģents 2 konfigurācijas failiem.

plugin.Runner *Runner* saskarne nodrošina iespēju veikt inicializāciju, kad spraudnis tiek palaists (aktivizēts), un deinicializāciju, kad spraudnis tiek apturēts (deaktivizēts). Piemēram, spraudnis var palaist/apturēt kādu fona *goroutine*, ieviešot Runner saskarni.

Savienojuma diagramma Zabbix aģents 2 izveido divvirzienu savienojumu ar spraudņiem, izmantojot UNIX ligzdas Linux vidē un Named Pipes Windows vidē.

Tālāk redzamā savienojuma diagramma ilustrē saziņas procesu starp Zabbix aģentu 2 un ielādējamu spraudni, kā arī metriku vākšanas procesu.


Agent2
Plugin
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin
plugin registration
Initialize()
create
execute
RegisterMetrics()
register request
metrics, plugin interfaces / error
validate request
Validate()
nil / error
nil / error
set supported interfaces
RegisterMetrics()
kill
destroy
return nil / error
plugin startup
Configure()
create
execute
configure request
(if external plugin implements Configurator)
Configure()
Start()
start request
start request
(if external plugin implements Runner)
Start()
plugin operation
Export()
export request
export request
Export()
return value / error
return value / error
return value / error
return value / error
logging (possible at any time during plugin lifetime)
Tracef(), Debugf(), Infof(),
Warningf(), Errf(), Critf()
log request
write to Agent2 log output (file, terminal, ...)
plugin shutdown
Stop()
destroy
terminate request
Stop()
(if plugin implements runner)
Scheduler
LoadablePlugin
Controller
PluginContainer
Plugin

Python bibliotēka Zabbix համակարգai

zabbix_utils ir oficiālā Zabbix Python bibliotēka.

Ar zabbix_utils jūs varat:

- Izmantot Zabbix API, lai pārvaldītu Zabbix objektus (izveidotu hostus, atjauninātu vienumus, iegūtu notikumus u.c.).
- Ievākt datus no Zabbix aģenta (līdzīgi kā Zabbix get).
- Nosūtīt datus uz Zabbix serveri vai starpniekserveri (līdzīgi kā Zabbix sender).

Bibliotēka atbalsta gan sinhronu, gan asinhronu skriptu izpildi, ļaujot tai apstrādāt gan vienkāršas darbplūsmas, gan liela mēroga paralēlas operācijas, piemēram, masveida datu ievākšanu, lielapjoma eksportēšanu, brīdinājumu apstrādi un efektīvu sistēmas mērogošanu.

Bibliotēka ir saderīga ar atbalstītajām Zabbix servera versijām un mūsdienīgām Python laidienu versijām.

Lai sāktu darbu, instalējiet bibliotēku un pēc tam sekojiet ātrās darba sākšanas ceļvedim.

Instalēšana

zabbix_utils var instalēt, izmantojot jebkuru no tālāk aprakstītajām metodēm.

Prasības:

- Zabbix 6.0 vai jaunāka versija (testēts ar 6.0, 7.0, 7.2, 7.4)
- Python 3.8 vai jaunāka versija (testēts ar 3.8–3.13)
- aiohttp asinhronais HTTP ietvars (nepieciešams tikai asinhronajam režīmam)

Python Package Index (PyPI)

Šī ir visizplatītākā metode lielākajai daļai vidi:

```
pip install zabbix_utils
pip install zabbix_utils[async] # Tikai async atbalstam
```

Zabbix repozitorijs

Izmantojiet šo metodi, ja vēlaties pārvaldīt atkarības, izmantojot savas sistēmas pakotņu pārvaldnieku.

Lejupielādējiet Zabbix pakotnes savai Linux distribūcijai un pēc tam izpildiet tālāk norādītās komandas.

Uz RHEL un atvasinājumiem:

```
dnf install python3-zabbix-utils
dnf install epel-release # Tikai async atbalstam
dnf install python3-aiohttp # Tikai async atbalstam
```

Uz Debian/Ubuntu un atvasinājumiem:

```
apt install python3-zabbix-utils
apt install python3-aiohttp # Tikai async atbalstam
```

Avots (GitHub)

Izmantojiet šo metodi, ja dodat priekšroku jaunākajai izstrādes versijai:

```
git clone https://github.com/zabbix/python-zabbix-utils
cd python-zabbix-utils/
pip install -r requirements.txt # Tikai async atbalstam
python3 setup.py install
```

Lai instalētu asinhronajam režīmam nepieciešamās atkarības, varat izmantot arī kādu no iepriekš aprakstītajām metodēm.

Ātrās darba sākšanas ceļvedis

Pēc tam, kad esat instalējis zabbix_utils, varat to izmantot savā skriptā.

Šis ātrās darba sākšanas ceļvedis parāda, kā:

- Iegūt hosta nosaukumu no Zabbix aģenta.
- Izveidot hostu un trapper vienumu, izmantojot Zabbix API.
- Nosūtīt vērtību uz vienumu.

Ceļvedī skripts ir parādīts soli pa solim, izskaidrojot katru daļu tās ieviešanas brīdī. Līdz ar to pilnais skripts ir pieejams lapas beigās.

Ceļvedī arī tiek pieņemts, ka jūsu Zabbix serveris, aģents un API darbojas lokāli.

Iegūstiet hosta nosaukumu no Zabbix aģenta

Sāciet ar tās sistēmas hosta nosaukuma iegūšanu, kurā darbojas Zabbix aģents. Šis hosta nosaukums būs nepieciešams, lai, izmantojot Zabbix API, izveidotu hostu.

1. Importējiet `Getter` klasi no zabbix_utils. Šī klase darbojas līdzīgi kā Zabbix get un ļauj pieprasīt datus no Zabbix aģenta.

2. Izveidojiet `Getter` instanci, kas izveido savienojumu ar lokālo Zabbix aģentu adresē `127.0.0.1`, portā `10050`.

3. Izsauciet `get()` metodi `Getter` instancei, lai pieprasītu hosta nosaukumu no Zabbix aģenta.

- `get()` metode kā parametru pieņem vienuma atslēgu un nosūta pieprasījumu Zabbix aģentam šim vienumam.
- `get()` metode atgriež objektu, un hosta nosaukums tiek glabāts šī objekta atribūtā `value`.
- `system.hostname` ir iebūvēta Zabbix aģenta atslēga, kas atgriež hosta nosaukumu.

```
from zabbix_utils import Getter

agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value
```

Pievienojieties un piesakieties Zabbix API

Tālāk pievienojieties Zabbix API. Tas ļauj jūsu skriptam pārvaldīt Zabbix objektus, piemēram, hostus un vienumus.

1. Importējiet `ZabbixAPI` klasi no zabbix_utils.

2. Izveidojiet `ZabbixAPI` instanci un norādiet savas Zabbix tīmekļa lietotāja saskarnes URL.

3. Izsauciet `login()` metodi `ZabbixAPI` instancei un norādiet savu lietotājvārdu un paroli. Kontam jābūt atļaujai piekļūt Zabbix API.

4. Izsauciet `logout()` metodi, lai pēc API darbībām aizvērtu sesiju.

```
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()
```

Izveidojiet hostu Zabbix

Tagad, kad esat ieguvis hosta nosaukumu no Zabbix aģenta un esat izveidojis savienojumu ar Zabbix API, varat izveidot jaunu hostu Zabbix.

1. Izsauciet API metodi `host.create()` `ZabbixAPI` instancē un norādiet hosta informāciju:

- `host` — iestatiet uz mainīgo `hostname`, kas satur hosta nosaukumu, kuru ieguvāt no Zabbix aģenta.
- `interfaces` — satur savienojuma informāciju par Zabbix aģentu, kas darbojas hostā.
- `groups` — satur vismaz vienu hostu grupu, kurai hostam jāpieder.

2. Tā kā Zabbix API atgriež jaunizveidotā hosta ID, saglabājiet šo ID mainīgajā `host_id`, lai to izmantotu vēlāk.

3. Izdrukājiet ziņojumu, lai apstiprinātu, ka hosts tika izveidots.

```
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}")
```

Izveidojiet jaunu vienumu Zabbix

Pēc hosta izveides varat tam pievienot vienumu.

1. Definējiet unikālu atslēgu izveidojamajam vienumam.

2. Izsauciet `item.create()` metodi `ZabbixAPI` instancē un norādiet vienuma informāciju:

- `hostid` - iestatiet uz `host_id` mainīgo, kas satur tikko izveidotā hosta ID.
- `name` - vienuma nosaukums.
- `key_` - iestatiet uz `item_key` mainīgo, kuru tikko definējāt.
- `type` - iestatiet uz 2 (trapper vienums); tas ir nepieciešams, lai saņemtu vērtības, ko jūsu skripts nosūtīs nākamajās darbībās.
- `value_type` - iestatiet uz 3 (bezparaksta skaitlisks), tas ir datu tips, ko šis vienums glabā.

3. Tā kā Zabbix API atgriež jaunizveidotā vienuma ID, saglabājiet šo ID mainīgajā `item_id`, lai to izmantotu vēlāk.

4. Izdrukājiet ziņojumu, lai apstiprinātu, ka vienums tika izveidots.

```
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}")
```

Nosūtiet vērtību uz hostu

Tagad, kad esat izveidojis savu hostu un vienumu, varat uz to nosūtīt datus.

1. Importējiet `time` klasi un pirms datu nosūtīšanas uzgaidiet dažas sekundes. Tas nodrošina, ka Zabbix ir pilnībā apstrādājis jūsu jauno hostu un vienumu.

2. Importējiet `Sender` klasi no zabbix_utils. Šī klase darbojas līdzīgi kā Zabbix sender un ļauj jūsu skriptam nosūtīt datus uz Zabbix.

3. Izveidojiet `Sender` instanci, kas pieslēdzas lokālajam Zabbix serverim `127.0.0.1` adresē, portā `10051`.

4. Izsauciet `send_value()` metodi `Sender` instancē, lai nosūtītu vērtību uz jūsu hosta vienumu, un norādiet šādu informāciju:

- `hostname` - iestatiet uz `hostname` mainīgo, kas satur hosta nosaukumu, kuru saņēmāt no Zabbix aģents.
- `item_key` - iestatiet uz `item_key` mainīgo, ko tikko iepriekš definējāt savā skriptā.
- 1 - nosūtāmā vērtība.

4. Izdrukājiet ziņojumu, lai apstiprinātu, ka vērtība tika nosūtīta.

```
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}")
```

Pēc tam, kad šis kods būs veiksmīgi izpildīts, jums vajadzētu redzēt veiksmes ziņojumu, kas norāda, ka Zabbix saņēma jūsu vērtību:

```
Sender response: {"processed": 1, "failed": 0, "total": 1, "time": "0.000151", "chunk": 1}
```

Tagad varat arī pārbaudīt savu Zabbix tīmekļa saskarni (*Monitoring* > *Latest data*), lai redzētu vērtību.

Pilns skripts

Zemāk ir pilns skripts, kas apvieno visas darbības: hosta nosaukuma iegūšanu, hosta un vienuma izveidi, izmantojot Zabbix API, un datu nosūtīšanu uz Zabbix.

```
import time
from zabbix_utils import Getter, ZabbixAPI, Sender

### Iegūt hosta nosaukumu no Zabbix aģenta
agent = Getter(host='127.0.0.1', port=10050)
hostname = agent.get('system.hostname').value

### Izveidot savienojumu ar Zabbix API un pieteikties
api = ZabbixAPI(url='127.0.0.1/zabbix')
api.login(user='Admin', password='zabbix')

### Izveidot hostu 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"Hosts '{hostname}' izveidots ar ID {host_id}")

### Izveidot vienumu 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"Vienums '{item_key}' izveidots ar ID {item_id}")

### Atteikties no API
api.logout()

### Gaidīt, līdz Zabbix apstrādā jauno hostu
time.sleep(10)

### Nosūtīt vērtību hostam
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value(hostname, item_key, 1)
```

```
print(f"Sender atbilde: {response}")
```

Zabbix API izmantošana

zabbix_utils ļauj izmantot Zabbix API, lai pārvaldītu Zabbix objektus, tostarp izveidotu hostus, atjauninātu vienumus, izgūtu notikumus un veiktu citas darbības.

API pieprasījumus var veikt sinhronajā vai asinhronajā režīmā:

- Sinhronajā režīmā jūsu Python skripts nosūta pieprasījumu un pirms turpināšanas gaida atbildi, kas ir piemēroti vienkāršām, secīgām un paredzamām darbībām.
- Asinhronajā režīmā skripts nosūta pieprasījumus, negaidot katru atbildi, ļaujot citām darbībām turpināties paralēli; tas ir efektīvāk lēniem pieprasījumiem vai lielām datu paketēm.

Šīs lapas piemēri ir vērsti uz sinhrono režīmu, lai gan asinhronais režīms izmanto līdzīgus principus. Papildu piemēri ir pieejami GitHub repozitorijā zabbix_utils.

Importēšana

Lai izmantotu zabbix_utils darbam ar Zabbix API, importējiet `ZabbixAPI` klasi savā skriptā:

```
from zabbix_utils import ZabbixAPI
```

> **Note:**
> Ja jau izmantojat kopienas Python bibliotēku, parasti varat aizstāt šo importu ar `ZabbixAPI` no zabbix_utils.

Pieteikšanās

Pirms API pieprasījumu veikšanas jums jāizveido `ZabbixAPI` instance un jāpiesakās Zabbix API.

Izvēlieties metodi, kas vislabāk atbilst tam, kā pārvaldāt akreditācijas datus. Varat izmantot savu lietotājvārdu un paroli vai API tokenus.

Inicializācijas laikā

Veidojot `ZabbixAPI` instanci, varat uzreiz norādīt Zabbix tīmekļa saskarnes URL un savus akreditācijas datus:

```
### Lietotājvārds un parole:
api = ZabbixAPI(url="127.0.0.1/zabbix", user="Admin", password="zabbix")

### API marķieris:
api = ZabbixAPI(url="127.0.0.1/zabbix", token="your_api_token")
```

Ja vēlaties grupēt savienojuma parametrus, varat tos nodot kā Python vārdnīcu un izpakot, izmantojot Python atslēgvārdu argumentu sintaksi:

```
ZABBIX_AUTH = {
    "url": "127.0.0.1/zabbix",
    "user": "Admin",
    "password": "zabbix"
}

api = ZabbixAPI(**ZABBIX_AUTH)
```

`login()` izmantošana

Vispirms varat izveidot `ZabbixAPI` instanci, norādot tikai Zabbix tīmekļa saskarnes URL, un vēlāk izmantot `login()` metodi ar saviem akreditācijas datiem:

```
### Lietotājvārds un parole:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")

### API marķieris:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(token="your_api_token")
```

Vides mainīgo izmantošana

Vispirms varat saglabāt savus akreditācijas datus kā vides mainīgos, piemēram, izmantojot terminālī:

```
### Lietotājvārds un parole:
export ZABBIX_USER="Admin"
export ZABBIX_PASSWORD="zabbix"

### Token:
export ZABBIX_TOKEN="your_api_token"
```

Pēc tam, ja palaižat savu skriptu no tās pašas termināļa sesijas, skriptā esošajai `ZabbixAPI` instancei ir nepieciešams tikai Zabbix tīmekļa lietotāja saskarnes URL:

```
api = ZabbixAPI(url="127.0.0.1/zabbix")
```

URL varat arī saglabāt vides mainīgajā:

```
### Lietotājvārds un parole:
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"
```

Ja URL un akreditācijas dati visi ir iestatīti kā vides mainīgie, skriptā esošajai `ZabbixAPI` instancei vispār nav nepieciešami nekādi argumenti:

```
api = ZabbixAPI()
```

Izrakstīšanās

Ja pieteicāties, izmantojot lietotājvārdu un paroli, pēc API darbību pabeigšanas izsauciet `logout()` metodi:

```
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:
Izsaukt `logout()` metodi nav nepieciešams, ja izmantojat API tokenus.

API pieprasījumi

Pēc pieteikšanās jūs varat veikt jebkurus API pieprasījumus, izsaucot metodes, kas aprakstītas Zabbix API metožu atsaucē.

API metodes tiek izsauktas, izmantojot šādu formātu:

```
api_instance.zabbix_object.method(parameters)
```

Note:
Dažu Zabbix API metožu vai objektu nosaukumos tiek izmantoti vārdi, kas Python ir rezervēti atslēgvārdi (piemēram, `import`). Lai izvairītos no Python kļūdām, pievienojiet pasvītrojuma zīmi metodes vai objekta nosaukumam, kad to izmantojat Python (piemēram, `api.configuration.import_`).

Piemēram, lai skatītu hostu sarakstu ar `host.get`:

```
### 1. Importējiet ZabbixAPI no zabbix_utils:
from zabbix_utils import ZabbixAPI

### 2. Izveidojiet ZabbixAPI instanci un piesakieties:
api = ZabbixAPI(url="127.0.0.1/zabbix")
api.login(user="Admin", password="zabbix")
```

```
### 3. Iegūstiet filtram atbilstošos hostus:
hosts = api.host.get(
    filter={
        "host": [
            "Zabbix server",
            "Linux server"
        ]
    }
)

### 4. Iterējiet cauri atgrieztajiem hostiem un izdrukājiet katra hosta informāciju:
for host in hosts:
    print(host)

### 5. Izrakstieties no API, lai aizvērtu sesiju:
api.logout()
```

Asinhronais režīms

Asinhronais režīms ļauj jūsu skriptam sūtīt API pieprasījumus, negaidot, līdz katrs no tiem tiks pabeigts. Tas var padarīt jūsu skriptu efektīvāku, ja tam jāizpilda daudzi API pieprasījumi vai ja dažu pieprasījumu izpilde aizņem ilgu laiku.

Izmantojot asinhrono režīmu, salīdzinājumā ar sinhrono režīmu ir vairākas būtiskas atšķirības:

- Importējiet Python `asyncio` moduli (vispirms jāinstalē nepieciešamās atkarības).
- Importējiet `AsyncZabbixAPI` `ZabbixAPI` vietā.
- Rakstiet savu kodu `async` funkcijā.
- Izmantojiet `await`, izsaucot API metodes, tostarp `login()` un `logout()`.

> **Attention:**
> Veidojot `AsyncZabbixAPI` instanci, jūs nevarat pieteikties inicializācijas laikā.

Piemēram, lai asinhronajā režīmā skatītu hostu sarakstu ar `host.get`:

```
### 1. Import asyncio for asynchronous mode, and AsyncZabbixAPI from zabbix_utils:
import asyncio
from zabbix_utils import AsyncZabbixAPI

### 2. Define the main async function where all API operations will run:
async def main():

    # 3. Create an AsyncZabbixAPI instance and log in (must await):
    api = AsyncZabbixAPI(url="127.0.0.1")
    await api.login(user="User", password="zabbix")

    # 4. Retrieve hosts matching a filter (must await):
    hosts = await api.host.get(
        filter={
            "host": [
                "Zabbix server",
                "Linux server"
            ]
        }
    )

    # 5. Iterate over returned hosts and print each host's details:
    for host in hosts:
        print(host)

    # 6. Log out from the API to close the session (must await):
    await api.logout()

### 7. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Piemēri

Tālāk sniegtie piemēri parāda biežāk sastopamos Zabbix API uzdevumus, izmantojot `zabbix_utils` bibliotēku.

Papildu piemēri ir pieejami GitHub repozitorija zabbix_utils `examples` direktorijā.

Datu vākšana no Zabbix aģenta

Pārskats

zabbix_utils ļauj vākt datus no Zabbix aģenta (līdzīgi kā Zabbix get).

Datus var vākt sinhronā vai asinhronā režīmā:

- Sinhronajā režīmā jūsu Python skripts pieprasa datus un gaida to saņemšanu, pirms turpināt darbu, kas ir piemēroti vienkāršām, secīgām un paredzamām darbībām.
- Asinhronajā režīmā skripts pieprasa datus, negaidot katru atbildi, tādējādi ļaujot citām darbībām notikt paralēli; tas ir efektīvāk lēniem pieprasījumiem vai lielām datu paketēm.

Šīs lapas piemēri ir vērsti uz sinhrono režīmu, lai gan asinhronais režīms izmanto līdzīgus principus. Papildu piemēri ir pieejami GitHub repozitorijā zabbix_utils.

Importēšana

Lai izmantotu zabbix_utils vienumu vērtību iegūšanai, importējiet `Getter` klasi savā Python skriptā:

```
from zabbix_utils import Getter
```

Pieprasījuma dati

Lai pieprasītu vienuma vērtību:

1. Izveidojiet `Getter` instanci, norādot sava Zabbix aģenta IP adresi un portu.
2. Izsauciet `get()` metodi `Getter` instancē, norādot tā vienuma atslēgu, kuru vēlaties iegūt.

Piemēram, lai pieprasītu datus vienumam `system.uname`:

```
agent = Getter(host='192.0.2.0', port=10050)
response = agent.get('system.uname')
```

Nestandarta IP adreses izmantošana

Ja serverim, kurā darbojas jūsu skripts, ir vairākas IP adreses, varat norādīt `source_ip`, ko `Getter` izmantos, pieslēdzoties Zabbix aģentam:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    source_ip='10.10.7.1'
)
```

Taimauta izmantošana

Jūs varat iestatīt atbildes `timeout` objektam `Getter`, lai noteiktu, cik ilgi jūsu skriptam jāgaida atbilde no Zabbix aģenta, pirms pārtraukt gaidīšanu:

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    timeout=30
)
```

Šifrēšanas izmantošana

`Getter` neietver iebūvētu šifrēšanas atbalstu, taču jūs varat to nodrošināt, izveidojot ietvaru, izmantojot trešo pušu bibliotēkas:

```
def psk_wrapper(sock, tls):
    # ...
    # TLS PSK ietvara ieviešana ligzdai
    # ...
```

```
agent = Getter(
    host='192.0.2.0',
    port=10050,
    socket_wrapper=psk_wrapper
)
```

Atbilde

Atbilde no Zabbix aģenta tiek apstrādāta bibliotēkā un atgriezta kā `AgentResponse` objekts:

```
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
```

Asinhronais režīms

Asinhronais režīms ļauj jūsu skriptam vākt vērtības, negaidot, līdz katra no tām tiek saņemta. Tas var padarīt jūsu skriptu efektīvāku, ja tam ir jāvāc daudzas vērtības vai ja dažu vērtību savākšana aizņem ilgu laiku.

Izmantojot asinhrono režīmu, salīdzinājumā ar sinhrono režīmu ir vairākas būtiskas atšķirības:

- Importējiet Python `asyncio` moduli (vispirms jāinstalē nepieciešamās atkarības).
- Importējiet `AsyncGetter` `Getter` vietā.
- Rakstiet savu kodu `async` funkcijā.
- Izmantojiet `await`, izsaucot `get()` metodi.

Piemēram, lai savāktu vienu vērtību, izmantojot asinhrono režīmu:

```
### 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())
```

Sūtīt datus uz Zabbix serveri vai starpniekserveri

zabbix_utils ļauj sūtīt vienumu vērtības uz trapper vienumu Zabbix serverī vai starpniekserverī (līdzīgi kā Zabbix sender).

Varat sūtīt vienu vērtību, vairākas vērtības vai pat mērķēt uz vairākiem Zabbix klasteriem.

Datus var sūtīt sinhronā vai asinhronā režīmā:

- Sinhronajā režīmā jūsu Python skripts nosūta vērtības un pirms turpināšanas gaida atbildi; tas ir piemērots vienkāršām, secīgām un paredzamām darbībām.
- Asinhronajā režīmā skripts nosūta vērtības, negaidot katru atbildi, tādējādi ļaujot citām darbībām turpināties paralēli; tas ir efektīvāk lēniem pieprasījumiem vai lielām datu paketēm.

Šīs lapas piemēri ir vērsti uz sinhrono režīmu, lai gan asinhronais režīms izmanto līdzīgus principus. Papildu piemēri ir pieejami GitHub repozitorijā zabbix_utils.

Importēšana

Lai izmantotu zabbix_utils vienumu vērtību sūtīšanai, importējiet savā skriptā Sender klasi:

```
from zabbix_utils import Sender
```

Lai sūtītu vairākas vērtības, varat importēt arī ItemValue klasi:

```
from zabbix_utils import Sender, ItemValue
```

Nosūtīt vienu vērtību

Lai nosūtītu vienuma vērtību:

1. Izveidojiet Sender instanci, norādot sava Zabbix servera vai starpniekservera IP adresi un portu.
2. Izsauciet send_value() metodi Sender instancē, izmantojot šādu formātu:

```
sender_instance.send_value('host', 'item.key', 'value', optional_timestamp, optional_nanoseconds)
```

Piemēram, lai nosūtītu 1 uz service.status trapper vienumu hostā Linux server:

```
sender = Sender(server='127.0.0.1', port=10051)
response = sender.send_value('Linux server', 'service.status', 1)
```

Noklusējuma IP adreses neizmantošana

Ja serverim, kurā darbojas jūsu skripts, ir vairākas IP adreses, varat norādīt source_ip, ko Sender izmantos, sūtot vērtības uz Zabbix serveri vai starpniekserveri:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    source_ip='10.10.7.1'
)
```

Timeout izmantošana

Jūs varat iestatīt atbildes timeout parametrā Sender, lai noteiktu, cik ilgi jūsu skriptam jāgaida atbilde no Zabbix servera vai starpniekservera, pirms pārtraukt gaidīšanu:

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    timeout=30
)
```

Aģenta konfigurācijas faila izmantošana

Varat ļaut zabbix_utils nolasīt Server vai ServerActive parametrus no lokāla Zabbix aģenta vai aģenta 2 konfigurācijas faila. Šādos gadījumos, veidojot Sender instanci, nav nepieciešams norādīt savienojuma parametrus:

```
sender = Sender(
    use_config=True,
    config_path='/etc/zabbix/zabbix_agent2.conf'
)
```

Attention:
Ja ServerActive satur vienu vai vairākus Zabbix klasterus ar vairākām servera instancēm, Sender nosūta datus uz pirmo pieejamo serveri katrā klasterī. Ja ServerActive nav iestatīts, tiek izmantota adrese no Server ar noklusējuma portu (10051).

Šifrēšanas izmantošana

Sender neietver iebūvētu šifrēšanas atbalstu, taču to var nodrošināt, izveidojot ietvaru, izmantojot trešo pušu bibliotēkas:

```
def psk_wrapper(sock, tls):
    # ...
    # TLS PSK ietvara ieviešana ligzdai
    # ...
```

```
sender = Sender(
    server='127.0.0.1',
    port=10051,
    socket_wrapper=psk_wrapper
)
```

Atbilde vienai vērtībai

Atbildi, ko atgriež Zabbix serveris vai starpniekserveris, apstrādā bibliotēka un atgriež kā `TrapperResponse` objektu:

```
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
```

Asinhronais režīms

Asinhronais režīms ļauj jūsu Python skriptam sūtīt vērtības, negaidot atbildi no Zabbix serveris vai starpniekserveris. Tas var padarīt jūsu skriptu efektīvāku, ja tam nepieciešams nosūtīt daudzas vērtības vai ja dažu vērtību nosūtīšana aizņem ilgu laiku.

Izmantojot asinhrono režīmu, salīdzinājumā ar sinhrono režīmu ir dažas būtiskas atšķirības:

- Importējiet Python `asyncio` moduli (vispirms jāinstalē nepieciešamās atkarības).
- Importējiet `AsyncSender` `Sender` vietā.
- Rakstiet kodu `async` funkcijā.
- Izmantojiet `await`, izsaucot `send_value()` metodi.

Piemēram, lai, izmantojot asinhrono režīmu, nosūtītu vienu vērtību:

```
### 1. Import asyncio for asynchronous mode, and AsyncSender from zabbix_utils:
import asyncio
from zabbix_utils import AsyncSender

### 2. Define the main async function where all data sending operations (must await) will be executed:
async def main():
    sender = AsyncSender(server='127.0.0.1', port=10051)
    response = await sender.send_value('Linux server', 'service.status', 1)

    # 3. Print the response returned by Zabbix server or proxy:
    print(response)

### 4. Run the async main() function using asyncio's event loop:
asyncio.run(main())
```

Vairāku vērtību nosūtīšana

Lai nosūtītu vairākas vērtības:

1. Sagatavojiet `ItemValue` objektu masīvu, kur katrs izmanto to pašu formātu kā metode `send_value()`.
2. Izveidojiet `Sender` instanci, norādot sava Zabbix servera vai starpniekservera IP adresi un portu.
3. Izsauciet metodi `send()` (metodes `send_value()` vietā) `Sender` instancē, norādot objektu masīvu ar nosūtāmajām vērtībām.

Piemēram, lai nosūtītu piecas vērtības uz dažādiem hostiem:

```
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)
```

Pielāgota gabala lieluma izmantošana

Ja jums ir jānosūta vairāk vērtību, nekā trapera vienums var pieņemt vienā pieprasījumā, varat tās sadalīt gabalos.

Pēc noklusējuma gabala lielums ir 250 vērtības. To var mainīt, iestatot parametru `chunk_size`, veidojot `Sender` instanci.

Piemēram, lai nosūtītu piecas vērtības trīs gabalos (2-2-1), iestatiet parametru `chunk_size` uz 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)
```

Sūtīt vērtības uz vairākām Zabbix kopām

Lai sūtītu vērtības uz vairākām Zabbix kopām:

1. Sagatavojiet Zabbix kopu masīvu. Ja kopai ir vairāki mezgli, vērtība tiks nosūtīta uz katras kopas pirmo **pieejamo** mezglu.
2. Izveidojiet `Sender`, norādot savu Zabbix kopu masīvu.
3. Izsauciet `send_value()` metodi `Sender` instancei, izmantojot to pašu formātu kā `send_value()` metodei.

Piemēram, lai nosūtītu vērtību uz pirmo pieejamo mezglu katrā kopā:

```
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)
```

Atbilde vairāku vērtību gadījumā

Pēc noklusējuma `Sender` atgriež apkopotu rezultātu par vērtību nosūtīšanu visos hostos vai klasteros:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}
```

Ja jums nepieciešama detalizētāka informācija, varat pārbaudīt rezultātus katram klasterim un katram gabalam, izmantojot atribūtu `response.details`:

```
print(response)
### {"processed": 2, "failed": 0, "total": 2, "time": "0.000108", "chunk": 2}

if response.failed == 0:
    print(f"Vērtība veiksmīgi nosūtīta {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"Apstrādāti {resp.processed} no {resp.total} mezglā {node.address}:{node.port}")
            # Apstrādāti 1 no 1 mezglā 127.0.0.1:10051
            # Apstrādāti 1 no 1 mezglā zabbix.example.local:10051
```

Atkļūdošanas žurnalēšana

Lai novērstu problēmas, strādājot ar Zabbix API, vācot datus no Zabbix aģenta vai sūtot datus uz Zabbix serveri vai starpniekserveri, varat iespējot atkļūdošanas žurnalēšanu.

Bibliotēka zabbix_utils izmanto Python standarta moduli `logging`.

Lai redzētu detalizētus atkļūdošanas ziņojumus no bibliotēkas:

1. Importējiet Python moduli `logging` savā skriptā.
2. Konfigurējiet žurnalēšanas sistēmu tā, lai ziņojumi tiktu izvadīti lasāmā formātā, un iestatiet žurnalēšanas līmeni uz 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)
```

Ja atkļūdošanas žurnalēšana ir iespējota, tiek izdrukāta detalizēta informācija par pieprasījumiem un atbildēm, piemēram:

```
[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
```

Moduļi

Kas ir PHP lietotāja saskarnes modulis?

- Modulis ir vienība ar unikālu ID, nosaukumu, aprakstu, autoru un citiem laukiem, kas definēti tā manifest failā, kā arī ar PHP, Javascript un citiem failiem, kas atrodas vienā direktorijā jūsu Zabbix lietotāja saskarnes instalācijas *modules* direktorijā (piemēram, *zabbix/ui/modules*).
- Modulim jāatbilst vienkāršiem noteikumiem, lai garantētu pareizu darbību.
- Modulim jābūt instalētam (izpakotam) un iespējotam lietotāja saskarnē, ko veic administrators.

Kam moduli var izmantot

- Jaunas funkcionalitātes pievienošanai, izmantojot pielāgotas lietotāja saskarnes sadaļas;
- Pielāgotu paneļa logrīku tipu izveidei (skatiet widget modules);
- Esošās funkcionalitātes pārrakstīšanai vai paplašināšanai.

Kam moduli nevar izmantot

- Jaunas API metodes reģistrēšanai vai esošas metodes modificēšanai.

Kā darbojas moduļi

- Iespējots modulis tiek palaists ar katru HTTP pieprasījumu pirms darbības koda izpildes.
- Modulis reģistrēs jaunas darbības vai pārdefinēs esošās.
- Modulis pievienos jaunas lietotāja saskarne sadaļas un noņems vai pārdefinēs esošās.
- Modulis, ja nepieciešams, piesaistīsies lietotāja saskarne notikumiem, piemēram, onBeforeAction un onTerminate.
- Pieprasītā darbība galu galā tiek izpildīta, palaižot darbības kodu — vai nu noklusēto, vai moduļa definēto.

Kur doties tālāk Neatkarīgi no tā, vai dodat priekšroku mācīties praktiski vai vispirms izlasīt vadlīnijas, šajās lapās ir ietverta informācija un darbības, kas nepieciešamas, lai izveidotu savus moduļus:

- Soli pa solim pamācības pirmā moduļa izveidei
- Moduļa failu struktūra
- Logrīku moduļu specifika
- Atkārtoti izmantojami moduļu piemēri

Moduļa failu struktūra

Viss kods, kas saistīts ar moduli, tiek glabāts vienā direktorijā jūsu Zabbix lietotāja saskarnes instalācijas direktorijā **modules** (piemēram, *zabbix/ui/modules*).

```
example_module_directory/          (obligāti)
   manifest.json                   (obligāti)  Metadati un darbību definīcija.
   Module.php                                  Moduļa inicializācija un notikumu apstrāde.
   actions/                                    Darbību kontrolieru faili.
       SomethingView.php
       SomethingCreate.php
       SomethingDelete.php
       data_export/
           ExportAsXml.php
           ExportAsExcel.php
   views/                                      Skatu faili.
       example.something.view.php
       example.something.delete.php
   assets/                                     Jebkuri papildu faili izmantošanai skatos. Jānorāda failā m
       js/                                     JavaScript faili, kas tiek izmantoti skatos.
           example.something.view.js.php
       css/                                    CSS faili, kas tiek izmantoti skatos.
           example.something.css
        image.png                              Attēli, kas tiek izmantoti skatos.
        example.something.file                 Jebkurš fails izmantošanai skatos.
```

Moduļa failu koks

Moduļa izveide Moduļa izveides procesa piemērs sastāv no šādām darbībām (ja pieejams, noklikšķiniet uz faila vai mapes nosaukuma, lai skatītu papildu informāciju par darbību):

1. Izveidojiet jaunu direktoriju modulim mapē **zabbix/ui/modules/**.
2. Pievienojiet failu manifest.json ar moduļa metadatiem.
3. Izveidojiet mapi *views* un definējiet moduļa skatu(s).
4. Izveidojiet mapi *actions* un definējiet moduļa darbību(as).

5. Izveidojiet failu Module.php (vai Widget.php informācijas paneļa logrīkiem) un definējiet inicializācijas un notikumu apstrādes noteikumus.
6. Izveidojiet mapi *assets* JavaScript failiem (ievietojiet mapē *assets/js*), CSS stiliem (ievietojiet mapē *assets/css*) vai jebkuriem citiem papildu failiem.
7. Pārliecinieties, ka manifest.json failā ir norādīti nepieciešamie skati, darbības un resursu faili.
8. Reģistrējiet moduli Zabbix lietotāja saskarnē un sāciet to izmantot.

Note:
Jūs varat reģistrēt un iespējot moduli, tiklīdz ir izveidots fails manifest.json. Kad modulis ir iespējots, varat nekavējoties priekšskatīt visas moduļa failos veiktās izmaiņas, atsvaidzinot Zabbix lietotāja saskarni.

manifest.json

Jebkuram modulim ir nepieciešams fails manifest.json. Failam jāatrodas moduļa galvenajā direktorijā (piemēram, *zabbix/ui/modules/module_name/manifest.json*).

Kā minimālā prasība manifest.json jānorāda šādi lauki:

```
{
    "manifest_version": 2.0,
    "id": "my_ip_address",
    "name": "My IP Address",
    "namespace": "MyIPAddress",
    "version": "1.0"
}
```

manifest.json atbalstītie parametri (nospiediet uz parametra nosaukuma, lai skatītu detalizētu aprakstu):

Parameter	Description	Required
manifest_version	Moduļa manifesta versija.	Yes
id	Unikāls moduļa ID.	
name	Moduļa nosaukums, kas tiks parādīts sadaļā Administration.	
namespace	PHP nosaukumtelpa moduļa klasēm.	
version	Moduļa versija.	
type	Moduļa tips. Logrīkam jābūt iestatītam uz *widget*	Yes for widgets, otherwise no
widget	Logrīka konfigurācija. Tiek izmantota tikai logrīkiem.	
actions	Darbības, ko reģistrēt ar moduli.	
assets	Iekļaujamie CSS stili un JavaScript faili.	No
author	Moduļa autors.	
config	Noklusējuma vērtības pielāgotām moduļa opcijām.	
description	Moduļa apraksts.	
url	Saite uz moduļa aprakstu.	

manifest_version

Moduļa manifesta versija. Pašlaik atbalstītā versija ir **2.0**.

Tips: Double

Piemērs:

```
"manifest_version": 2.0
```

id

Moduļa ID. Tam jābūt unikālam. Lai izvairītos no nosaukumu konfliktiem nākotnē, moduļiem ieteicams izmantot prefiksu (autora vai uzņēmuma nosaukumu, vai jebkuru citu). Piemēram, ja modulis ir piemērs mācību stundām un moduļa nosaukums ir "My module", tad ID būs "example_my_module".

Type: String

Example:

```
"id": "example_my_module"
```

nosaukums

Moduļa nosaukums, kas tiks parādīts sadaļā Administrēšana.

Type: String

Example:

```
"name": "My module"
```

nosaukumtelpa

PHP nosaukumtelpa moduļa klasēm.

Tips: String

Piemērs:

```
"namespace": "ClockWidget"
```

versija

Moduļa versija. Versija tiks parādīta sadaļā Administrēšana.

Tips: String

Piemērs:

```
"version": "1.0"
```

type

Moduļa tips. Nepieciešams logrīkiem, un tam jābūt vienādam ar ”widget”.

Tips: String

Noklusējums: ”module”

Piemērs:

```
"type": "widget"
```

darbības

Darbības, kas jāreģistrē modulī. Katrai darbībai ir obligāti jādefinē *class* objekta atslēga, pārējās darbību atslēgas ir neobligātas.

Tips: Object

Atbalstītās objekta atslēgas, ja *type* ir *module*:

- **write.your.action.name** (object) - darbības nosaukums; tas jāraksta ar mazajiem burtiem [a-z], atdalot vārdus ar punktiem. Atbalsta šādas atslēgas:
  - **class** (string; required) - darbības klases nosaukums.
  - **layout** (string) - darbības izkārtojums. Atbalstītās vērtības: *layout.json*, *layout.htmlpage* (noklusējums), *null*.
  - **view** (string) - darbības skats.

Piemērs:

```
"actions": {
    "module.example.list": {
        "class": "ExampleList",
        "view": "example.list",
        "layout": "layout.htmlpage"
        }
    }
```

Atbalstītās objekta atslēgas, ja *type* ir *widget*:

- **widget.{id}.view** (object) - faila un klases nosaukums logrīka skatam. Aizstājiet **{id}** ar logrīka id vērtību (piemēram, *widget.example_clock.view*). Atbalsta šādas atslēgas:
  - **class** (string; required) - darbības klases nosaukums logrīka skata režīmam, lai paplašinātu noklusēto CControllerDashboardWidgetView klasi. Klases pirmkoda failam jāatrodas direktorijā *actions*.
  - **view** (string) - logrīka skats. Tam jāatrodas direktorijā *views*. Ja skata fails ir *widget.view.php*, kas pēc noklusējuma tiek sagaidīts, šo parametru var izlaist. Ja tiek izmantots cits nosaukums, norādiet to šeit.
- **widget.{id}.edit** (object) - faila nosaukums logrīka konfigurācijas skatam. Aizstājiet **{id}** ar logrīka id vērtību (piemēram, *widget.example_clock.edit*). Atbalsta šādas atslēgas:

- **class** (string; required) - darbības klases nosaukums logrīka konfigurācijas skata režīmam. Klases pirmkoda failam jāatrodas direktorijā *actions*.
- **view** (string) - logrīka konfigurācijas skats. Tam jāatrodas direktorijā *views*. Ja skata fails ir *widget.edit.php*, kas pēc noklusējuma tiek sagaidīts, šo parametru var izlaist. Ja tiek izmantots cits nosaukums, norādiet to šeit.

Piemērs:

```
"actions": {
    "widget.tophosts.view": {
        "class": "WidgetView"
    },
    "widget.tophosts.column.edit": {
        "class": "ColumnEdit",
        "view": "column.edit",
        "layout": "layout.json"
    }
}
```

assets

Iekļaujamie CSS stili un JavaScript faili.

Tips: Object

Atbalstītās objekta atslēgas:

- **css** (array) - Iekļaujamie CSS faili. Failiem jāatrodas mapē *assets/css*.
- **js** (array) - Iekļaujamie JavaScript faili. Failiem jāatrodas mapē *assets/js*.

Piemērs:

```
"assets": {
    "css": ["widget.css"],
    "js": ["class.widget.js"]
}
```

autors

Moduļa autors. Autors tiks parādīts sadaļā Administrēšana.

Type: String

Piemērs:

```
"author": "John Smith"
```

config

Noklusējuma vērtības moduļa opcijām. Objekts var saturēt jebkādas pielāgotas atslēgas. Ja norādīts, šīs vērtības moduļa reģistrācijas laikā tiks ierakstītas datubāzē. Jaunas mainīgās vērtības, kas pievienotas vēlāk, tiks ierakstītas pirmajā izsaukuma reizē. Pēc tam mainīgo vērtības var mainīt tikai tieši datubāzē.

Type: Object

Piemērs:

```
"config": {
    "username": "Admin",
    "password": "",
    "auth_url": "https://example.com/auth"
}
```

apraksts

Moduļa apraksts.

Type: String

Piemērs:

```
"description": "This is a clock widget."
```

logrīks

Logrīka konfigurācija. Tiek izmantota, ja *type* ir iestatīts uz *widget*.

Tips: Object

Atbalstītās objekta atslēgas:

- **name** (string) - tiek izmantots logrīku sarakstā un kā noklusējuma galvene. Ja tukšs, tiks izmantots moduļa parametrs "name".
- **size** (object) - noklusējuma logrīka izmēri. Atbalsta atslēgas:
  - *width* (integer) - noklusējuma logrīka platums.
  - *height* (integer) - noklusējuma logrīka augstums.
- **form_class** (string) - klase ar logrīka lauku formu. Tai jāatrodas *includes* direktorijā. Ja klase ir *WidgetForm.php*, kas pēc noklusējuma tiek sagaidīta, šo parametru var izlaist. Ja tiek izmantots cits nosaukums, norādiet to šeit.
- **js_class** (string) - JavaScript klases nosaukums logrīka skata režīmam, lai paplašinātu noklusējuma CWidget klasi. Klase tiks ielādēta kopā ar paneli. Klases avota failam jāatrodas assets/js direktorijā. Skatiet arī: assets.
- **use_time_selector** (boolean) - nosaka, vai logrīkam ir nepieciešams paneļa laika atlasītājs. Atbalstītās vērtības: *true*, *false* (noklusējums).
- **refresh_rate** (integer) - logrīka atsvaidzināšanas intervāls sekundēs (noklusējums: 60).

Piemērs:

```
"widget": {
    "name": "",
    "size": {
        "width": 12,
        "height": 5
    },
    "form_class": "WidgetForm",
    "js_class": "CWidget",
    "use_time_selector": false,
    "refresh_rate": 60
}
```

url

Saite uz moduļa aprakstu. Logrīkiem šī saite tiks atvērta, noklikšķinot uz palīdzības ikonas ? logā *Pievienot logrīku* vai *Rediģēt logrīku*. Ja **url** nav norādīts, noklikšķinot uz palīdzības ikonas, tiks atvērta vispārīgā lapa Paneļa logrīki.

Tips: String

Piemērs:

```
"url": "http://example.com"
```

Darbības

Darbības ir atbildīgas par moduļa "biznesa loģiku". Darbība parasti sastāv no kontroliera un darbības skata.

Modulis var:

- Izsaukt darbības, kas jau ir definētas Zabbix lietotāja saskarnē.
- Aizstāt noklusējuma darbības ar pielāgotām darbībām.
- Definēt pilnīgi jaunas darbības.

Lai aizstātu noklusējuma darbības uzvedību ar pielāgotu uzvedību, moduļa konfigurācijā definējiet darbību ar tādu pašu nosaukumu. Kad darbība tiek izsaukta, noklusējuma Zabbix darbības vietā tiks izpildīta moduļa darbība.

Darbību faili jāglabā mapē *actions*. Darbības jānorāda manifest.json.

Kontrolieris

Darbību kontroliera darbplūsma:

1. Pārbaudiet, vai visi HTTP pieprasījumā nodotie parametri ir derīgi:

- Izsauciet kontroliera *checkInput()* metodi;
- Izmantojiet validācijas noteikumus, kas definēti CNewValidator.php;
- Izsauciet *validateInput()* metodi.

2. Pārbaudiet lietotāja atļaujas.

3. Sagatavojiet datus atbilstoši nodotajiem parametriem: ja *checkInput()* atgriež true, Zabbix izsauc kontroliera *doAction()* metodi.
4. Sagatavojiet skatam paredzēto **$data** masīvu. Izmantojiet *CControllerResponseData* un *setResponse()* metodi, lai saglabātu atbildi **$data** masīvā.

Piemērs:

```
/**
 * 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:
Pilnu pieejamo kontrolieru klašu sarakstu varat skatīt Zabbix pirmkodā.

Skati

Skata fails saņem datus no kontrollera un pēc tam sagatavo to HTML izskatu.

Note:
Skata(-u) definēšana lietotāja saskarnes modulim nav obligāta, ja vien modulis nav logrīks.
 Paneļa logrīkiem ir nepieciešami vismaz divi skati: viens rediģēšanas režīmam un viens skatīšanas režīmam (tie jāglabā *views* direktorijā).

Skatā ir iespējams izmantot iepriekš definētas Zabbix HTML klases (no */zabbix/ui/include/classes/html*), kā arī pievienot jaunas HTML un CSS klases. Jaunās klases jāglabā moduļa *assets* mapē.

Piemērs:

```
...
(new CColHeader(_('Name')))
```

Tas pievienos jaunu kolonnas nosaukumu *Name* un noformēs tabulas augšējo rindu tāpat kā citās Zabbix lapās.

Darbības skats

Šis ir atsauces fails darbības skata definēšanai.

```
<?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();
```

Resursi

Mape *assets* var saturēt jebkādus failus un apakšmapes, kas nepieder citām direktorijām. To var izmantot šādam saturam:

- JavaScript stiliem (jāatrodas mapē *assets/js*);
- CSS stiliem (jāatrodas mapē *assets/css*);
- Attēliem;
- Fontiem;
- Jebkam citam, kas jums jāiekļauj.

assets/js

Direktorija *assets/js* ir rezervēta, un tajā drīkst atrasties tikai JavaScript faili. Lai tos izmantotu logrīks, norādiet šos failus *manifest.json*.

Piemēram:

```
"assets": {
    "js": ["class.widget.js"]
}
```

assets/css

assets/css ir rezervēts un tajā drīkst atrasties tikai CSS stilu faili. Lai tos izmantotu logrīks, norādiet šos failus *manifest.json*.

Piemēram:

```
"assets": {
    "css": ["mywidget.css"]
}
```

CSS stili

CSS faili var saturēt pielāgotu atribūtu `theme`, lai definētu atšķirīgu stilu konkrētām lietotāja saskarnes tēmām.

Pieejamās tēmas un to atribūtu vērtības:

- **Blue** - [theme='blue-theme']
- **Blue (classic)** - [theme='blue-classic-theme']
- **Dark** - [theme='dark-theme']
- **Dark (classic)** - [theme='dark-classic-theme']
- **High-contrast light** - [theme='hc-light']
- **High-contrast dark** - [theme='hc-dark']

Piemērs:

```
.widget {
    background-color: red;
}

[theme='dark-theme'] .widget {
    background-color: green;
}
```

Reģistrēt jaunu moduli

Šajā sadaļā ir paskaidrots, kā pievienot jaunu moduli Zabbix lietotāja saskarnei.

Priekšnosacījumi Pirms turpināt, pārliecinieties, ka:

- Modulis atrodas jūsu Zabbix lietotāja saskarnes instalācijas *modules* direktorijā (piemēram, *zabbix/ui/modules*).
- Modulī ir vismaz manifest.json faila pamata versija.
- Jums ir piekļuve Zabbix izvēlnes sadaļai Administration (nepieciešams lietotāja lomas tips Super admin).

> **Note:**
> Lietotāja saskarne neinstalēs un pat neatpazīs nesaderīgus moduļus.

Moduļa pievienošana Atveriet lapu *Administrēšana→Vispārīgi→Moduļi* un nospiediet *Skenēt direktoriju*.

Atrodiet savu moduli sarakstā un aktivizējiet to.

Lai aktivizētu moduli, nospiediet hipersaiti *Atspējots* — moduļa stāvoklis mainīsies uz *Iespējots*.

Nospiediet uz moduļa nosaukuma, lai skatītu papildu informāciju par moduli, piemēram, autoru, versiju vai īsu aprakstu (ja tas ir definēts manifestā).

Logrīka priekšskatījums Logrīku moduļi pēc pievienošanas uzreiz kļūst redzami paneļa logrīku sarakstā.

Jūs varat atvērt paneli, pārslēgties uz rediģēšanas režīmu un pievienot logrīku panelim kā parasti.

Kad veicat kādas izmaiņas logrīkā, atsvaidziniet paneli, lai redzētu, kā logrīks izskatās ar jaunākajiem atjauninājumiem.

Logrīki

Logrīki ir Zabbix lietotāja saskarnes moduļi, kas tiek izmantoti paneļiem. Ja vien nav norādīts citādi, visas moduļu vadlīnijas ir attiecināmas arī uz logrīkiem.

Tomēr logrīks būtiski atšķiras no moduļa. Lai izveidotu logrīku:

- failā manifest.json norādiet tipu ”widget” (”type”: ”widget”);
- iekļaujiet vismaz divus skatus: vienu logrīka attēlošanas režīmam un vienu logrīka konfigurācijas režīmam (example.widget.view.php un example.widget.edit.php);
- un kontrolieri logrīka attēlošanai (WidgetView.php);
- izmantojiet un paplašiniet noklusējuma logrīku klases.

Konfigurācija

Šajā lapā ir aprakstītas klases, kuras var izmantot, lai izveidotu logrīka konfigurācijas skatu ar pielāgotiem konfigurācijas laukiem. Logrīka konfigurācijas skats ir logrīka daļa, kas ļauj lietotājam konfigurēt logrīka parametrus attēlošanai.

Logrīks

Primārā logrīka klase, paplašina visu paneļa logrīku bāzes klasi — *CWidget*. Nepieciešama, lai pārrakstītu noklusējuma logrīka uzvedību.

Klasei *Widget* jāatrodas logrīka saknes direktorijā (piemēram, *zabbix/ui/modules/my_custom_widget*).

Widget.php piemērs

```
<?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

Klase *WidgetForm* paplašina noklusējuma klasi *CWidgetForm* un satur *CWidgetField* lauku kopu, kas ir nepieciešama, lai definētu logrīka konfigurācijas glabāšanas struktūru datubāzē un apstrādātu ievades validāciju.

Klasei *WidgetForm* jāatrodas direktorijā *includes*. Ja klasei ir cits nosaukums, tas jānorāda parametrā *widget/form_class* failā *manifest.json*.

includes/WidgetForm.php piemērs

```
<?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

Klase *CWidgetFormView* ir nepieciešama, lai norādītu klasē *WidgetForm* definēto lauku attēlošanas loģiku, nosakot to izskatu un darbību, kad tie tiek atveidoti konfigurācijas skatā.

Klase *CWidgetFormView* atbalsta šādas metodes:

- *addField()* — kā parametru saņem klases *CWidgetFieldView* instanci; katrai *CWidgetField* klasei ir atbilstoša *CWidgetFieldView* klase izmantošanai logrīka konfigurācijas skatā.
- *addFieldset()* — kā parametru saņem klases *CWidgetFieldsGroupView* instanci, kas apvieno laukus sakļaujamā konteinerā.
- *addFieldsGroup()* — kā parametru saņem *CWidgetFormFieldsetCollapsibleView* instanci, kas vizuāli (ar apmali) apvieno laukus grupā.
- *includeJsFile()* — ļauj pievienot JavaScript failu logrīka konfigurācijas skatam.
- *addJavaScript()* — ļauj pievienot iekļautu JavaScript, kas tiks izpildīts, tiklīdz logrīka konfigurācijas skats tiks ielādēts.

Klasei *CWidgetFormView* jāatrodas direktorijā *views*.

views/widget.edit.php piemērs

```
<?php

/**
 * Mana pielāgotā logrīka formas skats.
 *
 * @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(_('Papildu konfigurācija')))
            ->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

JavaScript klase var tikt izmantota, lai logrīka konfigurācijas skatam pievienotu dinamisku darbību un interaktivitāti. Piemēram, var inicializēt krāsu atlasītāju, kas definēts klasē *CWidgetFormView*.

JavaScript klase ir jāielādē kopā ar formu, tāpēc uz to jāatsaucas klasē *CWidgetFormView*, izmantojot metodes *includeJsFile()* un *addJavaScript()*.

Tālāk redzamajā piemērā singleton klases instance tiek nekavējoties izveidota un saglabāta ar nosaukumu *window.my_custom_widget_form*. Tādējādi, atverot formu otro reizi, instance tiks izveidota no jauna.

JavaScript klasei jāatrodas direktorijā *views*.

views/widget.edit.js.php piemērs

```
<?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

Klase *CWidgetField* ir bāzes klase, no kuras tiek mantotas visas formas lauku klases (*CWidgetFieldCheckBox*, *CWidgetFieldTextArea*, *CWidgetFieldRadioButtonList* u.c.). Klases, kas paplašina *CWidgetField*, ir atbildīgas par logrīka konfigurācijas vērtību saņemšanu, saglabāšanu un validēšanu.

Ir pieejamas šādas *CWidgetField* klases.

CWidgetField class	Database field type	Description
CWidgetFieldCheckBox	int32	Viena izvēles rūtiņa.
CWidgetFieldCheckBoxList	array of int32	Vairākas izvēles rūtiņas vienā konfigurācijas laukā.
CWidgetFieldColor	string	Krāsas izvēles lauks.
CWidgetFieldDatePicker	string	Datuma izvēles lauks.
CWidgetFieldIntegerBox	int32	Lauks vesela skaitļa ievadei. Var izmantot minimālās un maksimālās vērtības konfigurēšanai.
CWidgetFieldLatLng	string	Teksta lauks, kas ļauj ievadīt ar komatiem atdalītu platumu, garumu un kartes tālummaiņas līmeni.
CWidgetFieldMultiSelect	-	Bāzes klase vairākatlases laukiem, ko paplašina visas *CWidgetFieldMultiSelect** klases.
CWidgetFieldMultiSelectAction	ID	Vairākatlases lauks darbību izvēlei no darbību saraksta, kas definēts sadaļā *Brīdinājumi* → *Darbības*.
CWidgetFieldMultiSelectGraph	ID	Vairākatlases lauks pielāgotu grafiku izvēlei.
CWidgetFieldMultiSelectGraphPrototype	ID	Vairākatlases lauks pielāgotu grafiku prototipu izvēlei.
CWidgetFieldMultiSelectGroup	ID	Vairākatlases lauks hostu grupu izvēlei.
CWidgetFieldMultiSelectHost	ID	Vairākatlases lauks hostu izvēlei.
CWidgetFieldMultiSelectHostInventory	array of int32	Vairākatlases lauks hostu inventāra lauku izvēlei.
CWidgetFieldMultiSelectItem	ID	Vairākatlases lauks vienumu izvēlei.
CWidgetFieldMultiSelectItemPrototype	ID	Vairākatlases lauks vienumu prototipu izvēlei.
CWidgetFieldMultiSelectMap	ID	Vairākatlases lauks karšu izvēlei.
CWidgetFieldMultiSelectMediaType	ID	Vairākatlases lauks mediju tipu izvēlei.
CWidgetFieldMultiSelectOverrideHost	ID	Vairākatlases lauks datu avota izvēlei hostiem no paneļa vai cita logrīka.
CWidgetFieldMultiSelectService	ID	Vairākatlases lauks pakalpojumu izvēlei.
CWidgetFieldMultiSelectSla	ID	Vairākatlases lauks SLA izvēlei.
CWidgetFieldMultiSelectUser	ID	Vairākatlases lauks lietotāju izvēlei.
CWidgetFieldNavTree	array of (multiple mixed)	Saglabā karšu navigācijas koka struktūru izmantošanai logrīkā *Kartes navigācijas koks*.
CWidgetFieldNumericBox	string	Lauks peldošā komata skaitļa ievadei.
CWidgetFieldPatternSelect	-	Bāzes klase šablonu atlases laukiem, ko paplašina visas *CWidgetPatternSelect** klases.

CWidgetField class	Database field type	Description
CWidgetFieldPatternSelectHost	string	Vairākatlases lauks hostu izvēlei. Atbalsta hostu nosaukumu šablonus.
CWidgetFieldPatternSelectItem	string	Vairākatlases lauks vienumu izvēlei. Atbalsta vienumu nosaukumu šablonus.
CWidgetFieldRadioButtonList	int32	Radiopogu grupa, kas sastāv no vienas vai vairākām radiopogām.
CWidgetFieldRangeControl	int32	Slīdnis vesela skaitļa tipa vērtības izvēlei.
CWidgetFieldReference	string	Saglabā šī logrīka unikālu identifikatoru panelī; izmanto logrīku saziņai.
CWidgetFieldSelect	int32	Nolaižamais atlases lauks.
CWidgetFieldSeverities	array of int32	*CWidgetFieldCheckBoxList* ar iepriekš iestatītām trigeru nopietnības pakāpēm.
CWidgetFieldSparkline	array of (multiple mixed)	Ļauj konfigurēt sparkline diagrammu.
CWidgetFieldTags	array of (string, int32, string)	Ļauj konfigurēt vienu vai vairākas birku filtra rindas.
CWidgetFieldTextArea	string	Teksta apgabals vairākrindu teksta ievadei.
CWidgetFieldTextBox	string	Teksta lauks vienas rindas teksta ievadei.
CWidgetFieldThresholds	array of (string, string)	Ļauj konfigurēt krāsu un skaitļu pārus.
CWidgetFieldTimePeriod	array of string	Laika perioda izvēles lauks.
CWidgetFieldTimeZone	string	Nolaižamais saraksts ar laika joslām.
CWidgetFieldUrl	string	Teksta lauks, kas ļauj ievadīt URL.

Note:
Maksimālais garums `int32` laukiem ir 10 rakstzīmes, `ID` laukiem — 20 rakstzīmes, bet `string` laukiem — 65535 rakstzīmes. Saliktajiem tipiem (piemēram, `array of (string, int32, string)`) ierobežojums attiecas uz katru elementu atsevišķi. Laukam specifiskus garuma ierobežojumus var iestatīt, izmantojot tādas metodes kā `setMaxLength()`.

Attēlojums

Šajā lapā ir aprakstīti komponenti, kurus var izmantot, lai izveidotu logrīka attēlojuma skatu. Logrīka attēlojuma skats ir logrīka daļa, kas saņem datus atbilstoši tā konfigurācijai un attēlo tos informācijas panelī konteinerā.

Attēlojuma skats sastāv no trim daļām:

- Logrīka darbība
- Logrīka skats
- JavaScript

Logdaļas darbība

Logdaļas darbības klase (*WidgetView*) satur metodes darbībām ar logdaļām attēlojuma skata režīmā. Lielākā daļa logdaļu darbību izmanto un/vai paplašina noklusējuma kontroliera klasi *CControllerDashboardWidgetView*.

Logdaļas darbības klasei jāatrodas *actions* direktorijā un tai jābūt norādītai *actions* parametrā (*actions/widget.{id}.view/class*) failā *manifest.json*.

Piemērs actions/WidgetView.php (ieviests Zabbix iebūvētajā *System information* logdaļā)

```
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()
            ]
        ]));
    }
}
```

Logrīka skats

Logrīka skata klase (*CWidgetView*) ir atbildīga par logrīka attēlojuma skata izveidi.

Logrīka skata klasei jāatrodas *views* direktorijā. Ja fails, kas satur logrīka skata klasi, tiek nosaukts citādi nekā pēc noklusējuma (*widget.view.php*), tad tas jānorāda *manifest.json* faila *actions* parametrā (*actions/widget.{id}.view/view*).

views/widget.view.php piemērs

```
<?php

/**
 * Mans pielāgotais logrīka skats.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['name'])
    )
    ->show();
```

JavaScript

JavaScript klase ir atbildīga par logrīka darbības noteikšanu, piemēram, logrīka datu atjaunināšanu, logrīka izmēra maiņu, logrīka elementu attēlošanu utt.

Visas JavaScript darbības izmanto un/vai paplašina visu paneļa logrīku pamata JavaScript klasi — *CWidget*. Klase *CWidget* satur metožu kopu ar noklusējuma ieviešanu logrīka darbībai. Atkarībā no logrīka sarežģītības šīs metodes var izmantot esošajā veidā vai paplašināt.

Klase *CWidget* satur šādas metodes:

- Metodes, kas nosaka logrīka dzīves ciklu: *onInitialize()*, *onStart()*, *onActivate()*, *onDeactivate()*, *onDestroy()*, *onEdit()*.
- Metodes, kas apstrādā logrīka datu atjaunināšanu un attēlošanu: *promiseUpdate()*, *getUpdateRequestData()*, *processUpdateResponse(response)*, *processUpdateErrorResponse(error)*, *setContents(response)*.
- Metodes, kas maina logrīka izskatu: *onResize()*, *hasPadding()*.

JavaScript klasei jāatrodas direktorijā *assets/js*, un tai jābūt norādītai *assets* (*assets/js*) parametrā failā *manifest.json*.

Dzīves cikla metodes

Logrīka dzīves cikla metodes izsauc informācijas panelis dažādos logrīka dzīves cikla posmos tā pastāvēšanas laikā informācijas panelī.

Metode ***onInitialize()*** definē logrīka sākotnējo stāvokli un/vai vērtības, neveicot nekādas HTML vai datu manipulācijas. Šī metode tiek izsaukta, kad logrīks tiek izveidots (tiek inicializēts logrīka objekts), parasti pievienojot logrīku paneļa lapai vai ielādējot paneļa lapu.

Piemērs:

```
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;
}
```

Metode ***onStart()*** definē logrīka HTML struktūru, neveicot nekādu datu apstrādi. Šī metode tiek izsaukta pirms pirmās paneļa lapas aktivizēšanas, tas ir, pirms panelis un tā logrīki tiek pilnībā parādīti lietotājam.

Piemērs:

```
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`
        );
    }
}
```

Metode ***onActivate()*** padara logrīku aktīvu un interaktīvu, iespējojot pielāgotus notikumu klausītājus (lai reaģētu uz lietotāja darbībām) un uzsākot logrīka atjaunināšanas ciklu (lai uzturētu tā saturu aktuālu). Šī metode tiek izsaukta, kad paneļa lapa tiek aktivizēta, tas ir, kad tā ir pilnībā attēlota lietotāja saskarnē.

Ņemiet vērā, ka pirms *onActivate()* metodes izsaukšanas logrīks ir neaktīvā stāvoklī (`WIDGET_STATE_INACTIVE`). Pēc veiksmīgas izsaukšanas logrīks pāriet aktīvā stāvoklī (`WIDGET_STATE_ACTIVE`). Aktīvajā stāvoklī logrīks reaģē, uztver notikumus, periodiski atjaunina savu saturu un var mijiedarboties ar citiem logrīkiem.

Piemērs:

```
onActivate() {
    this._startClock();

    this._resize_observer = new ResizeObserver(this._events.resize);
    this._resize_observer.observe(this._target);
}
```

onDeactivate() metode aptur visas logrīka darbības un interaktivitāti, deaktivizējot pielāgotos notikumu klausītājus un apturot logrīka atjaunināšanas ciklu. Šī metode tiek izsaukta, kad paneļa lapa tiek deaktivizēta, tas ir, kad no tās pārslēdzas prom vai tā tiek dzēsta, vai arī kad logrīks tiek dzēsts no paneļa lapas.

Ņemiet vērā, ka pirms *onDeactivate()* metodes izsaukšanas logrīks ir aktīvā stāvoklī (`WIDGET_STATE_ACTIVE`). Pēc veiksmīgas izsaukšanas logrīks pāriet neaktīvā stāvoklī (`WIDGET_STATE_INACTIVE`).

Piemērs:

```
onDeactivate() {
    this._stopClock();
    this._resize_observer.disconnect();
}
```

onDestroy() metode veic tīrīšanas uzdevumus, pirms logrīks tiek dzēsts no paneļa, kas var ietvert datubāzes savienojuma aizvēršanu, kas tika izveidots logrīka inicializācijas laikā, pagaidu datu dzēšanu, lai atbrīvotu sistēmas atmiņu un izvairītos no resursu noplūdēm, ar izmēra maiņas notikumiem vai pogu klikšķiem saistīto notikumu klausītāju reģistrācijas atcelšanu, lai novērstu nevajadzīgu notikumu apstrādi un atmiņas noplūdes, utt. Šī metode tiek izsaukta, kad logrīks vai paneļa lapa, kurā tas atrodas, tiek dzēsta.

Ņemiet vērā, ka pirms *onDestroy()* metodes izsaukšanas logrīks aktīvā stāvoklī (`WIDGET_STATE_ACTIVE`) vienmēr tiek deaktivizēts, izsaucot *onDeactivate()* metodi.

Piemērs:

```
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);
    }
}
```

onEdit() metode nosaka logrīka izskatu un darbību, kad panelis pārslēdzas rediģēšanas režīmā. Šī metode tiek izsaukta, kad panelis pārslēdzas rediģēšanas režīmā, parasti tad, kad lietotājs mijiedarbojas ar logrīka pogu *Edit* vai paneļa pogu *Edit dashboard*.

Piemērs:

```
onEdit() {
    this._deactivateGraph();
}
```

Atjaunināšanas procesa metodes

Logrīka atjaunināšanas procesa metodes ir atbildīgas par atjauninātu datu iegūšanu no Zabbix serveris vai jebkura cita datu avota un to attēlošanu logrīkā.

promiseUpdate() metode uzsāk datu atjaunināšanas procesu, iegūstot datus, parasti izmantojot tīmekļa pieprasījumus vai API izsaukumus. Šī metode tiek izsaukta, kad tiek parādīta paneļa lapa, un pēc tam periodiski, līdz paneļa lapa tiek pārslēgta uz citu paneļa lapu.

Tālāk ir sniegts piemērs noklusējuma *promiseUpdate()* metodes ieviešanai, ko izmanto lielākā daļa Zabbix vietējo logrīku. Noklusējuma ieviešanā *promiseUpdate()* metode ievēro vispārīgu modeli datu iegūšanai no servera. Tā izveido jaunu `Curl` objektu ar atbilstošo URL un pieprasījuma parametriem, nosūta `POST` pieprasījumu, izmantojot *fetch()* metodi ar datu objektu, ko izveido *getUpdateRequestData()* metode, un attiecīgi apstrādā atbildi (vai kļūdas atbildi) ar *processUpdateResponse(response)* vai *processUpdateErrorResponse(error)*. Šī ieviešana ir piemērota lielākajai daļai logrīku, jo tie parasti iegūst datus JSON formātā un apstrādā tos konsekventā veidā.

```
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);
        });
    }
```

Metode ***getUpdateRequestData()*** sagatavo servera pieprasījuma datus logrīka atjaunināšanai, apkopojot dažādas īpašības un to atbilstošās vērtības (logrīka identifikatorus, filtra iestatījumus, laika diapazonus u.c.) no logrīka stāvokļa un konfigurācijas, un izveidojot datu objektu, kas attēlo nepieciešamo informāciju nosūtīšanai serverim atjaunināšanas pieprasījumā. Šī metode tiek izsaukta tikai kā daļa no noklusējuma *promiseUpdate()* metodes, tas ir, logrīka atjaunināšanas procesa laikā.

Noklusējuma ieviešana:

```
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
    };
}
```

processUpdateResponse(response) metode apstrādā atbildi, kas saņemta no servera pēc atjaunināšanas pieprasījuma, un, ja

atjaunināšanas process ir bijis veiksmīgs un bez kļūdām, notīra logrīka datus un parāda jauno saturu ar *setContents()* metodi. Šī metode tiek izsaukta tikai kā daļa no noklusējuma *promiseUpdate()* metodes, tas ir, logrīka atjaunināšanas procesa laikā.

Noklusējuma implementācija:

```
processUpdateResponse(response) {
    this._setHeaderName(response.name);

    this._updateMessages(response.messages);
    this._updateInfo(response.info);
    this._updateDebug(response.debug);

    this.setContents(response);
}
```

processUpdateErrorResponse(error) metode apstrādā atbildi, kas saņemta no servera pēc atjaunināšanas pieprasījuma, ja atbilde ir kļūda, un parāda kļūdas ziņojumu/-us. Šī metode tiek izsaukta tikai kā daļa no noklusējuma *promiseUpdate()* metodes, tas ir, logrīka atjaunināšanas procesa laikā.

Noklusējuma ieviešana:

```
processUpdateErrorResponse(error) {
    this._updateMessages(error.messages, error.title);
}
```

setContents(response) metode parāda logrīka saturu, ja logrīka atjaunināšanas process ir bijis veiksmīgs un bez kļūdām, kas var ietvert DOM elementu manipulēšanu, UI komponentu atjaunināšanu, stilu vai formatējuma piemērošanu utt. Šī metode tiek izsaukta tikai kā daļa no noklusējuma *processUpdateResponse(response)* metodes, tas ir, apstrādājot no servera saņemto atbildi pēc atjaunināšanas pieprasījuma.

Noklusējuma ieviešana:

```
setContents(response) {
    this._body.innerHTML = response.body ?? '';
}
```

Logrīka attēlojuma modificēšanas metodes

Logrīka attēlojuma modificēšanas metodes ir atbildīgas par logrīka izskata modificēšanu.

Metode ***onResize()*** ir atbildīga par logrīka vizuālo elementu pielāgošanu, lai tie atbilstu jaunajam logrīka izmēram, kas var ietvert elementu pārkārtošanu, elementu izmēru pielāgošanu, teksta saīsināšanu, atlikto ielādi, lai uzlabotu atsaucību izmēra maiņas laikā, u.c. Šī metode tiek izsaukta, kad logrīka izmērs tiek mainīts, piemēram, kad lietotājs manuāli maina logrīka izmēru vai kad tiek mainīts pārlūka loga izmērs.

Piemērs:

```
onResize() {
    if (this.getState() === WIDGET_STATE_ACTIVE) {
        this._startUpdating();
    }
}
```

Metode ***hasPadding()*** ir atbildīga par 8px vertikālās atkāpes piemērošanu logrīka apakšdaļā, ja tas ir konfigurēts rādīt savu galveni. Šī metode tiek izsaukta, kad tiek aktivizēta paneļa lapa, tas ir, kad tā kļūst par lietotāja saskarnē attēloto lapu.

Noklusējuma ieviešana:

```
hasPadding() {
    return this.getViewMode() !== ZBX_WIDGET_VIEW_MODE_HIDDEN_HEADER;
}
```

Dažiem logrīkiem ir nepieciešams izmantot visu pieejamo logrīka vietu, lai, piemēram, konfigurētu pielāgotu fona krāsu. Tālāk ir parādīts *hasPadding()* metodes ieviešanas piemērs, kas tiek izmantots Zabbix iebūvētajā *Vienuma vērtība* logrīkā.

```
hasPadding() {
    return false;
}
```

Pamācības

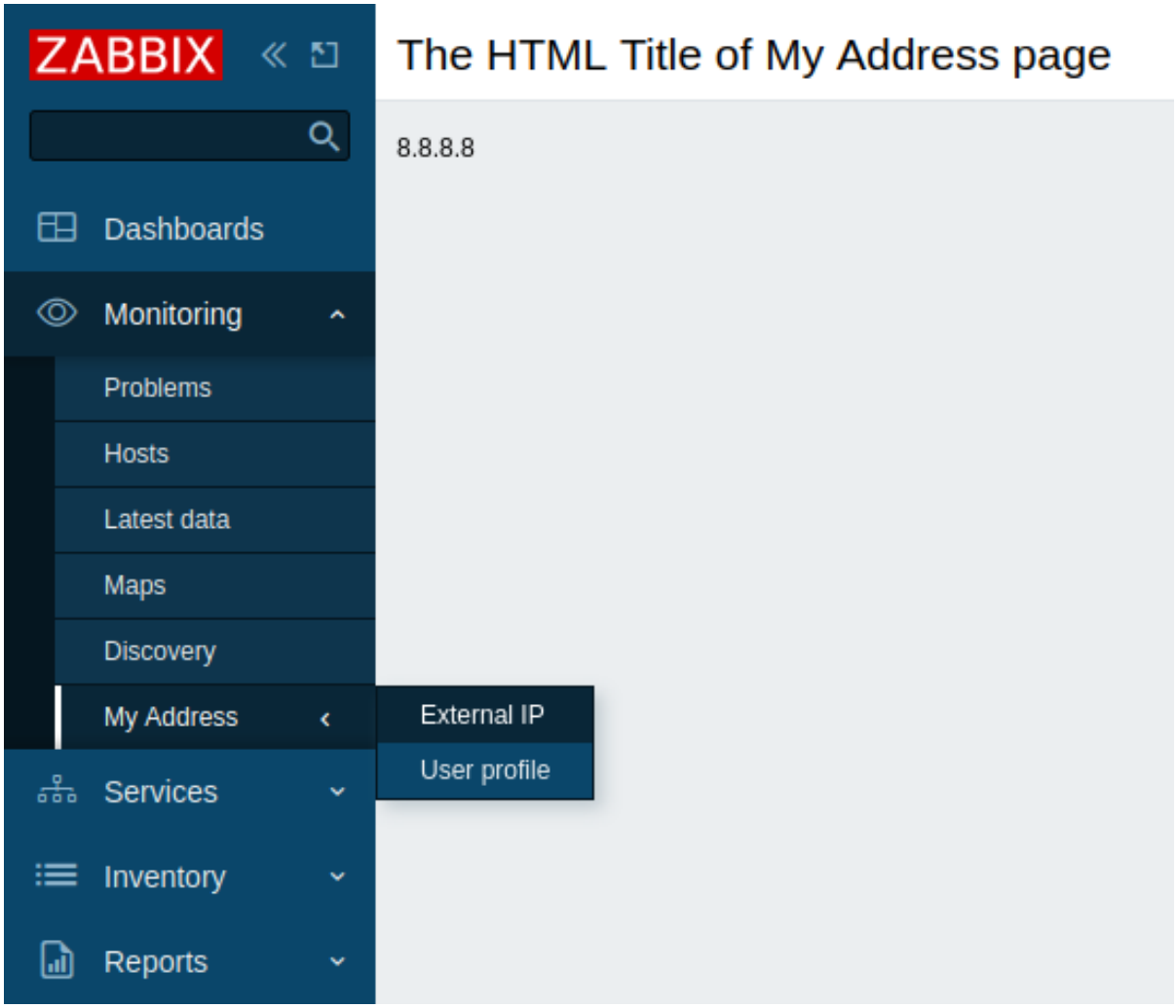
Šajā sadaļā ir iekļautas praktiskas soli pa solim izstrādātas pamācības, lai parādītu, kā izveidot pielāgotu moduli un logrīku Zabbix.

Create a module (tutorial)

Šī ir soli pa solim apmācība, kurā parādīts, kā izveidot vienkāršu Zabbix lietotāja saskarnes moduli. Jūs varat lejupielādēt visus šī moduļa failus kā ZIP arhīvu: MyAddress.zip.

Ko jūs izveidosiet

Šīs pamācības laikā jūs vispirms izveidosiet lietotāja saskarnes moduli, kas pievieno jaunu *Mana adrese* izvēlnes sadaļu, un pēc tam pārveidosiet to par modernāku lietotāja saskarnes moduli, kas veic HTTP pieprasījumu uz *https://api.seeip.org* un jaunā lapā jaunizveidotajā *Mana adrese* izvēlnes sadaļā attēlo atbildi — jūsu datora IP adresi. Lūk, kā izskatīsies pabeigtais modulis:



I daļa — Jauna izvēlnes sadaļa

Pievienojiet tukšu moduli Zabbix lietotāja saskarnei

1. Izveidojiet direktoriju *MyAddress* savas Zabbix lietotāja saskarnes instalācijas *modules* direktorijā (piemēram, *zabbix/ui/modules*).
2. Izveidojiet *manifest.json* failu ar moduļa pamata metadatiem (skatiet atbalstīto parametru aprakstu).

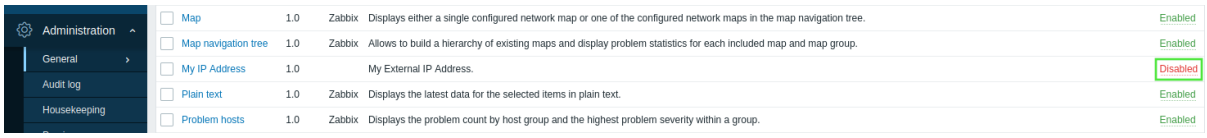
ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
```

```
    "namespace": "MyAddress",
    "description": "My External IP Address."
}
```

3. Zabbix lietotāja saskarnē atveriet sadaļu *Administration* → *General* → *Modules* un noklikšķiniet uz pogas *Scan directory*.

4. Sarakstā atrodiet jauno moduli *My IP Address* un noklikšķiniet uz hipersaites "Disabled", lai mainītu moduļa statusu no "Disabled" uz "Enabled" (ja modulis nav norādīts sarakstā, skatiet sadaļu troubleshooting).



Modulis tagad ir reģistrēts lietotāja saskarnē. Tomēr tas vēl nav redzams, jo jums joprojām ir jādefinē moduļa funkcionalitāte. Kad pievienosiet saturu moduļa direktorijam, pēc lapas atsvaidzināšanas izmaiņas nekavējoties būs redzamas Zabbix lietotāja saskarnē.

Izveidojiet izvēlnes sadaļu

1. Izveidojiet *Module.php* failu *MyAddress* direktorijā.

Šis fails ievieš jaunu *Module* klasi, kas paplašina noklusējuma *CModule* klasi. *Module* klase galvenajā izvēlnē ievietos jaunu *My Address* izvēlnes sadaļu.

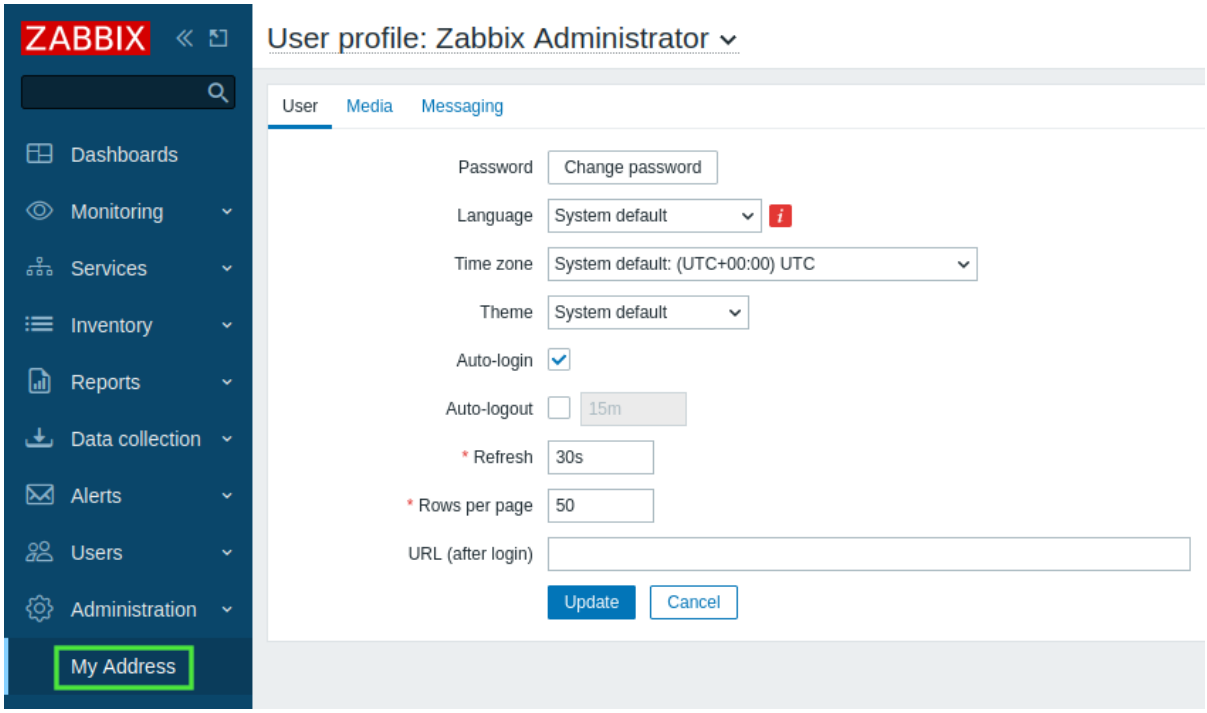
Metode *setAction()* norāda darbību, kas tiks izpildīta, noklikšķinot uz izvēlnes sadaļas. Sākumā varat izmantot iepriekš definēto darbību *userprofile.edit*, kas atvērs *Lietotāja profils* lapu. Šīs pamācības III daļā jūs uzzināsiet, kā izveidot pielāgotu darbību.

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:
Varat aizstāt `'userprofile.edit'` ar citām darbībām, piemēram, `'charts.view'` (atver pielāgotus grafikus), `'problems.view'` (atver *Uzraudzība* → *Problēmas*), vai `'report.status'` (atver *Sistēmas informācija* atskaiti).

3. Atsvaidziniet Zabbix lietotāja saskarne. Tagad Zabbix galvenās izvēlnes apakšā ir pieejama jauna *My Address* sadaļa. Noklikšķiniet uz *My Address*, lai atvērtu *Lietotāja profils* lapu.



II daļa — izvēlnes sadaļas atrašanās vietas maiņa

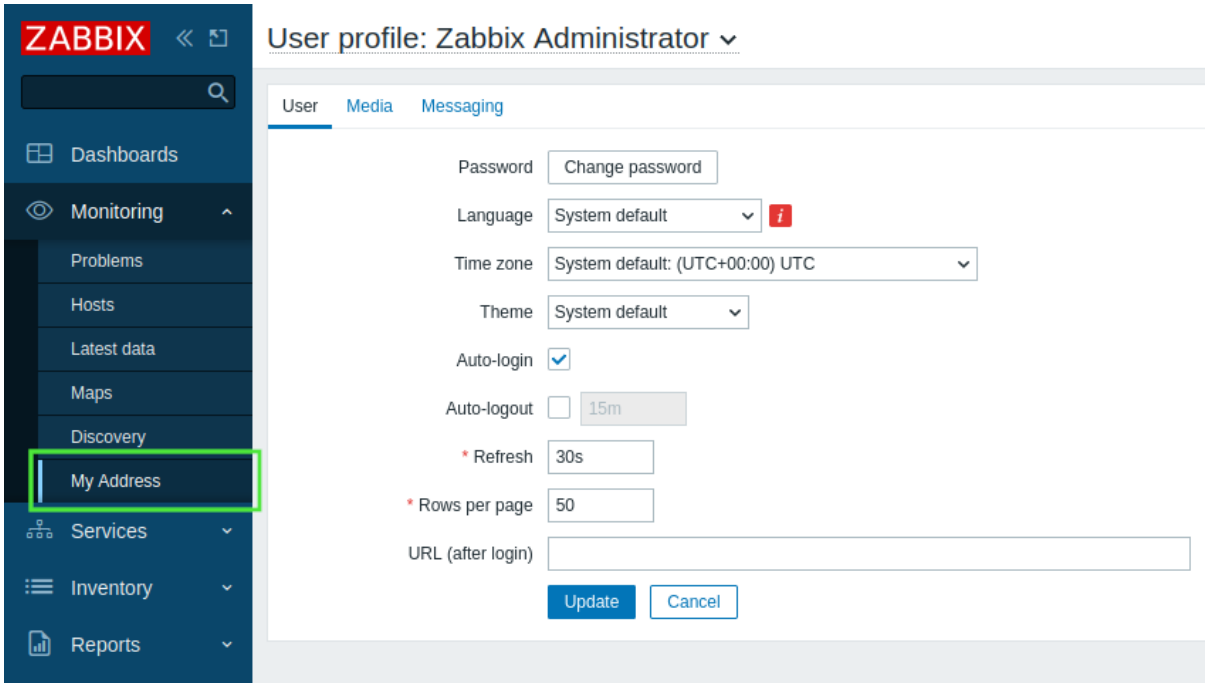
Šajā daļā jūs pārvietosiet izvēlnes sadaļu *My Address* uz sadaļu *Monitoring* un pēc tam tai pievienosiet ligzdotu izvēlni. Rezultātā lietotāji varēs piekļūt divām apakšizvēlnes lapām no izvēlnes sadaļas *Monitoring* → *My Address*.

1. Atveriet un rediģējiet failu *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. Atsvaidziniet Zabbix lietotāja saskarni. Izvērsiet izvēlnes sadaļu *Monitoring* un pārliecinieties, ka sadaļa *My address* tagad atrodas zem sadaļas *Discovery*.



3. Lai izvēlnes sadaļai *My Address* pievienotu ligzdotas lapas, vēlreiz atveriet un rediģējiet failu *Module.php*.

Šajā solī tiek izveidotas divas apakšsadaļas:

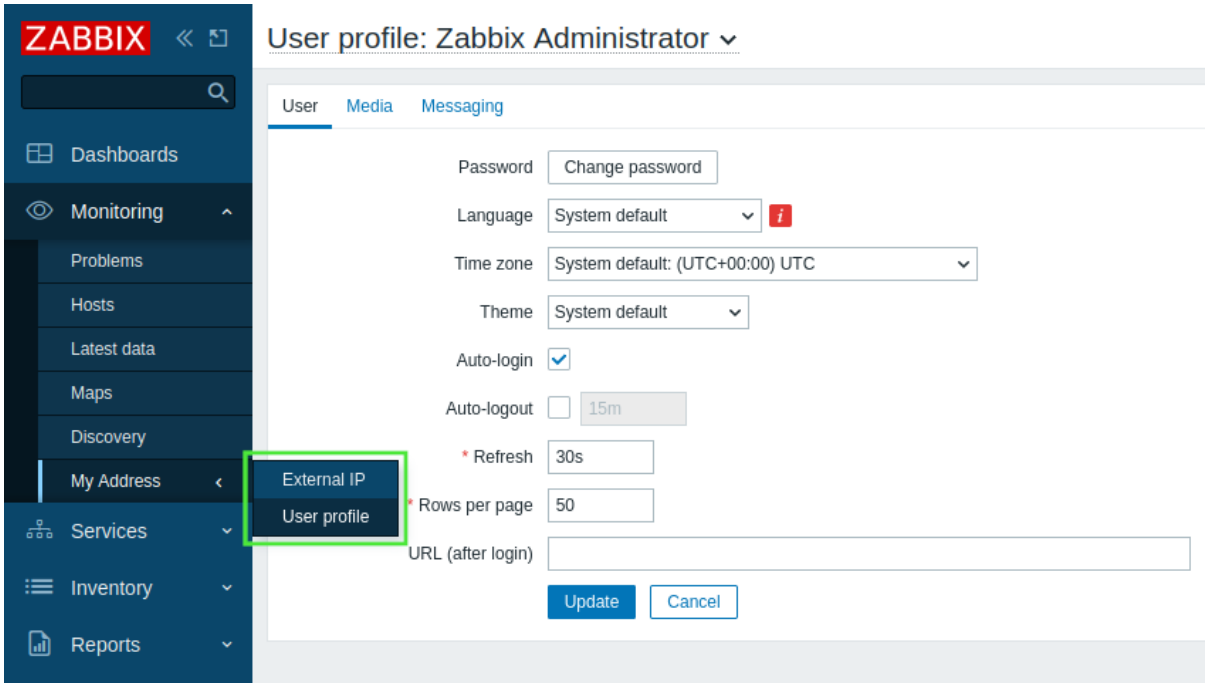
- *External IP*, kas izpilda jaunu darbību 'my.address', kura tiks definēta nākamajos soļos;
- *User profile*, kas izpilda iepriekš definēto darbību 'userprofile.edit', lai atvērtu lapu *User profile*.

Ņemiet vērā, ka ligzdotajai izvēlnei papildus iepriekšējos soļos izmantotajām klasēm ir jāizmanto arī klase *CMenu*.

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. Atsvaidziniet Zabbix lietotāja saskarni. Pārliecinieties, ka izvēlnes sadaļā *My address* tagad ir trešā līmeņa izvēlne ar divām lapām — *External IP* un *User profile*.



III daļa — moduļa darbība

Darbība tiek ieviesta divos failos — *actions/MyAddress.php* un *views/my.address.php*. Fails ***actions/MyAddress.php*** nodrošina biznesa loģikas ieviešanu, savukārt fails ***views/my.address.php*** ir atbildīgs par skatu.

1. Izveidojiet direktoriju *actions* direktorijā *MyAddress*.
2. Izveidojiet failu *MyAddress.php* direktorijā *actions*.

Darbības loģika tiks definēta klasē *MyAddress*. Šī darbības klase ieviesīs četras funkcijas: *init()*, *checkInput()*, *checkPermissions()* un *doAction()*. Zabbix lietotāja saskarne izsauc funkciju *doAction()*, kad tiek pieprasīta darbība. Šī funkcija ir atbildīga par moduļa biznesa loģiku.

> **Attention:**
> Dati ir jāorganizē kā asociatīvs masīvs. Masīvs var būt daudzdimensionāls un var saturēt jebkādus datus, ko sagaida skats.

ui/modules/MyAddress/actions/MyAddress.php

```
<?php

namespace Modules\MyAddress\Actions;

use CController,
    CControllerResponseData;

class MyAddress extends CController {

    public function init(): void {
        $this->disableCsrfValidation();
    }

    protected function checkInput(): bool {
        return true;
    }

    protected function checkPermissions(): bool {
        return true;
    }

    protected function doAction(): void {
        $data = ['my-ip' => file_get_contents("https://api.seeip.org")];
        $response = new CControllerResponseData($data);
        $this->setResponse($response);
```

```
    }
}
```

3. Izveidojiet jaunu direktoriju *views* direktorijā *MyAddress*.
4. Izveidojiet failu *my.address.php* direktorijā *views* un definējiet moduļa skatu.

Ņemiet vērā, ka mainīgais `$data` ir pieejams skatā bez īpašas tā definēšanas. Ietvars automātiski nodod asociatīvo masīvu skatam.

ui/modules/MyAddress/views/my.address.php

```
<?php

(new CHtmlPage())
    ->setTitle(_('The HTML Title of My Address Page'))
    ->addItem(new CDiv($data['my-ip']))
    ->show();
```

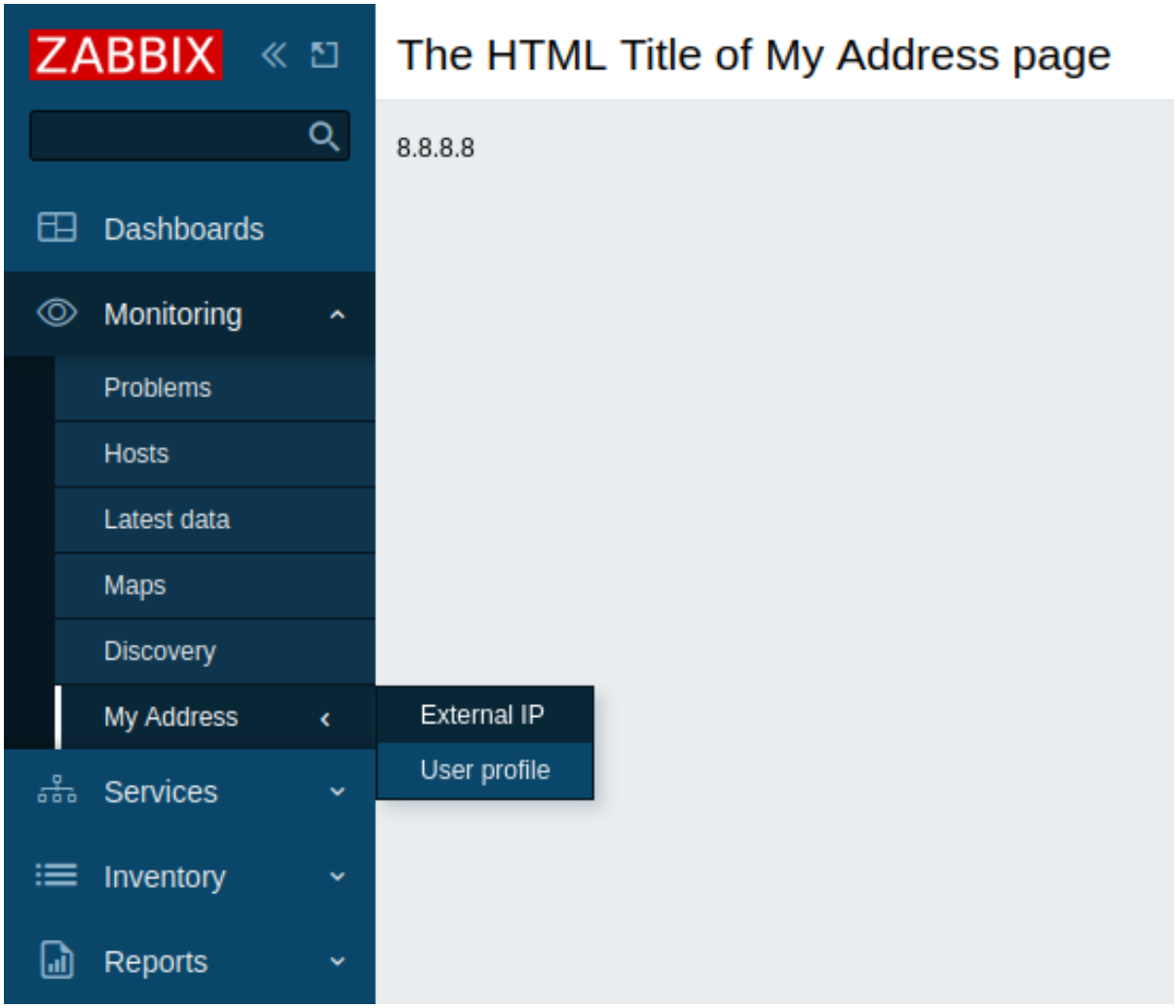
5. Moduļa darbība ir jāreģistrē failā *manifest.json*. Atveriet *manifest.json* un pievienojiet jaunu objektu `actions`, kas satur:

- darbības atslēgu ar darbības nosaukumu, kas rakstīts ar mazajiem burtiem (a-z) un ar vārdiem, kas atdalīti ar punktiem (piemēram, `my.address`);
- darbības klases nosaukumu (`MyAddress`) kā vērtību objekta `my.address` atslēgai `class`;
- darbības skata nosaukumu (`my.address`) kā vērtību objekta `my.address` atslēgai `view`.

ui/modules/MyAddress/manifest.json

```
{
    "manifest_version": 2.0,
    "id": "my-address",
    "name": "My IP Address",
    "version": "1.0",
    "namespace": "MyAddress",
    "description": "My External IP Address.",
    "actions": {
        "my.address": {
            "class": "MyAddress",
            "view": "my.address"
        }
    }
}
```

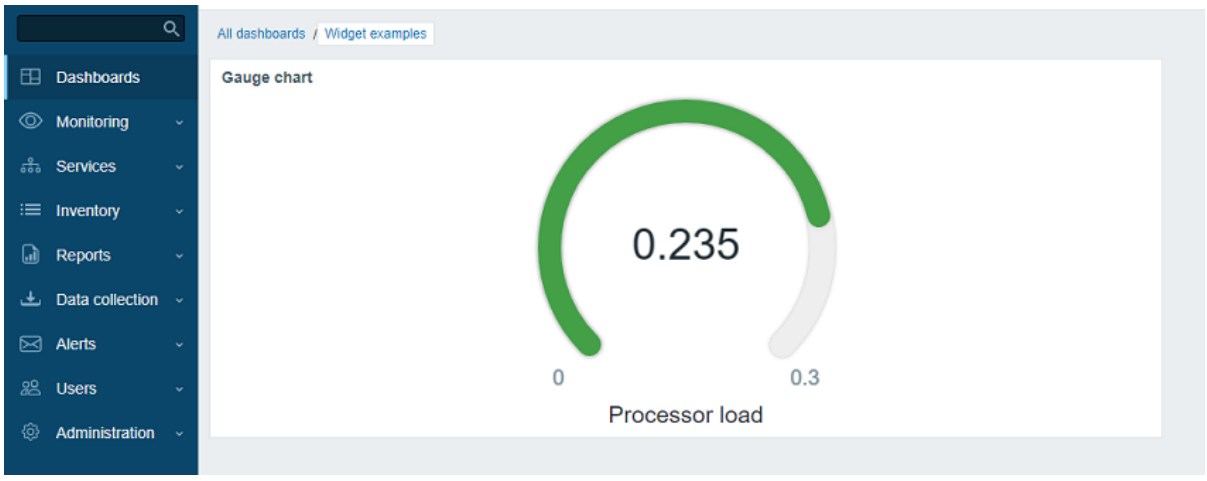
6. Atsvaidziniet Zabbix lietotāja saskarni. Noklikšķiniet uz *My address* → *External IP*, lai redzētu sava datora IP adresi.



Create a widget (tutorial)

Šī ir soli pa solim apmācība, kurā parādīts, kā izveidot vienkāršu paneļa logrīku. Jūs varat lejupielādēt visus šī logrīka failus kā ZIP arhīvu: lesson_gauge_chart.zip.

Ko jūs izveidosiet Šīs pamācības laikā jūs vispirms izveidosiet vienkāršu "Hello, world!" logrīku un pēc tam pārveidosiet to par modernāku logrīku, kas attēlo vienuma vērtību kā mērinstrumenta diagrammu. Lūk, kā izskatīsies pabeigtais logrīks:



Part I - "Hello, world!" In this section you will learn how to create the minimum required widget elements and add a new widget to Zabbix frontend.

Pievienojiet tukšu logrīku Zabbix lietotāja saskarnei

1. Izveidojiet direktoriju *lesson_gauge_chart* savas Zabbix lietotāja saskarnes instalācijas *modules* direktorijā (piemēram, *zabbix/ui/modules*).

> **Note:**
> Visi pielāgotie logrīki tiek uzskatīti par ārējiem moduļiem, un tie jāpievieno savas Zabbix lietotāja saskarnes instalācijas *modules* direktorijai (piemēram, *zabbix/ui/modules*). Direktorija *zabbix/ui/widgets* ir rezervēta Zabbix iebūvētajiem logrīkiem un tiek atjaunināta kopā ar Zabbix UI.

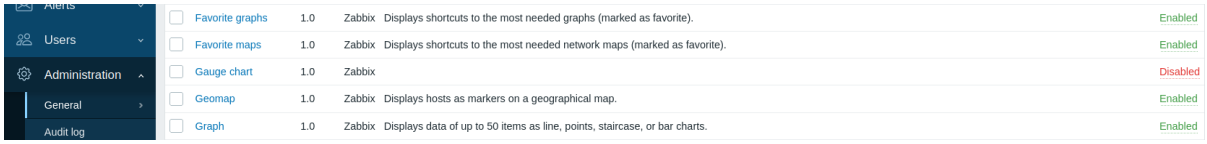
2. Izveidojiet failu *manifest.json* ar logrīka pamata metadatiem (skatiet atbalstīto parametru aprakstu).

ui/modules/lesson_gauge_chart/manifest.json

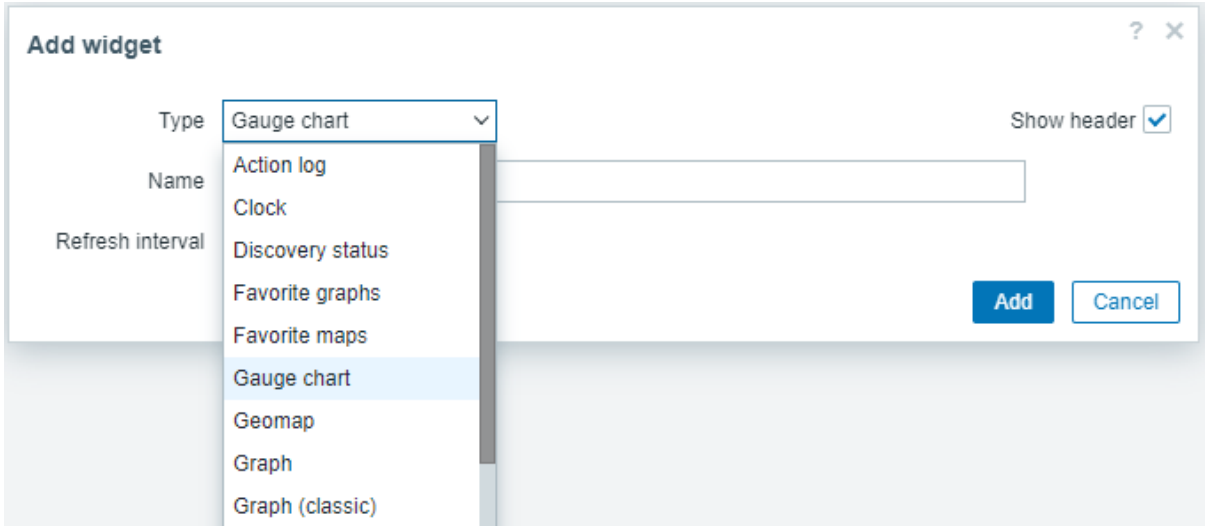
```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.1",
    "author": "Zabbix"
}
```

3. Zabbix lietotāja saskarnē atveriet sadaļu *Administration* → *General* → *Modules* un noklikšķiniet uz pogas *Scan directory*.

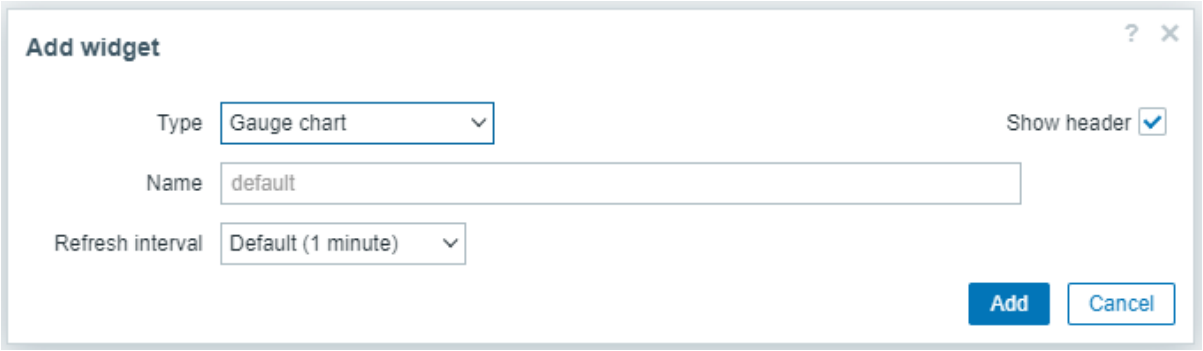
4. Sarakstā atrodiet jauno moduli *Gauge chart* un noklikšķiniet uz hipersaites "Disabled", lai mainītu moduļa statusu no "Disabled" uz "Enabled" (ja modulis nav norādīts sarakstā, skatiet sadaļu troubleshooting).



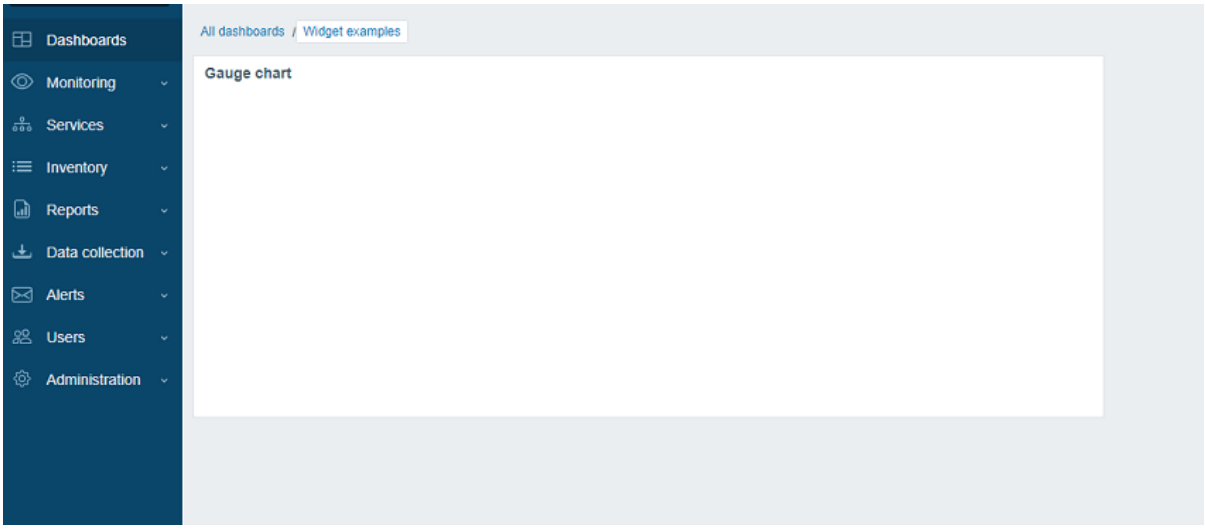
5. Atveriet informācijas paneli, pārslēdziet to rediģēšanas režīmā un pievienojiet jaunu logrīku. Laukā *Type* atlasiet "Gauge chart".



6. Šajā brīdī logrīka *Gauge chart* konfigurācijā ir tikai kopīgie logrīka lauki *Name* un *Refresh interval*. Noklikšķiniet uz *Add*, lai pievienotu logrīku informācijas panelim.



7. Informācijas panelī jāparādās tukšam logrīkam. Lai saglabātu informācijas paneli, augšējā labajā stūrī noklikšķiniet uz *Save changes*.



Pievienojiet logrīka skatu

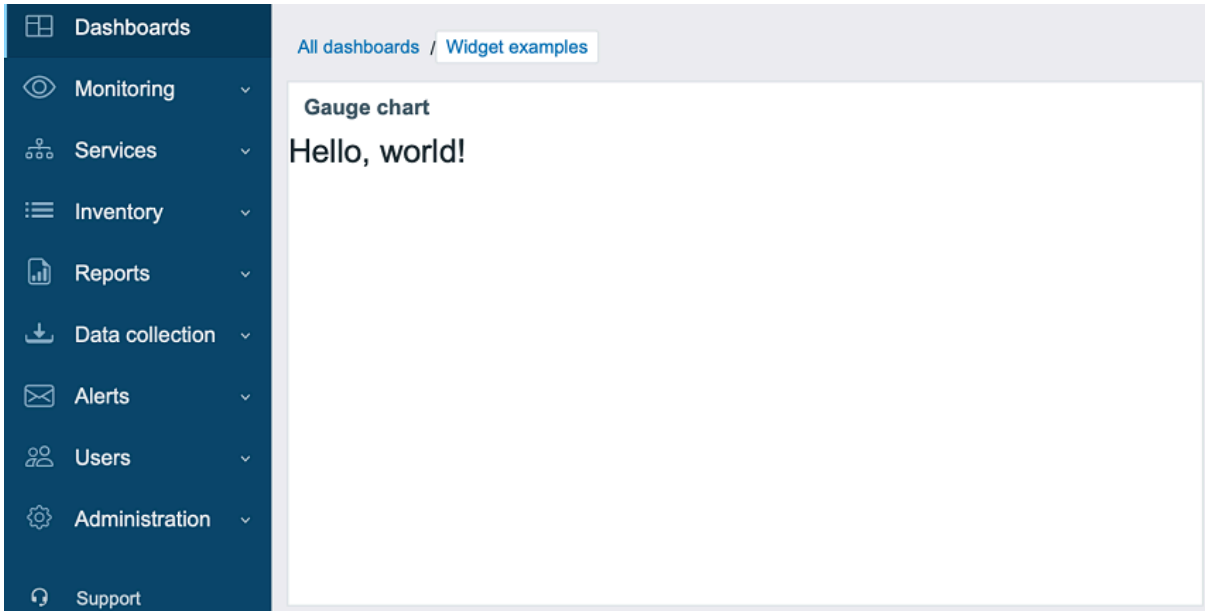
Note:
Logrīka **skata** failam jāatrodas direktorijā *views* (šajā pamācībā — *ui/modules/lesson_gauge_chart/views/*). Ja failam ir noklusējuma nosaukums *widget.view.php*, tas nav jāreģistrē failā *manifest.json*. Ja failam ir cits nosaukums, norādiet to faila manifest.json sadaļā *actions/widget.lesson_gauge_chart.view*.

1. Izveidojiet direktoriju *views* direktorijā *lesson_gauge_chart*.
2. Izveidojiet failu *widget.view.php* direktorijā *views*.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart logrīka skats.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, 'Hello, world!')
    )
    ->show();
```

3. Atsvaidziniet paneli. Logrīks *Gauge chart* tagad attēlo "Hello, world!".



Part II - Gauge chart Pievienojiet iestatījumus konfigurācijas skatam un izmantojiet tos logrīka skatā

Šajā sadaļā jūs uzzināsiet, kā pievienot logrīka konfigurācijas lauku un parādīt ievadīto vērtību logrīka skatā kā tekstu.

Logrīka konfigurācija sastāv no formas (*Zabbix\Widgets\CWidgetForm*) un logrīka formas skata (*widget.edit.php*). Lai pievienotu laukus (*Zabbix\Widgets\CWidgetField*), jums jāizveido *WidgetForm* klase, kas paplašinās *Zabbix\Widgets\CWidgetForm*.

Forma satur dažādu tipu lauku kopu (*Zabbix\Widgets\CWidgetField*), kas tiek izmantoti lietotāja ievadīto vērtību validēšanai. Formas lauks (*Zabbix\Widgets\CWidgetField*) katram ievades elementa tipam pārveido vērtību vienotā formātā, lai to saglabātu datubāzē.

> **Note:**
> Logrīka **formas** failam jāatrodas direktorijā *includes* (šajā pamācībā — *ui/modules/lesson_gauge_chart/includes/*). Ja failam ir noklusējuma nosaukums *WidgetForm.php*, tas nav jāreģistrē failā *manifest.json*. Ja failam ir cits nosaukums, norādiet to faila manifest.json sadaļā *widget/form_class*.

1. Izveidojiet jaunu direktoriju *includes* direktorijā *lesson_gauge_chart*.
2. Izveidojiet failu *WidgetForm.php* direktorijā *includes*.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\CWidgetForm;

class WidgetForm extends CWidgetForm {
}
```

3. Pievienojiet logrīka konfigurācijas formai lauku *Description*. Tas ir parasts teksta lauks, kurā lietotājs var ievadīt jebkuru rakstzīmju kopu. Tam var izmantot klasi *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'))
            );
    }
}
```

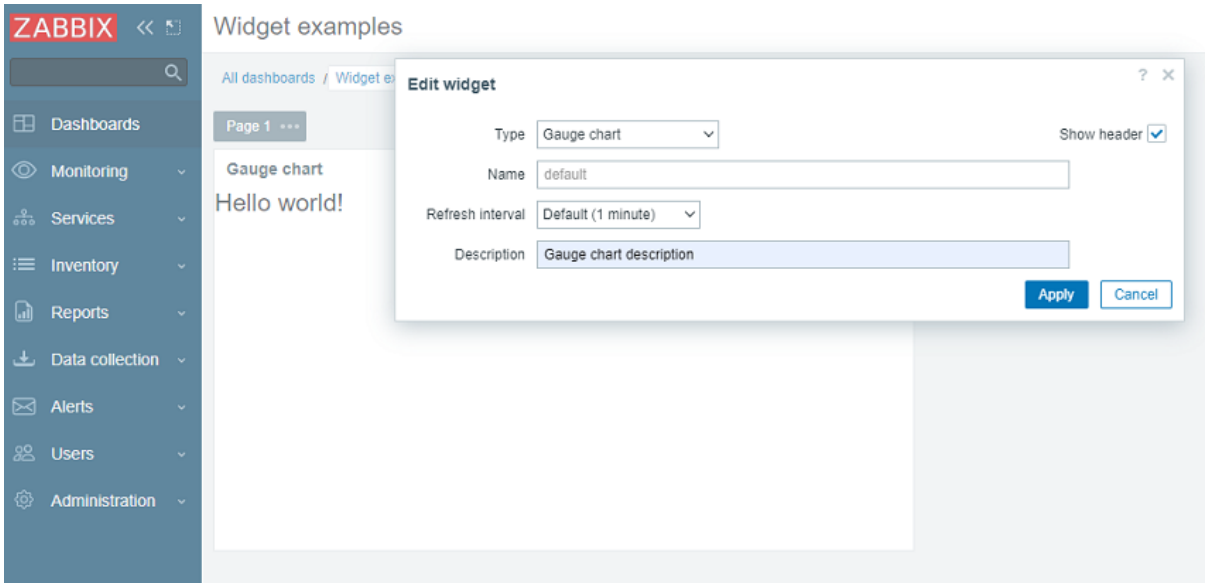
4. Direktorijā *views* izveidojiet logrīka konfigurācijas skata failu *widget.edit.php* un pievienojiet skatu jaunajam laukam *Description*. Lauka klasei *CWidgetFieldTextBox* skats ir *CWidgetFieldTextBoxView*.

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. Dodieties uz paneli un noklikšķiniet uz zobrata ikonas logrīkā, lai atvērtu logrīka konfigurācijas formu.

6. Logrīka konfigurācijas forma tagad satur jaunu teksta lauku *Description*. Ievadiet jebkuru vērtību, piemēram, *Gauge chart description*.



7. Logrīka konfigurācijas formā noklikšķiniet uz *Apply*. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Save changes*, lai saglabātu paneli. Ņemiet vērā, ka jaunais apraksts nekur nav redzams un logrīks joprojām attēlo "Hello, world!".

Lai jaunais apraksts tiktu parādīts logrīkā, lauka *Description* vērtība ir jāiegūst no datubāzes un jānodod logrīka skatam. Lai to izdarītu, jums jāizveido darbības klase.

8. Izveidojiet jaunu direktoriju *actions* direktorijā *lesson_gauge_chart*.

9. Izveidojiet failu *WidgetView.php* direktorijā *actions*. Darbības klase *WidgetView* paplašinās klasi *CControllerDashboardWidgetView*.

Logrīka konfigurācijas lauku vērtības tiek glabātas darbības klases īpašībā **$fields_values**.

ui/modules/lesson_gauge_chart/actions/WidgetView.php

```
<?php
```

```
namespace Modules\LessonGaugeChart\Actions;

use CControllerDashboardWidgetView,
    CControllerResponseData;

class WidgetView extends CControllerDashboardWidgetView {

    protected function doAction(): void {
        $this->setResponse(new CControllerResponseData([
            'name' => $this->getInput('name', $this->widget->getName()),
            'description' => $this->fields_values['description'],
            'user' => [
                'debug_mode' => $this->getDebugMode()
            ]
        ]));
    }
}
```

10. Atveriet *manifest.json* un reģistrējiet *WidgetView* kā darbības klasi sadaļā *actions/widget.lesson_gauge_chart.view*.

ui/modules/lesson_gauge_chart/manifest.json

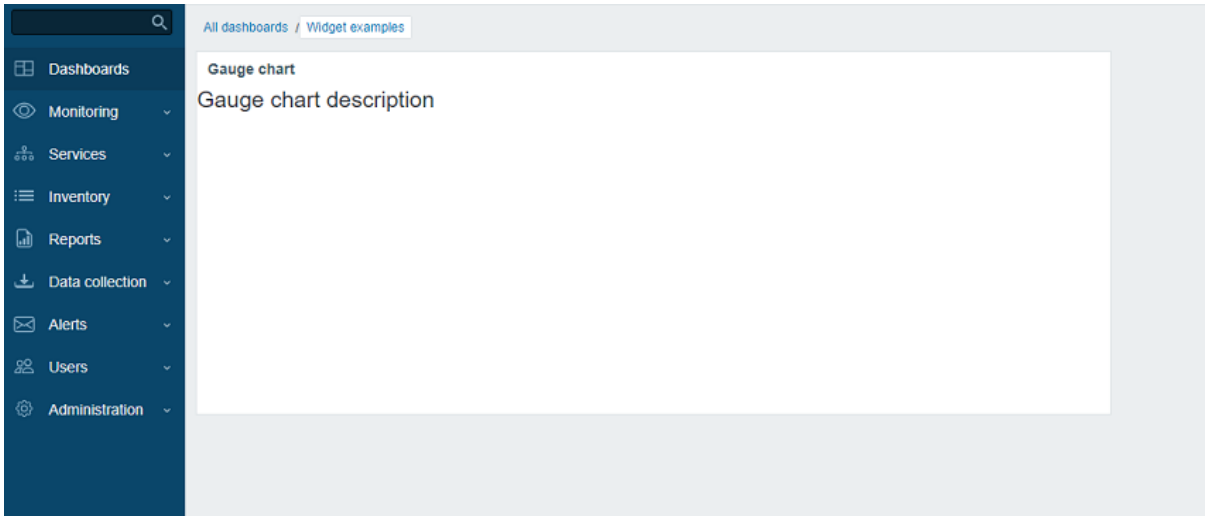
```
{
    "manifest_version": 2.0,
    "id": "lesson_gauge_chart",
    "type": "widget",
    "name": "Gauge chart",
    "namespace": "LessonGaugeChart",
    "version": "1.0",
    "author": "Zabbix",
    "actions": {
        "widget.lesson_gauge_chart.view": {
            "class": "WidgetView"
        }
    }
}
```

11. Tagad logrīka skatā varat izmantot apraksta lauka vērtību, kas atrodas *$data['description']*. Atveriet *views/widget.view.php* un aizstājiet statisko tekstu "Hello, world!" ar *$data['description']*.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart widget view.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem(
        new CTag('h1', true, $data['description'])
    )
    ->show();
```

12. Atsvaidziniet paneļa lapu. Tagad jums vajadzētu redzēt logrīka apraksta tekstu "Hello, world!" vietā.



Iegūt vienuma vērtību, izmantojot API

Logrīkam jāparāda lietotāja izvēlēta vienuma pēdējā vērtība. Lai to izdarītu, logrīka konfigurācijai jāpievieno iespēja atlasīt vienumus.

Šajā sadaļā jūs uzzināsiet, kā logrīka formai pievienot vienuma atlases lauku un kā pievienot šī lauka vizuālo daļu konfigurācijas skatā. Pēc tam logrīka kontrolieris varēs iegūt vienuma datus un tā vērtību, izmantojot API pieprasījumu. Pēc saņemšanas vērtību varēs attēlot logrīka skatā.

1. Atveriet *includes/WidgetForm.php* un pievienojiet lauku *CWidgetFieldMultiSelectItem*. Tas ļaus konfigurācijas formā atlasīt vienumu.

ui/modules/lesson_gauge_chart/includes/WidgetForm.php

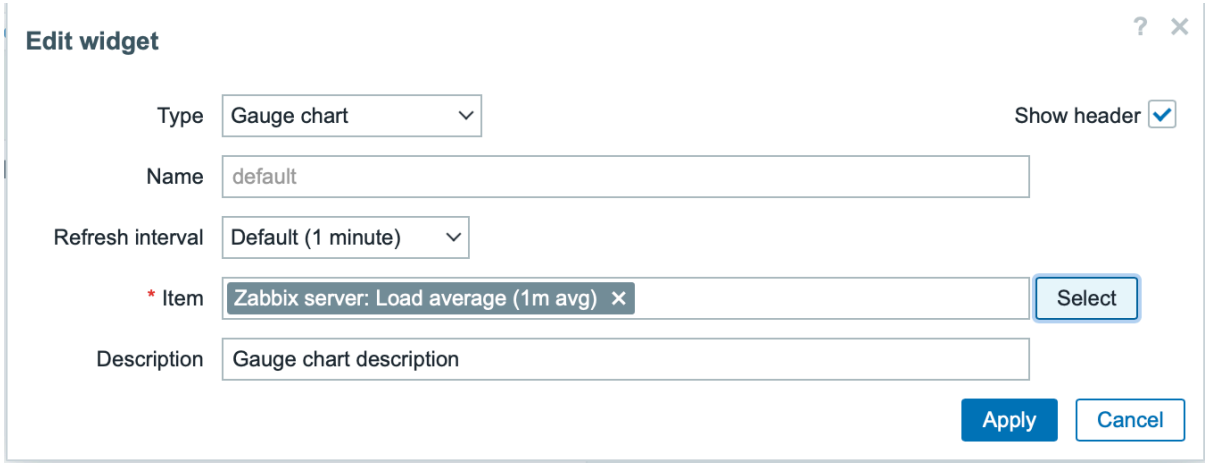
```
<?php

namespace Modules\LessonGaugeChart\Includes;

use Zabbix\Widgets\{
    CWidgetField,
    CWidgetForm
};

use Zabbix\Widgets\Fields\{
    CWidgetFieldMultiSelectItem,
    CWidgetFieldTextBox
};

/**
 * Mērinstrumenta diagrammas logrīka forma.
 */
class WidgetForm extends CWidgetForm {

    public function addFields(): self {
        return $this
            ->addField(
                (new CWidgetFieldMultiSelectItem('itemid', _('Vienums')))
                    ->setFlags(CWidgetField::FLAG_NOT_EMPTY | CWidgetField::FLAG_LABEL_ASTERISK)
                    ->setMultiple(false)
            )
            ->addField(
                new CWidgetFieldTextBox('description', _('Apraksts'))
            );
    }
}
```

2. Atveriet *views/widget.edit.php* un pievienojiet lauka vizuālo komponenti konfigurācijas skatam.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```php
<?php

/**
 * Mērinstrumenta diagrammas logrīka formas skats.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetFormView($data))
    ->addField(
        new CWidgetFieldMultiSelectItemView($data['fields']['itemid'])
    )
    ->addField(
        new CWidgetFieldTextBoxView($data['fields']['description'])
    )
    ->show();
```

3. Atgriezieties panelī un noklikšķiniet uz zobrata ikonas logrīkā, lai atvērtu logrīka konfigurācijas formu.
4. Logrīka konfigurācijas forma tagad satur jaunu ievades lauku *Vienums*. Atlasiet hostu "Zabbix server" un vienumu "Load average (1m avg)".



5. Logrīka konfigurācijas formā noklikšķiniet uz *Piemērot*. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Saglabāt izmaiņas*, lai saglabātu paneli.
6. Atveriet un modificējiet *actions/WidgetView.php*.

Turpmāk vienuma ID būs pieejams logrīka kontrolierī *$this->fields_values['itemid']*. Kontroliera metode *doAction()* apkopo vienuma datus (nosaukumu, vērtības tipu, mērvienības), izmantojot API metodi *item.get*, un vienuma pēdējo vērtību, izmantojot API metodi *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]
```

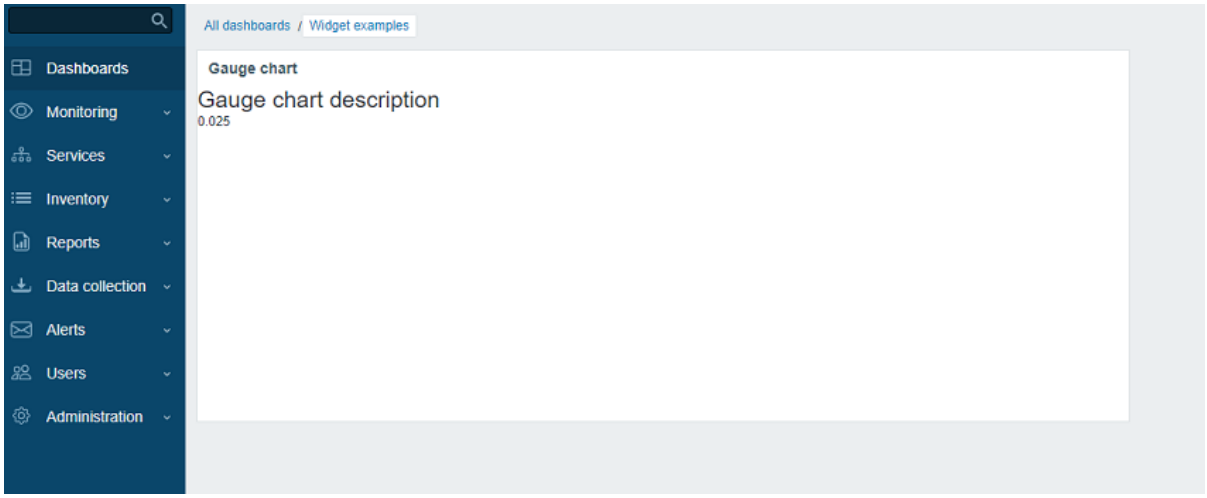
```
            ]
        ]);

        $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. Atveriet *views/widget.view.php* un pievienojiet vienuma vērtību logrīka skatam.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Mērinstrumenta diagrammas logrīka skats.
 *
 * @var CView $this
 * @var array $data
 */

(new CWidgetView($data))
    ->addItem([
        new CTag('h1', true, $data['description']),
        new CDiv($data['value'] !== null ? $data['value']['value'] : _('Nav datu'))
    ])
    ->show();
```

8. Atsvaidziniet paneļa lapu. Logrīks attēlos jaunāko vienuma vērtību.



Pievienojiet papildu konfigurācijas iestatījumus konfigurācijas skatam

Šajā sadaļā jūs uzzināsiet, kā pievienot izvēršamu/sakļaujamu sadaļu *Advanced configuration* ar neobligātiem parametriem, piemēram, krāsu, minimālo un maksimālo vērtību, mērvienībām un iepriekš izveidoto lauku *Description*.

1. Izveidojiet failu *Widget.php* galvenajā logrīka direktorijā *lesson_gauge_chart*, lai izveidotu jaunu klasi *Widget*.

Klase *Widget* paplašinās bāzes klasi *CWidget*, lai pievienotu/aizstātu noklusējuma logrīka iestatījumus (šajā gadījumā — tulkojumus). Tālāk norādītais JavaScript parāda virkni "No data", ja dati nav pieejami. Virkne "No data" ir iekļauta Zabbix lietotāja saskarnes tulkojumu failos.

Ja ir kādas logrīka konstantes, ieteicams tās norādīt arī klasē *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. Atveriet *includes/WidgetForm.php* un pievienojiet jaunus laukus *Color* (krāsu atlasītājs), *Min* (skaitlisks lauks), *Max* (skaitlisks lauks) un *Units* (atlase), kā arī definējiet noklusējuma krāsu paleti krāsu atlasītājam, lai to varētu izmantot nākamajās darbībās.

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
};

/**
 * Mērinstrumenta diagrammas logrīka forma.
 */
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. Atveriet *views/widget.edit.php* un pievienojiet lauku vizuālos komponentus konfigurācijas skatam.

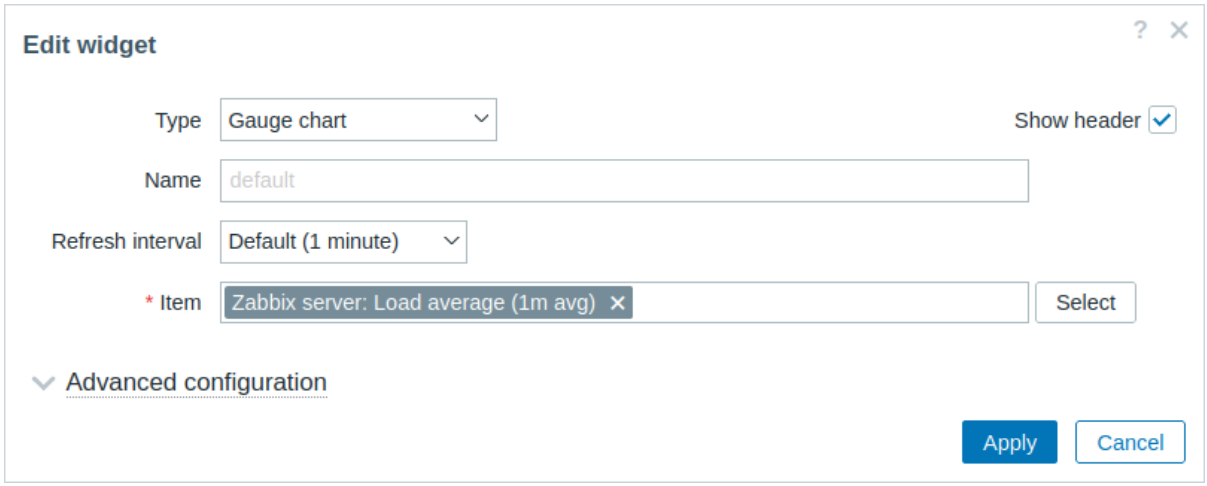
ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Mērinstrumenta diagrammas logrīka formas skats.
 *
 * @var CView $this
```

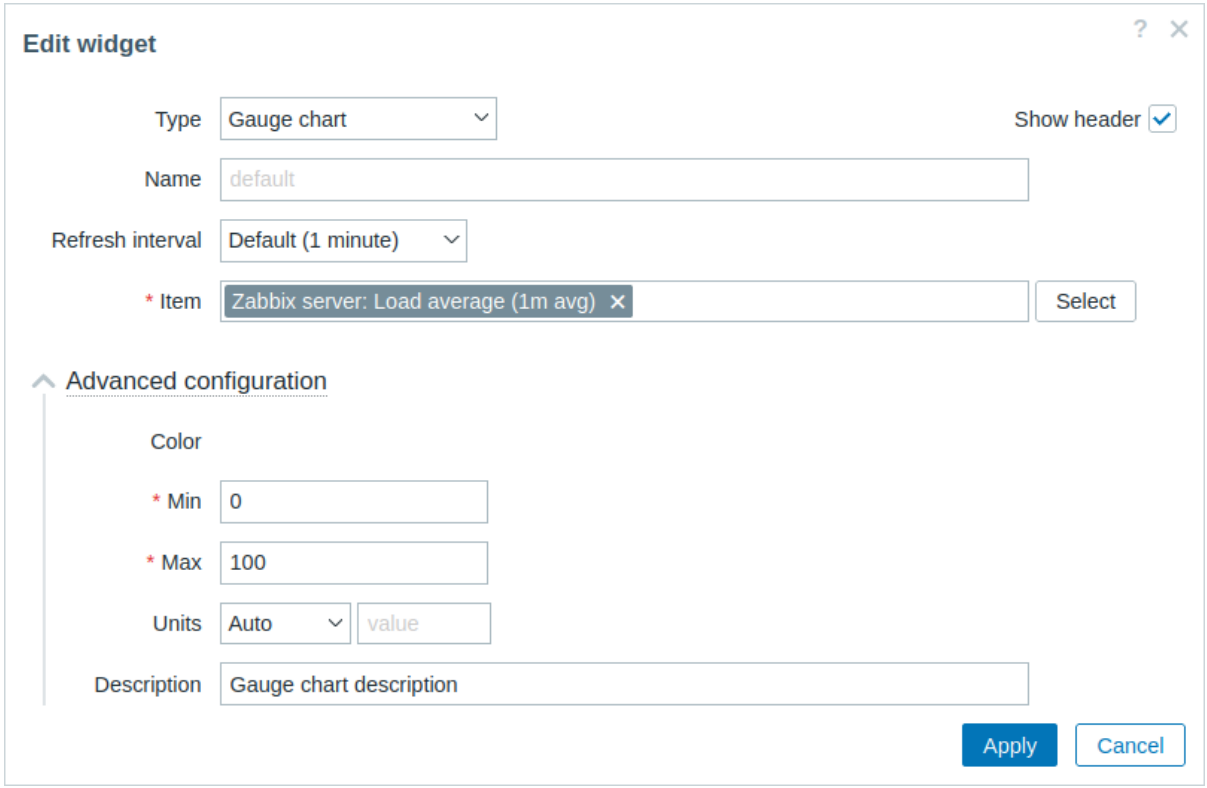
```
 * @var array $data
 */

$lefty_units = new CWidgetFieldSelectView($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:
Klases *CWidgetFormView* metode *addField()* kā otro parametru pieņem CSS klases virkni.

4. Atgriezieties panelī, pārslēdzieties uz rediģēšanas režīmu un noklikšķiniet uz zobrata ikonas logrīkā, lai atvērtu logrīka konfigurācijas formu. Logrīka konfigurācijas forma tagad satur jaunu izvēršamu/sakļaujamu sadaļu *Advanced configuration*.



5. Izvērsiet sadaļu *Advanced configuration*, lai redzētu papildu logrīka konfigurācijas laukus. Ņemiet vērā, ka laukam *Color* vēl nav krāsu atlasītāja. Tas ir tāpēc, ka krāsu atlasītājs ir jāinicializē ar JavaScript, kas tiks pievienots nākamajā sadaļā — *Add JavaScript to the widget*.



JavaScript pievienošana logrīkam

Šajā sadaļā jūs uzzināsiet, kā pievienot mērinstrumenta diagrammu (gauge chart), kas izveidota, izmantojot JavaScript, un parāda, vai jaunākā vērtība ir normāla vai pārāk augsta/zema.

1. Izveidojiet failu *widget.edit.js.php* direktorijā *views*.

JavaScript būs atbildīgs par krāsu izvēlētāja inicializēšanu konfigurācijas skatā.

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();
    }
```

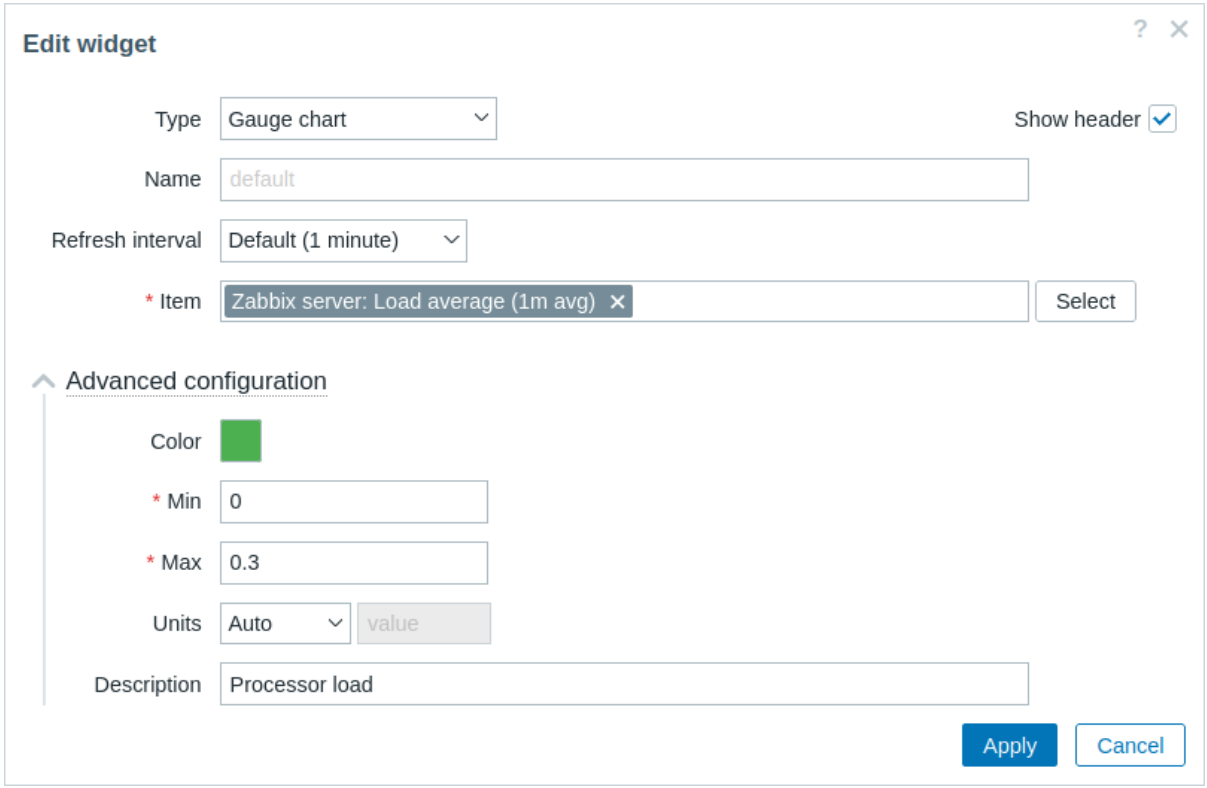
```
    updateForm() {
        this._unit_value.disabled = this._unit_select.value == <?= Widget::UNIT_AUTO ?>;
    }
};
```

2. Atveriet *views/widget.edit.php* un pievienojiet failu *widget.edit.js.php* ar JavaScript konfigurācijas skatam. Lai to izdarītu, izmantojiet metodi *includeJsFile()*. Lai pievienotu iekļauto JavaScript, izmantojiet metodi *addJavaScript()*.

ui/modules/lesson_gauge_chart/views/widget.edit.php

```
<?php

/**
 * Gauge chart logrīka formas skats.
 *
 * @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. Atgriezieties panelī, noklikšķiniet uz zobrata ikonas logrīkā, lai atvērtu logrīka konfigurācijas formu. Tagad izvērsiet sadaļu *Advanced configuration*, lai redzētu inicializēto krāsu izvēlētāju. Aizpildiet laukus ar vērtībām un izvēlieties krāsu mērinstrumenta diagrammai.



4. Logrīka konfigurācijas formā noklikšķiniet uz *Apply*. Pēc tam augšējā labajā stūrī noklikšķiniet uz *Save changes*, lai saglabātu paneli.

5. Atveriet *actions/WidgetView.php* un atjauniniet kontrolieri.

Īpašība **$this->fields_values** tagad satur visu *Advanced configuration* lauku vērtības. Pabeidziet kontrolieri, lai iespējotu konfigurācijas un atlasītā vienuma vērtības nodošanu logrīka skatam.

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. Atveriet un modificējiet *views/widget.view.php*.

Jums jāizveido konteiners mērinstrumenta diagrammai, kuru zīmēsiet nākamajos soļos, un konteiners aprakstam.

Lai nodotu vērtības JavaScript kā JSON objektu, izmantojiet metodi *setVar()*.

ui/modules/lesson_gauge_chart/views/widget.view.php

```
<?php

/**
 * Gauge chart logrīka skats.
 *
 * @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. Izveidojiet jaunu direktoriju *assets* direktorijā *lesson_gauge_chart*. Šī direktorija tiks izmantota JavaScript, CSS un, iespējams, citu resursu, piemēram, fontu vai attēlu, glabāšanai.
8. Logrīka skata JavaScript vajadzībām izveidojiet direktoriju *js* direktorijā *assets*.
9. Izveidojiet failu *class.widget.js* direktorijā *assets/js*.

Šī JavaScript logrīka klase paplašinās visu paneļa logrīku pamata JavaScript klasi — *CWidget*.

Panelis paļaujas uz korektu logrīka implementāciju un nodod visu attiecīgo informāciju logrīkam, izsaucot atbilstošās JavaScript metodes. Panelis arī sagaida, ka logrīks ģenerēs notikumus, kad notiks kāda mijiedarbība. Tādēļ klase *CWidget* satur metožu kopu ar noklusēto logrīka darbības implementāciju, kuru var pielāgot, paplašinot šo klasi.

Šajā gadījumā ir nepieciešama zināma pielāgošana, tāpēc tiks ieviesta pielāgota loģika šādai logrīka darbībai:

- logrīka inicializācija, kas ir atbildīga par logrīka sākotnējā stāvokļa noteikšanu (skatiet metodi *onInitialize()*);

- logrīka satura attēlošana (tas ir, mērinstrumenta diagrammas zīmēšana), ja logrīka atjaunināšanas process ir bijis veiksmīgs un bez kļūdām (skatiet metodi *processUpdateResponse(response)* un saistītās metodes *_resizeChart()* un *_updatedChart()*)
- logrīka izmēra maiņa (skatiet metodi *onResize()* un saistīto metodi *_resizeChart()*)

Citos mērinstrumenta diagrammas logrīka aspektos tiks izmantota noklusētā logrīka darbības implementācija. Lai uzzinātu vairāk par klases *CWidget* JavaScript metodēm, skatiet: JavaScript.

Tā kā šis JavaScript ir nepieciešams logrīka skatam, tas jāielādē kopā ar paneļa lapu. Lai iespējotu JavaScript ielādi, jums būs jāatjaunina parametri *assets/js* un *js_class* failā ***manifest.json***, kā parādīts 10. solī.

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. Atveriet *manifest.json* un pievienojiet:

- faila nosaukumu (*class.widget.js*) masīvam sadaļā *assets/js*;
- klases nosaukumu (*WidgetLessonGaugeChart*) parametram *js_class* sadaļā *widget*.

Klase *WidgetLessonGaugeChart* tagad tiks automātiski ielādēta kopā ar paneli.

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"]
    }
}
```

Pievienojiet logrīkam CSS stilus

Šajā sadaļā jūs uzzināsiet, kā pievienot pielāgotus CSS stilus, lai logrīks izskatītos pievilcīgāk.

1. Logrīka stiliem izveidojiet jaunu direktoriju *css* direktorijā *assets*.
2. Izveidojiet failu *widget.css* direktorijā *assets/css*. Lai stilizētu logrīka elementus, izmantojiet selektoru *div.dashboard-widget-{widget id}*. Lai konfigurētu CSS visam logrīkam, izmantojiet selektoru *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;
}
```

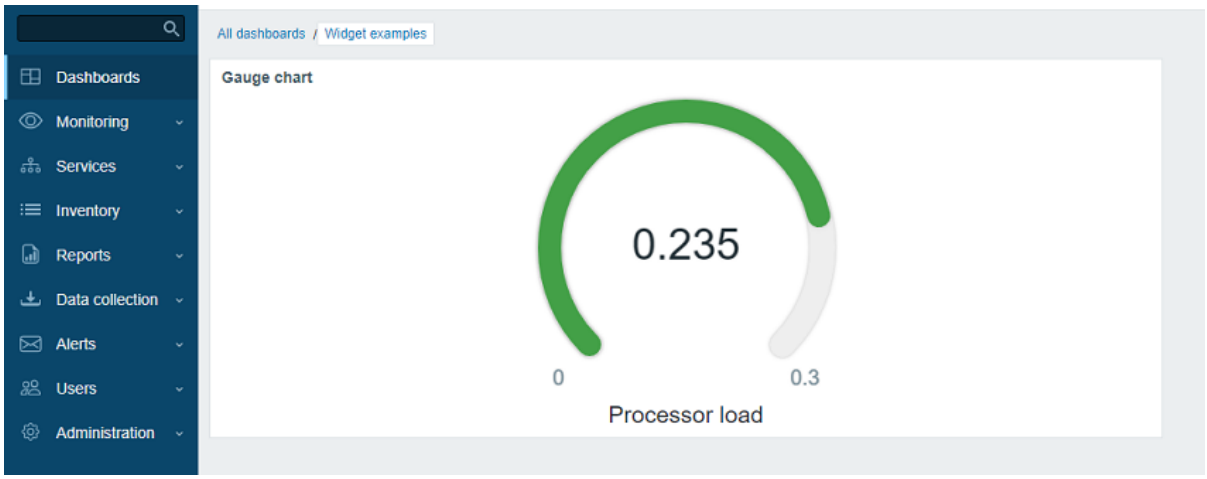
3. Atveriet *manifest.json* un pievienojiet CSS faila nosaukumu (*widget.css*) masīvam sadaļā *assets/css*. Tas ļaus failā *widget.css* definētajiem CSS stiliem ielādēties kopā ar paneļa lapu.

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. Atsvaidziniet paneļa lapu, lai redzētu pabeigto logrīka versiju.



Piemēri

Šajā sadaļā ir pieejami paraugmoduļu un logrīku faili, kurus varat izmantot kā pamatu saviem pielāgotajiem moduļiem.

Lai izmantotu moduli:

1. Lejupielādējiet ZIP arhīvu.
2. Izpakojiet saturu atsevišķā direktorijā savas Zabbix lietotāja saskarne instalācijas *modules* direktorijā (piemēram, *zabbix/ui/modules*).
3. Reģistrējiet moduli Zabbix lietotāja saskarnē.

Moduļa piemērs

- Veidojot hostu grupu, piešķiriet lasīšanas atļaujas konfigurētajām lietotāju grupām - hg_auto_perm.zip

Logrīku piemēri

- Minimāls logrīks - widget_min.zip
- "Hello, world" logrīks, izmantojot tikai CSS - hello_world_css.zip
- "Hello, world" logrīks, izmantojot tikai JavaScript - hello_world_js.zip
- "Hello, world" logrīks, izmantojot PHP - hello_world_php.zip

> **Note:**
> Kā piemērus varat izmantot arī Zabbix vietējos logrīkus.

Izmaiņas paplašinājumu izstrādē

Izmaiņas no 7.4 uz 8.0 ZBXNEXT-10445 Maksimālais `string` logrīka lauku garums ir palielināts no 2048 līdz 65535 rakstzīmēm. Noteiktiem `string` laukiem Zabbix vietējos logrīkos tagad ir arī savi garuma ierobežojumi (piemēram, iestatīti ar funkciju `setMaxLength()`).

Zabbix rokasgrāmatas lapas

Atsauces dokumentācija Zabbix komandrindas rīkiem un pakalpojumiem, kurā uzskaitītas pieejamās komandas, opcijas un lietošanas piemēri.

zabbix_agent2

Sadaļa: Uzturēšanas komandas (8)
Atjaunināts: 2019-01-29
Indekss Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_agent2 — Zabbix aģents 2

KOPSAVILKUMS

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

APRAKSTS

zabbix_aģents2 ir lietojumprogramma dažādu pakalpojumu parametru monitorēšanai.

IESPĒJAS

-c, **--konfig** *konfig-fails* Izmantojiet alternatīvo *konfig-failu*, nevis noklusējuma failu.

-R, **--izpildlaika-kontrole** *izpildlaika-iespēja* Veic administratīvās funkcijas saskaņā ar *izpildlaika-iespēju*.

Izpildlaika vadības opcijas: userparameter_reload
Pārlādēt lietotāja parametrus no konfigurācijas faila

log_level_increase
Palielināt žurnāla līmeni

log_level_decrease
Samazināt žurnāla līmeni

help
Uzskaitīt pieejamās izpildlaika vadības opcijas

metrics
Uzskaitīt pieejamās metriskās vērtības

version
Parādīt versiju

-p, **--print**
Izdrukāt zināmos vienumus un iziet. Katram vienumam tiek izmantotas vai nu vispārīgās noklusējuma vērtības, vai arī testēšanai paredzētās specifiskās noklusējuma vērtības. Šīs noklusējuma vērtības ir norādītas kvadrātiekavās kā vienuma atslēgas parametri. Atgrieztās vērtības ir ietvertas kvadrātiekavās, un pirms tām ir norādīts atgrieztās vērtības tips, atdalīts ar vertikālu svītru. Lietotāja parametriem tips vienmēr ir **t**, jo aģents nevar noteikt visas iespējamās atgrieztās vērtības. Vienumi, kas tiek parādīti kā strādājoši, negarantē, ka tie darbosies no Zabbix serveris vai zabbix_get, vaicājot palaistu aģenta dēmonu, jo atļaujas vai vide var atšķirties. Atgriezto vērtību tipi ir:

d
Skaitlis ar decimāldaļu.

m
Nav atbalstīts. To var izraisīt vaicājums par vienumu, kas darbojas tikai aktīvajā režīmā, piemēram, žurnāla uzraudzības vienums, vai vienumu, kam nepieciešamas vairākas savāktās vērtības. Atļauju problēmas vai nepareizi lietotāja parametri arī var izraisīt stāvokli ”nav atbalstīts”.

s
Teksts. Maksimālais garums nav ierobežots.

t
Teksts. Tas pats, kas **s**.

u
Neparakstīts vesels skaitlis.

-t, **--test** *item-key*
Pārbaudīt vienu vienumu un iziet. Izvades aprakstu skatiet pie **--print**.

-v, **--verbose**
Iespējot detalizētu izvadi, testējot vienumu vai izdrukājot zināmos vienumus.

-T, **--test-config**
Validēt konfigurācijas failu un iziet.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

DATNES

/usr/local/etc/zabbix_agent2.conf
Zabbix aģents 2 konfigurācijas faila noklusējuma atrašanās vieta (ja tā netika mainīta kompilēšanas laikā).

SKATĪT ARĪ

Dokumentāciju https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(8), **zabbix_js**(8), **zabbix_proxy**(8), **zabbix_sender**(8), **zabbix_server**(8)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

FAILI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 14:07:57 GMT, 2021. gada 22. novembrī

zabbix_agentd

Sadaļa: Uzturēšanas komandas (8)
Atjaunināts: 2019-01-29
Index Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_agent2 — Zabbix aģents 2

KOPSAVILKUMS

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

APRAKSTS

zabbix_agentd ir dēmons dažādu servera parametru uzraudzībai.

IESPĒJAS

-c, **--config** *config-file* Izmantojiet alternatīvo *config-file*, nevis noklusējuma failu.

-f, **--priekšplāns** Palaidiet Zabbix aģentu priekšplānā.

-R, **--runtime-control** *izpildlaika opcija* Veikt administratīvās funkcijas saskaņā ar *runtime-option*.

Izpildlaika vadības opcijas userparameter_reload
Pārlādēt lietotāja parametrus no konfigurācijas faila

log_level_increase[=*target*]
Palielināt žurnālošanas līmeni; ietekmē visus procesus, ja target nav norādīts

log_level_decrease[=*target*]
Samazināt žurnālošanas līmeni; ietekmē visus procesus, ja target nav norādīts

Žurnāla līmeņa vadības mērķi *process-type*
Visi norādītā tipa procesi (aktīvās pārbaudes, kolektors, klausītājs)

process-type,N
Procesa tips un numurs (piemēram, listener,3)

pid
Procesa identifikators, līdz 65535. Lielākām vērtībām norādiet mērķi kā ”process-type,N”

-p, **--print**
Izdrukāt zināmos vienumus un iziet. Katram vienumam tiek izmantotas vai nu vispārīgās noklusējuma vērtības, vai arī testēšanai paredzētas specifiskas noklusējuma vērtības. Šīs noklusējuma vērtības ir norādītas kvadrātiekavās kā vienuma atslēgas parametri. Atgrieztās vērtības ir ietvertas kvadrātiekavās, un pirms tām ir norādīts atgrieztās vērtības tips, atdalīts ar vertikālo svītru. Lietotāja parametriem tips vienmēr ir **t**, jo aģents nevar noteikt visas iespējamās atgrieztās vērtības. Vienumi, kas tiek parādīti kā strādājoši,

netiek garantēti kā strādājoši no Zabbix serveris vai zabbix_get, vaicājot darbojošos aģenta dēmonu, jo atļaujas vai vide var atšķirties. Atgriezto vērtību tipi ir:

d
Skaitlis ar decimāldaļu.

m
Nav atbalstīts. To var izraisīt vaicājums vienumam, kas darbojas tikai aktīvajā režīmā, piemēram, žurnāla uzraudzības vienums, vai vienumam, kam nepieciešamas vairākas savāktās vērtības. Atļauju problēmas vai nepareizi lietotāja parametri arī var izraisīt stāvokli "nav atbalstīts".

s
Teksts. Maksimālais garums nav ierobežots.

t
Teksts. Tas pats, kas **s**.

u
Neparakstīts vesels skaitlis.

-t, **--test** *item-key*
Pārbaudīt vienu vienumu un iziet. Izvades aprakstu skatiet pie **--print**.

-T, **--test-config**
Pārbaudīt konfigurācijas failu un iziet.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

FAILI

/usr/local/etc/zabbix_agentd.conf
Zabbix aģenta konfigurācijas faila noklusējuma atrašanās vieta (ja tā nav mainīta kompilēšanas laikā).

SKATIET ARĪ

Dokumentācija https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_get**(1), **zabbix_js**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

FAILI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 20:50:13 GMT, 2021. gada 22. novembrī

zabbix_get

Sadaļa: Lietotāja komandas (1)
Atjaunināts: 2022-01-06
Rādītājs Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_get - Zabbix iegūšanas utilīta

KOPSAVILKUMS

zabbix_get -s *hosta-nosaukums-vai-IP* [**-p** *porta-numurs*] [**-I** *IP-adrese*] [**-t** *taimauts*] **-k** *vienuma-atslēga*
zabbix_get -s *hosta-nosaukums-vai-IP* [**-p** *porta-numurs*] [**-I** *IP-adrese*] [**-t** *taimauts*] **--tls-connect cert --tls-ca-file** *CA-fails* [**--tls-crl-file** *CRL-fails*] [**--tls-agent-cert-issuer** *sertifikāta-izdevējs*] [**--tls-agent-cert-subject** *sertifikāta-subjekts*] **--tls-cert-file** *sertifikāta-fails* **--tls-key-file** *atslēgas-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *vienuma-atslēga*
zabbix_get -s *hosta-nosaukums-vai-IP* [**-p** *porta-numurs*] [**-I** *IP-adrese*] [**-t** *taimauts*] **--tls-connect psk --tls-psk-identity** *PSK-identitāte* **--tls-psk-file** *PSK-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *vienuma-atslēga*
zabbix_get -h
zabbix_get -V

APRAKSTS

zabbix_get ir komandrindas utilīta datu iegūšanai no Zabbix aģenta.

OPCIJAS

-s, **--host** *host-name-or-IP*
Norādiet hosta nosaukumu vai hosta IP adresi.

-p, **--port** *port-number*
Norādiet aģenta porta numuru, kas darbojas hostā. Noklusējuma vērtība ir 10050.

-I, **--source-address** *IP-address*
Norādiet avota IP adresi.

-t, **--timeout** *seconds*
Norādiet noildzi. Derīgais diapazons: 1-600 sekundes (noklusējums: 30)

-k, **--key** *item-key*
Norādiet vienuma atslēgu, kuram iegūt vērtību.

-P, **--protocol** *value*
Protokols, ko izmanto saziņai ar aģentu. Vērtības:

auto izveidot savienojumu, izmantojot JSON protokolu; ja tas neizdodas, atkārtot mēģinājumu, izmantojot vienkāršā teksta protokolu (noklusējums)

json izveidot savienojumu, izmantojot JSON protokolu

plaintext izveidot savienojumu, izmantojot vienkāršā teksta protokolu, kur tiek nosūtīta tikai vienuma atslēga (6.4.x un vecākās laidienu versijas)

--tls-connect *value*
Kā izveidot savienojumu ar aģentu. Vērtības:

nešifrēts
savienoties bez šifrēšanas (noklusējums)

psk
savienoties, izmantojot TLS un iepriekš koplietotu atslēgu

cert
savienoties, izmantojot TLS un sertifikātu

--tls-ca-file *CA-file*
Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta pārbaudei.

--tls-crl-file *CRL-file*
Pilns ceļš uz failu, kas satur atsauktos sertifikātus.

--tls-agent-cert-issuer *cert-issuer*
Atļautais aģenta sertifikāta izdevējs.

--tls-agent-cert-subject *cert-subject*
Atļautais aģenta sertifikāta subjekts.

--tls-cert-file *cert-file*
Pilns ceļš uz failu, kas satur sertifikātu vai sertifikātu ķēdi.

--tls-key-file *key-file*
Pilns ceļš uz failu, kas satur privāto atslēgu.

--tls-psk-identity *PSK-identity*
PSK identitātes virkne.

--tls-psk-file *PSK-file*
Pilns ceļš uz failu, kas satur iepriekš koplietoto atslēgu.

--tls-cipher13 *cipher-string*
Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 vajadzībām. Pārraksta noklusējuma šifru kopas atlases kritērijus. Šī opcija nav pieejama, ja OpenSSL versija ir mazāka par 1.1.1.

--tls-cipher *cipher-string*
GnuTLS prioritāšu virkne (TLS 1.2 un jaunākām versijām) vai OpenSSL šifru virkne (tikai TLS 1.2). Pārraksta noklusējuma šifru kopas atlases kritērijus.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

PIEMĒRI

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 aģents" --tls-psk-file /home/zabbix/zabbix_agentd.psk

SKATĪT ARĪ

Dokumentācija 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)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

PIEMĒRI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 08:42:29 GMT, 2021. gada 11. jūnijā

zabbix_js

Sadaļa: Lietotāja komandas (1)
Atjaunināts: 2022-01-06
Saturs Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_js - Zabbix JS utilīta

KOPSAVILKUMS

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

APRAKSTS

zabbix_js ir komandrindas utilīta, ko var izmantot iegulto skriptu testēšanai.

OPCIJAS

-s, **--script** *script-file*
Norādiet izpildāmā skripta faila nosaukumu. Ja kā faila nosaukums ir norādīts '-', skripts tiks nolasīts no stdin.

-p, **--param** *input-param*
Norādiet ievades parametru.

-i, **--input** *input-file*
Norādiet ievades parametra faila nosaukumu. Ja kā faila nosaukums ir norādīts '-', ievade tiks nolasīta no stdin.

-w, **--webdriver** *url*
Norādiet webdriver URL.

-l, **--loglevel** *log-level*
Norādiet žurnālošanas līmeni.

-t, **--timeout** *timeout*
Norādiet noildzi sekundēs. Derīgais diapazons: 1-600 sekundes (noklusējums: 10)

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

PIEMĒRI

zabbix_js -s script-file.js -p example

SKATĪT ARĪ

Dokumentācija https://www.zabbix.com/manuals

zabbix_agent2(8), **zabbix_agentd**(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_server**(8)

Saturs

NAME

SYNOPSIS

DESCRIPTION

OPTIONS

EXAMPLES

SEE ALSO

Šis dokuments tika izveidots: 21:23:35 GMT, 2020. gada 18. martā

zabbix_proxy

Sadaļa: Uzturēšanas komandas (8)
Atjaunināts: 2020-09-04
Index Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_proxy - Zabbix starpniekservera dēmons

KOPSAVILKUMS

zabbix_proxy [**-c** *config-file*]
zabbix_proxy [**-c** *config-file*] **-R** *runtime-option*
zabbix_proxy [**-c** *config-file*] **-T**
zabbix_proxy -h
zabbix_proxy -V

APRAKSTS

zabbix_proxy ir dēmons, kas apkopo uzraudzības datus no ierīcēm un nosūta tos uz Zabbix serveri.

OPCIJAS

-c, **--config** *config-file*
Izmantot alternatīvo *config-file* noklusējuma faila vietā.

-f, **--foreground**
Palaist Zabbix starpniekserveri priekšplānā.

-R, **--runtime-control** *runtime-option*
Veikt administratīvās funkcijas saskaņā ar *runtime-option*.

Izpildlaika vadības opcijas config_cache_reload
Pārlādēt konfigurācijas kešatmiņu.
Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta.
Aktīvais Zabbix starpniekserveris izveidos savienojumu ar Zabbix serveris un pieprasīs konfigurācijas datus.
Lai atrastu PID failu, tiks izmantots noklusējuma konfigurācijas fails (ja vien nav norādīta opcija **-c**), un signāls tiks nosūtīts procesam, kas norādīts PID failā.

snmp_cache_reload
Pārlādēt SNMP kešatmiņu.
Ņemiet vērā, ka Zabbix kešo SNMPv3 EngineID→IP atbilstības un var automātiski atkārtoti izmantot EngineID, lai samazinātu aptaujāšanas pieskaitāmās izmaksas.

housekeeper_execute
Izpildīt housekeeper.
Tiek ignorēts, ja housekeeper pašlaik tiek izpildīts.

diaginfo[=*section*]
Reģistrēt norādītās sadaļas iekšējo diagnostikas informāciju žurnālā.
Sadaļa var būt *historycache*, *preprocessing*, *locks*.
Pēc noklusējuma žurnālā tiek reģistrēta visu sadaļu diagnostikas informācija.

log_level_increase[=*target*]
Palielināt žurnāla līmeni; ietekmē visus procesus, ja mērķis nav norādīts.

log_level_decrease[=*target*]
Samazināt žurnāla līmeni; ietekmē visus procesus, ja mērķis nav norādīts.

Žurnāla līmeņa vadības mērķi *process-type*
Visi norādītā tipa procesi (pieejamības pārvaldnieks, pārlūka aptaujātājs, konfigurācijas sinhronizētājs, datu sūtītājs, atklāšanas pārvaldnieks, vēstures sinhronizētājs, housekeeper, http aptaujātājs, icmp pinger, ipmi pārvaldnieks, ipmi aptaujātājs, java aptaujātājs, odbc aptaujātājs, aptaujātājs, aģenta aptaujātājs, http aģenta aptaujātājs, snmp aptaujātājs, priekšapstrādes pārvaldnieks, pašuzraudzība, snmp trapper, uzdevumu pārvaldnieks, trapper, nesasniedzamo hostu aptaujātājs, vmware kolektors)

process-type,N
Procesa tips un numurs (piemēram, poller,3)

pid
Procesa identifikators, līdz 65535. Lielākām vērtībām norādiet mērķi kā ”process-type,N”

Profilēšanas vadības mērķi

process-type
Visi norādītā tipa procesi (konfigurācijas sinhronizētājs, datu sūtītājs, atklāšanas pārvaldnieks, vēstures sinhronizētājs, housekeeper, http aptaujātājs, priekšapstrādes pārvaldnieks, icmp pinger, ipmi pārvaldnieks, ipmi aptaujātājs, java aptaujātājs, aptaujātājs, aģenta aptaujātājs, http aģenta aptaujātājs, snmp aptaujātājs, pašuzraudzība, snmp trapper, uzdevumu pārvaldnieks, trapper, nesasniedzamo hostu aptaujātājs, vmware kolektors, vēstures aptaujātājs, pieejamības pārvaldnieks, odbc aptaujātājs)

process-type,N
Procesa tips un numurs (piemēram, history syncer,1)

pid
Procesa identifikators, līdz 65535. Lielākām vērtībām norādiet mērķi kā ”process-type,N”

scope
Profilēšanas tvērums (rwlock, mutex, processing) var tikt izmantots kopā ar procesa tipu (piemēram, history syncer,1,processing)

-T, **--test-config**
Pārbaudīt konfigurācijas failu un iziet.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

FAILI

/usr/local/etc/zabbix_proxy.conf
Zabbix starpniekservera konfigurācijas faila noklusējuma atrašanās vieta (ja tā nav mainīta kompilēšanas laikā).

SKATIET ARĪ

Dokumentācija https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_sender**(1), **zabbix_server**(8), **zabbix_js**(1), **zabbix_agent2**(8)

Indekss

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

FAILI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 16:12:22 GMT, 2020. gada 04. septembrī

zabbix_sender

Sadaļa: Lietotāja komandas (1)
Atjaunināts: 2021-06-01
Index Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_sender - Zabbix nosūtīšanas utilīta

KOPSAVILKUMS

zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] **-s** *hosts* **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] **-s** *hosts* **--tls-connect cert** **--tls-ca-file** *CA-fails* [**--tls-crl-file** *CRL-fails*] [**--tls-server-cert-issuer** *sertifikāta-izdevējs*] [**--tls-server-cert-subject** *sertifikāta-subjekts*] **--tls-cert-file** *sertifikāta-fails* **--tls-key-file** *atslēgas-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect cert** **--tls-ca-file** *CA-fails* [**--tls-crl-file** *CRL-fails*] [**--tls-server-cert-issuer** *sertifikāta-izdevējs*] [**--tls-server-cert-subject** *sertifikāta-subjekts*] **--tls-cert-file** *sertifikāta-fails* **--tls-key-file** *atslēgas-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect cert** **--tls-ca-file** *CA-fails* [**--tls-crl-file** *CRL-fails*] [**--tls-server-cert-issuer** *sertifikāta-izdevējs*] [**--tls-server-cert-subject** *sertifikāta-subjekts*] **--tls-cert-file** *sertifikāta-fails* **--tls-key-file** *atslēgas-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect cert** **--tls-ca-file** *CA-fails* [**--tls-crl-file** *CRL-fails*] [**--tls-server-cert-issuer** *sertifikāta-izdevējs*] [**--tls-server-cert-subject** *sertifikāta-subjekts*] **--tls-cert-file** *sertifikāta-fails* **--tls-key-file** *atslēgas-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] **-s** *hosts* **--tls-connect psk** **--tls-psk-identity** *PSK-identitāte* **--tls-psk-file** *PSK-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-z** *serveris* [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect psk** **--tls-psk-identity** *PSK-identitāte* **--tls-psk-file** *PSK-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect psk** **--tls-psk-identity** *PSK-identitāte* **--tls-psk-file** *PSK-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] **-k** *atslēga* **-o** *vērtība*
zabbix_sender [**-v**] **-c** *config-file* [**-z** *serveris*] [**-p** *ports*] [**-I** *IP-adrese*] [**-t** *taimauts*] [**-s** *hosts*] **--tls-connect psk** **--tls-psk-identity** *PSK-identitāte* **--tls-psk-file** *PSK-fails* [**--tls-cipher13** *šifra-virkne*] [**--tls-cipher** *šifra-virkne*] [**-T**] [**-N**] [**-r**] [**-g**] **-i** *ievades-fails*
zabbix_sender -h
zabbix_sender -V

APRAKSTS

zabbix_sender ir komandrindas utilīta monitoringa datu sūtīšanai uz Zabbix serveri vai starpniekserveri. Zabbix serverī jāizveido vienums ar tipu **Zabbix trapper** un atbilstošu atslēgu. Ņemiet vērā, ka ienākošās vērtības tiks pieņemtas tikai no hostiem, kas norādīti šī vienuma laukā **Allowed hosts**.

OPCIJAS

-c, **--config** *config-file*
Izmantot *config-file*. **Zabbix sender** nolasa servera informāciju no agentd konfigurācijas faila. Pēc noklusējuma **Zabbix sender** nenolasa nevienu konfigurācijas failu. Tiek atbalstīti tikai parametri **Hostname**, **ServerActive**, **SourceIP**, **TLSConnect**, **TLSCAFile**, **TLSCRLFile**, **TLSServerCertIssuer**, **TLSServerCertSubject**, **TLSCertFile**, **TLSKeyFile**, **TLSPSKIdentity** un **TLSPSKFile**. Hosts, kas definēts ar parametru **HostnameItem**, netiks izmantots; šajā gadījumā hosta nosaukums jānorāda komandrindā (skatiet opciju -s). Datu sūtīšanai tiek izmantotas visas adreses, kas definētas aģenta konfigurācijas parametrā **ServerActive**. Ja pakešdatu sūtīšana uz vienu adresi neizdodas, nākamās paketes uz šo adresi netiek sūtītas.

-z, **--zabbix-server** *server*
Zabbix servera hosta nosaukums vai IP adrese. Ja hosts tiek uzraudzīts ar starpniekserveri, tā vietā jāizmanto starpniekservera hosta nosaukums vai IP adrese. Lietojot kopā ar **--config**, pārraksta aģenta konfigurācijas failā norādītās parametra **ServerActive** vērtības.

-p, **--port** *port*
Norādiet serverī darbojošā Zabbix servera trapper porta numuru. Noklusējuma vērtība ir 10051. Lietojot kopā ar **--config**, pārraksta aģenta konfigurācijas failā norādītās parametra **ServerActive** porta vērtības.

-I, **--source-address** *IP-address*
Norādiet avota IP adresi. Lietojot kopā ar **--config**, pārraksta aģenta konfigurācijas failā norādīto parametru **SourceIP**.

-t, **--timeout** *seconds*
Norādiet noildzi. Derīgais diapazons: 1-300 sekundes (noklusējums: 60)

-s, **--host** *host*
Norādiet hosta nosaukumu, kuram pieder vienums (kā reģistrēts Zabbix lietotāja saskarnē). Hosta IP adrese un DNS nosaukums nedarbosies. Lietojot kopā ar **--config**, pārraksta aģenta konfigurācijas failā norādīto parametru **Hostname**.

-k, **--key** *key*
Norādiet vienuma atslēgu, uz kuru nosūtīt vērtību.

-o, **--value** *value*
Norādiet vienuma vērtību.

-i, **--input-file** *input-file*
Ielādēt vērtības no ievades faila. Norādiet **-** kā **<input-file>**, lai nolasītu vērtības no standarta ievades. Katrā faila rindā ir ar atstarpēm atdalīti lauki: **<hostname> <key> <value>**. Katra vērtība jānorāda atsevišķā rindā. Katrā rindā jābūt 3 ar atstarpēm atdalītiem ierakstiem: **<hostname> <key> <value>**, kur ”hostname” ir uzraudzītā hosta nosaukums, kā reģistrēts Zabbix lietotāja saskarnē, ”key” ir mērķa vienuma atslēga un ”value” — nosūtāmā vērtība. Norādiet **-** kā **<hostname>**, lai izmantotu hosta nosaukumu no aģenta konfigurācijas faila vai no argumenta **--host**.

Ievades faila rindas piemērs:

”Linux DB3” db.connections 43

Vērtības tipam jābūt pareizi iestatītam vienuma konfigurācijā Zabbix lietotāja saskarnē. Zabbix sender vienā savienojumā nosūtīs līdz 250 vērtībām. Izmēra ierobežojums vērtību sūtīšanai no ievades faila ir atkarīgs no izmēra, kas aprakstīts Zabbix komunikācijas protokolā. Ievades faila saturam jābūt UTF-8 kodējumā. Visas vērtības no ievades faila tiek sūtītas secīgi no augšas uz leju. Ierakstiem jābūt formatētiem saskaņā ar šādiem noteikumiem:

- Tiek atbalstīti ieraksti pēdiņās un bez pēdiņām.
- Dubultpēdiņa ir citēšanas rakstzīme.
- Ieraksti ar atstarpēm jāliek pēdiņās.
- Dubultpēdiņas un atpakaļslīpsvītras rakstzīmes citētā ierakstā jāatceļ ar atpakaļslīpsvītru.
- Atcelšana netiek atbalstīta ierakstos bez pēdiņām.
- Rindpārejas atcelšanas secības (\n) tiek atbalstītas citētās virknēs.
- Rindpārejas atcelšanas secības tiek noņemtas no ieraksta beigām.

-T, **--with-timestamps**
Šo opciju var izmantot tikai kopā ar opciju **--input-file**.

Katrā ievades faila rindā jābūt 4 ar atstarpēm atdalītiem ierakstiem: **<hostname> <key> <timestamp> <value>**. Laika zīmogs jānorāda Unix timestamp formātā. Ja mērķa vienumam ir trigeri, kas uz to atsaucas, visiem laika zīmogiem jābūt augošā secībā, pretējā gadījumā notikumu aprēķins nebūs pareizs.

Ievades faila rindas piemērs:

”Linux DB3” db.connections 1429533600 43

Papildinformāciju skatiet opcijā **--input-file**.

Ja hostam, kas atrodas uzturēšanas režīmā ar tipu “nav datu”, tiek nosūtīta vērtība ar laika zīmogu, šī vērtība tiks atmesta; tomēr ir iespējams nosūtīt vērtību ar laika zīmogu par uzturēšanas periodu, kam beidzies termiņš, un tā tiks pieņemta.

-N, **--with-ns**
Šo opciju var izmantot tikai kopā ar opciju **--with-timestamps**.

Katrā ievades faila rindā jābūt 5 ar atstarpēm atdalītiem ierakstiem: **<hostname> <key> <timestamp> <ns> <value>**.

Ievades faila rindas piemērs:

”Linux DB3” db.connections 1429533600 7402561 43

Papildinformāciju skatiet opcijā **--input-file**.

-r, **--real-time**
Sūtīt vērtības pa vienai, tiklīdz tās tiek saņemtas. To var izmantot, lasot no standarta ievades.

-g, **--group**
Grupēt vērtības pēc hostiem un sūtīt katram hostam atsevišķā paketē.

--tls-connect *value*
Kā izveidot savienojumu ar serveri vai starpniekserveri. Vērtības:

nešifrēts
savienoties bez šifrēšanas (noklusējums)

psk
savienoties, izmantojot TLS un iepriekš koplietotu atslēgu

cert
savienoties, izmantojot TLS un sertifikātu

--tls-ca-file *CA-file*
Pilns ceļš uz failu, kas satur augstākā līmeņa CA sertifikātus vienādranga sertifikāta verifikācijai.

--tls-crl-file *CRL-file*
Pilns ceļš uz failu, kas satur atsauktos sertifikātus.

--tls-server-cert-issuer *cert-issuer*
Atļautais servera sertifikāta izdevējs.

--tls-server-cert-subject *cert-subject*
Atļautais servera sertifikāta subjekts.

--tls-cert-file *cert-file*
Pilns ceļš uz failu, kas satur sertifikātu vai sertifikātu ķēdi.

--tls-key-file *key-file*
Pilns ceļš uz failu, kas satur privāto atslēgu.

--tls-psk-identity *PSK-identity*
PSK identitātes virkne.

--tls-psk-file *PSK-file*
Pilns ceļš uz failu, kas satur iepriekš koplietoto atslēgu.

--tls-cipher13 *cipher-string*
Šifru virkne OpenSSL 1.1.1 vai jaunākai versijai TLS 1.3 vajadzībām. Aizstāj noklusējuma šifru kopas atlases kritērijus. Šī opcija nav pieejama, ja OpenSSL versija ir mazāka par 1.1.1.

--tls-cipher *cipher-string*
GnuTLS prioritāšu virkne (TLS 1.2 un jaunākām versijām) vai OpenSSL šifru virkne (tikai TLS 1.2). Aizstāj noklusējuma šifru kopas atlases kritērijus.

-v, **--verbose**
Izvērstais režīms, **-vv** papildu detalizācijai.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

IZEJAS STATUSS

Izejas statuss ir 0, ja vērtības tika nosūtītas un serveris tās visas veiksmīgi apstrādāja. Ja dati tika nosūtīti, bet vismaz vienas vērtības apstrāde neizdevās, izejas statuss ir 2. Ja datu nosūtīšana neizdevās, izejas statuss ir 1.

PIEMĒRI

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -k mysql.queries -o 342.45

Nosūtīt **342.45** kā vērtību uzraudzītā hosta **mysql.queries** vienumam. Izmantot aģenta konfigurācijas failā definēto uzraudzīto hostu un Zabbix serveri.

zabbix_sender -c /etc/zabbix/zabbix_agentd.conf -s ”Monitored Host” -k mysql.queries -o 342.45

Nosūtīt **342.45** kā vērtību hosta **Monitored Host mysql.queries** vienumam, izmantojot aģenta konfigurācijas failā definēto Zabbix serveri.

zabbix_sender -z 192.168.1.113 -i data_values.txt

Nosūtīt vērtības no faila **data_values.txt** uz Zabbix serveri ar IP adresi **192.168.1.113**. Hostu nosaukumi un atslēgas ir definētas failā.

echo ”- hw.serial.number 1287872261 SQ4321ASDF” | zabbix_sender -c /usr/local/etc/zabbix_agentd.conf -T -i -

Nosūtīt laika zīmogu saturošu vērtību no komandrindas uz Zabbix serveri, kas norādīts aģenta konfigurācijas failā. Domuzīme ievades datos norāda, ka arī hosta nosaukums jāizmanto no tā paša konfigurācijas faila.

echo ’”Zabbix server” trapper.item ””’ | zabbix_sender -z 192.168.1.113 -p 10000 -i -

Nosūtīt tukšu vienuma vērtību uz Zabbix serveri ar IP adresi **192.168.1.113** portā **10000** no komandrindas. Tukšas vērtības jānorāda ar tukšām dubultpēdiņām.

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

Nosūtīt **342.45** kā vērtību hosta **Monitored Host mysql.queries** vienumam uz serveri ar IP **192.168.1.113**, izmantojot TLS ar sertifikātu.

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

Nosūtīt **342.45** kā vērtību hosta **Monitored Host mysql.queries** vienumam uz serveri ar IP **192.168.1.113**, izmantojot TLS ar iepriekš koplietotu atslēgu (PSK).

SKATĪT ARĪ

Dokumentācija 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)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

IZEJAS STATUSS

PIEMĒRI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 08:42:39 GMT, 2021. gada 11. jūnijā

zabbix_server

Sadaļa: Uzturēšanas komandas (8)
Atjaunināts: 2020-09-04
Saturs Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_server - Zabbix servera dēmons

KOPSAVILKUMS

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

APRAKSTS

zabbix_server ir Zabbix programmatūras galvenais dēmons.

OPCIJAS

-c, **--config** *config-file*
Izmantot alternatīvo *config-file* noklusējuma faila vietā.

-f, **--foreground**
Palaist Zabbix serveri priekšplānā.

-R, **--runtime-control** *runtime-option*
Veikt administratīvās funkcijas saskaņā ar *runtime-option*.

-T, **--test-config**
Pārbaudīt konfigurācijas failu un iziet.

-h, **--help**
Parādīt šo palīdzību un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

Zabbix servera palaišanas piemēri ar komandrindas parametriem:

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

IZPILDLAIKA VADĪBA

Izpildlaika vadības opcijas:

config_cache_reload
Pārlādēt konfigurācijas kešatmiņu. Tiek ignorēts, ja kešatmiņa pašlaik tiek ielādēta. Lai atrastu PID failu un nosūtītu signālu procesam, kas norādīts PID failā, tiks izmantots noklusējuma konfigurācijas fails (ja vien nav norādīta opcija **-c**).

snmp_cache_reload
Pārlādēt SNMP kešatmiņu, notīrīt SNMP rekvizītus (dzinēja laiku, dzinēja sāknēšanas reižu skaitu, dzinēja ID, akreditācijas datus) visiem hostiem. Ņemiet vērā, ka Zabbix kešatmiņā saglabā SNMPv3 EngineID→IP atbilstības un var automātiski atkārtoti izmantot EngineID, lai samazinātu aptaujāšanas pieskaitāmās izmaksas.

housekeeper_execute
Izpildīt housekeeper. Tiek ignorēts, ja housekeeper pašlaik tiek izpildīts.

trigger_housekeeper_execute
Izpildīt trigeru housekeeper (noņemt problēmas dzēstiem trigeriem). Tiek ignorēts, ja pakalpojumiem paredzētais trigeru housekeeper pašlaik tiek izpildīts.

diaginfo[=*section*]
Reģistrēt norādītās sadaļas iekšējo diagnostikas informāciju žurnālā. Sadaļa var būt *historycache*, *preprocessing*, *alerting*, *lld*, *valuecache*, *locks* un *connector*. Pēc noklusējuma žurnālā tiek reģistrēta visu sadaļu diagnostikas informācija.

ha_status
Reģistrēt augstas pieejamības (HA) klastera statusu žurnālā.

ha_remove_node[=*target*]
Noņemt augstās pieejamības (HA) mezglu, kas norādīts pēc tā nosaukuma vai ID. Ņemiet vērā, ka aktīvos/gaidstāves mezglus nevar noņemt.

ha_set_failover_delay[=*delay*]
Iestatīt augstās pieejamības (HA) pārslēgšanās aizkavi. Tiek atbalstīti laika sufiksi, piemēram, 10s, 1m.

proxy_config_cache_reload[=*target*] Pārlādēt starpniekservera konfigurācijas kešatmiņu.

secrets_reload
Pārlādēt noslēpumus no Vault.

service_cache_reload
Pārlādēt pakalpojumu pārvaldnieka kešatmiņu.

prof_enable[=*target*]
Iespējot profilēšanu. Ietekmē visus procesus, ja target nav norādīts. Iespējota profilēšana nodrošina detalizētu informāciju par visām rwlock/mutex funkcijām pēc funkcijas nosaukuma.

prof_disable[=*target*]
Atspējot profilēšanu. Ietekmē visus procesus, ja target nav norādīts.

log_level_increase[=*target*]
Palielināt žurnāla līmeni, ietekmē visus procesus, ja target nav norādīts

log_level_decrease[=*target*]
Samazināt žurnāla līmeni, ietekmē visus procesus, ja target nav norādīts

Žurnāla līmeņa vadības mērķi *process-type*
Visi norādītā tipa procesi (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
Procesa tips un numurs (piemēram, poller,3)

pid
Procesa identifikators, līdz 65535. Lielākām vērtībām norādiet mērķi kā ”process-type,N”

Profilēšanas vadības mērķi

process-type
Visi norādītā tipa procesi (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
Procesa tips un numurs (piemēram, history syncer,1)

pid
Procesa identifikators, līdz 65535. Lielākām vērtībām norādiet mērķi kā ”process-type,N”

scope
Profilēšanas tvērums (rwlock, mutex, processing) var tikt izmantots kopā ar procesa tipu (piemēram, history syncer,1,processing)

FAILI

/usr/local/etc/zabbix_server.conf
Zabbix serveris konfigurācijas faila noklusējuma atrašanās vieta (ja tā netika mainīta kompilēšanas laikā).

SKATĪT ARĪ

Dokumentācija https://www.zabbix.com/manuals

zabbix_agentd(8), **zabbix_get**(1), **zabbix_proxy**(8), **zabbix_sender**(1), **zabbix_js**(1), **zabbix_agent2**(8)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

FAILI

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 16:12:14 GMT, 2020. gada 04. septembrī

zabbix_web_service

Sadaļa: Uzturēšanas komandas (8)
Atjaunināts: 2019-01-29
Saturs Atgriezties uz galveno saturu

NOSAUKUMS

zabbix_web_service - Zabbix tīmekļa pakalpojums

KOPSAVILKUMS

zabbix_web_service [**-c** *config-file*]
zabbix_web_service [**-c** *config-file*] **-T**
zabbix_web_service -h
zabbix_web_service -V

APRAKSTS

zabbix_web_service ir lietotne, kas nodrošina tīmekļa pakalpojumus Zabbix komponentiem.

OPCIJAS

-c, **--config** *config-file*
Izmantojiet alternatīvo *config-file* noklusējuma faila vietā.

-T, **--test-config**
Pārbaudīt konfigurācijas failu un iziet.

-h, **--help**
Parādīt šo palīdzības informāciju un iziet.

-V, **--version**
Izvadīt versijas informāciju un iziet.

FAILI

/usr/local/etc/zabbix_web_service.conf
Zabbix tīmekļa pakalpojuma konfigurācijas faila noklusējuma atrašanās vieta (ja tā nav mainīta kompilēšanas laikā).

SKATĪT ARĪ

Dokumentācija 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)

Saturs

NOSAUKUMS

KOPSAVILKUMS

APRAKSTS

OPCIJAS

DATNES

SKATĪT ARĪ

AUTORS

Šis dokuments tika izveidots: 12:58:30 GMT, 2021. gada 11. jūnijā

Autortiesību paziņojums

Uz Zabbix dokumentāciju NEATTIECAS AGPL-3.0 licence. Uz Zabbix dokumentācijas izmantošanu attiecas šādi noteikumi:

Jūs drīkstat izveidot šīs dokumentācijas drukātu kopiju tikai savai personīgai lietošanai. Pārveidošana citos formātos ir atļauta, ja vien faktiskais saturs nekādā veidā netiek mainīts vai rediģēts. Jūs nedrīkstat publicēt vai izplatīt šo dokumentāciju jebkādā formā vai jebkādā datu nesējā, izņemot gadījumus, kad izplatāt dokumentāciju līdzīgā veidā, kā to izplata Zabbix (tas ir, elektroniski lejupielādei Zabbix tīmekļa vietnē) vai USB vai līdzīgā datu nesējā, tomēr ar nosacījumu, ka dokumentācija tiek izplatīta kopā ar programmatūru tajā pašā datu nesējā. Jebkurai citai izmantošanai, piemēram, drukātu kopiju izplatīšanai vai šīs dokumentācijas pilnīgai vai daļējai izmantošanai citā publikācijā, ir nepieciešama iepriekšēja rakstiska piekrišana no pilnvarota Zabbix pārstāvja. Zabbix patur visas tiesības uz šo dokumentāciju, kas nav skaidri piešķirtas iepriekš.